Contributions à l'étude de modèles, de langages et d'algorithmes pour le raisonnement et la prise de décision en intelligence artificielle

Jérôme Lang
Institut de Recherche en Informatique de Toulouse
CNRS - Université Paul Sabatier

Préambule

Ce document est un rapport d'habilitation à diriger des recherches. Il présente une grande partie des travaux que j'ai réalisés entre 1991 et 2002, et les agence de manière synthétique en les regroupant en cinq chapitres thématiques.

Le premier chapitre fait exception : je n'y présente pas mes contributions à un domaine de recherche, mais j'y tente, afin de donner le plus de cohérence possible au document, de construire une taxonomie *incomplète* des principaux problèmes, en intelligence artificielle, liés au raisonnement et à la prise de décision. Elle n'est pas sans rappeler, en de nombreux points, celle que Sandewall a échafaudée dans [202] afin de clarifier l'ensemble des travaux existant dans le domaine du raisonnement sur le temps et l'action. Un objectif ambitieux, qui va certes au-delà de la taxonomie que je propose, serait d'effectuer un travail similaire à celui de Sandewall, pour le domaine de la prise de décision en intelligence artificielle (qui est encore plus vaste que le domaine du raisonnement sur l'action, puisqu'il l'inclut, et en outre empiète sur la planification, la satisfaction de contraintes, la théorie de la décision, la robotique cognitive, les agents intelligents). Le premier chapitre est une tentative pour aller dans cette direction, mais il est évident que beaucoup de chemin reste à faire, notamment parce que des pans entiers de problèmes liés à la prise de décision (je pense notamment à l'apprentissage et à la décision à partir de cas) n'y figurent pas, tout simplement parce que je n'ai proposé aucune contribution dans ces domaines de recherche.

Je me sers ensuite de cette taxonomie pour organiser la présentation de mes travaux de recherche. Les chapitres 2 à 6 décrivent ainsi mes contributions à différents sous-domaines de l'intelligence artificielle, ainsi que les perspectives de recherche que j'envisage (ces dernières figurent généralement à la fin de la section ou de la sous-section pertinente).

Ce document est un rapport d'habilitation, pas une monographie (même incomplète) : ainsi, le découpage proposé en cinq chapitres thématiques n'a de sens que pour l'organisation de mes travaux de recherche de manière équilibrée et ne constitue en aucun cas un découpage représentatif de l'ensemble des travaux de la communauté dans ce domaine. Pour des raisons analogues, ces chapitres ne contiennent pas un état de l'art du domaine – quand je mentionne d'autres travaux que les miens, c'est avant tout pour mieux les situer (cependant, j'accorderai une importance aux travaux réalisés dans mon environnement proche, en particulier à l'IRIT).

Ce document n'est pas non plus un rapport de recherche : ainsi, le contenu de mes contributions est exposé brièvement et informellement, et ne comporte ni preuves, ni exemples (à quelques exceptions près pour ces derniers) ; je réfèrerai systématiquement aux articles correspondant aux sujets évoqués. Je joins à ce document une sélection d'articles à peu près représentative des différents sujets auxquels j'ai contribué.

Toile de fond et notations

Structures de base

$\subseteq$ désigne l'inclusion et $\subset$ l'inclusion *stricte*. $\times$ désigne le produit cartésien. $|.|$ désigne la cardinalité.

Ordres

préordre un préordre sur un ensemble X est une relation binaire réflexive et transitive sur X, notée $\geq$;

préordre strict relation binaire transitive et asymétrique ($\forall x, y \in X$, $x > y$ implique non$(y > x)$) sur X, notée $>$;

préordre total préordre vérifiant $\forall x, y \in X$, $x \geq y$ ou $y \geq x$;

ordre préordre antisymétrique ($\forall x, y \in X$, $x \geq y$ et $y \geq x$ implique $x = y$) ;

ordre total préordre total antisymétrique.

Soit $\geq$ un préordre sur X ; l'ordre strict généré par $\geq$ est défini par : $x > y$ si et seulement si ($x \geq y$ et non$(y \geq x)$) ; la relation d'équivalence générée par $\geq$, notée $\sim$, est définie par : $x \sim y$ si et seulement si ($x \geq y$ et $y \geq x$).

Distances

pseudo-distance faible une pseudo-distance faible sur X est une fonction $d : X \times X \to \mathbb{R}^+$, symétrique, vérifiant ($\forall x, d(x, x) = 0$) ;

pseudo-distance pseudo-distance faible vérifiant la propriété de séparation ($d(x, y) = 0$ si et seulement si $x = y$) ;

distance faible pseudo-distance faible vérifiant l'inégalité triangulaire ($d(x, z) \leq d(x, y) + d(y, z)$) ;

distance pseudo-distance vérifiant l'inégalité triangulaire.

Soit d une [pseudo-]distance [faible] sur X, $x \in X$ et $A, B \subseteq X$. On définit $d(x, A) = \min_{y \in A} d(x, y)$ et $d(A, B) = \min_{x \in A, y \in B} d(x, y)$.

Logique propositionnelle

Soit VAR un ensemble fini de variables propositionnelles. $\mathcal{L}_{VAR}$ est le langage propositionnel construit à partir de VAR, des connecteurs logiques habituels, et des symboles $\top$ et $\bot$. Les formules de $\mathcal{L}_{VAR}$ sont notées par des lettres grecques $\alpha, \beta, \varphi, \psi$ etc. On note $Var(\varphi)$ l'ensemble des variables propositionnelles apparaissant dans φ.

La relation de conséquence logique dans une logique L est notée $\vdash_L$; celle de la logique classique propositionnelle est notée $\vdash_{CL}$ ou $\vdash$ lorsqu'il n'y a pas d'ambiguïté.

Une formule φ de $\mathcal{L}_{VAR}$ est un *littéral* s'il existe une variable $v \in VAR$ telle que $\varphi = v$ ou $\varphi = \neg v$. Une *clause* est une disjonction de littéraux (en particulier, $\bot$ lorsque l'ensemble des littéraux est vide). Un *cube* est une conjonction de littéraux (en particulier, $\top$ lorsque l'ensemble des littéraux est vide). Une *CNF* est une conjonction de clauses. Une *DNF* est une disjonction de termes. Une *NNF* est une formule dans laquelle le symbole $\neg$ n'apparaît que devant des variables (et donc jamais devant une parenthèse). On note $Lit(\varphi)$ l'ensemble des littéraux apparaissant dans une forme NNF de φ.

$\Omega_{VAR} = 2^{VAR}$ est l'ensemble des interprétations de $\mathcal{L}_{VAR}$; lorsqu'il n'y a pas d'ambiguïté on le note simplement Ω. Les interprétations sont notées ω, ω' etc. La satisfaction d'une formule φ par une interprétation ω est notée $\omega \models \varphi$. $Mod(\varphi) = \{\omega \in \Omega, \omega \models \varphi\}$ est l'ensemble des modèles de φ . On note $for(\omega)$ la formule propositionnelle (unique à l'équivalence logique près) dont l'unique modèle est ω.

Soit $X \subseteq VAR$. Une X-interprétation, notée ω_X, est une interprétation sur $\mathcal{L}_X$, c'est-à-dire l'affectation d'une valeur de vérité aux variables de X. On note les interprétations [partielles] ainsi : $[a, \neg b]$ est la $\{a, b\}$-interprétation qui donne la valeur vrai (resp. faux) à a (resp. à b). On identifie parfois une interprétation [partielle] avec le terme qui lui correspond ($a \wedge \neg b$ pour l'interprétation qui précède).

Si $\varphi \in \mathcal{L}_{VAR}$ et x est une variable, $\varphi_{x\leftarrow\top}$ (resp. $\varphi_{x\leftarrow\bot}$) est le résultat de la substitution de x par $\top$ (resp. $\bot$) dans φ . L'oubli de l'ensemble de variables V dans la formule propositionnelle φ (voir [173]), noté $\exists V.\varphi$, est défini inductivement comme suit :

- $\exists\emptyset.\varphi \equiv \varphi$;
- $\exists\{x\}.\varphi \equiv \varphi_{x\leftarrow\bot} \vee \varphi_{x\leftarrow\top}$;
- $\exists\{x\} \cup V.\varphi \equiv \exists V.(\exists\{x\}.\varphi)$.

Complexité computationnelle

Dans ce document, je mentionnerai souvent des classes de complexité au-delà de NP et de coNP. Pour plus de détails sur ces classes et leurs problèmes complets, on peut voir le livre de Papadimitriou [188].

Si C et C′ sont deux classes de complexité, coC est le complémentaire de C et $\mathsf{C}^{\mathsf{C}'}$ est la classe des langages reconnaissables par une machine de Turing pour C munie d'oracles C', où un oracle C' résout une instance quelconque d'un problème dans C' ou coC' en temps unité.

- *hiérarchie booléenne* : BH_2 est la classe de tous les langages L tels que $L = L_1 \cap L_2$, où L_1 est dans NP et L_2 dans coNP (et donc, coBH_2 est la classe de tous les langages L tels que $L = L_1 \cup L_2$, où L_1 est dans NP et L_2 dans coNP) ; BH_3 est la classe de tous les langages L tels que $L = L_1 \cap (L_2 \cup L_3)$, où L_1 et L_2 sont dans NP et L_3 dans coNP ;
- *hiérarchie polynomiale* :
  - $\Delta_2^p = \mathsf{P}^{\mathsf{NP}}$ est la classe des langages reconnaissables en temps polynomial par une machine de Turing déterministe munie d'oracles NP ;
  - $\Theta_2^p = \Delta_2^p[\mathcal{O}(\log n)]$ est le sous-ensemble de Δ_2^p obtenu en requiérant que le nombre d'oracles NP soit borné par une fonction logarithmique de la taille de l'input ;
  - $\Sigma_2^p = \mathsf{NP}^{\mathsf{NP}}$; $\Pi_2^p = \text{co-}\Sigma_2^p$;
  - $\Delta_3^{\mathsf{P}} = \mathsf{P}^{\Sigma_2^p}$; $\Pi_3^p = \mathsf{coNP}^{\Sigma_2^p}$;
- PSPACE est l'ensemble de tous les langages reconnaissables par une machine de Turing (déterministe ou non) en utilisant un espace polynomial ;
- EXPTIME (resp. NEXPTIME) est l'ensemble de tous les langages reconnaissables en temps exponentiel par une machine de Turing déterministe (resp. non-déterministe) ;
- EXPSPACE est l'ensemble de tous les langages reconnaissables par une

machine de Turing (déterministe ou non) en utilisant un espace exponentiel.

On peut noter les inclusions suivantes :

$$
\begin{array}{ll}
 & \mathsf{P} \\
\subseteq & \mathsf{NP},\ \mathsf{coNP} \\
\subseteq & \mathsf{BH}_2,\ \mathsf{coBH}_2 \\
\subseteq & \mathsf{BH}_3 \\
\subseteq & \Theta_2^p \\
\subseteq & \Delta_2^p \\
\subseteq & \Sigma_2^p,\ \Pi_2^p \\
\subseteq & \Delta_3^p \\
\subseteq & \Sigma_3^p,\ \Pi_3^p \\
\subseteq & ... \\
\subseteq & \mathsf{PSPACE} \\
\subseteq & \mathsf{EXPTIME} \\
\subseteq & \mathsf{EXPSPACE}
\end{array}
$$

Théories de l'incertain

Soit S un ensemble fini.

On note $PROB_S$ l'ensemble des distributions de probabilité sur S. Une distribution (resp. mesure) de probabilité sur S (resp. sur 2^S) est notée p (resp. P).

Une *distribution de possibilité* sur S est une fonction $\pi : S \to [0,1]$ telle que $\max_{s \in S} \pi(s) = 1$. On note $POSS_S$ l'ensemble des distributions de possibilité sur S. La *mesure de possibilité* (resp. de *nécessité*) induite par π est la fonction $\Pi : 2^S \to [0,1]$ (resp. $N : 2^S \to [0,1]$) définie par $\Pi(A) = \max_{s \in A} \pi(s)$ (resp. $N(A) = \min_{s \in A}(1 - \pi(s)) = 1 - \Pi(\bar{A})$).

Une *fonction de masse* sur 2^S est une fonction $m : 2^S \setminus \{\emptyset\} \to [0,1]$. La *mesure de croyance* (resp. de *plausibilité*) induite par m est la fonction $Bel : 2^S \to [0,1]$ (resp. $Pl : 2^S \to [0,1]$) définie par $Bel(A) = \sum_{X \subseteq A} m(X)$ (resp. $Pl(A) = \sum_{X \cap A \neq \emptyset} m(X) = 1 - Bel(\bar{A})$).

Multiensembles ; ordres leximin et leximax ; OWA

Un multiensemble sur un ensemble de référence E est une fonction $\mu : E \to I\!N$. μ est fini si et seulement si $\{x \mid \mu(x) > 0\}$ est fini et dans ce cas on définit $|\mu| = \sum_{x \in E} \mu(x)$. Soient μ, μ' deux multi-ensembles sur E. On définit :

- $\mu \subseteq \mu'$ si et seulement si $\forall x \in E, \mu(x) \leq \mu'(x)$;
- $(\mu \cup \mu')(x) = \mu(x) + \mu'(x)$.

Supposons maintenant que E soit muni d'une relation d'ordre $\geq$ et soit μ un multi-ensemble fini sur E. Le réarrangement croissant de μ, noté $\mu^{\uparrow}$, est le vecteur de $E^{|\mu|}$ défini par : $\mu^{\uparrow}(i)$ est l'unique $x \in E$ tel que $\begin{cases} \sum_{y<x} \mu(y) < i \\ \sum_{y \leq x} \mu(y) \geq i \end{cases}$; le réarrangement décroissant de μ, noté $\mu^{\downarrow}$, est le vecteur de $E^{|\mu|}$ défini par $\mu^{\uparrow}(i)$ est l'unique $x \in E$ tel que $\begin{cases} \sum_{y>x} \mu(y) < i \\ \sum_{y \geq x} \mu(y) \geq i \end{cases}$.

L'*ordre leximin* est défini sur l'ensemble des multi-ensembles finis sur E par $\mu >_{leximin} \mu'$ si et seulement si il existe un entier k tel que $\mu^{\uparrow}(k) < \mu'^{\uparrow}(k)$ et $\forall i < k$, $\mu^{\uparrow}(i) = \mu'^{\uparrow}(i)$; puis $\mu \leq_{leximin} \mu'$ si et seulement si non $(\mu >_{leximin} \mu')$. Similairement, l'*ordre leximax* est défini par $\mu >_{leximax} \mu'$ si et seulement si il existe un entier k tel que $\mu^{\downarrow}(k) < \mu'^{\downarrow}(k)$ et pour tout $i < k$, $\mu^{\downarrow}(i) = \mu'^{\downarrow}(i)$; puis $\mu \leq_{leximax} \mu'$ si et seulement si non $(\mu >_{leximax} \mu')$. $\leq_{leximin}$ et $\leq_{leximax}$ sont des préordres totaux.

Soit $n > 0$ et $\vec{w} = \langle w_1, \ldots, w_n \rangle \in [0,1]^N$ tel que $\sum_{i=1} w_i = 1$. L'opérateur de *moyenne ordonnée pondérée* (OWA) induit par $\vec{w}$ est la fonction $\mathsf{OWA}_{\vec{w}}$ qui à tout multiensemble fini μ sur $[0,1]$ de cardinalité n, associe $\mathsf{OWA}_{\vec{w}}(\mu) = \sum_{i=1}^{n} w_i.\mu^{\uparrow}(i)$.

Logiques modales de la croyance et de la connaissance

Le langage de la logique des croyances KD45 propositionnelle (respectivement, de la logique des connaissances S5 propositionnelle) est généré à partir d'un ensemble de variables propositionnelles, des connecteurs logiques habituels et de la modalité **Bel** (respectivement de la modalité **K**). Les formules de KD45 ou S5 sont notées par des lettres grecques majuscules (Φ, Ψ etc.). Une formule de KD45 ou de S5 est *objective* si et seulement si elle ne contient aucune occurrence de **Bel** (resp. de **K**). Les formules objectives

sont notées par des lettres grecques minuscules (φ , ψ etc.).

Un modèle de Kripke pour S5 ou KD45 est un triplet $M = \langle W, val, R\rangle$ où W est un ensemble de mondes, $val : W \to \Omega$ est une fonction de valuation et R une relation d'accessibilité symétrique et transitive (et, en outre, réflexive dans le cas de S5). La satisfaction d'une formule par un couple (M, w) est définie de la façon inductive usuelle – en particulier, $(M, w) \models \mathbf{K}\ \Phi$ (resp. $(M, w) \models \mathbf{Bel}\ \Phi$) si et seulement si $\forall w'$ tel que $R(w, w')$ on a $M, w' \models \Phi$.

Normes et conormes triangulaires

Une *norme triangulaire* (ou *t-norme*) est une loi de composition interne $\star : [0, 1] \times [0, 1] \to [0, 1]$, associative, commutative, non-décroissante et ayant 1 pour élément neutre (et – comme propriété dérivée – 0 pour élément absorbant).

Une *conorme triangulaire* (ou *t-conorme*) est une loi de composition interne $\star : [0, 1] \times [0, 1] \to [0, 1]$, associative, commutative, non-décroissante et ayant 0 pour élément neutre (et – comme propriété dérivée – 1 pour élément absorbant).

Un *renversement* est une fonction $\nu : [0, 1] \to [0, 1]$ strictement décroissante et involutive.

Chapitre 1

Une taxonomie des processus de prise de décision en intelligence artificielle

Une partie importante de l'intelligence artificielle (IA) s'intéresse à la modélisation, la simulation et/ou à la programmation d'un *agent* face à un système. Une autre partie importante de l'IA s'intéresse aux collectivités d'agents en interaction mais dans ce mémoire, on considèrera par défaut un agent unique (appelé *l'agent* tant qu'il n'y aura pas d'ambiguïté) – cette hypothèse ne sera relâchée qu'en quelques endroits, notamment au chapitre 4.

L'intelligence artificielle considère généralement deux grands types de comportement de l'agent face au système : le comportement que l'on pourrait qualifier de "passif" (on verra à la fin de ce chapitre que ce terme n'est pas approprié), qui consiste à raisonner sur le système sans agir sur celui-ci ; et le comportement actif : l'agent effectue des actions, soit pour faire évoluer le système dans une direction souhaitable, soit pour acquérir de nouvelles informations sur ce système. On appellera généralement ces deux types de comportement, respectivement, *raisonnement* et *prise de décision* – mais cette dichotomie n'est pas dénuée d'ambiguïté, comme on va le voir souvent au long de ce mémoire.

Le terme "agent" est lui aussi ambigu. L'agent est une entité (partiellement ou totalement) autonome, dotée de capacités de perception et/ou de raisonnement, capable d'effectuer certaines actions et censée prendre des décisions (c'est-à-dire choisir les actions à effectuer) "rationnellement" (encore

un terme ambigu). Deux grandes classes d'agents sont à considérer :

- l'agent est un individu (ou un collectif indivisible d'individus – une entreprise, par exemple) ;
- l'agent est une entité autonome artificielle : un robot, un logiciel ayant des fonctions de décision, ou encore un système automatique même des plus simples.

À première vue, on peut supposer que les sciences humaines et sociales (économie et psychologie cognitives, linguistique) s'intéressent exclusivement aux agents humains et les sciences de l'ingénieur (automatique et commande de systèmes, robotique, informatique "traditionnelle") exclusivement aux agents artificiels. C'est sans doute vrai dans une bonne mesure. Cela dit, l'intelligence artificielle (et également la partie de l'informatique qui s'intéresse à l'interaction homme-machine) est concernée par les deux aspects de la question[1]. On peut dire que la notion d'agent autonome rationnel (humain ou artificiel) intervient en informatique au moins dans trois grands domaines d'applications :

- l'*aide à la décision* : il s'agit d'effectuer des tâches préparatoires (calcul, inférence, classification ...) en vue de donner à l'utilisateur humain des éléments pertinents lui facilitant la prise de décision (qu'il prendra alors seul) – il peut s'agir notamment de proposer à l'agent des décisions pertinentes ; les agents considérés dans ce contexte sont humains (individuels ou en collectivité) ;
- la *robotique cognitive* : le système informatique "intelligent" est couplé à des capteurs et des effecteurs qui interagissent avec un monde réel sans requérir une intervention humaine, et il effectue des tâches complexes nécessitant des phases de *raisonnement* (par opposition à la robotique et à l'automatique classique, où le raisonnement a été fait au préalable et compilé dans une loi de commande) : les agents considérés sont ici artificiels ;
- la *communication homme-machine* : le système informatique a ici, comme pour l'aide à la décision, un rôle d'assistance à l'utilisateur humain (ou à un groupe d'utilisateurs) mais de façon plus active : il doit dialoguer avec lui et raisonner sur ses croyances, ses préférences et

[1] Je ne peux m'empêcher ici de remarquer que l'informatique au CNRS a longtemps fait partie des "Sciences pour l'ingénieur" et constitue depuis peu un département à part qui s'intitule "Sciences et technologies de l'information et de la communication", et qui n'est pas loin d'être à égale distance thématique entre les sciences pour l'ingénieur et les sciences humaines et sociales (on peut y ajouter les mathématiques, et depuis peu la biologie – mais ce n'est pas mon propos).

ses intentions ; il y a cette fois deux types d'agents autonomes concernés, humain(s) et artificiels(s).

Bien que ces trois domaines d'application soient bien distincts (ce que confirme l'existence de communautés de recherche assez séparées), la structure du document ne suit pas cette trichotomie, parce qu'en définitive les outils méthodologiques (modèles, langages, algorithmes) utilisés dans ces domaines sont souvent communs. Par exemple, planifier dans le but d'agir soi-même ou planifier dans le but de conseiller une action à un utilisateur sont deux problèmes proches. Dans le même ordre d'idées, les modèles des sciences de la décision (économie mathématique, psychologie cognitive) sont utiles à chacun de ces domaines :

- la robotique cognitive les reprend à son compte (voir la littérature abondante ces dix dernières années sur la "planification fondée sur la théorie de la décision"), puisqu'il s'agit certes d'agir sans intervention humaine mais cependant d'agir *pour* un humain et donc de prendre les décisions qu'il aurait prises lui-même (voire de meilleures) ;
- les systèmes de communication homme-machine, qui se doivent de comprendre ce que veut dire, ce que croit, ce que désire l'utilisateur, doivent avoir une bonne connaissance de son mode de raisonnement et de son comportement, et pour cette raison doivent faire appel à des économistes cognitifs, des psychologues cognitifs et des linguistes ;
- c'est encore plus évident pour les systèmes d'aide à la décision, dont l'efficacité dépend de la pertinence de la modélisation et de l'agrégation des préférences de(s) (l')utilisateur(s).

Si l'IA gagne à faire usage des modèles normatifs et descriptifs des sciences de la décision, elle requiert par ailleurs d'autres modèles et outils méthodologiques, puisqu'il faut tenir compte des aspects *computationnels* qui sont peu pertinents pour les sciences humaines et sociales[2]. C'est là la spécificité de l'IA, sur laquelle nous n'allons pas cesser de revenir.

Revenons sur la distinction informelle faite un peu plus haut entre "raisonnement" et "décision". On aurait pu organiser le document qui suit en distinguant radicalement ces deux types de tâches. Cependant, on procédera différemment, pour deux raisons : la première est que certains processus com-

[2]Il faut toutefois noter que les notions de rationalité limitée et de charge mentale, qui ont à voir avec les aspects computationnels tels que l'informatique les entend, intéressent les chercheurs en psychologie et en économie cognitives.

plexes, comme la planification en environnement partiellement observable (traitée au chapitre 6), font intervenir les deux types de processus de façon étroitement mêlée ; la seconde est qu'il est possible (voir la fin de ce chapitre) de considérer un processus de raisonnement "pur" comme un processus de prise de décision.

1.1 Processus de prise de décision

De façon très générale, un processus de (prise de) décision ("decision making process"), pour un agent donné, consiste à faire un choix parmi une liste d'actions disponibles, en fonction de *connaissances* (ou de *croyances*) qu'il possède sur le système et sa dynamique, et de préférences qu'il a sur les états du système et les actions qu'il entreprend. Par "système" on entend tout ce qui est extérieur à l'agent et qui est pertinent pour son processus de prise de décision. Dans cette formulation, beaucoup de paramètres sont laissés très vagues. La suite de ce chapitre vise à les préciser un à un, de façon structurée.

Un *modèle* d'un processus de décision (mono-agent) est constitué du modèle du système (avec sa dynamique) et de l'agent (avec ses connaissances, les actions qui lui sont disponibles, et ses préférences). Formellement, c'est un 10-uplet $\mathcal{M} = \langle \mathcal{C}, \mathcal{H}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{B}, b_{init}, exec, evol, \mathcal{G} \rangle$ où

- $\mathcal{C}$ est la *classe* du processus ;
- $\mathcal{H}$ est l'*horizon* (du processus) ;
- $\mathcal{S}$ est l'*ensemble des états*, dit encore *espace d'états* ;
- $\mathcal{A}$ est l'*ensemble des actions* ;
- $\mathcal{O}$ est l'*ensemble des observations* ;
- $\mathcal{B}$ est l'*ensemble des états de croyance* ;
- $b_{init} \in \mathcal{B}$ (parfois noté b_0) est l'*état de croyance initial* ;
- $exec : \mathcal{S} \times \mathcal{H} \to \mathcal{A}$ est la *fonction d'exécutabilité* ;
- *evol* est la *fonction d'évolution*, ou encore de *progression* ;
- $\mathcal{G}$ est la *structure préférentielle*.

Cette définition est pour l'instant très vague. Chacune de ces composantes donnera lieu à des précisions et des explications dans les paragraphes qui suivent (sauf pour la notion de classe de processus, qui ne sera expliquée en détail qu'en Section 1.9).

Cela dit, le modèle n'est pas tout. La spécification d'un *problème de*

prise de décision[3] nécessite aussi le choix d'un *langage de représentation* suffisamment expressif mais également suffisamment pratique (efficace) pour permettre de coder le problème tout en facilitant sa résolution. Nous y reviendrons en Section 1.6.

Dans un premier temps, nous allons élaborer une typologie des *processus de décision* eux-mêmes, c'est-à-dire des différents *modèles*, avant d'élaborer une autre typologie, cette fois des *langages de représentation* ; nous finirons ce chapitre en parlant plus rapidement des méthodes de résolution (algorithmes) envisageables.

La taxonomie des processus de décision que nous allons élaborer est inspirée (surtout pour ce qui concerne les actions et l'inertie) de celle que Sandewall a mise en place pour les systèmes dynamiques [202]. Cependant, la nôtre ne se focalise pas sur les mêmes paramètres (nous introduisons des paramètres relatifs à la prise de décision, en particulier l'observabilité et les préférences, qui sont absents dans [202], et inversement, nous choisissons d'ignorer certains paramètres des actions, comme les effets retardés, le temps continu, etc.).

Comme chez Sandewall, nous faisons une distinction entre les paramètres (dits encore *spécialités*) *ontologiques* qui sont relatifs au système lui-même et à la façon dont l'agent peut agir sur celui-ci, et les paramètres *épistémiques* relatifs à la connaissance que l'agent a sur le système. Nous introduirons une troisième classe de paramètres, relative aux *préférences* que l'agent a sur les états et évolutions possibles du système.

Une instance donnée de chacun des paramètres définira une classe particulière de processus de décision, notée $\mathcal{C}$, qui aura des langages de représentation et des techniques de calcul appropriées.

[3] Cette terminologie est dangereuse pour les lecteurs informaticiens. Un problème de prise de décision n'a rien à voir avec un problème de décision au sens informatique du terme ; c'est d'ailleurs pour éviter d'entretenir une confusion que l'on utilise depuis le début du document "prise de décision" (traduction certes un peu maladroite de "decision making"). Malheureusement, dans ce document on parlera beaucoup de complexité computationnelle et on aura donc *aussi* besoin de se référer à des problèmes de décision au sens informatique... On essaiera de faire en sorte qu'il n'y ait jamais d'ambiguïté.

1.2 Paramètres ontologiques

Nous devons d'abord introduire les notions de base d'un processus de décision, à savoir l'horizon, les états, les actions et les trajectoires.

1.2.1 Horizon

L'*horizon* du processus est l'ensemble $\mathcal{H}$ des étapes pertinentes pour le contrôle et l'observation du processus. Différentes hypothèses sur l'horizon donnent naissance à des modèles mathématiques bien différents. Lorsque l'horizon est continu, les modèles (étudiés principalement en automatique[4] requièrent des outils mathématiques puissants provenant de l'algébre linéaire ou du calcul différentiel et sont bien loin de ceux que nous allons passer en revue dans ce mémoire. Pour cette raison nous supposerons dans toute la suite du mémoire que *le temps est représenté sur une échelle discrète* (ce qui est une hypothèse courante en intelligence artificielle). Par ailleurs, nous supposerons que le temps a une origine, qui sera par une convention toute naturelle $t_0 = 0$. Cela nous laisse apparemment deux grandes hypothèses possibles en ce qui concerne l'horizon du processus :

- *horizon fini* : $\mathcal{H} = \{0, ..., T\}$. La première étape de décision a lieu à l'instant 0 et la dernière au plus tard à l'instant $T - 1$ (après que la dernière décision a conduit à l'exécution d'une action à l'instant $T-1$, le système évolue encore une fois et l'ultime instant pertinent est donc T). Nous aurons parfois besoin de nommer l'ensemble des étapes de décision, $\mathcal{T}_A = \{0, ..., T-1\}$ ($= \emptyset$ lorsque $T = 0$). Deux cas particuliers seront considérés assez souvent dans ce mémoire – et ils sont assez particuliers pour qu'on en fasse des hypothèses à part :
  - *staticité* : $\mathcal{H} = \{0\}$. Le système n'évolue pas, il est statique. Dans ce cas, $\mathcal{T}_A = \emptyset$, l'agent n'a aucun contrôle sur le système, et il n'y a aucune étape de décision *à proprement parler*. On y reviendra en Section 1.8.
  - *mono-étape* ("single-stage") : $\mathcal{H} = \{0,1\}$. Une seule étape de décision, deux instants ("avant" et "après").
- *horizon infini* : $\mathcal{H} = \mathbb{N}$. On le cite pour mémoire mais on ne considèrera (presque) pas de processus à horizon infini dans ce mémoire[5].

[4] Ceci ne veut pas dire que l'automatique ne s'intéresse pas aux processus en temps discret !

[5] Un cas intermédiaire est celui des processus à horizon *indéterminé* où le nombre d'étapes est fini mais non connu à l'avance et non borné. On modélise généralement ces processus comme des processus à horizon infini (voir [31]).

Au chapitre 6 on s'intéressera à des classes de processus à horizon fini pour lesquels l'horizon est borné par une fonction polynomiale de la taille des données d'entrée ("horizon polynomial")[6].

1.2.2 États, actions, trajectoires et conséquences

États

Un *état* est la description du système à un instant donné. Tout comme l'horizon, l'espace d'états peut être continu, infini dénombrable, ou fini. Comme je m'intéresse avant tout à des problèmes combinatoires et à leur complexité algorithmique, je supposerai cependant dans tout le document que *l'espace d'états* $\mathcal{S}$ *est fini* (sauf précision contraire). L'état initial du système est noté s_0.

Actions

Une *action* est destinée à faire évoluer l'état du système ou les croyances de l'agent. Pour l'instant nous allons d'abord examiner les actions destinées à changer (en général) l'état du système, appelées dans la suite du document *actions effectives* (parce qu'elles requièrent des effecteurs) – on pourrait aussi les appeler *ontiques* ou *physiques* . Les *actions épistémiques* sont destinées à faire évoluer les croyances de l'agent ; elles seront l'objet du chapitre 5. Un troisième type d'action, plus marginal (les *actions délibératives*) sera introduit en Section 1.8 (il n'est pas utile d'en dire plus pour l'instant). Les actions sont par définition sous le contrôle de l'agent[7] : on supposera qu'à chaque étape de décision, l'agent décide de l'exécution d'une certaine action (qui peut être "ne rien faire" – cette action particulière sera notée λ) et le système évolue en conséquence.

Trajectoires

Une *trajectoire* du système, notée τ , est une séquence d'états, d'actions et d'observations[8] : lorsque l'horizon est fini,

[6] Cette définition ne s'applique pas à un processus de décision fixé – cela n'aurait pas de sens – mais à une *classe* de processus.

[7] Les actions effectuées par d'autres agents peuvent être vues comme des événements exogènes (voir le paragraphe 1.2.3)

[8] On pourrait discuter de la pertinence de parler des observations dans la Section qui concerne les paramètres ontologiques plutôt que dans celle qui concerne les paramètres épistémiques ; ce choix n'est pas simple à faire (voir le paragraphe 1.3.4).

$$\tau = \langle\langle s_0, o_0, a_0\rangle, \ldots, \langle s_{T-1}, o_{T-1}, a_{T-1}\rangle, \langle s_T, o_T\rangle\rangle \quad ;$$

lorsque l'horizon est infini, $\tau = \langle\langle s_i, o_i, a_i\rangle, i \in I\!N\rangle$. On note $\tau(i) = \langle s_i, o_i, a_i\rangle$ pour $i < T$ et $\tau(T) = \langle s_T, o_T\rangle$. La *trajectoire d'états* associée à τ est la suite d'états $\langle s_i, i \in \mathcal{H}\rangle$; c'est donc la projection de τ sur les états. *Par abus de notation et de langage nous emploierons le terme "trajectoire" et la notation τ pour la trajectoire d'états, lorsque seuls les états sont pertinents.* De façon similaire, la *trajectoire observable* associée à τ ne contient que les actions et les observations et la *trajectoire objective* associée à τ ne contient que les actions et les états. On note $TRAJ_T$ (respectivement $OBSTRAJ_T$) l'ensemble des trajectoires (respectivement des trajectoires observables) de longueur T (donc d'horizon $T-1$).

Conséquences

On a choisi de ne pas distinguer (dans les notations) les espaces d'états à chacun des instants : $\mathcal{S}$ est l'espace d'états (l'univers des mondes possibles) pour l'instant 0, l'instant 1, etc. (alors qu'on aurait pu choisir de noter S_0, S_1, etc.) ; cela n'entraîne pas de perte de généralité. Cela dit, dans le cas particulier où les préférences de l'agent dépendent exclusivement de l'état final (obtenu en bout de trajectoire), on utilisera le terme *conséquences* pour désigner les états finaux possibles, et on notera $\mathcal{X}$ (plutôt que $\mathcal{S}$) l'ensemble des conséquences. La raison de ce choix de notation (qui est habituel en théorie de la décision) est un gain de clarté lorsqu'il s'agira d'exprimer des préférences (notamment au chapitre 4).

1.2.3 Inertie

L'inertie est la propension du système à rester stationnaire en l'absence d'actions entreprises par l'agent.

- *inertie forte* : en l'absence d'actions, le système n'évolue pas.
- *inertie faible*, ou encore *tendance à l'inertie* : en l'absence d'actions, le système *tend* à rester stationnaire : cette hypothèse de staticité est faite par défaut à chaque fois que rien ne la contredit.
- cas général (non-inertie) : le système est soumis à une évolution qui lui est propre, indépendamment des actions entreprises par l'agent.

Lorsque le système n'est pas inerte, il est sujet à l'occurrence d'*événements exogènes* qui sont indépendants de la volonté de l'agent. La tendance à l'inertie se traduit par le fait que les événements sont exceptionnels (dans [202]

ils sont appelés *surprises*)[9].

1.2.4 Exécutabilité

À un instant donné t et dans un état donné s du système, l'ensemble des actions disponibles est $exec(s,t) \subseteq \mathcal{A}$. Plusieurs hypothèses :

- *exécutabilité totale* : $exec(s,t)$ ne dépend ni de s ni de t : $exec(s,t) = \mathcal{A}$ pour tous s et t.
- *absence de préconditions*, ou encore *exécutabilité inconditionnelle* : $exec(s,t)$ ne dépend que de t.
- *exécutabilité stationnaire* : $exec(s,t)$ ne dépend que de s.

Par défaut, on fera l'hypothèse d'exécutabilité totale. Cette hypothèse peut sembler *a priori* très restrictive (dans la littérature sur la planification en intelligence artificielle, par exemple, les actions ont des préconditions qui définissent les états dans lesquelles elles sont exécutables). Cependant, elle ne provoque pas vraiment de perte de généralité : en effet, lorsqu'on dit qu'une action n'est pas exécutable dans un certain état ou à un certain instant, on est en pratique presque toujours dans l'une de ces quatre situations :

(i) l'action n'a aucun effet ;
(ii) l'action a un effet désastreux (ou encore, l'action est illégale) ;
(iii) l'action a un effet imprévisible ;
(iv) l'action est physiquement impossible.

Dans les quatre cas, on peut modéliser l'action comme totalement exécutable, pourvu que le modèle soit quelque peu élaboré : dans le cas (i), c'est simple ; dans le cas (ii), il faut avoir un modèle préférentiel suffisamment élaboré pour représenter des utilités fortement négatives ; dans le cas (iii), il faut pouvoir représenter des actions non-déterministes dont l'effet est l'ignorance totale sur les valeurs des variables qu'elles concernent ; enfin, dans le cas (iv), il suffit de considérer, à la place de l'action a, l'action "tenter de réaliser a" : lorsque cela n'est pas possible, on se ramène alors au cas (i). Des exemples :

- déplacer le cube C sur la table lorsque C est recouvert par un autre cube : (i) et (iv) ;

[9] La différence entre tendance à l'inertie et non-inertie, liée à la fréquence des changements, conduit, dans le premier cas (et seulement le premier) à des stratégies de minimisation des changements (voir [202], chapitres 9-11).

– mesurer l'intensité du courant lorsque l'ampèremètre est en panne : (i) et (iv) ;
– prendre le médicament M (alors qu'il est incompatible avec M' déjà pris par le patient) : (ii) ou (iii) ;
– faire le quotient A/B (pour un programme informatique) lorsque $B = 0$: (iii) si l'on considère l'effet sur le registre concerné (en général imprévisible).

1.2.5 Fonctions d'évolution, markovianité et stationnarité

Afin de pouvoir garder un niveau de généralité suffisant, nous définissons une *fonction d'évolution* du système qui, à une trajectoire correspondant au passé et au présent du système (comprenant donc l'état courant) et à l'action entreprise, associe un état de croyance sur le nouvel état du système. La terminologie "état de croyance" est volontairement vague, et sera précisée par la suite (en Section 1.3) ; nous supposerons donc qu'il existe un espace (abstrait pour l'instant) d'états de croyance possibles, noté $\mathcal{B}$. (Donnons toutefois un exemple pour faciliter la lecture de ce paragraphe : si on choisit une modélisation probabiliste des croyances sur le monde et son évolution, $\mathcal{B}$ est l'ensemble des distributions de probabilité sur $\mathcal{S}$). Une *fonction d'évolution* est alors une fonction

$$evol : TRAJ_{\mathcal{T}} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{B}$$

Cette définition est lourde car très générale. Elle ne va pas cesser d'être rendue plus simple par des hypothèses sur la modélisation du système (markovianité, stationnarité, déterminisme ...) et plus spécifique par des choix sur la structure des états de croyance.

La modélisation du système est *markovienne* si l'évolution du système est indépendante de son histoire, c'est-à-dire si les croyances que l'agent possède sur l'état s_{t+1} ne dépendent que de l'état du système s_t à l'instant t et de l'action entreprise a_t. Formellement, cela revient à imposer que la fonction *evol* puisse s'exprimer ainsi :

$$evol(\tau, s, a) = evolm(|\tau|, s, a)$$

où *evolm* est une fonction de $\mathcal{H} \times \mathcal{S} \times \mathcal{A}$ dans $\mathcal{B}$.

La modélisation du système est *markovienne et stationnaire*[10] si et seulement si l'évolution du système ne dépend ni de la trajectoire passée, ni de l'instant considéré. Formellement, cela revient à imposer que la fonction *evol* ne dépende pas du tout de τ, c'est-à-dire

$$evol(\tau, s, a) = evolms(s, a)$$

où $evolms$ est une fonction de $\mathcal{S} \times \mathcal{A}$ dans $\mathcal{B}$.

Par exemple, lorsque la modélisation des croyances est probabiliste, la markovianité signifie que l'évolution du système peut être représentée par des probabilités de transition $p_t(.|a)$ sur $\mathcal{S}$; si de plus le modèle est stationnaire alors les probabilités de transition ne dépendent pas de t et il suffit alors, pour chaque action a et chaque état s, d'une distribution de probabilité $p(.|s, a)$.

Il est important de noter que la markovianité et la stationnarité ne sont pas des propriétés du système lui-même mais de sa *modélisation*[11]. Intuitivement, pour rendre un modèle markovien et stationnaire il suffit de coder suffisamment d'information sur le passé du système et sur t dans l'état courant (on peut se référer à [31]).

Dans la suite du document, on fera toujours les hypothèses de markovianité et de stationnarité, sauf bien entendu si le contraire est indiqué. La remarque qui précède nous montre que cela n'entraine pas vraiment de perte de généralité.

1.2.6 Effets des actions : déterminisme, non-déterminisme, effets conditionnels

On touche maintenant à une question délicate : la discussion sur le déterminisme des actions doit-elle figurer dans la Section sur les paramètres ontologiques ou dans celle sur les paramètres épistémiques ? Sandewall [202] choisit de la faire figurer dans les paramètres ontologiques. Notons cependant

[10] Plus généralement on peut définir la stationnarité indépendemment de la markovianité, et avoir ainsi des modèles stationnaires mais non markoviens. Pour simplifier, on ne le fera pas dans ce document.

[11] De toute façon , on ne traitera jamais que la modélisation d'un système, jamais le système lui-même ...

qu'il ne considère pas le déterminisme comme un paramètre à part entière : dans sa taxonomie, les actions sont réparties en deux grandes classes, à savoir, les actions à effet unique et les actions à effets alternatifs ; cette seconde classe regroupe les actions non-déterministes et les actions à effets conditionnels (dépendant simplement de l'état dans lequel l'action a été exécutée). *A priori*, il semble fondamental de faire une distinction primaire entre non-déterminisme et effets conditionnels (et nous le ferons) – et surtout dans une optique de prise de décision. Cependant, nous verrons que cette distinction n'est pas toujours si évidente (nous y reviendrons en Section 1.3.4).

Actions déterministes et non-déterministes

Une action a est dite *déterministe* si et seulement si pour tout instant t, pour toute trajectoire passée $\tau = \langle s_0, ..., s_{t-1} \rangle \in TRAJ_T$ du processus et pour tout état s_t dans lequel a est exécutable, il n'y a qu'un seul état résultant possible s_{t+1}. En d'autres termes, la fonction d'évolution *evoldet* d'une action déterministe peut être décrite en utilisant l'espace des croyances précises $\mathcal{B} = \mathcal{S}$, c'est-à-dire $evoldet(\tau, s, a) \in \mathcal{S}$. Sous les hypothèses de markovianité et de stationnarité, la dynamique d'une action déterministe a peut donc être écrite par une fonction $evoldetms : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$; on notera plus simplement $res(s, a)$ au lieu de $evoldetms(s, a)$. Notons bien que si l'état subséquent $res(s, a)$ est complètement déterminé, *il n'est cependant pas nécessairement complètement connu de l'agent* (voir en 1.3.3 et la discussion de 1.3.4). Toute action qui n'est pas déterministe est dite *non-déterministe*.

Actions à effets conditionnels ou inconditionnels

Une action a (déterministe ou non) est à effets conditionnels si ses effets sur le système dépendent de s. Sous les hypothèses de markovianité et de stationnarité, une action est à effets *inconditionnels* si la fonction $evolms$ ne dépend pas de s, c'est-à-dire $evolms(s, a) = evolmsi(a)$ où $evolmsi$ est une fonction de $\mathcal{A}$ dans $\mathcal{B}$. Lorsque a est de plus déterministe, c'est très simple : a est à effets conditionnels si et seulement si la fonction res qui à un état s_t et une action a associe l'état suivant s_{t+1} n'est pas constante. Lorsque a est non-déterministe, elle est inconditionnelle si, intuitivement, la donnée de l'état s_t n'apporte aucune information quant à l'état ultérieur s_{t+1}. Par exemple, dans le cas d'une modélisation probabiliste des croyances, a est à effets inconditionnels si et seulement si sa probabilité de transition $p(.|s, a, t)$

ne dépend pas de s.

1.3 Paramètres épistémiques

Les paramètres épistémiques concernent les hypothèses sur la nature et la quantité des *croyances* que l'agent a sur l'état du monde à chaque instant.

1.3.1 État initial, état courant : incertitude statique

L'agent a une croyance sur l'état initial du monde qui peut être complète ou incomplète. Lorsque la croyance est complète, l'agent connait avec précision l'état du monde, et sa croyance peut dont être représentée directement par cet état s_0 . Lorsque la croyance est incomplète, plusieurs états initiaux possibles sont (plus ou moins) compatibles avec l'état de croyance de l'agent. On va introduire un état de croyance initial abstrait b_0 comme un élément de $\mathcal{B}$, où $\mathcal{B}$ est l'espace des états de croyance. Ce qui s'applique à l'état initial s'applique de la même façon à l'état courant à n'importe quelle étape t (d'autant plus que s_t n'est autre que l'état initial du sous-processus de décision qui reste lorsque t unités de temps se sont écoulées). Pour chaque instant t, l'agent aura un état de croyance b_t, complet ou incomplet, sur l'état courant s_t.

Selon les capacités que l'agent a de quantifier ou de comparer les croyances qu'il a relativement à cet état courant, on lui associera donc un modèle de l'incertitude, c'est-à-dire un espace des états de croyance $\mathcal{B}$. Nous allons passer en revue quelques modèles, mais il faut avoir à l'esprit que cette liste n'a pas de prétention à l'exhaustivité.

- *états de croyance complets.* Comme on l'a vu, plus haut $\mathcal{B} \equiv \mathcal{S}$ et b_t peut être identifié à un état unique s_t.

- *états de croyance incomplets "tout ou rien"* (incertitude binaire). Lorsque l'agent n'est pas capable ni de comparer, ni, encore moins, de quantifier les croyances incomplètes qu'il a en certaines propriétés du monde, sa croyance est représentée tout simplement par un ensemble non vide d'états possibles $b_t \subseteq \mathcal{S}$. On a donc $\mathcal{B} \equiv 2^{\mathcal{S}} \setminus \{\emptyset\}$. Chaque état $s \in b_t$ est un état courant plausible et chaque état de $\mathcal{S} \setminus b_t$ ne peut en aucun cas, selon l'agent, être l'état courant.
  Étant donné un état de croyance b_t et un sous-ensemble A de $\mathcal{S}$, l'une exactement de ces situations se produit :

1. l'agent croit que l'état réel est dans A, ce qui se traduit par $b_t \subseteq A$;
2. l'agent croit que l'état réel n'est pas dans A, ce qui se traduit par $b_t \cap A = \emptyset$;
3. l'agent considère comme possible que l'état réel soit dans A, et également qu'il n'y soit pas, ce qui se traduit par $b_t \cap A \neq \emptyset$ et $b_t \cap \bar{A} \neq \emptyset$.

Un état de croyance binaire peut aussi être vu comme un modèle de Kripke de la logique doxastique KD45 ou de la logique épistémique S5 – les trois situations précédentes correspondant à la vérité dans ce modèle de Kripke des formules, respectivement, **Bel** φ, **Bel** $\neg\varphi$, $\neg$**Bel** $\varphi \wedge \neg$**Bel** $\neg\varphi$ dans KD45 (ou **K** φ, **K** $\neg\varphi$, $\neg$**K** $\varphi \wedge \neg$**K** $\neg\varphi$ dans S5).
Le propos qui précède suggère qu'il est temps de clarifier l'usage jusqu'ici un peu anarchique de "croyances" et de "connaissances". Selon la distinction faite par les logiciens, la connaissance diffère de la simple croyance en ceci qu'une connaissance est vraie dans le monde réel (l'axiome **K** $\varphi \rightarrow \varphi$ de S5) alors qu'une croyance ne l'est pas forcément. Par souci d'homogénéité terminologique, je préfère employer "croyance" plutôt que connaissance, mais c'est un abus (et il serait lourd d'écrire sans cesse "croyance ou connaissance"). Quand il sera important de choisir (notamment aux chapitres 5 et 6 où nous utiliserons abondamment la logique épistémique), on le précisera clairement.

– *états de croyance bayésiens.* $\mathcal{B}$ est l'ensemble des distributions de probabilité sur $\mathcal{S}$, noté $PROB_{\mathcal{S}}$ (rappelons que $\mathcal{S}$ est fini). On note respectivement p et P les distributions de probabilité sur $\mathcal{S}$ (resp. les mesures de probabilité sur $2^{\mathcal{S}}$). C'est le modèle le plus fréquemment utilisé ; il est d'autant plus pertinent que le processus étudié est par nature mesurable et expérimentable de façon répétée. Mais il est de nombreuses situations où il n'est pas raisonnable de supposer que l'agent dispose d'une telle distribution de probabilité pour quantifier sa croyance sur l'état courant du monde. D'autres modèles ont été élaborés, qui permettent de pallier cette limitation. Certains sont des extensions du modèle probabiliste, et d'autres remettent en cause la nature quantitative des croyances relatives de l'agent.

– *états de croyance probabilistes non bayésiens.* La modèle bayésien suppose l'*unicité* de la distribution de probabilité. Relâcher cette hypo-

thèse d'unicité permet de définir un état de croyance comme une *famille de distributions de probabilité*. Des modèles moins généraux que ce dernier mais toutefois plus généraux que le modèle bayésien peuvent être obtenus en imposant des contraintes sur les familles de probabilités (par exemple, les *probabilités intervalles*).

– *états de croyance évidentiels*. Cette autre généralisation du modèle bayésien (qui peut en fait être vue comme un cas particulier du précédent) consiste à répartir une masse unitaire non plus sur les différents états du monde, mais sur différents sous-ensembles non vides d'états du monde (ou encore, différentes propriétés ou informations relatives au monde). Cette fonction de masse m est donc une fonction de $2^{\mathcal{S}} \rightarrow [0,1]$ telle que $m(\emptyset) = 0$[12] et $\sum_{X \in 2^S} m(X) = 1$. C'est la *théorie des fonctions de croyance* de Dempster-Shafer. La croyance en la propriété "l'état courant est dans A" est quantifée par une *fonction de croyance* Bel définie par $Bel(A) = \sum_{X \subseteq A} m(X)$. [13]

– *états de croyance ordinaux et possibilistes*. Il est de nombreux cas où cela a peu de sens de quantifier les croyances d'un agent par une distribution de probabilité ou plus généralement par un modèle quantitatif. Pour autant, le modèle "tout ou rien" est bien trop peu expressif puisqu'il ne permet pas d'exprimer des comparaisons entre intensités de croyance. Les modèles ordinaux sont en quelque sorte un intermédiaire entre des modèles (trop) quantitatifs et des modèles (trop) peu expressifs, et constituent ainsi parfois un bon compromis. Un état de croyance *ordinal* est une relation de *préordre* (c'est-à-dire une relation réflexive et transitive) sur $\mathcal{S}$. Un état de croyance ordinal *total* est un préordre total sur $\mathcal{S}$. $\mathcal{B}$ est donc selon le cas l'ensemble des préordres ou des préordres totaux sur $\mathcal{S}$. La *théorie des possibilités*, du moins sa version de base, coïncide avec le modèle ordinal total défini plus haut. Un état de croyance possibiliste est une distribution de possibilité π [14], qui

[12] À quelques endroits dans le document on relâchera cette hypothèse. Par défaut, on la suppose vérifiée.

[13] On définit de façon duale la *plausibilité* de A par $Pl(A) = \sum_{X \cap A \neq \emptyset} m(X) = 1 - Bel(\overline{A})$.

[14] Attention à la notation ! Dans ce mémoire on utilisera π pour les distributions de possibilité et le caractère légèrement différent ϖ pour désigner les politiques d'action. Il fallait choisir entre éviter une notation ambiguë et utiliser les notations usuelles, j'ai choisi la seconde option. Cette ambiguïté de notation ne sera véritablement gênante qu'en Section 7.2. où les deux notions (et notations, donc) seront simultanément requises. Par chance pour le lecteur, il ne sera jamais question de trigonométrie dans ce document.

est une fonction de $\mathcal{S}$ dans $[0,1]$ qui satisfait (sauf précision contraire) l'hypothèse de *normalisation* $\max_{s\in\mathcal{S}} \pi(s) = 1$ (rappelons que S est supposé fini, ce qui justifie l'emploi de la notation max). $\pi(s)$ quantifie l'absence de contradiction entre les croyances de l'agent et l'énoncé "l'état du monde est s" : ainsi, $\pi(s) = 1$ dès que rien dans les informations dont l'agent dispose ne contredit l'hypothèse que l'état réel soit s. Étant donnée une distribution de possibilité π, la *mesure de possibilité* attachée à l'événement $A \subseteq S$ est définie par $\Pi(A) = \max_{s\in A} \pi(s)$ et sa *mesure de nécessité* par $N(A) = 1 - \Pi(\bar{A}) = \min_{s\notin A} 1 - \pi(s)$. Le caractère ordinal de ce modèle est une conséquence directe de l'utilisation des seuls opérateurs min et max. [15] [16]

Il existe encore d'autres structures possibles pour les états de croyance, en particulier les probabilités non-standard (faisant appel à des infinitésimaux)[17], mais on s'en tiendra là.

1.3.2 Effets des actions sur le monde

Les croyances sur les effets des actions sont sujettes à la même taxonomie que les croyances sur l'état du monde. C'est de toute façon évident si l'on pense que parler des effets d'une action n'est rien d'autre que de parler de l'état du monde qui sera obtenu après l'exécution de celle-ci dans l'état courant. On a déjà parlé du déterminisme à la Section précédente, parlons maintenant des différents modèles épistémiques du non-déterminisme, qui sont naturellement calqués sur les modèles de l'incertitude : on obtient un modèle du non-déterminisme en spécifiant l'espace des états de croyance $\mathcal{B}$ dans lequel la fonction d'évolution *evol* prend ses valeurs. Par souci de sim-

[15] L'opération $x \mapsto 1 - x$ (appelé renversement d'ordre) ne contrarie pas l'ordinalité, du moins tant qu'on ne compare jamais des quantités $\pi(s)$ avec des quantités $1 - \pi(s')$.

[16] Cette notion d'"ordinalité" pourrait être formalisée davantage. Formellement, un modèle constitué d'objets de base (ici, les distributions de possibilité) et d'un ensemble de constructeurs unaires $\mathcal{C}$ (ici, les opérations d'induction $Poss : \pi \mapsto \Pi$ et $Nec : \pi \mapsto N$ – d'autres constructeurs seront ajoutés par la suite) est ordinal si et seulement si pour tout π et tout π' ordre-équivalents (à savoir, $\forall s, s' \in S, \pi(s) \leq \pi(s')$ ssi $\pi'(s) \leq \pi'(s')$), alors pour chaque $C \in \mathcal{C}$, $C(\pi)$ et $C(\pi')$ sont ordre-équivalents. Cette définition peut facilement être généralisée lorsque les constructeurs ne sont plus unaires. Il n'est pas difficile de voir que le modèle possibiliste ainsi défini satisfait l'hypothèse d'ordinalité. Bien évidemment, les modèles quantitatifs évoqués plus haut ne le sont pas.

[17] En ce qui concerne le modèle des fonctions conditionnelles ordinales [211], encore appelées "fonctions *kappa*", rien ne permet pour l'instant de les différencier du modèle possibiliste (seul le conditionnement le permettra, et nous n'avons pas encore parlé de conditionnement).

plicité on fait les hypothèses de markovianité et de stationnarité et par abus de notation on emploiera la notation *evol* au lieu de *evolms*.

- *non-déterminisme "tout ou rien"* : la fonction *evol* est à valeurs dans l'espace des états de croyance tout-ou-rien, c'est-à-dire

  $$evol \ : \mathcal{S} \times \mathcal{A} \rightarrow 2^{\mathcal{S}} \setminus \{\emptyset\}$$

  On a donc, pour une action donnée a et un état donné s, un ensemble non vide $evol(s, a)$ d'états successeurs possibles.

- *non-déterminisme stochastique* : la fonction *evol* est à valeurs dans l'espace des états de croyance bayésiens, c'est-à-dire

  $$evol \ : \mathcal{S} \times \mathcal{A} \rightarrow PROB_{\mathcal{S}}$$

  Pour une action donnée a et un état donné s, l'état de croyance $evol(s, a)$ résultant de l'application de a dans s est une distribution de probabilité sur $\mathcal{S}$. Par la suite on utilisera la notation $p(.|s, a)$ au lieu de $evol(s, a)$; on a donc, pour tout $s' \in \mathcal{S}$, $p(s'|s, a) = evol(s, a)(s')$. Étant donnée une indexation des états $S = \{s_i \mid 1 \leq i \leq |\mathcal{S}|\}$, on peut également représenter les effets d'une action a par une matrice stochastique de taille $|\mathcal{S}|$ dont le coefficient $a_{i,j}$ de la ligne i et de la colonne j est égal à $p(s_i|s_j, a)$.

- *non-déterminisme ordinal* : la fonction *evol* est à valeurs dans l'espace des états de croyance ordinaux, c'est-à-dire l'ensemble des relations de préordre sur $\mathcal{S}$. La relation de préordre $evol(s, a)$ sera notée $\geq_{s,a}$: $s_1 \geq_{s,a} s_2$ signifie ainsi que l'application de a dans s rend le résultat s_1 plus plausible que s_2. Le cas particulier non-déterminisme ordinal total correspond de manière univoque au *non-déterminisme possibiliste* où les effets d'une action a sont modélisés par une famille de distributions de possibilité normalisées $\pi(.|s, a)$, pour $s \in \mathcal{S}$. $\pi(s'|s, a)$ quantifie la possibilité d'obtenir l'état s' après avoir exécuté l'action a dans l'état s.

On n'ira pas plus loin dans l'énumération des types de non-déterminisme obtenus en faisant varier l'espace des états de croyance. Cela ne présente de toute façon aucune difficulté.

Sous les hypothèses de markovianité et de stationnarité, la définition que nous avons donnée de la fonction d'évolution prend comme argument

une action et un état initial ; or, dans la pratique, cet état initial n'est pas nécessairement connu. Il importe donc de déterminer, en fonction de $evolms$, le nouvel état de croyance b' résultant de l'application de l'action a dans l'état de croyance b. Cette transformation d'un état de croyance en un autre par une action sera appelée *progression*, et notée $prog$:

$$prog \ : \mathcal{B} \times \mathcal{A} \rightarrow \mathcal{B}$$

Bien entendu, la définition formelle de $prog$ dépend de la nature de l'espace de croyance $\mathcal{B}$. *De plus, elle suppose que les modèles d'incertitude statique et d'incertitude dynamique (non-déterminisme) choisis sont identiques.*

Cette restriction semble fort limitative : il se peut en effet que les croyances sur l'état courant et les croyances sur les actions soient de nature différente. Cependant, cette restriction n'est pas si limitative qu'elle en a l'air : lorsque cette situation d'hétérogénéité des modèles apparait (un modèle M_E des croyances sur l'état courant et un modèle M_A des croyances sur les effets des actions), on commencera par identifier un modèle commun plus général des états de croyance dans le processus de décision, comme le plus petit modèle au moins aussi général que M_A et M_E, c'est-à-dire le modèle $M_A \vee M_E$ dans par le treillis des modèles de l'incertitude de la Figure 1. Ainsi, par exemple, si les croyances de l'agent sur l'état initial du monde sont modélisées par un ensemble d'états (incertitude "tout ou rien") et les effets des actions par des distributions de probabilité, alors le modèle adéquat pour l'ensemble du processus de décision sera la théorie des fonctions de croyance.

Il reste maintenant à expliciter la fonction de progression pour chacun des modèles possibles. Pour certains modèles, il existe un choix très naturel pour $prog$; pour d'autres ce n'est pas le cas et plusieurs choix pertinents sont possibles.

- dans le modèle "tout-ou-rien", le choix de la fonction $prog$ est évident : un état s' est considéré (par l'agent) comme possible après l'exécution de l'action a dans l'état de croyance $b \subseteq \mathcal{S}$ si et si seulement s'il existe un état $s \in b$ tel que s' est un résultat possible de l'action a dans s, c'est-à-dire
$$prog(b, a) = \bigcup_{s \in b} evol(s, a)$$
- dans le modèle bayésien, le choix évident est obtenu en identifiant le modèle du processus à une chaine de Markov : $prog(p_t, a)$ est la

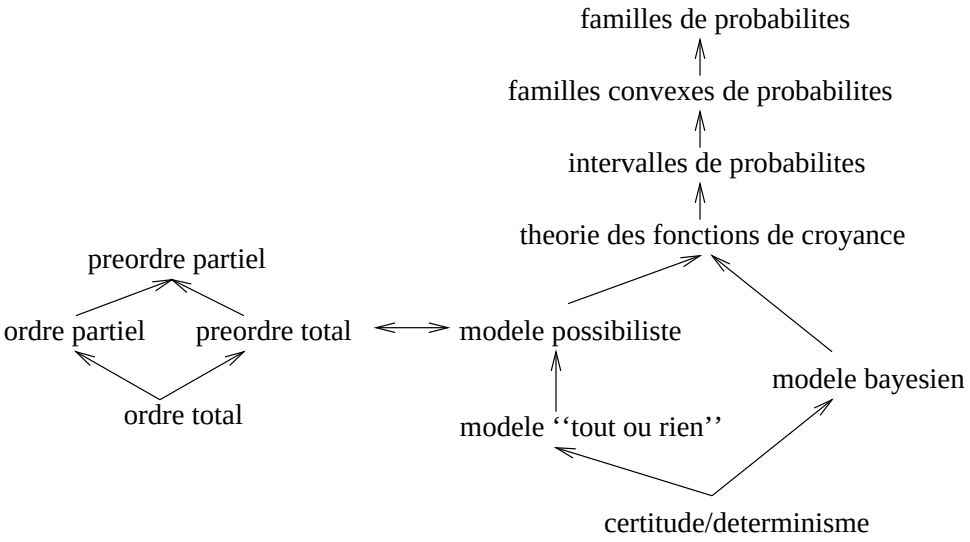

FIG. 1.1 – Modèles d'incertitude

distribution de probabilité p_{t+1} définie par

$$p_{t+1}(s') = \sum_{s \in \mathcal{S}} p_t(s).p(s'|s,a)$$

– dans le modèle possibiliste, les deux choix les plus courants sont le modèle min-max $prog_{min} : \pi_t, a \mapsto \pi_{t+1}$ avec

$$\pi_{t+1}(s') = \max_{s \in \mathcal{S}} \min(\pi_t(s), \pi(s'|s,a))$$

et le modèle produit $prog_{\times} : \pi_t, a \mapsto \pi_{t+1}$ avec

$$\pi_{t+1}(s') = \max_{s \in \mathcal{S}} \pi_t(s).\pi(s'|s,a)$$

Le modèle min-max est le plus proche des modèles ordinaux, tandis que le modèle produit correspond de façon univoque aux fonctions conditionnelles ordinales. On reviendra là-dessus au chapitre 6.

À ma connaissance, il existe très peu de travaux étudiant la progression par une action d'un état de croyance purement ordinal[18] ou d'un état de

[18]La mise à jour s'en rapproche mais seuls les effets de l'action sont ordinaux, pas l'état de croyance initial. Il se pose dans ce cas une question d'agrégation de relations d'ordres.

croyance évidentiel (voir cependant le travail récent de Guéré [112]).

Notons enfin qu'une fois fixés le modèle de l'incertitude statique et le modèle de l'incertitude dynamique, on sait comment évaluer l'incertitude associée aux *trajectoires*. Ainsi :

- dans le modèle tout-ou-rien (pour l'incertitude statique et dynamique), la donnée d'un état de croyance initial b_0, d'une suite d'actions $\langle a_0, \ldots, a_{T-1} \rangle$ et d'une fonction $evol : \mathcal{S} \times \mathcal{A} \rightarrow 2^{\mathcal{S}}$ induit une partition des trajectoires de $TRAJ_T$ en deux sous-ensembles (trajectoires possibles, trajectoires impossibles) ; une trajectoire $\tau = \langle s_1, \ldots, s_T \rangle$ est possible si et seulement si pour tout $i \leq T-1$ on a $s_{i+1} \in evol(s_i, a_i)$;
- dans le modèle bayésien (pour l'incertitude statique et dynamique), la donnée d'un état de croyance initial p_0, d'une suite d'actions $\langle a_0, \ldots, a_{T-1} \rangle$ et d'une famille de probabilités de transition $p(.|s, a)$ induit une distribution de probabilité sur les trajectoires définie par

$$p(\tau) = \prod_{t=0}^{T-1} p(s_{t+1}|s_t, a_t)$$

- dans le modèle possibiliste, la donnée d'un état de croyance initial π_0, d'une suite d'actions $\langle a_0, \ldots, a_{T-1} \rangle$ et d'une famille de possibilités de transition $\pi(.|s, a)$ induit une distribution de possibilité sur les trajectoires définie par

$$p(\tau) = \min\{\pi(s_{t+1}|s_t, a_t) \mid t = 0, \ldots, T-1\}$$

  dans le modèle min-max et par

$$p(\tau) = \prod_{t=0}^{T-1} \pi(s_{t+1}|s_t, a_t)$$

  dans le modèle produit.

1.3.3 Observabilité

Tandis qu'au paragraphe précédent on a parlé de la croyance qu'a l'agent des effets des actions sur l'état du monde (appelés aussi effets ontiques), dans celui-ci on va parler de leurs effets sur les croyances de l'agent. L'observabilité concerne la nature de ces effets (appelés aussi *effets épistémiques*).

Les actions qui n'ont que des effets épistémiques sont des *actions purement épistémiques*. (On reviendra sur les actions épistémiques et on en affinera la typologie au chapitre 5).

Les actions que l'agent décide d'exécuter sont fonction, non pas *directement* de l'état du système (à la connaissance duquel l'agent n'a pas nécessairement accès), mais de ce qu'il en aura observé *juste auparavant*. Idéalement, l'état et ce qu'on en observe coïncident, mais cette hypothèse est loin d'être la norme.

On définit alors un *espace des observations* $\mathcal{O}$, supposé fini tout au long de ce document. Les observations représentent le feedback donné par le système et sont donc une projection de l'état réel (non nécessairement observé) du système. Deux hypothèses extrêmes sont courantes :

- *observabilité totale* : l'espace des observations est identifié à celui des états ($\mathcal{O} \equiv \mathcal{S}$) : l'agent connait donc à tout instant l'état courant du système avec précision.

- *non-observabilité* : l'espace des observations est un singleton $\{o^*\}$, où o^* est une observation fictive ("observation vide") [31]. Le système ne donne aucun feedback.

En l'absence de l'une de ces deux hypothèses, on est dans la situation plus générale d'*observabilité partielle*[19], où les observations et les états sont liés par une structure de *corrélation observation-état* dont le modèle varie en fonction de deux paramètres relativement indépendants : d'une part, la nature qualitative de la relation de corrélation ("many-to-many", "one-to-many" etc.) et le modèle de l'incertitude (tout-ou-rien, bayésien, ordinal etc.).

- *corrélation tout-ou-rien "many-to-many"*[20] : l'agent dispose d'une relation de corrélation $R_C \subseteq \mathcal{O} \times \mathcal{S}$ qui traduit la compatibilité entre observations et états : $(o, s) \in R_C$ traduit le fait que s est un état compatible avec l'observation o. On impose l'hypothèse de double normalisation $\forall s, R_C^{-1}(s) \neq \emptyset$ et $\forall o, R_C^{-1}(o) \neq \emptyset$, puisqu'il doit exister au moins une observation compatible avec un état donné et *vice versa*.

[19] On reviendra en détail sur les différentes hypothèses d'observabilité au chapitre 6.

[20] Essayez donc de traduire many-to-many, one-to-many etc. en français. Il y a peut-être des expressions consacrées mais je ne les connais pas.

– *corrélation tout-ou-rien "one-to-many"* : cas particulier du précédent où R_C est telle que pour tout état, il existe une unique observation o compatible avec lui. Une observation peut alors être vue comme un ensemble non vide d'états et $\mathcal{O}$ comme une abstraction de $\mathcal{S}$.

– *corrélation bijective (one-to-one)* : à chaque observation correspond un unique état et *vice-versa*. C'est (voir plus haut) le cas extrême d'observabilité totale.

– *corrélation one-to-all* : il n'y a qu'une seule observation possible ("rien") et elle est compatible avec tous les états. C'est (voir plus haut) le cas extrême de non-observabilité.

– *corrélation bayésienne* : plutôt qu'une relation de compatibilité, l'agent dispose d'une distribution de probabilité $p(.|o)$ sur $\mathcal{S}$ pour chaque observation o. $p(s|o)$ est la probabilité que l'état courant soit s lorsque l'on a observé o[21]. Comme pour les corrélations tout-ou-rien, on peut distinguer le cas one-to-many où pour chaque état s il existe une observation unique o telle que $p(s|o) > 0$.

– *corrélation ordinale ou possibiliste* : on a cette fois une relation de préordre $\geq_o$ sur $\mathcal{S}$ (ou encore une distribution de possibilité $\pi(.|o)$) pour chaque observation o. Ce modèle est bien entendu une généralisation du modèle tout-ou-rien many-to many et on peut, comme précédemment, distinguer le cas one-to-many. Un exemple intuitif de corrélation ordinale est le modèle de la perception potentiellement erronée ("misperception") défini comme suit : $\mathcal{O}$ est isomorphe à $\mathcal{S}$ (à chaque s on a une observation $obs(s)$) mais avec misperception possible quoiqu'exceptionnelle : $\pi(s|obs(s)) = 1$ et pour tout $s' \neq s$, $\pi(s'|obs(s)) = \alpha(s, s') < 1$. $\alpha(s, s')$ peut être constant ou plus généralement tenir compte de la similarité entre états qui rend plus plausible la misperception quand les états sont difficilement distinguables.

Une fois que la nature des corrélations états-observations est connue, il reste à savoir comment déterminer un nouvel état de croyance à partir

[21] Certains modèles, en particulier les POMDP, supposent plutôt qu'on se donne initialement les probabilité $p(o|s)$ – mais cela revient au même, puisque le modèle permet également de calculer $p(s)$ pour tout s, ce qui ce qui permet de calculer $p(s|o)$ par la formule de Bayes.

d'un état attendu (résultant d'un ancien état de croyance et d'une action) et d'une observation (voir le schéma 1.2, réminiscent des schémas de systèmes en boucle fermée comme on en voit en automatique).

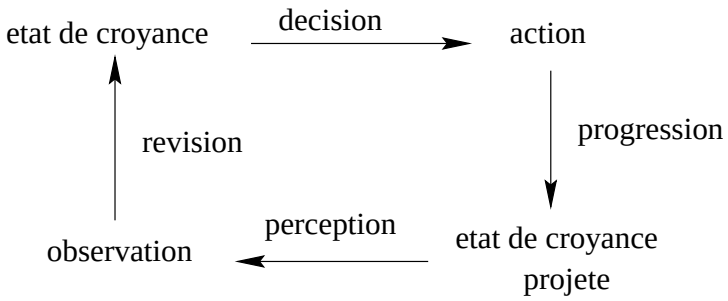

FIG. 1.2 – boucle de rétroaction

Ce dernier processus est appelé *révision*, ou encore, dans certains modèles, *conditionnement*; il sera noté $*$. L'opération standard de révision pour des états de croyance bayésiens est le conditionnement probabiliste (règle de Bayes), à savoir : pour toute probabilité p et toute observation o telle que $P(o) \neq 0$,

$$(p * o)(s) = p(s|o) = \frac{p(o|s)p(s)}{\sum_{s' \in S} p(o|s')p(s')}$$

Pour le modèle tout-ou-rien, l'opération standard est l'intersection, à savoir, pour tout $b \subseteq \mathcal{S}$, et toute observation o telle que $b \cap R_C(o) \neq \emptyset$,

$$(b * o)(s) = b \cap R_C(o)$$

On trouvera dans [77] plusieurs opérateurs de conditionnement pour des états de croyance possibilistes.

Les opérateurs de révision "standard" que nous venons de donner sont volontairement non définis lorsque l'observation est en conflit total avec l'état de croyance. La question de leur définition dans ce cas est un problème délicat, connu sous le nom de *révision des croyances*. Il sera abordé plusieurs fois dans la suite de ce document (chapitres 2 et 5).

1.3.4 Le non-déterminisme à la lumière de l'observabilité

Nous abordons maintenant une question délicate : *le non-déterminisme est-il un paramètre ontique ou épistémique ?*

On peut reformuler la question ainsi : le fait qu'une action soit déterministe ou non est-elle une propriété du système ou des croyances de l'agent ? Nous allons tenter d'y répondre brièvement (et sans entrer dans des considérations philosophiques qui nous dépassent largement). Une action peut être qualifiée d'*objectivement non-déterministe* si elle est non-déterministe pour n'importe quel agent rationnel "réaliste" ; si, de plus, la croyance sur les états ultérieurs est la même pour tous ces agents, on dira qu'elle est *fortement objectivement non-déterministe*. Ces agents "raisonnablement concevables" sont des êtres humains individuels ou en collectivité, ou des robots munis de capteurs : il paraît donc naturel de dire que des actions comme "lancer une pièce", "générer un nombre aléatoire" ou "tenter de poser le cube A sur le cube B avec un bras manipulateur imparfait" sont objectivement non-déterministes. En voici une qui ne l'est pas. L'ascenseur de mon laboratoire (ce dernier a 4 étages) ne peut pas être localisé de l'extérieur (on n'a aucun moyen de savoir à quel étage il est en l'observant – on peut seulement savoir s'il est en déplacement, grâce au voyant "occupé") mais il le peut si l'on se trouve à l'intérieur. L'action "faire se déplacer l'ascenseur vers le troisième étage" (considérée comme une action identique pour un agent à l'intérieur de l'ascenseur et pour un agent au troisième étage et à l'extérieur de l'ascenseur – au niveau des effets il se passe de toute façon la même chose dans les deux cas) est non-déterministe pour un agent qui se trouve au troisième étage à l'extérieur de l'ascenseur (les effets possibles sont : la porte s'ouvre immédiatement, ou au bout de 1, 2, ou 3 unités de temps, en supposant que l'ascenseur parcourt une unité de temps par étage), mais elle est déterministe pour l'agent qui se trouve à l'intérieur, ou encore pour l'agent qui vient de monter les escaliers et de voir la porte de l'ascenseur s'ouvrir au second étage, et qui n'a pas vu le voyant "occupé" se rallumer depuis.

Cette distinction entre non-déterminisme subjectif ou objectif n'est pas sans rapport avec la distinction entre probabilités objectives et subjectives. Les probabilités objectives, obtenues comme des fréquences limites de réalisation d'événements, sont caractéristiques du système lui-même, et sont en principe les mêmes pour tout agent extérieur à celui-ci. Si l'on considère que l'action "lancer la pièce" a pu être expérimentée autant de fois que l'on veut (en théorie, une infinité de fois), on peut aisément concevoir que les probabilités associées à ses effets sont objectives. Une telle action est alors *objectivement stochastique*. Cela suppose, évidemment, que tout agent concevable aura été capable de réaliser ce grand nombre d'expériences sur la pièce lui permettant de déterminer (approximativement bien entendu) les probabilités associées aux effets. C'est la même chose pour "tenter de poser

le cube A sur le cube B". C'est encore la même chose pour le problème de l'ascenseur si l'on se restreint à l'ensemble des agents qui sont situés à l'extérieur de l'ascenseur. Au contraire, une probabilité subjective [204] est la propriété exclusive d'un individu (j'évite ici le terme "agent" parce que je ne sais pas très bien ce que signifierait la probabilité *subjective* d'un robot), et si deux individus ont la même, c'est une coïncidence. L'action "acheter 10000 actions de *Titi et Léa S.A.*", qui a pour effets possibles un enrichissement ou un appauvrissement de l'individu dès le lendemain, si elle peut aisément être considérée comme objectivement non-déterministe, ne l'est pas *fortement*; de surcroît, même s'il peut être raisonnable de modéliser les croyances (du point de vue d'un agent donné) sur ses effets par une distribution de probabilité, l'action n'est certainement pas "objectivement stochastique" au sens où, si l'on choisit de modéliser l'incertitude sur les effets de l'action par une distribution de probabilité pour un agent donné, c'est un choix, qui est contestable (et qui implique que les probabilités sont subjectives) : le choix d'autres modèles (par exemple un modèle ordinal, ou la théorie de l'évidence) serait peut-être également adéquat.

Revenons au non-déterminisme. On peut avancer que *la distinction entre non-déterminisme ontologique et non-déterminisme épistémique est avant tout une question d'observabilité*. Considérons à nouveau le problème de l'ascenseur, et soient les actions $a_{j,k}$ "faire se déplacer l'ascenseur vers le j$^{\text{ème}}$ étage et attendre k unités de temps". Vues par un agent de l'intérieur, ce sont des actions déterministes à effets conditionnels; par exemple, $a_{3,1}$ est traduite par la fonction de transition déterministe suivante : $res(s_3, a_{3,1}) = res(s_2, a_{3,1}) = res(s_4, a_{3,1}) = s_3$, $res(s_1, a_{3,1}) = s_2$, $res(s_0, a_{3,1}) = s_1$. Un agent de l'extérieur peut tout-à-fait considérer la même fonction de transition, *mais l'état initial est cette fois non observable*, ce qui implique que l'agent sait seulement que l'ensemble des résultats possibles est $\{s_1, s_2, s_3\}$ (au moins, s_0 et s_4 sont exclus – c'est mieux que rien) – éventuellement, il connaitra la probabilité objective sur la position courante de l'ascenseur, pourvu qu'il ait expérimenté celui-ci ou qu'on lui ait fourni des statistiques fiables (ainsi, si l'état de croyance b_0 sur la position initiale de l'ascenseur est la distribution de probabilité uniforme sur $\{s_0, s_1, s_2, s_3, s_4\}$ alors $p(s_3|b_0, a_{3,1}) = \frac{3}{5}$ et $p(s_1|b_0, a_{3,1}) = p(s_2|b_0, a_{3,1}) = \frac{1}{5}$). Plus généralement, on sait que toute action non-déterministe peut être modélisée comme une action déterministe pourvu que l'on introduise une variable supplémentaire non-observable (et *vice versa*). Cela n'a évidemment pas toujours un sens, intuitivement parlant : coder "jeter une pièce" comme une action déterministe en introduisant la variable non-observable x telle que

$p(pile|x = vrai, JeterPiece) = 1$, $p(face|x = faux, JeterPiece) = 1$, et $p(x = vrai) = p(x = faux) = \frac{1}{2}$, où x a la valeur vrai signifie que le prochain jeter de pièce va la faire tomber sur pile, n'est pas très satisfaisant intuitivement (mais l'est techniquement). Notons enfin que l'introduction d'une variable supplémentaire multiplie la taille de l'espace d'états par la taille du domaine de cette variable (c'est-à-dire le nombre d'effets possibles de l'action) : ainsi, dans l'exemple précédent, d'un espace d'états à deux éléments $S = \{pile, face\}$ nous passons à un espace d'états à 4 éléments $S' = \{(pile, x = vrai), (pile, x = faux), (face, x = vrai), (face, x = faux)\}$. Nous reviendrons sur ce genre de problèmes lorsque nous aborderons les langages de représentation (en Section 1.4). Pour conclure ce paragraphe, on voit qu'il existe un phénomène de vases communicants entre déterminisme avec effets conditionnels et non-déterminisme (qui justifie a posteriori le choix de Sandewall de considérer un paramètre global "effets alternatifs" qui englobe les deux), via le passage de variables du statut observable au statut non-observable ou vice-versa. En corollaire, on pourrait dire que le non-déterminisme d'une action est objectif si les variables observables "pertinentes" pour l'action considérée sont les mêmes pour tous les agents.

1.4 Paramètres préférentiels

Dans l'optique d'un processus de prise de décision, ce n'est pas tout d'exprimer la dynamique du système et les croyances de l'agent, il faut aussi se donner les objectifs que l'agent cherche à atteindre[22]. On emploiera plusieurs termes qui se réfèrent tous aux préférences de l'agent : *préférences*, *objectifs*, *buts*, *désirs*. Les termes *structure préférentielle*, *utilité*, *disutilité*, *satisfaction*, *récompense*, *coût*, *pénalité* se réfèrent quant à eux à la mesure des préférences.

Dans cette section on élabore une taxonomie des types de préférences sur les effets du processus de décision, c'est-à-dire les trajectoires.

De façon générale, les préférences de l'agent portent sur l'ensemble de la trajectoire effectivement réalisée par le système. Cependant, dans de nombreux cas, elles ne portent que sur l'état final s_T du système, alors noté plus simplement x (voir le paragraphe 1.2.2) ; il est évident que cette hypothèse n'a de sens que lorsque l'horizon est fini ou, éventuellement, indéfini – mais pas infini. Nous allons d'abord élaborer une typologie des préférences lorsque

[22]Il va de soi que lorsque l'agent est artificiel, "ses" objectifs ne lui sont pas propres mais lui ont été transmis par l'agent qui l'a programmé.

cette hypothèse simplificatrice est faite.

1.4.1 Préférences statiques

Les préférences de l'agent portent sur l'ensemble des conséquences $\mathcal{X}$. Selon la nature de ces préférences, la *structure préférentielle* $\mathcal{G}$ c'est-à-dire l'objet mathématique décrivant ces préférences, varie.

- *préférences "tout ou rien"* : $\mathcal{G}$ est une *partition de* $\mathcal{X}$ *en deux sous-ensembles*. Lorsque l'agent exprime un but, ou un ensemble indivisible de buts, à satisfaire, il partitionne l'ensemble des états en deux classes : les *états gagnants* (dits encore *états buts*, *états satisfaisants*, ou *états objectifs*), et les *états perdants*, dits encore états non-buts, *états non satisfaisants*, etc.[23].
  Les préférences tout-ou-rien sont bien peu expressives puisqu'elle ne permettent pas d'exprimer des intensités de préférence, c'est-à-dire une forme de gradualité dans la satisfaction des objectifs. Or, une grande partie des processus de décision utilise une notion de préférence qui, justement, est graduelle.

- *préférences cardinales* (ou *quantitatives*, *évaluationnelles*) : la structure préférentielle est une *fonction d'utilité*. Une fonction d'utilité u est une fonction de $\mathcal{X}$ dans un ensemble Val qui est en général une partie de $\mathbb{R}$ (parfois finie), ou un produit cartésien d'échelles numériques ($\mathbb{R}^n$, etc.). Le choix exact de Val dépend de la nécessité d'avoir ou non des éléments de Val représentant respectivement l'indifférence absolue (ou neutralité), la satisfaction maximale (existence d'un maximum), et la dissatisfaction maximale (existence d'un minimum). Voici quelques choix fréquents pour Val :
  - $Val = \mathbb{R}$ permet d'exprimer l'indifférence (représentée par 0) et permet donc par-là d'exprimer la satisfaction positive (utilités positives) et la satisfaction négative ou dissatisfaction (utilités négatives) : on dit dans ce cas que l'échelle Val est *bipolaire* ;
  - $Val = [0, 1]$, où 0 et 1 expriment respectivement la dissatisfaction et la satisfaction maximales ;
  - $Val = \overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$, bien qu'isomorphe à $[0, 1]$, est, intuitivement parlant, légèrement différent puisque sa bipolarité ne fait pas de

[23] Nous favorisons la terminologie gagnant/perdant parce que c'est la seule qui évite l'utilisation de "non-", et parce que la terminologie fréquente buts/non-buts est ambiguë : les "buts" partitionnent les états en "états-buts" et "états non-buts".

doute ;
- on utilisera parfois $Val = \mathbb{R}^-$ (échelle monopolaire sans minimum) et $Val = \overline{\mathbb{R}}^- = \mathbb{R}^- \cup \{-\infty\}$ (échelle monopolaire avec minimum). Ces échelles d'utilités négatives ou nulles seront appelées échelles de *pénalités*.
- $Val = \mathbb{R}^n$ convient à des évaluations selon plusieurs critères sans se contraindre à faire le choix d'une fonction d'agrégation.

Une fonction d'utilité u induit une relation de préordre $\geq_u$ définie par $s \geq_u s'$ ssi $u(s) \geq u(s')$; et, par conséquent, $s >_u s'$ si et seulement si $u(s) > u(s')$; $s \sim_u s'$ ssi $u(s) = u(s')$; s, s' sont incomparables ssi $u(s) \not\geq u(s')$ et $u(s') \not\geq u(s)$.

- *préférences relationnelles* : la structure de base $\mathcal{G}$ est une relation binaire $R \subseteq \mathcal{S}^2$, la plupart du temps supposée *transitive* : on parlera alors de *préférences ordinales*. Attention cependant à la terminologie : R n'est en général qu'un préordre partiel. Dans ce cas, R sera notée $\geq$. On note usuellement $x > x'$ pour $x \geq x'$ et non $(x' \geq x)$ (préférence stricte de x' sur x). On exige parfois que $\geq$ soit de plus total (deux candidats doivent toujours pouvoir être comparés) – c'est alors un préordre total. Une structure de préférence ordinale ne met donc en évidence que des classes d'états de niveaux de satisfaction différents, sans qu'on puisse quantifier les intensités de préférence (on a donc la gradualité sans l'intensité). Notons qu'une structure de préférence avec incomparabilité peut être induite par plusieurs fonctions d'utilité incommensurables : ainsi, si l'on se donne n fonctions d'utilité sur $\mathcal{S}$, *l'ordre de Pareto* induit par $u_1, \ldots, u_n$, défini par $s \geq_{u_1,\ldots,u_n} s'$ si et seulement si $u_i(s) \geq u_i(s')$ est vrai pour chaque i, est un préordre partiel et induit donc une structure préférentielle avec incomparabilités. Enfin, parfois on requiert non seulement que l'incomparabilité, mais aussi l'indifférence, soient exclues ; R est alors antisymétrique (pas d'indifférence : $x \sim x'$ implique $x = x'$) ; ce choix de modèle est toutefois assez restrictif.

- *préférences floues* : $\mathcal{G}$ est une relation de préférence floue, c'est-à-dire une fonction $\mu_P : \mathcal{S} \times \mathcal{S} \to [0, 1]$. $\mu_P(s, s')$ exprime l'intensité de préférence de s à s'. Ce modèle recouvre en fait tous les précédents. La préférence n'est plus une notion absolue, mais *relative* ; une préférence floue permet d'exprimer des intensités de préférences sans avoir recours à une fonction d'utilité. On requiert habituellement certaines

hypothèses sur la fonction μ_P que nous n'allons pas écrire ici puisqu'on ne rencontrera plus ce type de préférences dans la suite du mémoire.

Dans la suite du document nous nous précupperons de préférences évaluationnelles avec une échelle numérique (en général $\mathbb{R}$ ou $\overline{\mathbb{R}}$) et de préférences relationnelles binaires transitives, c'est-à-dire de *structures ordinales* (relations de préordre – totales ou non).

On peut se poser la question suivante : du strict point de vue de l'IA[24], y a-t-il une différence essentielle entre une structure de préférence numérique et une structure de préférence ordinale ? Esquissons une réponse : s'il s'agit juste de choisir un élément non-dominé dans la structure de préférence, pas vraiment ; si, par contre, il s'agit de combiner entre elles des structures de préférences (voir chapitre 4) ou de combiner des structures de préférences avec des états de croyance (voir en Section 1.5), alors oui (voir [70] pour une discussion poussée sur l'ordinalité en décision dans l'incertain). La façon dont les structures de préférences peuvent être calculées comme l'agrégation de structures de préférences élémentaires, dans deux cas bien connus (décision de groupe et décision multi-critère) sera évoquée au chapitre 4.

1.4.2 Préférences dynamiques

Cessons à présent de supposer que les préférences de l'agent ne dépendent que de l'état final du système. Elles peuvent alors dépendre de tous les états s_0, ..., s_T rencontrés par le système ainsi que des actions a_0, ..., a_{T-1} entreprises. La structure préférentielle va donc porter sur les *trajectoires*. La taxonomie précédente (structures tout ou rien, ordinales ou quantitatives) s'applique aussi bien aux préférences dynamiques qu'aux préférences statiques. Nous allons plutôt distinguer différentes hypothèses relatives à *ce qui compte* pour le calcul des préférences dynamiques.

- *état final seulement* : on se ramène au paragraphe précédent ;
- *trajectoire d'états seulement* : les actions ne comptent pas.
- *état final et trajectoire d'actions seulement* : les états intermédiaires ne comptent pas.
- cas général : toute la trajectoire compte.

Allons maintenant plus avant dans la discussion dans le cas (de loin le plus courant pour les environnements dynamiques) où les préférences sont

[24] Bien entendu, l'économie cognitive, la psychologie cognitive et la philosophie accordent une importance significative à cette question.

décrites par des fonctions d'utilité sur $\mathbb{R}$. Il s'agit de déterminer une fonction d'utilité sur les trajectoires $\tau \mapsto u(\tau) \in \mathbb{R}$, à partir de données plus élémentaires comme les *utilités instantanées* qui associent une utilité à un état à un instant donné et comme les coûts des actions.

Commençons par le cas où l'horizon est fini.

Voici l'hypothèse la plus fréquente : la fonction d'utilité $u : TRAJ_{\mathcal{H}} \to \mathbb{R}$ est *temporellement décomposable* s'il existe une fonction $F : \mathbb{R}^{\mathcal{H}+1} \to \mathbb{R}$, des fonctions $u_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ pour $t = 0, ..., T-1$ et une fonction $u_T : \mathcal{S} \to \mathbb{R}$ telles que, pour toute trajectoire objective $\tau = \langle\langle s_0, a_0\rangle, \ldots, \langle s_{T-1}, a_{T-1}\rangle, s_T\rangle$ on a

$$u(\tau) = F(u_0(s_0, a_0), \ldots, u_{T-1}(s_{T-1}, a_{T-1}), u_T(s_T))$$

Le choix habituel pour F est la somme ; on parlera alors d'utilités *temporellement décomposables et additives*, ce qui signifie que l'utilité d'une trajectoire est la somme des utilités instantanées associées à chacun des états et des actions de la trajectoire. Une autre hypothèse fréquente est que les utilités instantanées u_i sont *stationnaires*, c'est-à-dire indépendantes de t. Ces utilités instantanées s'écriront souvent sous la forme $u(s, a) = R(s) - C(a, s)$, c'est-à-dire comme la différence entre la *récompense* $R(s)$ associée à l'état s et le coût $C(a, s)$ de l'action a dans l'état s[25].

Le modèle temporellement décomposable et additif recouvre bien des cas particuliers, entre autres le modèle, courant en planification, où les objectifs de l'agent sont simplement d'atteindre un état but : on a alors $R_t(s) = 0$ pour tout $t < T$ et $R_T(s) = 1$ si s est un état but, $R_T(s) = 0$ sinon. Ce modèle, tel qu'il est écrit, n'est pas stationnaire mais on peut le rendre facilement stationnaire en modélisant la dynamique du système de telle façon qu'après avoir atteint un état but, le système entre dans un état absorbant d'utilité nulle[26].

Passons maintenant au cas où l'horizon est infini.

[25] Il n'a pas échappé au lecteur attentif que cette formulation est incorrecte pour ce qui est de l'instant T ; mais il ne lui échappera pas non plus qu'il suffit de considérer une action fictive entreprise systématiquement à T (et de coût nul) pour que les choses rentrent dans l'ordre.

[26] On peut raffiner ce modèle en minimisant le coût total des actions entreprises : l'utilité d'un état but est alors un nombre positif suffisamment grand pour être toujours supérieur à la somme de toute somme de T coûts d'actions.

L'additivité simple n'est en général pas adaptée au calcul de l'utilité d'une trajectoire, puisque cette utilité pourrait diverger vers l'infini. En général, on introduit un *facteur d'atténuation* $\rho \in]0,1[$, et l'utilité de $\tau = \langle\langle s_i, a_i\rangle, i \in \mathbb{N}\rangle$ est alors donnée par $u_d(\tau) = \sum_{i\in\mathbb{N}} \rho^i u(s_i, a_i)$. Une alternative consiste à définir $u_m(\tau) = \lim_{t\longrightarrow+\infty} \frac{1}{t} \sum_{i=0}^{t} u(s_i, a_i)$ (*critère de la moyenne*)[27].

1.5 Paramètres épistémico-préférentiels

Jusqu'à présent on a vu comment exprimer des préférences entre états ou entre trajectoires. Dans un environnement sans incertitude (l'état initial étant connu et les actions déterministes), une suite d'actions détermine une trajectoire unique et on peut alors comparer deux suites d'actions simplement en comparant les deux trajectoires qu'elles induisent ; en d'autres termes, la structure préférentielle sur les trajectoires induit de façon évidente une structure préférentielle sur les plans d'actions. Dans l'hypothèse où l'environnement n'est pas sans incertitude, ce passage ne tient plus, puisque plusieurs trajectoires différentes peuvent être obtenues par l'exécution d'un plan d'actions. Pour comparer des plans d'actions, il faut par conséquent intégrer les préférences sur les trajectoires et les croyances sur l'état initial du système et les effets des actions. Ce problème est celui de la *décision dans l'incertain*.

Comme au paragraphe précédent, nous allons d'abord examiner le cas où $T = 1$. On supposera que l'état initial est connu et les actions non-déterministes (en vertu du principe des vases communicants entre non-déterminisme et effets conditionnels évoqué précédemment, on sait que cela n'induit pas de perte de généralité).

1.5.1 Utilité espérée (modèle probabiliste de base)

On suppose dans ce paragraphe que les actions sont stochastiques et la structure préférentielle numérique. Le critère de l'*utilité espérée* consiste à évaluer la valeur d'une action par l'espérance mathématique de l'utilité de la conséquence obtenue :

$$V_{s_0}(a) = \sum_{s_1\in\mathcal{S}} p(s_1|s_0, a)u(s_1)$$

[27] $\mathcal{S}$ et $\mathcal{A}$ étant fini, u est bornée, d'où le convergence de $u_d(\tau)$ et de $u_m(\tau)$ vers des limites finies.

Une action a est optimale (pour l'agent, bien entendu) si et seulement si son utilité espérée est maximale. On parle alors de *décision sous risque* si les probabilités sont objectives, de d*décision dans l'incertain* sinon. Le modèle de l'utilité espérée a été justifié axiomatiquement par von Neumann et Morgenstern [218] dans le cas où le non-déterminisme des actions est objectif (probabilités de transition objectives, indépendantes de l'agent) et par Savage [204] dans le cas où il est subjectif (nous n'allons pas entrer dans les détails).

1.5.2 Généralisations du critère de l'utilité espérée

Le modèle probabiliste de base, fondé sur le critère de l'utilité espérée, a souvent été mis en défaut (notamment à l'aide de paradoxes, dont les deux plus connus sont dûs à Allais et à Ellsberg). Les limitations du critère de l'utilité espérée ainsi mises en évidence ont conduit à deux types de travaux : des *généralisations*, qui recouvrent l'utilité espérée comme cas particulier, et des *modèles alternatifs*.

Je commence ici par ses généralisations. Je ne souhaite pas entrer ici dans le détail des différents modèles ; je me borne à citer les plus connus d'entre eux :

- la décision fondée sur les fonctions de croyance [128, 129] ;
- la décision fondée sur l'intégrale de Choquet [108, 208] et plus généralement les critères d'utilité dépendante du rang.

1.5.3 Modèle possibiliste (ordinal avec commensurabilité)

Je passe maintenant bièvement en revue deux classes de modèles alternatifs à l'utilité espérée, en commençant par le modèle possibiliste, qui consiste à faire le choix de structures ordinales à la fois pour l'incertain et les préférences, mais avec *commensurabilité* des deux structures.

On suppose dans ce paragraphe que la dynamique du système est représentée par des fonctions de transitions possibilistes, et que la structure préférentielle est numérique. Sans perte de généralité, on suppose que l'échelle U des valeurs d'utilité est $[0, 1]$ (si elle ne l'est pas initialement, cela suppose un changement d'échelle, voire la transformation d'une relation de préordre total en une fonction u à valeurs sur $[0, 1]$). Cette étape, parfois délicate, vise à rendre l'échelle d'incertitude et l'échelle de préférences *commensurables*.

Le critère de l'utilité ordinale de Dubois et Prade [79] consiste alors à calculer les deux quantités

$$V_P(a) = \min_{s_1 \in \mathcal{S}} \max(1 - \pi(s_1|s_0, a), u(s_1))$$

$$V_O(a) = \max_{s_1 \in \mathcal{S}} \min(\pi(s_1|s_0, a), u(s_1))$$

$V_P(a)$ et $V_O(a)$ sont appelées respectivement *utilité pessimiste* et *utilité optimiste* de a. Ces deux utilités, qui sont des contreparties possibilistes de l'utilité espérée, sont ensuite utilisées pour classer les actions (plusieurs choix sont possibles, le plus recommandé étant de classer en priorité selon l'utilité pessimiste et de n'utiliser l'utilité optimiste que pour raffiner le classement). Le critère pessimiste est une généralisation du *critère de Wald* [220].

On peut raffiner ces critères en remplaçant les opérateurs *min* et *max* par respectivement *leximin* et *leximax*, ou encore les généraliser (tout en restant dans un modèle ordinal avec commensurabilité) en remplaçant V_P et V_O par des intégrales de Sugeno [198, 80].

1.5.4 Modèles purement ordinaux

Lorsque les données initiales du processus sont purement ordinales, on peut soit se ramener arbitrairement à des structures quantitatives commensurables, soit refuser cette commensurabilisation. Dans ce dernier cas de figure, les choix possibles sont toutefois assez limités. Soient $\geq^C_{a,s_0}$ et $\geq^P$ les relations sur S correspondant respectivement aux croyances sur le résultat de a dans s_0, et aux préférences de l'agent. Boutilier [26] propose de comparer deux actions par la relation

$$a \geq_B b \text{ ssi } \forall s \in Max(\geq^C_{a,s_0}, \mathcal{S})\ \exists s' \in Max(\geq^C_{b,s_0}, \mathcal{S}) \text{ tel que } s \geq_P s'$$

c'est-à-dire que les actions sont comparées selon les pires de leurs résultats les plus plausibles (ce qui est immédiatement représentable par le critère V_P du paragraphe précédent). Dubois, Fargier, Perny et Prade [72, 100], quant à eux, préfèrent une action a à une action b si l'ensemble des états où a donne un meilleur résultat que b est plus plausible que l'ensemble des états où b donne un meilleur résultat que a :

$$a \geq_{DFPP} b \text{ ssi } \{s \mid res(s,a) >^P res(s,b)\} \geq^C \{s \mid res(s,b) >^P res(s,a)\}$$

1.5.5 Généralisation au cas multi-étapes

Lorsque $T > 1$, les choses sont similaires, à la différence qu'on ne compare plus des actions individuelles mais des *politiques d'action*. Sous les hypothèses de markovianité et de stationnarité, une *politique* $\pi : \mathcal{O} \times \mathcal{H} \longrightarrow \mathcal{A}$ (voir le paragraphe 1.7.1) associe à chaque instant t et chaque observation o une action a.

De la même façon qu'une action effectuée dans un état de croyance initial donné a plusieurs résultats possibles (éventuellement gradués en fonction de leur plausibilité), la mise en oeuvre d'une *politique*, dans un état de croyance initial, induit plusieurs *trajectoires* possibles, éventuellement graduées en fonction de leur plausibilité. Il suffit ensuite de généraliser les critères précédents en tenant compte cette fois des relations entre trajectoires.

Ainsi, par exemple, dans le cas où $\mathcal{H}$ est fini, où l'incertitude probabiliste et les préférences sont modélisées par une fonction d'utilité sur $TRAJ$, l'utilité espérée d'une politique est définie par

$$V(\pi) = \sum_{\tau \in (\mathcal{S}\times\mathcal{A})^T \times \mathcal{S}} p(\tau|\pi)u(\tau)$$

où, si $\tau = \langle\langle s_0, a_0\rangle, \ldots, \langle s_{T_1}, a_{T_1}\rangle, s_T\rangle$ alors $p(\tau|\pi) = p_0(s_0).\Pi_{t=0}^{T-1} p(s_{t+1}|s_t, a_t)$.

1.6 Langages de représentation

La taxonomie que nous avons élaborée jusqu'à présent ne concerne que les modèles (du système et de l'agent). Or, les données d'un processus de décision (connaissances et préférences évoquées jusqu'ici) doivent être codées dans un langage donné pour pouvoir être ensuite traitées informatiquement.

Dans de nombreux processus, le système est composé d'un ensemble de variables plus ou moins indépendantes, chacune d'entre elles prenant ses valeurs dans un domaine donné. Il est alors clair que l'espace d'états, qui est le produit cartésien des domaines des variables, est sujet à une explosion combinatoire. Il n'y a pas besoin de beaucoup de variables ni de gros domaines pour que le nombre d'états du système soit rédhibitoire. C'est parfois la même chose avec les actions, lorsqu'une action est composée de plusieurs micro-actions consistant chacune à choisir une valeur pour une variable de décision donnée.

Lorsque la taille de l'espace d'états et/ou de l'espace d'actions est prohibitive, il n'est alors pas question de coder l'input du processus en considérant les actions et les états de manière explicite : il suffit de voir que l'expression, par exemple, de probabilités de transitions par des matrices comme on le ferait en recherche opérationnelle, nécessite une taille des données en $\mathcal{O}(|S|^2.|A|)$, pour se convaincre de le nécessité d'élaborer des modes de représentation moins coûteux. Ceci doit être possible, puisque les données de processus apparemment très complexes si l'on regarde leur combinatoire sont exprimés en fait de façon très simple et très concise par des agents humains (on en verra des exemples plus loin dans ce document).

On va pour cela exploiter la structure interne des espaces d'états et d'actions sous-jacentes au processus. Cette structure, ce n'est rien d'autre que la décomposition de $\mathcal{S}$ et de $\mathcal{A}$ en produits cartésiens de domaines de variables. D'où la définition qui suit.

Un *langage de représentation structuré* pour un processus de prise de décision consiste en la donnée d'un espace d'états et/ou d'actions décomposés en produit cartésiens, et d'une méthode permettant d'exprimer des croyances et des préférences de manière implicite, au moyen de "morceaux de croyance ou de préférence" exprimés *localement*, c'est-à-dire ne se référant qu'aux variables réellement concernées par ce morceau de croyance ou de préférence. Ces langages seront donc non seulement *structurés*, *concis*, *compacts*, *économiques* mais aussi *implicites*, *locaux* et *proches de l'intuition*, c'est-à-dire de la façon dont l'agent humain les exprimerait en langage naturel. Au contraire, le langage qui consiste à ne pas décomposer $\mathcal{S}$ et $\mathcal{A}$ et à exprimer les données en listant explicitement les états et les actions dans des tables (il y a des cas où on ne peut faire autrement) est appelé *explicite* ou *plat*.

La qualité d'un langage de représentation se mesurera donc à ces trois paramètres :

1. compacité (économie de représentation) ;
2. lisibilité / localité ;
3. compatibilité avec les procédures de calcul.

Puisque les données d'un processus de décision se décomposent en trois ensembles de données : croyances statiques (sur le système), croyances sur la dynamique du système et préférences, il s'agit maintenant de passer en revue les différentes classes de langages pour chacun de ces types de données.

1.6.1 Croyances statiques

Les croyances statiques concernent aussi bien l'état initial du monde que les corrélations entre variables d'état ou entre états et observations.

État initial

La représentation plate des croyances sur l'état initial consiste simplement en la donnée explicite de l'état de croyance (par exemple, pour une distribution de probabilité p_0, la liste $\{\langle s, p_0(s)\rangle | s \in \mathcal{S}\}$). Passons maintenant rapidement en revue des classes de langages structurés.

Un *langage structuré* pour la description des croyances sur l'état initial consiste en la donnée d'un ensemble de variables d'état $V = \{v_1, ..., v_n\}$, chacune ayant un domaine de valeurs D_i. L'espace d'états est alors $\mathcal{S} = D_1 \times \ldots \times D_n$ ou éventuellement un sous-ensemble de $D_1 \times \ldots \times D_n$.

On appellera *problème de prise de décision* un couple $\langle \mathcal{P}, \mathcal{R} \rangle$ où $\mathcal{P}$ est un *processus de prise de décision* et $\mathcal{R}$ la représentation de son input dans un langage donné. Ainsi, un même processus de décision donnera naissance à *deux problèmes distincts* selon, par exemple, que son input est donné sous forme plate ou structurée. La raison de cette distinction est que les procédures de calcul aussi bien que la complexité de la recherche d'une politique d'actions vont être fonctions non seulement de la classe de processus à laquelle appartient $\mathcal{P}$ mais aussi à la façon dont il est représenté.

Nous allons maintenant examiner les *paramètres représentationnels* d'un problème de prise de décision. Au contraire des paramètres ontologiques, épistémiques ou préférentiels, ces paramètres sont attachés au problème $\langle \mathcal{P}, \mathcal{R} \rangle$ et non pas seulement au processus $\mathcal{P}$.

Le paramètre essentiel lié à la représentation concerne les *dépendances entre variables d'état*. Cette question n'a pas de sens lorsque la représentation est plate. De surcroît, la présence ou non de dépendances statiques entre variables (paramètre **D** chez Sandewall) dépend du type de représentation choisie (ce qui montre bien que c'est un paramètre lié au processus *et* à sa représentation).

– *dépendances logiques* (ou contraintes d'intégrité).
  C'est le type de dépendance le plus simple : les dépendances entre va-

riables sont codées par des formules propositionnelles. Par exemple, $\{a \wedge b \Rightarrow c, \neg a \Rightarrow \neg c\}$ exprime des dépendances entre a, b et c. On reviendra au paragraphe 2.3 sur les relations de dépendance et d'indépendance en logique propositionnelle.

– *dépendances probabilistes* : il y a deux façons de "probabiliser" une contrainte logique représentée par une formule propositionnelle.

(1) Prenons l'exemple précédent $\{a \wedge b \Rightarrow c, \neg a \Rightarrow \neg c\}$. Cette formule propositionnelle peut être vue comme une définition partielle de c à partir de a et b (partielle car dans le contexte $a \wedge \neg b$ on ne peut pas déduire la valeur de c)[28]. Probabilisée, cette dépendance devient une table de probabilité conditionnelle où l'on spécifie $P(c|a \wedge b)$, $P(c|a \wedge \neg b)$, $P(c|\neg a \wedge b)$, $P(c|\neg a \wedge \neg b)$. Un ensemble de telles tables de probabilités conditionnelles associé à un réseau sur l'ensemble de variables représentant la structure de dépendance est un *réseau bayésien* [190]. Dans un réseau bayésien, il existe une probabilité conditionnelle *pour chaque variable* qui donne la probabilité de cette variable sachant tout n-uplet de valeurs pour les *parents* de cette variable. La relation "être parent de" codée dans le réseau est acyclique. Les variables qui n'ont pas de parents (il en existe au moins une) ont une probabilité a priori. Ces conditions permettent de dire qu'un réseau bayésien est une *représentation compacte d'une distribution de probabilité*. Bien entendu, plus les variables sont indépendantes, plus la représentation est compacte.

(2) Revenons à l'exemple. Une deuxième façon de probabiliser $\{a \wedge b \Rightarrow c, \neg a \Rightarrow \neg c\}$ est de spécifier des contraintes sur la probabilité des formules $a \wedge b \Rightarrow c$ et $\neg a \Rightarrow \neg c$, par exemple $Pr(a \wedge b \Rightarrow c) = \alpha$ et $Pr(\neg a \Rightarrow \neg c) = \beta$. Ces contraintes sont la représentation compacte non plus d'*une* distribution de probabilité mais *d'une famille* (éventuellement vide, éventuellement réduite à un singleton) de distributions de probabilité. C'est le principe de la *logique probabiliste* [187].

– *dépendances ordinales* : les dépendances entre variables sont souvent mieux modélisées par une structure d'incertitude ordinale. C'est en particulier le cas de règles d'héritage de propriétés ou de règles avec exceptions. De façon analogue à ce qui précède, des réseaux ayant

[28] Bien entendu, on peut aussi la voir comme une définition partielle de a à partir de b et c ou de b à partir de a et c.

la même structure externe que des réseaux bayésiens mais spécifiant des distributions de possibilités conditionnelles permettent de spécifier une distribution de possibilité unique, tandis que des contraintes sur le degré de possibilité ou de nécessité de formules propositionnelles permettent de spécifier, en général, une famille de distributions de possibilités[29]. C'est le principe de la logique possibiliste, que j'ai étudiée en détail au cours de ma thèse [144] et qui ne sera que brièvement évoquée dans ce document.

La similarité des outils évoqués ci-dessus dans les modèles probabilistes et possibilistes suggère leur généralisation à des modèles d'incertitude plus généraux. Ce principe a été étudié de manière systématique dans le cas des contraintes sur des formules propositionnelles. On donnera à cette généralisation la dénomination générique de *logiques pondérées*. Une "logique pondérée" est un langage de représentation compact permettant la spécification d'une ou de plusieurs structures d'incertitudes, au moyen de l'expression de contraintes sur les mesures d'incertitude de formules propositionnelles données. On verra au chapitre 4 que les logiques pondérées constituent un outil de représentation compacte non seulement de mesures d'incertitude mais aussi de structures de préférences.

Corrélations états-observations

Spécifier des corrélations entre états et observations de manière compacte revient bien souvent à spécifier des dépendances entre variables d'état observables et variables d'état non observables, et le paragraphe précédent s'y applique tout-à-fait.

1.6.2 Croyances dynamiques

On a vu précédemment que la spécification explicite des effets d'une action ou d'un événement non-déterministe exige de donner la liste de tous les couples d'états, ce qui est (quadratiquement) pire que la spécification explicite de croyances statiques. D'où la nécessité de travailler là aussi avec des représentations structurées et concises. Ces langages de représentation d'effets d'actions sont justement une grosse sphère d'activité en représentation des croyances. On cherche à éviter non seulement une spécification expli-

[29]Notons cependant que dans le fragment le plus utilisé de la logique possibiliste (où les contraintes sont des bornes inférieures de degrés de nécessité), on se ramène à la spécification d'une distribution de possibilité unique, via le *principe du minimum de spécificité*.

cite de tous les couples d'états, mais également la description systématique des propriétés du système qui restent inchangées par l'action (*problème du décor*) et des changements indirects induits par des changements résultant directement des effets "primaires" de l'action (*problème de la ramification*).

Mes contributions au raisonnement sur l'action et le changement sont regroupées au chapitre 3.

1.6.3 Préférences

De la même façon qu'il est judicieux de représenter les connaissances sur le système et son évolution dans un langage structuré, il est (souvent) nécessaire de faire de même pour les préférences de l'agent. En effet, même des préférences temporellement décomposables sont de façon générale aussi coûteuses à spécifier explicitement que des croyances statiques, à savoir un espace en $\mathcal{O}(|\mathcal{S}|)$. Si elles ne sont pas temporellement décomposables, c'est bien pire...

Un chapitre entier (le quatrième) est consacré aux langages de représentation des préférences.

1.7 Résolution d'un problème de prise de décision. Politiques

Parler de la *résolution* d'un problème de prise de décision (à noter qu'on a bien dit "problème" et non "processus") consiste à se poser les deux questions suivantes :

1. *Quel est l'output attendu du problème ?*
2. *Par quel moyen le calculer ?*

L'output attendu du problème sera généralement une *politique d'action* ; mais il faudra distinguer des cas dégénérés où la "politique" est assez différente de ce qu'on entend habituellement par ce terme.

1.7.1 Politiques d'action

L'output attendu d'un problème de prise de décision est une *politique d'action* $\pi : \mathcal{H} \times OBSTRAJ_t \rightarrow \mathcal{A}$ qui, à chaque instant t et à chaque trajectoire observable possible $\tau_{0\rightarrow t}$ de 0 à t, associe une action à effectuer. Une politique d'actions spécifie donc (complètement) le comportement d'un

agent en situation de prise de décision.

Les politiques d'action deviennent des objets plus simples lorsque des hypothèses simplificatrices sont faites, comme on le constate dans le tableau qui suit (on note M pour markovianité et s_0 l'état initial).

hypothèses simplicatrices	politiques
M	$\pi : \mathcal{H} \times \mathcal{B} \to \mathcal{A}$
M + observabilité totale	$\pi : \mathcal{H} \times \mathcal{S} \to \mathcal{A}$
non-observabilité (ou déterminisme + s_0 connu)	$\pi : \mathcal{H} \to \mathcal{A}$
M + horizon infini + obs. totale + stationnarité[30]	$\pi : \mathcal{S} \to \mathcal{A}$
$\mathcal{H} = \{0, 1\}$	$\pi : \mathcal{O} \to \mathcal{A}$
$\mathcal{H} = \{0, 1\}$ + observabilité totale	$\pi : \mathcal{S} \to \mathcal{A}$
$\mathcal{H} = \{0, 1\}$ + non-observabilité (ou dét. + s_0 connu)	$\pi \in \mathcal{A}$

On voit que la complexité spatiale de la représentation explicite d'une politique varie énormément selon les hypothèses sur la nature du système, et peut devenir prohibitive : par exemple, si l'incertitude est binaire, la taille d'une politique est en $\mathcal{O}(|\mathcal{H}|.2^{|\mathcal{S}|})$, c'est-à-dire en $\mathcal{O}(2^{2^n})$ si l'input est exprimé dans un langage concis (on reparlera de la taille des politiques au chapitre 6). Lorsque la taille d'une politique est trop importante, il faut trouver une parade. Deux types de parade sont concevables :

- exprimer la politique d'actions sous forme compacte – ce qui nécessite la définition d'un langage de représentation de politiques ;
- renoncer à produire une politique optimale et chercher une politique approximative, non nécessairement optimale – l'abandon de l'optimalité permet en général de trouver des politiques bien plus concises ; cette politique approximative peut également être partielle, c'est-à-dire s'appliquer seulement à un sous-ensemble des trajectoires observables possibles, ou ne donner les actions à entreprendre que pour quelques instants et non tout l'horizon.

Il faudra donc introduire une notion de *rationalité limitée* pour la résolution d'un processus de prise de décision trop complexe, c'est-à-dire chercher à faire le mieux possible *compte tenu des ressources disponibles*. Attardons-nous un peu sur cette question.

[30] Lorsque l'horizon est infini et stationnaire, il existe une politique optimale qui est stationnaire (ce résultat ne tient pas lorsque l'horizon est fini).

1.7.2 Phase en ligne et phase hors-ligne. Paramètres de ressources.

Il arrive souvent que les données d'un processus de prise de décision – même à une seule étape de décision – ne soient pas toutes connues au même moment. C'est bien sûr le cas pour les observations obtenues comme feedback d'actions, qui ne peuvent évidemment pas être connues au début du processus, mais aussi pour les croyances sur l'état du système avant même que la première action soit entreprise. Considérons par exemple un système à diagnostiquer : lors de la modélisation du système, les croyances de l'agent ne concernent que le comportement des composants – donc les liens existant entre les différentes variables du système ; lors de la phase de décision proprement dite, l'agent a en outre à sa disposition des observations sur les sorties du système (variables directement observables). On voit là qu'on peut voir deux phases distinctes dans la résolution d'un processus de décision : la phase *hors ligne* ("off-line") où le système est décrit dans sa généralité et la phase *en ligne* ("on-line") où les premières observations sont connues. En général, on peut supposer que la phase hors-ligne a lieu suffisamment longtemps avant que les actions doivent être entreprises alors que la phase en-ligne se situe "dans le feu de l'action", les actions ayant alors un degré d'urgence plus marqué. L'output attendu de la phase hors-ligne est alors un "prétraitement" du problème, ou encore une "compilation", qui facilitera sa résolution en-ligne. On peut pousser la formalisation de cette séparation d'un problème en deux phases plus avant et définir des classes de complexité tenant compte du prétraitement (c'est le sujet de [169]). Naturellement, la répartition optimale de la charge computationnelle entre phase hors-ligne et phase en-ligne dépend du contexte du processus en termes de ressources. Intuitivement, plus le rapport entre (a) le coût du temps hors-ligne et de l'espace de stockage d'une politique et (b) le coût du temps en-ligne est élevé, plus l'intérêt de faire porter la charge computationnelle sur la phase hors-ligne est grand. À une extrémité, en situation d'extrême urgence en-ligne, l'idéal est de calculer hors-ligne une politique d'action complète (lorsque c'est raisonnable), ce qui permet de préparer au mieux la prise de décision en-ligne (cette dernière prendra, à chaque instant, seulement un temps logarithmique en la taille de la politique). À l'autre extrémité, lorsqu'il y a peu d'urgence, ou lorsqu'on a peu de mémoire en ligne, ou lorsque la prise de décision en ligne est facile, il peut être inutile de faire un pré-traitement – donc, de calculer ne serait-ce qu'un petit morceau de politique – et laisser tout le travail pour la phase en-ligne.

1.8 Décision ou raisonnement ?

Ce document traite aussi de processus où il n'y a pas à proprement parler de *décision*. On a l'habitude de parler de ces processus comme des processus de *raisonnement pur*. On va montrer que ces problèmes peuvent être formalisés comme des processus de décision (quelque peu dégénérés).

Le raisonnement pur vu comme de la prise de décision : rationalité limitée et actions délibératives

Jusqu'à présent on a mentionné deux classes d'action (non disjointes comme on verra aux chapitres 5 et 6) : les actions *effectives*, dont le but principal est de modifier l'état du système ; et les actions *épistémiques* qui ont pour but principal l'acquisition par l'agent de nouvelles croyances sur le système (et qui, en général, laissent inchangé l'état du système). On va maintenant introduire un troisième type d'action : les *actions délibératives*, et définir ensuite des classes de problèmes de prise de décision (qui vont se superposer à la taxonomie précédente) en fonction des actions que l'agent peut entreprendre.

Avant d'introduire les actions délibératives, il faut distinguer *états de croyance implicites et explicites*. Jusqu'à présent, nous n'avons pas fait la différence entre les deux parce que nous avons supposé que toutes les actions d'inférence que l'agent peut entreprendre à partir d'un état de croyance sont "gratuites" – ou encore, ce qui revient au même lorsque seules des tâches déductives doivent être entreprises, que l'état de croyance de l'agent est fermé pour la déduction.

Un état de croyance *explicite* est simplement un ensemble de données élémentaires et indépendantes (des "items propositionnels", c'est-à-dire des formules propositionnelles, éventuellement accompagnées d'un élément annexe – priorité, contexte d'application, degré d'incertitude etc.). Étant donnée une relation d'inférence, l'état de croyance *implicite* associé à un état de croyance explicite est l'ensemble de tout ce qui peut en être inféré. Une action délibérative est ce qui permet à l'agent de passer d'un état épistémique explicite à un état explicite plus complet.

Par exemple, si $K \models \varphi$, on peut dire que l'agent dont les croyances sont représentées *explicitement* par la formule K connait *implicitement* φ *mais pas explicitement*, du moins tant qu'il n'a pas prouvé que φ était conséquence

logique de K. Une fois qu'il l'aura prouvé, il l'ajoutera au stock de croyances explicites. (La définition précédente d'un état de croyance explicite ne va cependant pas suffire tout-à-fait à nos besoins, pour la raison que les tâches délibératives que l'agent effectue ne sont pas toutes déductives ; on y revient un peu plus loin).

Une *action délibérative* agit sur l'état de croyance *explicite* de l'agent (par exemple au moyen d'une règle d'inférence). Les actions délibératives qui consistent en l'application d'une règle d'inférence sont appelées *actions délibératives microscopiques*, ou encore *actions inférentielles*. Si l'on fait le choix de ne pas distinguer croyances explicites et croyances implicites (ce qu'on a d'ailleurs fait jusqu'ici), alors les actions délibératives ne sont pas pertinentes. Notons qu'une action délibérative n'a aucune influence, non seulement sur l'état du système (c'est bien clair), *ni sur les croyances implicites de l'agent*, au contraire des actions épistémiques.

Fagin et Halpern [90] ont proposé plusieurs logiques pour le raisonnement "limité" avec des états de croyance explicites, qui se fondent sur la notion de *conscience* ("awareness") de la prouvabilité d'une formule : intuitivement, un agent est conscient de φ si φ fait partie de son état de croyance explicite, ou en d'autres termes s'il a en tête une preuve de φ . Bien entendu, on peut être conscient d'un ensemble de formules sans être conscient de toutes ses conséquences logiques (ce qui permet d'exprimer, par exemple, qu'un agent n'est pas nécessairement conscient de tous les théorèmes logiques).

Une autre communauté s'intéresse (depuis une époque plus récente) aux actions de délibération : celle des "agents autonomes" ; on considère des agents ayant des capacités de raisonnement limitées, qui effectuent régulièrement des tâches d'*ordonnancement de la délibération* ("deliberation scheduling") afin de reconsidérer leur plan courant. L'article de Parsons et Wooldridge [189], par exemple, exprime la reconsidération d'intention comme un processus décisionnel markovien.

Introduire les états de croyance explicites et les actions délibératives dans la modélisation d'un processus de décision (non-dégénéré) permet d'exprimer formellement les principes de rationalité limitée dont on a parlé à la fin de la Section 1.6. Nous ne nous aventurons pas plus loin sur ce point mais il est évident que c'est un large champ de recherches, par exemple en ce qui concerne les questions de complexité computationnelle.

Allons maintenant un peu plus loin dans la formalisation de problèmes de raisonnement comme des problèmes de prise de décision. On a parlé un peu plus haut d'un premier type d'actions délibératives, à savoir celles qui consistent en l'application d'une règle d'inférence et qui sont appelées *microscopiques*. Voici maintenant les actions délibératives *macroscopiques*. Ces actions ont des effets plus complexes que l'application d'une simple règle d'inférence. Une action délibérative macroscopique peut être vue comme un *programme* construit à partir d'actions délibératives microscopiques, ou comme une action "opaque" et indivisible (que l'on ne cherche pas à décomposer en actions microscopiques). Voici enfin des exemples d'actions délibératives macroscopiques :

- *actions de déduction* : déterminer si une formule est conséquence logique d'une autre ;
- *actions de recherche de modèle(s)* : trouver un modèle d'une formule cohérente ; déterminer (sous forme compacte si possible) l'ensemble de *tous* les modèles d'une formule ;
- *actions de recherche de conséquence* : déterminer l'ensemble des conséquences d'une formule ayant une forme syntaxique donnée ;
- *actions de compilation* : réexprimer une formule de manière équivalente en vue de la rendre plus manipulable pour la suite du processus ;
- *actions d'abduction* : trouver une hypothèse permettant de déduire une certaine formule et satisfaisant certains critères (cohérence, critères syntaxiques ou de minimalité).

Il en existe bien d'autres (des exemples : déterminer si deux variables sont indépendantes modulo une base de croyances, décomposer une base de croyances en sous-bases vérifiant certaines propriétés, déterminer l'ensemble des sous-bases maximales cohérentes d'un ensemble donné de formules, etc.).

Comment modéliser les conséquences d'une action délibérative macroscopique ? Tant que l'on ne considère que des actions de déduction, c'est simple : les états de croyance explicites suffisent ; mais on ne peut pas exprimer, dans un état de croyance explicite, la cohérence d'une formule (qui correspond à la *non-croyance* de sa négation). On a donc besoin de généraliser quelque peu la notion d'état de croyance explicite. Un *état de croyance explicite généralisé* est un ensemble de croyances *et de conclusions*, où une conclusion peut être vue comme une méta-croyance. Par exemple, l'action qui a montré qu'une formule φ est satisfaisable modifie l'état de croyance explicite généralisé de l'agent en lui ajoutant la conclusion φ *satisfaisable* (ce qui, on l'a déjà dit, revient à exprimer une non-croyance en $\neg\varphi$). ω *est un modèle de* φ , H *est*

une explication minimale, *les impliqués premiers de* φ *sont* $\{\gamma_1, \ldots, \gamma_q\}$, A *est une sous-base maximale cohérente de* B sont des exemples de conclusions.

Enfin, il existe une action délibérative particulière : l'action *stop*, dont le rôle est de stopper le processus délibératif, soit parce que le but est atteint (apparition d'une formule souhaitée dans les croyances explicites, identification d'un modèle etc.), soit parce que les ressources computationnelles sont épuisées.

Pour alléger un peu la terminologie, nous n'emploierons plus l'adjectif "généralisé" : par défaut, les états de croyance explicites seront des états de croyance explicites généralisés.

Un processus de raisonnement "pur" peut être vu comme un problème de prise de décision concernant des états de croyance explicites et des actions délibératives, ce qui va être repris plus en détail au chapitre 2.

Types d'agents, d'actions, de processus

Pour résumer ce qui précède, nous distinguons désormais trois notions de base :

- l'*état du système* ;
- l'*état de croyance (implicite) de l'agent* est en général une construction complexe sur l'ensemble des états du système (par exemple, un sous-ensemble, une distribution de probabilité...) ;
- l'*état de croyance explicite* de l'agent est un ensemble de croyances, sans structure particulière, auquel un état de croyances implicite est sous-jacent.

Dans la suite du document, lorsque l'on parlera d'*états de croyances* (tout court), il s'agira d'états de croyance implicites (après ce chapitre on ne parlera presque plus d'états de croyance explicites).

Dans ce qui précède, on a également distingué trois types d'actions :

- les *actions (purement) délibératives* laissent inchangé le système ainsi que l'état de croyances implicite de l'agent ; elles ne peuvent modifier que son état de croyance explicite généralisé ; elles n'ont plus de sens si on ne fait pas de distinction entre états de croyance explicites et implicites.

– les *actions (purement) épistémiques* laissent inchangé l'état du système mais pas l'état de croyance implicite (ni explicite) de l'agent ;
– les *actions effectives* modifient généralement l'état du système, et par contre-coup, l'état (implicite et a fortiori explicite) des croyances de l'agent, et ceci même en l'absence de feedback du système.

On peut maintenant établir une typologie des agents *selon la nature de leurs préférences* :
– les *agents volitifs* ont une fonction d'utilité sur les états du système, qui est généralement non constante ;
– les *agents (purement) épistémiquement volitifs*, appelés plus simplement *agents juste curieux*, sont indifférents à l'état du monde mais ont une préférence pour la quantité de connaissance et possèdent donc une relation de préférence ou une fonction d'utilité *sur les états de croyance (implicites)*. On y reviendra au chapitre 5.
– les *agents inférentiellement volitifs* sont indifférents à l'état du monde et à l'état implicite de leurs croyances *mais pas à leur état de croyance explicite*. Par exemple, un agent qui cherche à savoir si soit ψ soit $\neg\psi$ est conséquence logique de ses croyances initiales aura une préférence pour les états de croyance explicites dans lesquels ψ ou $\neg\psi$ apparaît.
– les *agents totalement indifférents* n'ont aucune préférence, ni sur les états du système, ni sur l'état de leurs croyances implicites ou explicites.

On peut enfin établir une typologie de ces mêmes agents selon le(s) type(s) d'actions dont ils disposent :
– les *agents purement délibératifs*, ou encore *purement introspectifs*, n'ont à leur disposition que des actions purement délibératives ;
– les *agents inquisiteurs* ont à leur disposition des actions purement épistémiques (et des actions délibératives), mais pas d'actions effectives ;
– les *agents ontiquement actifs* ont à leur disposition des actions de n'importe quel type.

Enfin, on peut établir une typologie des processus de raisonnement et/ou de prise de décision selon le type d'agent qu'ils mettent en oeuvre :
– les *processus de raisonnement pur* mettent en jeu des agents purement délibératifs et inférentiellement volitifs ;
– les *processus de prise de décision épistémique* mettent en jeu des agents inquisiteurs et épistémiquement volitifs ;
– les *processus de prise de décision* dans le sens général mettent en jeu

des agents ontiquement actifs et volitifs.

Chacun de ces trois processus peut être vu comme un processus de prise de décision totalement observable, à condition de le modéliser d'une façon particulière. Ainsi, les problèmes de prise de décision purement épistémiques peuvent être vus comme des processus de décision totalement observables sur l'espace des états de croyance implicite de l'agent (ce qui est bien connu de la communauté de recherche sur les processus décisionnels markoviens totalement observables), et, moins classiquement, les problèmes de raisonnement pur peuvent être vus comme des processus de décision totalement observables sur l'espace des états de croyance explicites de l'agent (les coûts des actions délibératives peuvent être fonction du temps de calcul et/ou de l'espace mémoire utilisé).

1.9 Classes de processus et de problèmes ; introduction au reste du document

Le lecteur attentif aura remarqué qu'un terme et une notation ont été employés, qui n'ont pas encore été définis formellement.

Une *classe de processus* (de prise de décision) est un ensemble [d'instances] de processus de prise de décision – de la même façon qu'en informatique, un problème (de décision, de recherche, d'optimisation ...) au sens informatique est un ensemble d'instances. En général, une classe est définie en fixant certains paramètres du processus (ontologiques, épistémiques, préférentiels) – exactement comme pour les classes de problèmes de raisonnement sur l'action et le changement chez Sandewall [202].

De facon similaire, une *classe de problèmes* (de prise de décision) est un ensemble [d'instances] de problèmes défini non seulement en fixant certains paramètres du processus mais également un langage de représentation (ou éventuellement une famille de langages). La notion de classe de problèmes est probablement plus pertinente que celle de classe de processus, dans la mesure où à une classe donnée va correspondre une panoplie de méthodes de résolution adaptées (là encore, la démarche est similaire à celle de Sandewall).

Voici, pour terminer, quelques exemples de classes de processus et de problèmes :

– la classe des *processus décisionnels markoviens (*MDP) à horizon fini est

définie en fixant les paramètres suivants : (1) horizon fini ; (2) actions stochastiques ; (3) utilités additives, temporellement décomposables ; (4) critère de l'utilité espérée. En fixant en outre le langage, on obtient différentes classes de *problèmes*, chacune d'entre elles ayant des méthodes de résolution qui lui sont consacrées ; par exemple, la programmation dynamique (méthode de base pour les MDP avec représentation explicite) est inappplicable telle quelle lorsque la représentation est structurée (voir [32, 31]) ;

- la classe des *problèmes de diagnostic à base de modèles* est définie en fixant les paramètres suivants : (1) horizon statique ; (2) actions délibératives seulement ; (3) incertitude tout-ou-rien ; (4) utilité fonction des diagnostics minimaux contenus dans l'état de croyance explicite généralisé[31] ; (5) représentation compacte (logique propositionnelle). On obtient la classe de problèmes de *diagnostic avec tests* en ajoutant des actions épistémiques, puis la classe de problèmes de *diagnostic avec test et réparation* en ajoutant des actions ontiques de réparation de composants.
- la classe des *problèmes de planification "à la* STRIPS*"* est définie en fixant les paramètres suivants : (1) représentation compacte dans le langage STRIPS ; (2) horizon fini ; (3) actions déterministes, inconditionnelles, avec précondition ; (4) état initial connu ; (5) utilité tout-ou-rien définie par une conjonction de littéraux buts. On peut raffiner davantage en considérant la classe des *problèmes* STRIPS *à horizon polynomial* (on en reparlera au chapitre 6).

Le reste du document est structuré en cinq chapitres. Pour chacun d'entre eux, on fixe quelques paramètres et on se focalise sur les classes de processus et de problèmes ainsi obtenues.

- au chapitre 2, les agents sont purement délibératifs, l'horizon est statique, le langage de représentation est logique ; les problèmes obtenus sont donc des problèmes de raisonnement pur sur un système non évolutif ;
- le chapitre 3 est assez hétéroclite :
  - en Section 3.1, l'horizon est dynamique mono-étape ($\mathcal{H} = \{0,1\}$), le système est inerte (pas de changements qui ne soient provoqués par des actions) et les agents ont à leur disposition des actions ontiques (non-déterministes) mais ils n'ont pas de capacité de décision, ce qui revient à supposer qu'ils sont purement délibératifs (ils raisonnent

[31] Par exemple : le nombre ; ou alors, 1 s'ils y sont tous et 0 sinon.

sur les effets d'une action fixée), et la représentation est logique ;
  - en Section 3.2, les agents sont purement délibératifs, l'horizon est fini, le système n'est pas inerte (éventuellement, il est semi-inerte), et la représentation est logique ;
  - en Section 3.3, les agents sont purement délibératifs et le système est statique comme au chapitre 2 (on y parle de raisonnement sur l'espace et le changement, et j'ai choisi de mettre cette partie dans ce chapitre pour des raisons de proximité technique avec le raisonnement sur le temps et le changement) ;
- au chapitre 4, les agents sont volitifs (et tout le reste est simple : horizon mono-étape, actions déterministes, état initial connu) ; on ne s'intéresse par conséquent qu'aux préférences, et on fait varier les langages de représentation ; dans la dernière Section de ce chapitre, on lève pour la seule fois l'hypothèse d'un agent unique et on considère une collectivité d'agents en situation de décision de groupe ;
- le chapitre 5 est consacré aux processus purement épistémiques ;
- et enfin, le chapitre 6 parle de processus de prise de décision généraux (avec actions ontiques et épistémiques), avec un horizon fini, des actions non-déterministes, des environnements partiellement observables, et diverses représentations (explicite, logique propositionnelle classique, extensions de STRIPS, logiques modales, problèmes de satisfaction de contraintes).

Chapitre 2

Raisonnement plausible : incohérence, incertitude, inférences non-monotones, indépendance

Dans ce chapitre, nous faisons deux hypothèses considérablement simplificatrices : le système n'évolue pas (ce qui revient à imposer $\mathcal{H} = \{0\}$) et *l'agent est purement délibératif.*

Un agent délibératif un peu fruste est celui qui est équipé de la relation de déduction du calcul propositionnel (ou, au niveau microscopique, des règles d'inférence d'un système complet pour la logique propositionnelle). Ce n'est pas que les tâches d'inférence qu'il effectue soient simples (loin de là) ; mais la variété des types de connaissances avec lesquelles il est souvent nécessaire de composer rend la logique propositionnelle souvent trop peu expressive. D'un côté, il y a bien sûr la logique du premier ordre, qui permet d'exprimer des notions fonctionnelles et des quantifications. *Nous faisons cependant le choix d'ignorer la logique du premier ordre dans ce document.* C'est un choix guidé tout simplement par la nature des travaux que j'ai effectués jusqu'à présent : m'intéressant avant tout aux problèmes décidables, je me suis naturellement intéressé à la logique propositionnelle et à certaines de ses extensions mais pas à la logique des prédicats. D'un autre côté, de nombreux formalismes permettant des raisonnements plausibles développés dans la communauté de l'IA ont consisté en des extensions de la logique propositionnelle. Voici quelques-uns des problèmes de "raisonnement plausible" (en environnement

statique) soulevés par cette communauté (la liste n'est pas exhaustive) :

- raisonnement avec des exceptions, avec des règles plus ou moins spécifiques ;
- raisonnement avec des faits et règles empreints d'incertitude (provenant de données statistiques ou de connaissances expertes) ;
- raisonnement à partir de sources d'information multiples et éventuellement partiellement incohérentes ;
- raisonnement par analogie, à partir de cas.

Le reste du chapitre vise à exposer mes contributions au traitement de certains de ces problèmes, à savoir : (i) plusieurs formalismes de raisonnement à partir d'informations incohérentes ; (ii) les liens entre raisonnement dans l'incertain et raisonnement non-monotone ; (iii) la formalisation logique de plusieurs notions d'indépendance.

Notons enfin que le *raisonnement sur l'espace*, bien qu'il concerne un monde statique et des agents purement délibératifs, sera traité au chapitre 3, après le raisonnement sur le temps et l'action, en raison de la proximité technique de mes contributions dans ces deux domaines.

2.1 Raisonnement à partir de bases de croyances incohérentes

Il y a plusieurs situations assez différentes où on peut être amené à travailler avec des ensembles de formules incohérentes. Le projet collectif *FUSION* a contribué à élaborer une classification très fine de ces situations [24], dont nous faisons ici un bref résumé.

1. plusieurs experts ou capteurs qui fournissent des données conflictuelles ;
2. plusieurs avis divergents concernant l'évaluation d'un objet ou d'un candidat ;
3. plusieurs contextes ou plusieurs règles simultanément applicables dont les conclusions sont contradictoires ;
4. les hypothèses de bon fonctionnement simultané de composants d'un système dont la conjonction est contradictoire ;
5. plusieurs critères conduisant à des préférences incompatibles pour un agent donné ;
6. plusieurs agents dont les préférences sont incompatibles.

Dans ce chapitre, nous ne nous préoccupons que de raisonnement et pas de préférences. Nous laisserons donc les items 5 et 6 pour le chapitre 4 où nous traiterons les préférences conflictuelles[1]. Quant à l'item 4, nous le considérerons partiellement ici et le reprendrons au chapitre 5 lorsque nous parlerons de l'utilisation de tests en diagnostic de pannes.

L'agent (qu'il soit purement délibératif, inquisiteur ou ontiquement actif) équipé de la relation de déduction de la logique classique est impuissant face à l'incohérence, puisqu'il doit renoncer à exploiter sa base de croyances dès que celle-ci est incohérente : en effet, à partir d'une base de connaissances incohérente, la relation de déduction classique permet d'inférer n'importe quelle formule du langage.

Quelle attitude raisonnable peut-on alors envisager face à une base de croyances incohérentes ? Nous allons élaborer une taxonomie de ces attitudes.

1 *attitude active*. L'agent cherche à résoudre les incohérences, c'est-à-dire à les identifier (on pourrait presque dire "débugger la base de croyances"), en effectuant des actions de prise d'information (des tests) qui, idéalement, rendront *in fine* la base de croyances[2] cohérente. Ceci suppose que l'agent est semi-actif, et nous traiterons ce type d'approche au chapitre 5.
2 *attitude passive*. L'agent n'aura pas de nouvelles informations lui permettant de l'aider à raisonner avec l'incohérence. Il doit se débrouiller tant bien que mal avec sa base de croyances, et rien d'autre.

Puisque nous ne nous intéressons dans ce chapitre qu'au deuxième type d'attitude, allons plus loin dans la taxonomie des attitudes en en dégageant deux grandes classes : puisqu'il faut de toute façon s'arranger pour réduire l'ensemble des conséquences déduites des informations de la base (par rapport à ce que ferait la logique classique), il n'y a en définitive que deux solutions : soit on affaiblit les informations de la base de croyances de telle manière que leur conjonction va devenir cohérente, soit on affaiblit la relation

[1] Le point 2 nous parait plus proche du raisonnement sur l'état du monde (on raisonne sur la capacité d'un candidat à faire le travail demandé, ou celle d'un objet à répondre correctement à sa fonction) que du raisonnement sur des préférences (les experts – objectifs – qui donnent leur avis ne sont d'ailleurs pas nécessairement impliqués dans la décision qui sera prise concernant l'objet ou le candidat. Comment pourrait-on, sinon, parler d'avis objectifs ?)

[2] *Stricto sensu*, ce n'est pas la base de croyances initiale qui sera "rendue" cohérente, mais sa révision par les résultats des tests.

de conséquence logique[3].

2.1.1 Affaiblir les informations

Trois grandes classes de méthodes ont été proposées dans la littérature, et en particulier à l'IRIT. J'insisterai plus particulièrement sur les méthodes que j'ai contribuées à développer, ainsi que sur celles qui ont été développées par d'autres chercheurs de l'IRIT.

2.1.1.1 Affaiblissement par inhibition

C'est la méthode la plus ancienne. Elle consiste à calculer d'abord des sous-ensembles de la base de croyances B cohérents et maximaux pour un critère qui est à définir. Cette façon de procéder remonte à [195]. Le critère le plus évident est celui de l'inclusion : une *sous-base maximale cohérente* est tout simplement une sous-base cohérente de B dont tout sur-ensemble strict dans B est incohérent.

Il n'est pas suffisant de se fixer un critère de sélection ; il faut aussi savoir ce que l'on fera des informations déduites à partir des différentes sous-bases sélectionnées. Pour cela, on introduit ce que Cayrol et Lagasquie [43] appellent un *procédé d'inférence*. Le plus courant consiste à inférer une formule si et seulement si elle est déductible de *toute* sous-base préférée (inférence "universelle"). Des procédés plus faibles sont l'inférence "existentielle" qui consiste à inférer une formule si et seulement si elle est déductible d'au moins une sous-base préférée, et l'inférence "argumentative", qui consiste à inférer une formule si et seulement si elle est existentiellement inférable et sa négation ne l'est pas.

Bases de connaissances stratifiées et critères de sélection

Le critère de l'inclusion pure et simple n'est souvent pas très efficace, en particulier parce qu'il génère trop de sous-bases cohérentes, ce qui rend les relations d'inférence trop prudentes ou au contraire trop aventureuses. Un critère évident, dont la capacité de discrimination est meilleure, est la cardinalité. Cependant, l'un comme l'autre de ces critères ne permettent pas de tirer parti d'éventuelles informations sur la fiabilité des sources qui ont fourni les informations.

[3] On peut évidemment envisager de faire les deux à la fois.

J'ai ainsi contribué à définir des critères plus fins de sélection de sous-bases cohérentes. Dans [144, 74, 13], j'ai contribué à proposer un raffinement des critères de l'inclusion et de la cardinalité qui tiennent compte de degrés de priorité, certitude ou fiabilité associées aux informations de la base de cohérence. Une telle base de croyances est dite stratifiée : elle est partitionnée en sous-bases de niveaux différents, à savoir, $B = B_1 \cup \ldots \cup B_n$ (où B_1 contient ensemble les informations les plus certaines). Le critère de sélection dit *discrimin*[4], également proposé par Brewka [34], sélectionne les sous-bases $A = A_1 \cup \ldots \cup A_n$ (où $A_i \subseteq B_i$ pour tout i) telles que pour tout j, $A_1 \cup \ldots A_j$ est maximale cohérente pour l'inclusion dans $B_1 \cup \ldots \cup B_j$, tandis que le critère *leximin* sélectionne les sous-bases $A = A_1 \cup \ldots \cup A_n$ telles que pour tout j, $A_1 \cup \ldots A_j$ est maximale cohérente pour la cardinalité dans $B_1 \cup \ldots \cup B_j$. Ce second critère raffine le premier, c'est-à-dire que toute sous-base leximin-préférée est également discrimin-préférée.

Dans [74] et [144], ces critères sont vus comme des raffinements du critère de sélection sous-jacent à la relation de déduction en logique possibiliste. Ce critère sélectionne une sous-base cohérente unique (et c'est là son intérêt computationnel) en déterminant d'abord le plus petit indice j tel que $B_1 \cup \ldots \cup B_j$ est incohérente, puis sélectionne la sous-base cohérente $B_1 \cup \ldots \cup B_{j-1}$. Ce critère appelé "best-out" a l'inconvénient d'inhiber trop de formules, et en particulier des formules qui ne sont objectivement pas mélées aux contradictions (mais elle a l'avantage d'une complexité algorithmique plus raisonnable que les autres, et, ce qui y est lié, la production d'une sous-base cohérente unique). Un raffinement de cette méthode, proposé par Benferhat, Dubois et Prade dans [14], consiste à rajouter à B_{j-1} celles des formules de $B_j \cup ... \cup B_n$ qui ne font partie d'aucun conflit minimal (formules appelées "libres") ; cette méthode est plus satisfaisante que la précédente car elle a un pouvoir d'inférence plus important mais elle est computationnellement plus complexe (l'inférence est Π_2^p-complète).

Une interprétation en termes de probabilité de pannes

Lorsque B n'a qu'une seule strate, le critère `leximin` revient à préférer les sous-bases cohérentes de B de cardinalité maximale, ou de manière équivalente, à "expliquer" l'incohérence par les ensembles "candidats"[5] de cardi-

[4]Cette terminologie est postérieure aux travaux cités, elle apparait pour la première fois dans [73].

[5]Un ensemble candidat est un sous-ensemble C minimal de B tel que $B \backslash C$ est cohérent. C'est donc le complémentaire d'une sous-base maximale cohérente.

nalité minimale. Cette démarche est rigoureusement équivalente à celle qui, en diagnostic, privilégie les configurations de pannes impliquant un nombre minimal de composants défectueux (*principe de parcimonie*). La justification implicite de ce principe est que les probabilités *a priori* que les pannes des différents composants sont des événements indépendants dont les probabilités a priori sont très faibles et toutes égales. Cette analogie peut être poursuivie en ces termes : chaque information φ_i d'une base de connaissances B est associée à une source $sigma_i$, de la même façon qu'en diagnostic à base de modèles chaque composant c_i est associé à une formule φ_i modélisant son bon fonctionnement. La source σ_i est assimilée au composant c_i et l'hypothèse de bon fonctionnement de σ_i signifie que la formule φ_i est pertinente. Au contraire, si σ_i est en panne, alors φ_i n'est pas pertinente : elle peut être vraie (par hasard) aussi bien que fausse. Ce qui peut être représenté par la formule $\Sigma_B = \bigwedge_{i=1\ldots n}(\sigma_i \rightarrow \varphi_i)$, appelée Σ-transformation de B [157]. L'ensemble des conflits de B est la projection de Σ sur les variables $\{\sigma_i, i = 1 \ldots n\}$, c'est-à-dire $\Gamma_B = \exists Var(B).\Sigma_B$.

J'ai donné dans [145] une interprétation du principe de parcimonie en termes de probabilités infinitésimales. Ainsi, on suppose que les sources d'information σ_i ont une probabilité de panne $\varepsilon \ll 1$, et les événements "σ_i est en panne" sont indépendants. On calcule ensuite la probabilité *a posteriori* de chaque ensemble de sources en état de fonctionnement, appelé *scénario*, en conditionnant par la connaissance sur les conflits. Ainsi, supposons que les conflits minimaux de $B = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ soient $\{\{\sigma_2, \sigma_3\}, \{\sigma_2, \sigma_4\}\}$, ce qui implique que les scénarios maximaux (correspondant aux sous-bases maximales cohérentes de B) sont $\{\{\sigma_1, \sigma_3, \sigma_4\}, \{\sigma_1, \sigma_2\}\}$. La probabilité a posteriori du scénario $\{\sigma_1, \sigma_3, \sigma_4\}$ est $\frac{Pr((\sigma_1 \wedge \sigma_3 \wedge \sigma_4) \wedge (\neg(\sigma_2 \wedge \sigma_3) \wedge \neg(\sigma_2 \wedge \sigma_4)))}{Pr(\neg(\sigma_2 \wedge \sigma_3) \wedge \neg(\sigma_2 \wedge \sigma_4)))} = 1 - \mathcal{O}(\varepsilon)$ tandis que celle de $\{\sigma_1, \sigma_2\}$ est $\mathcal{O}(\varepsilon)$. On voit que le critère de parcimonie implique que la probabilité *a posteriori* du scénario de cardinalité maximale est proche de 1 (et, bien évidemment, celle des autres scénarios maximaux est proche de 0). Plus généralement, on montre que s'il existe exactement p scénarios de cardinalité maximale k, alors pour tout scénario E on a :

- si $|E| = k$ alors $Pr(E|\Gamma_B) = \frac{1}{p} + \mathcal{O}(\varepsilon)$;
- si $|E| < k$ alors $Pr(E|\Gamma_B) = \mathcal{O}(\varepsilon^{k-|E|})$

Par conséquent, seuls les scénarios de cardinalité maximale ont une probabilité qui ne tend pas vers 0, ce qui justifie le critère de la cardinalité pour la sélection de sous-bases cohérentes. Dans l'article [145], cette justification est faite de façon plus précise en associant à B la fonction de croyance Bel_B

dont les éléments focaux sont les scénarios de B avec leur probabilité a posteriori. La mesure de croyance $Bel_B(\psi)$ de toute formule ψ représente alors sa *probabilité de déductibilité* (voir [210]). Cette fonction de croyance peut se définir de façon équivalente par

$$Bel_B(\psi) = \sum_{E|E\models\psi} Pr(E|\Gamma_B)$$

Il est alors intéressant de voir que cette fonction de croyance, qui est induite par B seulement (et les hypothèses initiales sur les probabilités de panne, bien sûr), permet de caractériser de nombreuses relations d'inférence tolérantes à l'incohérence de la littérature. Ainsi, par exemple :

- $Bel_B(\psi) \longrightarrow_{\varepsilon\longrightarrow 0} 1$ si et seulement si ψ est prouvable dans toutes les sous-bases cohérentes de B de cardinalité maximale ;
- $Bel_B(\psi)$ ne tend pas vers 0 lorsque $\varepsilon \longrightarrow 0$ si et seulement si ψ est prouvable dans au moins une sous-base cohérente de B de cardinalité maximale ;
- $Bel_B(\psi) > 0$ si et seulement si ψ est prouvable dans au moins une sous-base cohérente de B maximale pour l'inclusion.
- $Bel_B(\psi) > \frac{1}{2}$ si et seulement si ψ est prouvable dans la majorité absolue des sous-base cohérentes de B de cardinalité maximale.

D'autres relations d'inférence (notamment les relations argumentatives) peuvent être obtenues par des moyens similaires (voir [145]). Cependant, la relation prudente associée au critère de l'inclusion, à savoir, ψ est inférée si et seulement si elle est prouvable dans toutes les sous-bases maximales (pour l'inclusion) cohérentes de B , ne peut pas être obtenue par une telle caractérisation[6].

Incohérences et pénalités

On a vu plus haut qu'il est important de pouvoir exprimer que certaines informations sont plus fiables que d'autres et qu'on doit pouvoir tenir compte de ces fiabilités relatives lors de la sélection de sous-bases cohérentes de B . Le problème avec les approches à base de stratification est qu'elles ne permettent pas de compensation entre informations de strates différentes : ainsi, le rejet d'un nombre quelconque de formules de la strate i sera toujours privilégié au rejet d'une seule formule de la strate $i-1$. Dans le cas extrême du critère

[6]Il faudrait pour cela lever l'hypothèse que les probabilités de panne sont égales et quantifier universellement sur celles-ci.

"best-out", on en arrive au point où seule la strate de l'information rejetée la plus fiable compte. Dans de nombreuses situations pratiques on a besoin de permettre une compensation entre le rejet de certaines informations et la satisfaction d'un certain nombre d'autres. Une telle approche consiste à associer à chaque φ_i de B une *pénalité* induite par son rejet : une *base de croyances avec pénalités* est un ensemble de couples $B = \{\langle \varphi, x_i \rangle, i = 1 \dots m\}$, où pour tout i, $x_i \in \mathbb{R}^+ \cup \{+\infty\}$. La pénalité associée à une sous-base de B est alors la somme des pénalités des formules qu'elle rejette (c'est-à-dire les formules de $B \setminus S$) tandis que la pénalité associée à un monde est la somme des pénalités des formules qu'il viole : $k_B(\omega) = \sum\{x_i | i \in 1 \dots m, \omega \models \neg x_i\}$; on voit aisément que la pénalité d'une sous-base *maximale* cohérente de B est identique à la pénalité d'un quelconque de ses modèles – ce qui n'est pas le cas pour une sous-base cohérente non maximale. Il est alors intuitif de définir des scénarios préférés (respectivement, des mondes préférés) comme ceux qui ont une pénalité minimale. Cette idée, très naturelle, a été utilisée dans plusieurs travaux, en particulier ceux de Pinkas [192], de Eiter et Gottlob dans un contexte abductif [87] ou de Schiex et al. [207] dans le contexte les CSP valués.

Nous avons effectué une étude systématique de l'inférence à partir de connaissances valuées par des pénalités (selon le principe qu'une formule est inférée si elle peut être déduite de tous les scénarios préférés, ou de manière équivalente si elle est vraie dans tous les mondes préférés) [84, 85]. Ce travail, qui comprend une étude algorithmique et une implémentation de la recherche de sous-bases de pénalité minimale, a été l'objet du DEA de Florence Dupin de Saint-Cyr [81].

On voit immédiatement que lorsque les pénalités sont unitaires, les scénarios préférés sont ceux de cardinalité maximale et le critère de la pénalité minimale correspond à celui de la cardinalité maximale. Plus généralement, le critère `leximin` peut facilement être recouvré comme cas particulier du critère de la pénalité minimale, et ce à l'aide d'une transformation polynomiale, pourvu que l'on connaisse un majorant M de la cardinalité des strates (si B comporte n strates, il suffit d'associer à chaque information de la strate i la pénalité $(M+1)^{|n-i|}$).

Le critère de la pénalité minimale semble naturel, et il est en tout cas facile à mettre en oeuvre. Il souffre cependant d'un handicap : quelle est la signification de ces pénalités en terme d'incertitude ? (On verra au chapitre 4 qu'en terme de préférences c'est bien plus simple : les pénalités sont des

disutilités). Nous avons donné une réponse dans [85] : tant que l'on ne se préoccupe que des pénalités associées aux mondes (ou, de façon équivalent, aux scénarios), les pénalités sont, à une transformation logarithmique près, des plausibilités de singletons au sens de la théorie de l'évidence de Dempster-Shafer (ces plausibilités de singletons $pls \ : \omega \mapsto pls(\omega) = Pl(\{\omega\})$ sont aussi appelées des "fonctions de contour"). Voici davantage de détails. Soit $B = \{\langle \varphi_i, x_i \rangle, i = 1...m\}$, où pour tout i, $x_i \in \mathbb{R}^+ \cup \{+\infty\}$, une base de croyances avec pénalités. À chacun des couples $\langle \varphi_i, x_i \rangle$ on associe une fonction de support simple m_i comme ci-dessous, et la combinaison de Dempster des m_i opère exactement comme la somme des pénalités, comme le montre le tableau de correspondances ci-dessous.

$$
\begin{array}{lcl}
m_i \ : \left\{ \begin{array}{ll} m_i(\varphi_i) & = 1 - \exp(-x_i) \\ m_i(\top) & = \exp(-x_i) \end{array} \right. & \Longleftrightarrow & k_i \ : \left\{ \begin{array}{ll} k_i(\omega) = 0 & \text{si } \omega \models \varphi_i \\ k_i(\omega) = x_i & \text{si } \omega \models \neg\varphi_i \end{array} \right. \\
m_B = m_1 \otimes ... \otimes m_n & \Longleftrightarrow & k_B = k_1 + ... + k_n \\
pl_B(\{\omega\}) = \Pi_i pl_i(\{\omega\}) & \Longleftrightarrow & k_B(\omega) = \sum_{i \ | \ \omega \models \neg\varphi_i} k_i(\omega) \\
 & \Longleftrightarrow & \\
 & \boxed{pl_B(\{\omega\}) = \exp(-k_B(\omega))} &
\end{array}
$$

Toutefois, on ne peut pas étendre aisément cette correspondance des mondes aux formules quelconques (une telle extension fonctionne comme cas limite lorsqu'on considère des fonctions de croyance infinitésimales ; voir [85]).

2.1.1.2 Affaiblissement par dilatation. Fusion d'informations

Le raisonnement à partir de sous-bases de B consistait à affaiblir les informations de B en les transformant en défauts dont la véritée est plausible mais pas certaine. Lorsqu'un défaut n'est pas satisfait, peu importe la façon dont il n'est pas satisfait (en d'autres termes, on ne fera pas de distinction entre un défaut "violé de peu" et un autre "violé de beaucoup"). Cette distinction est pourtant pertinente si l'on considère que l'ensemble des modèles peut être muni d'une structure de *pseudo-distance* : une information sera d'autant plus insatisfaite par un monde que la distance de ce monde φ au plus proche des modèles de φ est grande. Ce principe est la base de la *fusion de croyances* à base de distances (dans l'ordre chronologique : [196], [170], [136], puis [134]). Soit d une pseudo-distance propositionnelle, c'est-à-dire une fonction de $\Omega \times \Omega \rightarrow \mathbb{N}$ telle que $d(\omega, \omega) = 0$ et $d(\omega, \omega') = d(\omega', \omega)$. La distance de la formule φ au monde ω est définie

par $d(\omega, \varphi) = \min_{\omega' \models \varphi} d(\omega, \omega')$ et la distance entre φ et φ' par $d(\varphi, \varphi') = \min_{\omega \models \varphi, \omega' \models \varphi'} d(\omega, \omega')$. On définit alors la *fusion* $Merge(\varphi_1, ..., \varphi_n)$ de n formules φ_1, ..., φ_n par : $Mod(Merge(\varphi_1, ..., \varphi_n)) = \{\omega \mid *_{i=1...n} d(\omega, \varphi)$ is minimal $\}$, où $*$ est une opération commutative, associative et ayant 0 pour élément neutre (les deux choix les plus évidents pour $*$ étant la somme et le maximum). Dans un travail avec Sébastien Konieczny et Pierre Marquis [134], on propose un modèle très général de la fusion à base de distances (qui comprend non plus une seule mais deux étapes d'agrégation, pour des raisons que je n'évoque pas ici) et on fait une étude systématique de sa complexité.

Au vu de qui précède, la fusion consiste donc à affaiblir les informations φ_i en les *dilatant* : l'information φ, qui exprimait que le monde réel était à coup sûr dans $Mod(\varphi)$, est affaiblie en considérant que plus ω est "loin de φ", moins il est plausible.

2.1.1.3 Affaiblissement par oubli de variables

Après l'affaiblissement par inhibition et l'affaiblissement par dilatation, voici l'affaiblissement par oubli de variables, proposé dans un article écrit avec Pierre Marquis [158]. L'idée sous-jacente est que l'incohérence d'une base de croyances peut être causée par un excès d'information sur certaines variables dont l'importance n'est peut-être pas cruciale. On peut alors se focaliser sur ces variables, simplifier les informations en oubliant tout ce qui concerne ces variables "responsables de l'incohérence" et parvenir ainsi à restaurer la cohérence.

Oublier un ensemble de variables dans une formule permet de l'affaiblir tout en gardant le maximum de croyances sur les variables non oubliées. On a en effet montré dans [155] que $\exists V.\phi$ est la formule la plus forte (en terme de conséquence logique) qui soit indépendante de V[7]. Par conséquent, si on a une base de croyances $B = \{\varphi_1, ..., \varphi_n\}$ incohérente, oublier certaines variables dans chacune des φ_i peut rétablir la cohérence : si $\vec{V} = \langle V_1, \ldots, V_n \rangle$ où pour tout i ; $V_i \subseteq Var(V_i)$, on définit $\exists \vec{V}.B = \{\exists V_1.\varphi_1 \wedge \ldots \wedge \exists V_n.\varphi_n\}$. Étant donnée une relation de préférence $\preceq$ sur l'ensemble des vecteurs de variables admissibles, la relation d'inférence "à base d'oubli" est définie par : $B \mid\sim^F_{\preceq} \psi$ si et seulement si on a $\exists \vec{V}.B \models \psi$ pour tous les vecteurs $\vec{V}$ minimaux (selon $\preceq$) tels que $\exists \vec{V}.B$ soit cohérent.

[7] On verra un peu plus loin dans le document qu'une formule est indépendante de V si et seulement si elle est équivalente à une formule dans laquelle aucune variable de V n'apparait.

Comme pour l'inhibition de formules, le choix des sous-ensembles de variables à oublier (donc celui de $\preceq$) peut être guidé par des priorités ou des pénalités sur les variables (traduisant leur degré de pertinence). On peut également imposer des contraintes plus complexes sur l'oubli de variables, telles que l'homogénéité (une variable oubliée dans une des φ_i doit être oubliée dans toutes les φ_i) ou des dépendances entre l'oubli d'une variable et l'oubli d'une autre qui n'aurait plus de sens en l'absence de la première. Enfin, on montre que l'affaiblissement par oubli de variables généralise à la fois les approches par inhibition et les approches par dilatation ; cette généralité ne s'accompagne heureusement pas d'un saut de complexité.

2.1.2 Affaiblir directement la déduction

On peut distinguer deux sous-familles de méthodes d'affaiblissement de la déduction. La première famille (logiques "quasi-classiques") reste proche de la logique classique en ce qu'elle définit des relations d'inférence qui coincident avec l'inférence de la logique propositionnelle lorsque la prémisse est cohérente. La seconde famille est celle des logiques paraconsistantes, qui interprètent les connecteurs logiques de façon fort différente de la logique classique et, par cela, s'en éloigne bien plus que les logiques quasi-classiques.

Dans cette section, $\vdash_{CL}$ dénote la relation d'inférence de la logique classique propositionnelle.

Inférences quasi-classiques et construction d'arguments

Une première façon d'affaiblir la déduction est d'autoriser les règles d'inférence classiques tant qu'elle ne risquent pas de mener à l'indésirable *ex falso quodlibet sequitur* (abrégé en *efqs*). Je reprends ici la terminologie de Besnard et Hunter [18] dans un sens plus général en définissant une relation d'inférence *quasi-classique* comme une relation d'inférence $\vdash_q$ telle que pour toute formule φ de $\mathcal{L}$ telle que $\nvdash_{CL} \neg\varphi$ et pour toute formule ψ de $\mathcal{L}$, on a $\varphi \vdash_q \psi$ si et seulement si $\varphi \vdash_{CL} \psi$. Bien entendu, $\vdash_{CL}$ elle-même est quasi-classique. Une relation d'inférence quasi-classique qui ne satisfait pas le principe *efqs* mais qui n'est cependant pas très satisfaisante est la relation dite "classique non-triviale", définie par $\varphi \vdash_{CNT} \psi$ si et seulement si $\varphi \vdash_{CL} \psi$ et $\nvdash_{CL} \neg\varphi$. Une inférence quasi-classique bien plus satisfaisante est celle $\vdash_Q$ de Besnard et Hunter [18] – qui s'appelle, justement, quasi-classique – et

qui, *grosso modo*, opère de cette façon : soient C_φ et C_ψ des formes normales conjonctives de φ et ψ au sens classique (c'est-à-dire en appliquant les règles de distributivité et d'idempotence de $\vee$ et $\wedge$ ainsi que les règles de De Morgan et l'élimination des doubles négations), alors $\varphi \vdash_Q \psi$ si et seulement si il existe une dérivation par résolution puis affaiblissement de C_ψ à partir de C_φ, où la résolution est (comme d'habitude) la règle $\frac{\alpha \vee \beta \quad \neg\alpha \vee \gamma}{\beta \vee \gamma}$ et l'affaiblissement est l'introduction de la disjonction $\frac{\alpha}{\alpha \vee \beta}$. Cette idée, simple et élégante, donne une relation d'inférence quasi-classique ne satisfaisant pas le principe *efqs* ; de surcroît, montrer que $\varphi \vdash_Q \psi$ où ψ est en CNF est un problème seulement coNP-complet (voir [179]). Ce résultat tranche avec les autres relations d'inférences étudiées dans cette section, dont la complexité est au-delà du premier niveau de la hiérarchie polynomiale.

Les *inférences non-signées* de Besnard et Schaub [22] sont elles aussi quasi-classiques. Elles consistent à d'abord mettre la base de croyance initiale Σ en NNF, puis à la traduire en Σ^{+}_{-} en remplaçant chaque occurrence d'un littéral positif (resp. négatif) x (resp. $\neg x$) par un nouveau symbole x^+ (resp. x^-), et à maximiser ensuite l'ensemble de défauts $\{(x \leftrightarrow x^+) \wedge (\neg x \leftrightarrow x^-)\}$; l'inférence sceptique non-signée consiste alors à inférer ψ si et seulement si ψ est prouvable dans toutes les extensions de la théorie de défauts ainsi obtenue (de façon similaire, on peut définir des inférences non-signées prudente et crédule).

C'est ce même principe – qui consiste à autoriser les règles de déduction classiques mais en imposant certaines limites – qui est à l'oeuvre dans les approches *argumentatives* de l'inférence. Ici, il s'agit d'autoriser la déduction $\Delta \vdash \psi$, où Δ est un ensemble fini de formules de $\mathcal{L}$ si (informellement) ψ a des arguments acceptables (construits à partir de Δ) suffisamment forts et si les arguments acceptables de $\neg\psi$ sont suffisamment faibles par rapport à ceux-là. Les quatre points qui sont à préciser sont :

1. quelle est la *structure* d'un argument ?
2. comment définir l'*acceptabilité* d'un argument ?
3. comment définir la *force* d'un argument ou d'un ensemble d'arguments ?
4. comment définir une inférence à partir des arguments acceptables ?

Je ne vais pas faire une revue exhaustive des reponses à ces questions qui me mènerait trop loin de mon propos ; je donnerai seulement quelques exemples qui ont été étudiés en profondeur par des chercheurs de l'IRIT.

Le plus souvent, un argument pour φ à partir de Δ est un sous-ensemble minimal A de Δ, cohérent et permettant de déduire ψ (mais des définitions plus complexes existent, notamment chez [19] où les argument ont une structure d'arbre). Je fais l'impasse sur les diverses définitions de l'acceptabilité d'un argument (souvent fondées sur l'absence d'arguments en conflit plus ou moins direct avec lui).

Si tous les arguments cohérents sont acceptables, alors la relation d'inférence $\vdash_A$ définie par $\Delta \vdash_A \psi$ si et seulement s'il existe un argument acceptable pour ψ à partir de Δ, coincide avec la relation d'inférence aventureuse $\vdash_{inc}^{\exists}$ ([41] et [19]). Plus généralement, Cayrol [41] montre qu'il existe de nombreuses correspondances similaires à la précédente entre l'inférence à partir de sous-bases maximales cohérentes et l'inférence à partir d'arguments acceptables[8].

La comparaison des arguments selon leur force permet de raffiner les inférences qui en découlent, selon le principe informel qu'on autorise l'inférence de ψ à partir de Δ si la force des arguments acceptables de ψ vérifie une certaine propriété de dominance sur la force des arguments acceptables de $\neg\psi$. On peut également définir une inférence valuée (à savoir, inférer ψ avec une certaine force qui est fonction de la force des arguments pour ψ et des arguments pour $\neg\psi$).

Qu'est-ce que la force d'un argument ? L'intuition dit que plus un argument est court (en nombre de formules de Δ), meilleur il est. Les probabilités le disent également, puisque si les sources qui ont fourni les informations de Δ sont pondérées par une probabilité de fiabilité identique p et que ces sources sont indépendantes, alors la probabilité a priori qu'un argument A soit pertinent est $p^{|A|}$. Sous ces mêmes hypothèses, on peut calculer la force probabiliste d'un ensemble d'arguments[9]. La définition inductive de la force

[8] Ces résultats pourraient nous inciter à mettre en doute notre taxonomie des relations d'inférence tolérant l'incohérence ; cependant, la taxonomie parait acceptable si l'on considère que les classes de relations d'inférence ainsi définies n'ont pas à être disjointes. Même si les relations définies sont souvent (quoique pas toujours) les mêmes, sur un plan conceptuel, il parait assez différent d'affaiblir Δ en inhibant des ensembles minimaux de formules et d'affaiblir la déduction à partir de Δ en n'autorisant que les déductions supportées par un argument acceptable, ce qui va dans le sens des distinctions faites par Del Val et Shoham [60], Cayrol [41] et Benferhat, Dubois et Prade [15] entre "approches basées sur la cohérence" et "approches fondationnelles" – la taxonomie que je suis en train d'élaborer s'inspire d'ailleurs de celle-là, en la généralisant.

[9] C'est particulièrement simple si l'on suppose de surcroît que les probabilités de fiabilité

d'un ensemble d'arguments proposée par Besnard et Hunter [19] est dans son esprit assez proche de cette interprétation probabiliste, même si elle ne se borne pas à préférer les arguments les plus courts.

Au lieu de rechercher simplement les arguments les plus courts, on peut introduire des priorités entre arguments qui sont induites de priorités définies a priori entre les informations de Δ (cette question a fait l'objet de nombreux travaux à l'IRIT, notamment [2]).

Logiques paraconsistantes

Contrairement à toutes les approches que l'on a évoquées jusqu'ici, les logiques paraconsistantes ne sont pas quasi-classiques, et ont par conséquent un comportement bien différent de celui de la logique classique, même lorsque les bases de croyances considérées sont cohérentes. Ce qui revient à dire que les logiques paraconsistantes ne se contentent pas d'échapper au principe *efqs*, mais elles interprètent différemment les connecteurs (en particulier, $a \rightarrow b$ n'est plus un raccourci syntaxique de $\neg a \vee b$). Ces logiques sont substructurelles, c'est-à-dire que leurs théorèmes sont aussi des théorèmes de la logique classique. Une famille bien connue de logiques paraconsistantes est l'ensemble des logiques C_i de Da Costa, en particulier la logique C_1. C_1 [55] garde de la logique classique tous les schémas d'axiomes qui n'ont rien à voir avec la négation, suivant l'idée que l'information positive est fondamentale (on en saura plus en lisant l'article de Besnard et Laenens [23] sur l'intérêt des logiques paraconsistantes en représentation des connaissances). Du fait que $a \wedge \neg a$ est une formule cohérente de C_1, et qu'elle n'est pas équivalente à $b \wedge \neg b$, non seulement C_1 échappe au principe *efqs* mais elle permet aussi de localiser l'incohérence sur les variables qui en sont responsables, et seulement sur celles-ci. On utilise souvent le raccourci syntaxique $a^\circ \equiv \neg(a \wedge \neg a)$ qui peut être inteprété comme "a n'est pas concerné par l'incohérence".

Une autre famille de logiques paraconsistantes (suivant l'article fondateur de Belnap [11]) est fondée sur des sémantiques multivaluées. Enfin, les *inférences signées* de Besnard-Schaub [22] sont définies en prenant le même point de départ que pour les inférences non-signées (voir paragraphe 2.1.2) mais où ψ est inférée (sceptiquement) de Σ si *sa traduction* φ^{+}_{-} est prouvable dans toutes les extensions de la théorie de défauts obtenue. Contrairement aux inférences non-signées, les inférences signées ne sont pas quasi-classiques.

sont infiniment proches de 1.

La complexité de toutes ces inférences est est étudiée en détail dans [53].

L'approche paraconsistante pour le traitement de l'incohérence procède donc par une localisation de l'incohérence plutôt que par son évitement (à l'opposé des approches par affaiblissement). Elle ne permet toutefois pas de graduer les incohérences, ce que permettent les approches à base de priorités, de pondérations ou de distances. Afin d'avoir un traitement à la fois local et graduel de l'incohérence, j'ai développé avec Philippe Besnard une approche pour intégrer la théorie des possibilités (qui est à la base des approches par priorités) et la logique C_1. Rappelons qu'une mesure de nécessité classique[10] est une fonction $N : \mathcal{L} \to [0,1]$ qui vérifie les trois propriétés suivantes : (i) les théorèmes ont un degré de nécessité maximal (1 par convention) ; (ii) deux formules équivalentes ont le même degré de nécessité ; (iii) la nécessité d'une conjonction est la minimum des nécessités de ses membres. Il suffit d'appliquer cette définition en faisant varier la notion de théorème (et d'équivalence logique) pour obtenir toute une famille de mesures de nécessité. Ainsi, une C_1-nécessité est une fonction $N : \mathcal{L} \to [0,1]$ vérifiant

(Taut) si $\vdash_{C1} \varphi$ alors $N(\varphi) = 1$;

(Eq) si $\vdash_{C1} \varphi \leftrightarrow \psi$ alors $N(\varphi) = N(\psi)$;

(Conj) $N(\varphi \wedge \psi) = \min(N(\varphi), N(\psi))$

La première partie de l'article [20] est écrite dans cette optique (on pourrait par exemple considérer des nécessités intuitionnistes traduisant la force d'une preuve constructive) tandis que la seconde, ainsi qu'un article ultérieur [21], sont focalisées sur les nécessités paraconsistantes et leur rôle dans le traitement local et graduel de l'incohérence. On a examiné les propriétés des C_1-nécessités, étudié leur sémantique, généralisé le principe de minimum de spécificité de la logique possibiliste [75], et surtout, on a montré leur intérêt pratique pour la fusion d'informations et le raisonnement à partir de règles par défaut (et aussi la gestion de préférences).

Conclusions

Je reporte dans le tableau qui suit les caractéristiques des différentes approches du traitement non-trivial de l'incohérence évoquées dans cette section. Une approche est graduelle si elle permet de comparer des incohérences selon leur force. Elle est locale si elle permet explicitement de repèrer les variables concernées par l'incohérence. Elle est quasi-classique si elle coïncide

[10] sans la normalisation $N(\bot) = 0$. C'est un détail sur lequel nous ne nous attardons pas.

avec la logique classique lorsque les données sont cohérentes. Enfin, nous faisons figurer le type de langage propositionnel nécessaire. Les approches dont l'input est un ensemble de formules qui n'est pas équivalent à sa conjonction nécessitent le symbole supplémentaire "virgule" dans la syntaxe – il peut être considéré, si l'on veut, comme un connecteur. Certaines approches nécessitent l'explicitation de priorités ou de données du même acabit (ordres, pondérations). [11].

	gradualité	localité	quasi-cl.	langage
affaiblissement par inhibition	oui	non	oui	$\mathcal{L}$ + virgule {+ priorités}
affaiblissement par dilatation	oui	non	oui	$\mathcal{L}$ + virgule + distance implicite
affaiblissement par oubli de variables	oui	oui	oui	$\mathcal{L}$ + virgule {+ priorités}
logique quasi-classique	non	non	oui	$\mathcal{L}$ (partie droite de $\vdash$ en CNF)
argumentation	oui	non	oui	$\mathcal{L}$ + virgule {+ priorités}
inférences BS non-signées	non	oui	oui	$\mathcal{L}$ [12]
inférences BS signées	non	oui	oui	$\mathcal{L}$
logiques paraconsistantes	non	oui	non[13]	$\mathcal{L}$
nécessités paraconsistantes	oui	oui	non	$\mathcal{L}$ + priorités

[11] Il manque à cette table récapitulative plusieurs points pertinents, concernant notamment la *réflexivité* des relations d'inférence (peut-on inférer φ à partir de φ ?) et leur caractère *sain* (peut-on inférer à la fois φ et $\neg\varphi$?). Plus généralement on peut examiner les propriétés des relations d'inférences associées à ces formalismes – voir par exemple[43] pour certains d'entre eux.

[13] Les inférences BS signées et non-signées font certes appel à de nouveaux symboles, mais ils ne sont que des outiles intermédiaires – ils ne figurent ni dans l'input ni dans l'output des relations d'inférence.

[13] Ce n'est cependant pas vrai pour les logiques paraconsistantes qui *minimisent* l'incohérence, comme la logique LP_m de Priest.

2.2 Raisonnement sur des croyances incertaines et raisonnement révisable

Comme l'incohérence, l'incertitude peut avoir de multiples sources qui nécessitent différents traitements.

1. *incertitude concernant les règles génériques* :

   – existence d'*exceptions* à une règle. C'est, en général (!) autant une spécialité liée à la représentation qu'une caractéristique épistémique : en effet, les exceptions (ou une partie des exceptions) peuvent être connues mais pour des raisons d'économie de représentation, ainsi que pour être en mesure d'appliquer la règle à un objet dont on ne connait pas la nature exacte, on a intérêt à les coder en tant que telles. C'est l'exemple classique des oiseaux : "sauf exception, les oiseaux volent" ne signifie pas que l'agent ne connait pas les espèces d'oiseaux non volantes, mais seulement qu'il est utile de pouvoir appliquer cette règle par défaut à un individu dont on sait seulement qu'il est un oiseau. Cette règle par défaut peut très bien coexister avec des règles plus spécifiques comme "sauf exception, les oiseaux d'Antarctique ne volent pas" ; la priorité entre les règles sera alors (en principe) déterminée par sa spécificité (bien entendu, les règles les plus spécifiques ont la priorité).

   – règle *conjecturelle* (dont la validité est incertaine). Ces règles n'ont pas nécessairement d'exceptions, mais la connaissance de l'agent est insuffisante pour qu'il puisse les étiqueter comme certaines. C'est typiquement le cas lorsque la règle est issue d'un processus d'apprentissage et que jusqu'à présent aucune exception n'a été rencontrée. La règle "[en vertu de ce que j'ai observé je conjecture que] les moutons sont blancs" deviendra de plus en plus plausible au fur et à mesure que l'agent (un jeune enfant, par exemple) observera des moutons différents et sera rejetée dès qu'il apercevra un mouton noir. Ce type de règle n'est pas nécessairement le résultat d'un processus d'apprentissage, cela peut être tout simplement seulement une croyance subjective : ainsi la règle "les problèmes NP-complets ne sont pas dans P". A-t-on observé un grand nombre de problèmes NP-complets dont on sait qu'ils ne sont pas dans P ? Certainement pas. Ici, on est même certain que l'une de ces deux règles $\forall x$ npc(x)

$\Rightarrow \neg$ p(x) et $\forall x$ npc(x) $\Rightarrow$ p(x) est valide (avec une nette préférence pour la première !).

2. *incertitude concernant les connaissances factuelles.* Les origines possibles de cette incertitude sont diverses :
   - possibilité de *mauvaise perception* d'une information communiquée par un capteur ou un autre agent. On pourra alors avoir recours à des mesures de fiabilité.
   - possibilité de *mauvais fonctionnement* d'un composant dans un problème de diagnostic. Là encore, il s'agit de degrés de fiabilité.
   - croyances subjectives sur la réalisation d'un évènement passé ou futur.
   - instanciation d'une règle générique avec exceptions à un objet non complètement identifié.

Pour ce qui est des croyances factuelles, on peut envisager des approches à base de minimisation comme la logique des défauts, ou bien des approches quantitatives (par exemple, lorsqu'on dispose d'informations statistiques sur les pannes d'un composant, il serait dommage de s'en priver). La différence entre les deux types d'approches tient surtout à la quantité d'information disponible et à son caractère objectif ou subjectif.

Pour ce qui est des règles génériques, c'est plus complexe : il faut pouvoir être capable de distinguer entre règles avec exceptions et conjectures. Le calcul des probabilités permet de faire la différence entre les deux, grâce à l'existence de conditionnels : ainsi, il est aisé de distinguer entre $\forall$`x Prob(vole(x) | oiseau(x)) = 1 - `ε et `Prob(`$\forall$`x npc(x)` $\Rightarrow \neg$ `p(x)) = 1 - `ε . Cette distinction fondamentale entre implication et conditionnement est à l'origine du travail sur les objets conditionnels par Dubois et Prade [78]. Ces objets conditionnels, dont la sémantique est trivaluée ("vrai", "faux", "ne s'applique pas"), permettent de représenter des règles génériques avec exceptions de façon satisfaisante dans un cadre purement qualitatif tout comme dans un cadre quantitatif.

Abordons maintenant des aspects plus techniques de la représentation et du raisonnement avec des croyances incertaines. Les problèmes qu'on est amené à se poser concernent l'économie de la représentation (on en a déjà parlé au chapitre 1), la révision des croyances incertaines, et l'inférence. Pour cette dernière, on sera amené à distinguer les *inférences valuées* consistant à

inférer des connaissances sur la mesure de l'incertitude d'une formule donnée ("que peut-on dire de la probabilité de φ ?") des *inférences brutes* dont l'output est en oui-non ("étant donné la connaissance disponible, acceptons-nous φ comme plausible ?"). Tandis que les premières sont souvent monotones, les secondes sont non-monotones.

La suite de la section est focalisée sur mes travaux et s'articule selon un plan qui a trait aux outils employés plutôt qu'aux problématiques. Je parlerai d'abord des logiques pondérées pour la représentation de mesures d'incertitude, puis du raisonnement non-monotone à base de relations d'ordre et de ses liens avec les logiques des conditionnels.

2.2.1 Logiques pondérées pour la représentation de croyances d'incertaines

Intuitivement, une logique pondérée permet de représenter des contraintes sur la mesure de l'incertitude de formules données et d'inférer des connaissances sur l'incertitude d'autres formules. On verra au chapitre 4 que les logiques pondérées s'appliquent également à la représentation de préférences.

Une *théorie de l'incertain* est une famille $\mathcal{F}$ de mesures d'incertitude à valeurs dans une échelle Val – on choisit souvent $[0, 1]$ –, définie par une liste d'axiomes.

Une *logique pondérée pour l'incertitude* consiste en la donnée d'une théorie de l'incertain $\mathcal{F}$ et d'un langage logique $\mathcal{L}$ (propositionnel ou du premier ordre). Le langage d'une logique pondérée est constitué de couples $\langle \varphi, A \rangle$, où φ est une formule propositionnelle de $\mathcal{L}_{VAR}$ et A un sous-ensemble de Val. Si $f \in \mathcal{F}$ alors f satisfait $(\varphi\ ,\ A)$ si et seulement si $f(\varphi) \in A$. Si Φ est un ensemble de tels couples alors $\Phi \models \langle\ \psi\ ,\ X\ \rangle$ si et seulement si tout $f \in \mathcal{F}$ satisfaisant chacun des couples de Φ satisfait aussi $\langle\ \psi\ ,\ X\ \rangle$. On peut aussi parfois définir une relation d'inférence non-monotone qui nécessite la donnée d'une relation de préférence $\preceq$ sur $\mathcal{F}$: $\Phi \mid\!\approx \psi$ si et seulement si $Min(\preceq, Mod(\Phi)) \subseteq Mod(\langle \psi, X \rangle)$ – la relation de préférence peut être, par exemple, fondée sur la spécificité ou sur le degré d'incohérence lorsque $\mathcal{F}$ est la théorie des possibilités ou sur l'entropie lorsque $\mathcal{F}$ est la théorie des probabilités.

En voici des exemples. La logique possibiliste, que j'ai étudiée pendant ma thèse [144], est un exemple de logique pondérée où $\mathcal{F}$ est la famille des

mesures de nécessité sur $\mathcal{L}_{VAR}$. Ses variantes DISCRIMIN et LEXIMIN sont également des logiques pondérées, mais avec des mesures d'incertitude un peu plus complexes et n'étant plus à valeurs sur $[0, 1]$ mais sur des ensembles de vecteurs de $[0, 1]$. La logique des pénalités, la logique probabiliste [58, 187], et les logiques à base de fonctions de croyance comme celle de Saffiotti [201] sont des logiques pondérées pour l'incertitude.

Quelques mots sur les aspects computationnels. J'ai écrit un article de synthèse sur l'algorithmique de la logique possibiliste [148] (qui reprend pour l'essentiel des résultats que j'ai obtenus pendant ma thèse). Des résultats en ce qui concerne la logique des pénalités figurent dans le rapport de DEA de Florence Dupin de Saint-Cyr [81]. Enfin, on peut voir l'inférence dans certaines logiques pondérées comme un cas particulier de fusion à partir de distances, et ainsi réutiliser des résultats de complexité de l'article [134] – ceci dit, une étude plus générale de la complexité des logiques pondérées est une perspective fort intéressante.

2.2.2 Logiques des conditionnels et relations d'inférence non-monotone

C'est encore une fois le principe d'économie de représentation qui est la source du travail que j'ai effectué en collaboration avec Luis Fariñas del Cerro et Andreas Herzig [102, 103]. Les travaux de Kraus, Lehmann et Magidor d'une part [138], de Gärdenfors et Makinson d'autre part [107] ont montré que les relations d'inférence non-monotones vérifiant un certain nombre de "bonnes" propriétés sont nécessairement fondées sur une relation de préordre entre mondes ou entre formules. En particulier, les relations dites *comparatives* de Gärdenfors et Makinson sont fondées sur un préordre total $\geq$ sur $\mathcal{L}_{VAR}$ qui correspond (à un détail peu pertinent près) à la contrepartie qualitative des mesures de possibilité [67] (et que nous appellons *ordre de possibilité*) [14] L'inférence à partir d'un ordre de possibilité, dite *inférence comparative*, est définie par Gärdenfors et Makinson comme suit : $F \mid\!\approx_{\geq} G$ si et seulement si soit $F \vdash G$ soit $F \wedge G > F \wedge \neg G$.

Le problème de cette définition est qu'elle suppose connu l'ordre de possibilité et que la donnée explicite de cet ordre a, même dans le cas fini, un

[14] La donnée d'un ordre de possibilité est équivalente à la donnée d'un préordre total sur Ω dans le cas où $\mathcal{L}_{VAR}$ est généré par un nombre fini de symboles propositionnels mais pas dans les autres cas (voir là encore [67]).

coût exorbitant. C'est pourquoi nous nous sommes attachés à générer des relations d'inférence non plus à partir d'ordres de possibilité explicites mais à partir d'ensemble E de contraintes sur cet ordre (de la forme $A \geq B$ ou $A > B$) qui définissent implicitement un ensemble d'ordres de possibilité. La relation d'inférence non-monotone induite par E est définie comme suit : $A \mid\sim_E B$ si et seulement si pour tout ordre de possibilité $\geq$ satisfaisant E, on a $A \mid\sim_{\geq} B$. On montre que ces relations d'inférence ne sont en général plus comparatives (elles échappent à la règle de *monotonie rationnelle*) mais seulement *préférentielles* au sens de [138] – voir aussi [78].

Enfin, nous avons montré que déterminer si $A \mid\sim_E B$ est équivalent à un test de validité dans la logique des conditionnels VA de Lewis. On montre d'abord qu'un ordre de possibilité $\geq$ est équivalent à un système de sphères "absolu" $M_{\geq}$, puis on donne deux théorèmes de représentation utilisant la sémantique des sphères : le premier dit que $A \mid\sim_{\geq} B$ si et seulement si soit $A \vdash B$ soit $M_{\geq} \models A \succ A \wedge \neg B$; et le second, que $A \mid\sim_E B$ si et seulement si soit $A \vdash B$ soit $\mathcal{E} \models A \succ A \wedge \neg B$, où $\mathcal{E}$ est l'ensemble de formules conditionnelles représentant les contraintes de E. Ces résultats sont établis sans l'hypothèse de finitude de l'ensemble des symboles propositionnels.

2.3 Raisonnement sur l'indépendance

Il est de nombreux problèmes de raisonnement où il est primordial de déterminer et/ou d'exploiter des relations de dépendance entre variables propositionnelles.

Il y a deux façons de raisonner avec la dépendance.

La première suppose que les relations de dépendance (entre variables, entre actions et variables, entre thèmes et variables) sont explicites (donc faisant partie de l'input) et qu'on les exploite ensuite pour diverses tâches (notamment, la révision, la mise-à-jour, la qualification). C'est le point de vue développé par Fariñas del Cerro et Herzig [101] ainsi que par Castilho, Gasquet et Herzig [39] : ainsi, lorsqu'on met à jour une base de croyances par une nouvelle information, ou lorsqu'on exécute une action dans un état de croyances donné, l'état de croyances résultant est obtenu en supposant que seules les variables avec lesquelles la nouvelle information interfère peuvent changer de valeur de vérité [101].

La seconde suppose que les relations de dépendances (entre variables, entre actions et variables, entre formules et variables, entre formules) sont implicites. Elles ne font pas partie de l'input mais sont déterminées à partir de l'input, qui est une base de croyances (que l'on supposera propositionnelle sauf indication contraire) Σ. C'est cette seconde approche que nous avons développée avec Pierre Marquis et Paolo Liberatore [155, 151, 152].

2.3.1 Indépendance formule-variable

Il est ici question de savoir quelles sont les variables propositionnelles dont dépend une formule (propositionnelle) donnée, ou encore, dont elle "parle". Il est évident que $\varphi = a \wedge (b \vee \neg b)$ parle de a, mais parle-t-elle de b ? Oui si l'on considère la relation de dépendance purement syntaxique "apparait dans". Non si l'on considère la relation de dépendance sémantique définie comme suit [155] : la formule φ est indépendante de la variable v si et seulement si elle peut être réecrite en une formule équivalente φ' dans laquelle v n'apparait pas. On peut par ailleurs la raffiner en distiguant dépendances positives et négatives, c'est-à-dire en définissant une relation d'indépendance *formule-littéral* : φ est indépendante du littéral l si et seulement si elle peut être réécrite en une formule équivalente φ' en forme normale négative dans laquelle le littéral l n'apparait pas. On notera $DepVar(\varphi)$ (respectivement $DepLit(\varphi)$) l'ensemble des variables (respectivement, des littéraux) sémantiquement dépendant(e)s de φ.

Dans [152] on étudie en détail la complexité algorithmique de ces relations (ainsi, les indépendances sémantiques formule-variable et formule-littéral sont, malgré leur aspect extérieur de simplicité, coNP-complètes), on en a donné des caractérisations, en particulier en utilisant les impliquants premiers et l'oubli de variables, et on a montré que ces notions, ou des notions très proches, se retrouvaient dans plusieurs travaux indépendants sous des noms différents (par exemple, *influençabilité* ou *pertinence*). Cette notion d'indépendance jouera un rôle important au chapitre 3 (à propos du raisonnement sur l'action et de la mise-à-jour). L'indépendance formule-variable et les notions en relation avec elle (en particulier l'oubli de variables et de littéraux) ont fait l'objet de l'article [152].

2.3.2 Indépendance conditionnelle

L'indépendance conditionnelle entre formules propositionnelles est la contrepartie logique de l'indépendance probabiliste. Elle a été introduite par Dar-

wiche [56] à la suite de ses travaux avec Pearl sur les réseaux causaux [57]. La définition de Darwiche est la suivante : si X, Y et Z sont des ensembles deux-à-deux disjoints de variables propositionnelles et Σ une formule propositionnelle, alors X et Y sont indépendants selon Σ étant donné Z si et seulement si pour toutes affectations complètes $\vec{x}$, $\vec{y}$ et $\vec{z}$ des variables de X, Y et Z respectivement, la cohérence de $\vec{x} \wedge \vec{z} \wedge \Sigma$ et de $\vec{y} \wedge \vec{z} \wedge \Sigma$ implique la cohérence de $\vec{x} \wedge \vec{y} \wedge \vec{z} \wedge \Sigma$. On a étudié la complexité des ces relations d'indépendance conditionnelle (qui se situe en en général au second niveau de la hiérarchie polynomiale, sauf hypothèses supplémentaires sur X, Y , Z et/ou sur Σ). On a aussi montré que bien souvent, ce n'est pas cette notion d'indépendance dont on a besoin mais une notion plus forte (et justement appelée indépendance conditionnelle forte) : X et Y sont fortement indépendants selon Σ étant donné Z si et seulement si ils sont indépendants selon Σ étant donné Z' pour tout $Z' \subset Z$. Cette notion d'indépendance n'est pas moins complexe que la précédente (elle l'est même parfois plus) mais elle a une caractérisation agréable en termes d'impliqués premiers que n'a pas la précédente (X et Y sont fortement indépendants selon Σ étant donné Z si et seulement si il n'existe pas d'impliqué premier de Σ mentionnant (i) au moins une variable de X, (ii) au moins une variable de Y et (iii) aucune variable en-dehors de $X \cup Y \cup Z$). Une telle caractérisation permet de calculer l'indépendance à partir d'une compilation de Σ. On a donné une troisième définition, encore plus forte, que nous ne donnons pas ici. On a également étudié ces notions sous l'angle des propriétés des graphoïdes (propriétés que devraient satisfaire, en principe, des relations d'indépendance conditionnelle) et on a relié ces notions d'indépendance à des notions voisines apparues indépendamment dans la littérature (nouveauté, relevance entre sujets) ainsi qu'aux notions d'indépendance conditionnelle probabiliste, ordinale (comme dans les travaux de Ben Amor et al. [12]) ou préférentielle (voir [4]). Pour tout ceci voir l'article [151].

Chapitre 3

Raisonnement sur l'action, le changement, le temps et l'espace

Dans le chapitre précédent, le système était statique ; dans ce chapitre on va considérer des systèmes en évolution, sauf en Section 4 où on parlera d'"évolution" spatiale ; bien qu'il ne s'agisse pas d'évolution au sens propre du terme, on choisit de traiter le raisonnement sur le changement et l'espace dans ce chapitre, parce que les problématiques traitées (persistance, minimisation des changements, dépendances, problème du décor et ramification) sont très similaires à celles du raisonnement sur le temps et l'action.

Un mot maintenant sur la terminologie "raisonnement sur le temps et le changement". Dans la communauté de l'intelligence artificielle, l'expression "raisonnement temporel" s'applique à l'étude des relations qui existent entre objets temporels (points, intervalles), et à l'inférence à partir de telles relations (ce que nous n'aborderons pas dans ce mémoire) ; alors que l'expression "raisonnement sur le temps" concerne des systèmes en évolution, et s'intéresse de façon essentielle à l'étude de la vérité de propositions à travers le temps (et comprend en particulier le raisonnement sur l'action et le changement). Les deux expressions sont souvent le nom de deux sessions différentes dans les conférences généralistes d'intelligence artificielle ("raisonnement temporel" d'une part, "raisonnement sur [le temps], l'action, le changement" d'autre part). Ces domaines sont si distincts que je connais peu de travaux qui se situe réellement à leur intersection. Cette double terminologie peut d'ailleurs s'étendre à l'espace : le "raisonnement spatial" s'intéresse aux relations entre

objets dans l'espace tandis que le "raisonnement sur l'espace" traitera des changements d'une région à l'autre de l'espace.

Dans ce chapitre, on va s'intéresser aux problèmes suivants :

1. étant donné la description d'un état de connaissances et des effets d'une action sans feedback, quel est l'état de connaissances résultant ? En d'autres termes, on s'intéresse à la *projection*, ou encore *progression*, d'un état de connaissance par une action. Les approches abondent : les "langages d'action", la mise-à-jour des croyances, et, bien avant eux, la description d'actions pour la planification, abordent cette question. On va voir qu'ils l'abordent en fait de façon assez proche, et, après avoir montré les liens qui existent entre ces différentes approches, je tenterai d'expliquer pourquoi elles ont été développées de façon relativement séparée. *Je ne parlerai pas d'actions épistémiques dans ce chapitre*, avant tout pour des raisons d'équilibres des différentes parties de ce mémoire : les actions épistémiques, leurs effets, leur rôle en planification, feront l'objet du chapitre 5.
2. étant donné une suite d'états de connaissances à différents instants, comment peut-on "compléter" ces états de connaissances par des hypothèses sur l'évolution du système ? Cette question et la précédente sont toutes deux des instances de ce que Sandewall appelle la "complétion de scénario". Je vais montrer que cette seconde question (où l'on raisonne sur des observations en absence d'actions) est radicalement différente de la précédente (où l'on raisonne sur des effets d'actions en l'absence de feedback, c'est-à-dire d'observations – à l'exception de l'instant initial), et, en particulier, *il n'est pas question d'essayer de la traiter avec la mise-à-jour des croyances.* Cette dernière affirmation va nous conduire à une réflexion sur les différences ontologiques entre révision, mise-à-jour, certaines de leurs extensions (révision itérée, mise à jour généralisée, et ce que l'on appellera extrapolation), et il nous faudra au passage tordre (un peu) le cou à l'adage par trop simpliste "la révision s'occupe d'un monde statique et la mise-à-jour d'un monde en évolution".
3. enfin, on va voir que les problèmes et les méthodes liés au raisonnement sur l'action et le changement (minimisation des changements, problème du décor et de la ramification, non-déterminisme, concurrence, révision, mise-à-jour, distinction entre actions et événements, entre effets d'actions et observations) ont tous une une signification qui va au-delà du raisonnement sur un monde dynamique (au sens premier du terme) :

elles s'appliquent dès qu'il s'agit de raisonner sur l'état du monde en plusieurs contextes dépendants, que ces "contextes" soient des instants ou des intervalles, des points ou des régions de l'espace, des classes d'objets, des situations etc.

3.1 Action et mise-à-jour

Dans cette section nous faisons les hypothèses suivantes :

- il n'y a pas de *choix d'action* par l'agent ; ce dernier se contente de raisonner sur les effets attendus d'une action, sans que l'on tienne compte de ses préférences ni du processus qui l'a conduit à choisir cette action.
- les actions considérées sont seulement effectives, et sans feedback (donc sans effet sur les croyances de l'agent autres que la projection des effets sur le système).
- le temps est implicite (il intervient simplement parce qu'on distingue l'état qui *précéde* l'exécution d'une action de l'état qui en *résulte*).

Il s'agit donc, pour l'agent, de raisonner sur l'évolution du système entre un instant initial et un instant final, en fonction des croyances dont il dispose sur l'état initial, sur l'état final et sur ce qui s'est passé entre ces deux instants (action(s) ou événement(s))[1].

La *mise-à-jour des croyances* est un processus qui permet d'intégrer à la base de croyances un *changement* de l'état du système explicitement spécifié par une formule propositionnelle. Cette définition est encore pour l'instant un peu vague mais nous y reviendrons un peu plus tard. La mise-à-jour des croyances s'oppose à la *révision des croyances* où l'on intègre une nouvelle information sur le système à une base de croyances sur ce *même* système, c'est-à-dire en supposant qu'il n'a pas évolué. Cette distinction a été mise au clair par Katsuno et Mendelzon [131], bien que la mise-à-jour y fût antérieure [223, 118, 224].

Les articles liant explicitement mise-à-jour et raisonnement sur l'action ne sont pas légion. Une raison possible est que les travaux sur la mise-à-jour,

[1] Remarquons que puisque la phase de choix de l'action est ignorée, cela revient au même de confondre l'action avec un événement, dont l'occurrence est indépendante de la volonté de l'agent, et qui aurait les mêmes effets que l'action considérée. Ceci implique que les "langages d'actions" s'appliquent tout aussi bien à la description d'effets d'événements.

à commencer par ceux de Winslett puis de Katsuno et Mendelzon, se préoccupent principalement de sémantique tandis que les langages pour raisonner sur l'action s'intéressent avant tout à l'expression efficace et concise des effets d'une action. Katsuno et Mendelzon montrent que tout opérateur de mise-à-jour "convenable" (c'est-à-dire satisfaisant un certain nombre de postulats) est fondé sur une famille $\{\geq_\omega, \omega \in \Omega\}$ de préordres sur Ω exprimant les croyances sur l'évolution plausible du système. Si le langage contient n symboles propositionnels, la représentation explicite d'une telle collection de préordres nécessite un espace en $\mathcal{O}(2^{2n})$, ce qui implique la nécessité d'élaborer des langages compacts pour exprimer cette dynamique. Cette problématique, toutefois, a été un peu négligée dans les travaux concernant la mise à jour alors qu'ils sont centraux en ce qui concerne les langages d'action.

Un des rares articles à traiter à la fois de raisonnement sur l'action et de mise à jour est celui de Brewka et Hertzberg [35], qui définissent un langage de représentation compact au moyen de couples $\langle$ précondition, postcondition $\rangle$, où les préconditions sont mutuellement exclusives. Ces couples sont appelés des *transitions*. Les transitions permettent une description concise et intuitive de l'évolution du système (effets des actions ou évolution "naturelle").

3.1.1 Description compacte d'opérateurs de mise à jour : le modèle des transitions

L'article que j'ai écrit avec Marie-Odile Cordier fait suite à un article de Cordier et Siegel [47] proposant un langage à base de transitions à priorité pour la mise-à-jour. Certaines de ces transitions sont simplement des transitions de persistance : si `l` est un littéral, $\langle$ `l, l` $\rangle$ exprime que par défaut, `l` persiste d'un instant à l'autre. Ces transitions peuvent exprimer des effets d'actions comme des changements inhérents à l'évolution naturelle du système.

On ne suppose plus, comme dans [35] que les préconditions sont mutuellement incohérentes, ce qui rend possible l'existence de transitions simultanément applicables dont les effets se contredisent (soit que l'on exécute simultanément deux actions dont les effets s'opposent, soit que l'on exécute une action dont certains effets s'opposent à l'évolution naturelle du système). En conséquence, il arrive que l'ensemble des parties droites des transitions "applicables" soit incohérent. Ce formalisme nécessite par conséquent un mécanisme de gestion de l'incohérence. L'article [46] montre que ce mécanisme,

qui opère par maximisation de l'ensemble des transitions satisfaites, est parallèle à celui mis en oeuvre dans les approches du traitement de l'incohérence par inhibition de formules (cf. chapitre 2) et dans les opérateurs de "base revision" [185].

Lorsqu'on fait l'hypothèse (très restrictive) que l'état de croyances initial est complet, il est alors possible de traduire un opérateur de mise à jour à base de transitions en un opérateur de "base revision" [46]. Cette traduction nécessite une réification temporelle du langage propositionnel initial (que nous définissons ici parce que nous en aurons besoin à plusieurs reprises dans le document)[2] :

Réification temporelle : la réification temporelle d'un langage $\mathcal{L}_{VAR}$ selon un ensemble fini d'instants $\mathcal{H}$ est le langage propositionnel $\mathcal{L}_{VAR,\mathcal{H}}$ généré par l'ensemble de symboles propositionnels $\{x_t, x \in VAR, t \in \mathcal{H}\}$ (parfois notés $[t]x$) ; si $\varphi \in \mathcal{L}_{VAR}$ et $t \in \mathcal{H}$ alors φ_t (ou encore $[t]\varphi$) est la formule de $\mathcal{L}_{VAR,\mathcal{H}}$ obtenue en remplaçant chaque occurrence de toute variable x dans φ par x_t.

Le modèle de transitions TR est alors traduit en la base de croyances $B_{TR} = \{\varphi_0 \rightarrow \psi_1, \mid \langle \varphi, \psi \rangle \in TR\}$ et la mise-à-jour d'une base de croyances complète KB par α selon TR est identique à la "base revision" de B_{TR} par $KB_0 \wedge \alpha_1$. Cependant, l'intérêt de cette transformation est limité par l'hypothèse que l'état de croyance initial est complet. Ce n'est pas une surprise : c'est la condition nécessaire pour que révision et mise à jour puissent coïncider.

3.1.2 Mise-à-jour à base de dépendance et langages d'action

Mise-à-jour et dépendance

L'approche précédente est à base de *minimisation*, non pas seulement des changements comme dans la *PMA* de Winslett [224] mais des transitions non satisfaites, ce qui est plus général – d'ailleurs la *PMA* est un cas particulier des mise-à-jour à base de transitions. Cependant, la minimisation des changements n'est pas toujours souhaitable pour la mise-à-jour. En particulier, Herzig et Rifi [126] ont montré que les approches fondées sur la

[2]La réification temporelle est quelque chose de très classique (elle est à la base, par exemple, du codage en logique propositionnelle de problèmes de planification comme dans le cadre SATPLAN sur lequel nous reviendrons au chapitre 6).

minimisation des changements traitent mal le problème de la mise-à-jour par une disjonction ; plus formellement, un opérateur de mise-à-jour qui satisfait les postulats de Katsuno et Mendelzon *ne peut pas* traiter efficacement la disjonction (le postulat responsable est (U5)).

C'est pour cela que plusieurs chercheurs ont étudié des opérateurs de mise à jour non fondés sur la minimisation, et en particulier, la famille d'opérateurs de mise-à-jour *à base de dépendance* : une telle mise à jour d'une base de croyances B par une formule A consiste à d'abord "oublier" dans B toutes les informations "qui concernent" A (laissant inchangées les valeurs de vérité des variables qui ne sont pas concernées par la mise-à-jour), puis à ajouter A au résultat. Il reste maintenant à préciser ce qu'il faut entendre par "les informations qui concernent la formule A", et cette notion de pertinence est induite par une relation de dépendance entre formules : A concerne B si et seulement si B dépend de A ; la généralité de l'approche tient au fait que cette notion de dépendance entre formules peut varier.

La quasi-totalité des approches de la mise à jour à base de dépendance considère que cette relation de dépendance formule-formule est induite à partir d'une relation de dépendance *formule-variable* $Dep \subseteq \mathcal{L}_{VAR} \times VAR$, qui est ensuite ainsi étendue à $\mathcal{L}_{VAR} \times \mathcal{L}_{VAR}$: A et B sont indépendants si et seulement si il existe au moins une variable dont A et B dépendent. On revient donc à la notion de relation formule-variable dont nous avons déjà parlé en Section 2.3.1. On peut par exemple choisir

- la dépendance implicite syntaxique : $Dep(A)$ est l'ensemble des variables apparaissant dans A ;
- la dépendance implicite sémantique (voir Section 2.3.1)[3]
- la dépendance explicite syntaxique [120] : on se donne a priori une relation de dépendance (symétrique ou non) $dep \subseteq VAR \times VAR$ entre *variables* (deux variables sont dépendantes si, intuitivement, elles concernent le même thème) et on définit alors Dep par $Dep(A) = \cup_{v \in Var(A)} dep(v)$.
- la dépendance explicite sémantique : $Dep(A) = \cup_{v \in DepVar(A)} dep(v)$.

Quel que soit le choix de Dep, la mise à jour d'un monde ω par une formule A par rapport à Dep, notée $\omega \diamond_{Dep} A$, est l'ensemble de tous les mondes ω' tels que (i) $\omega' \models A$, et (ii) pour toute variable propositionelle x de PS

[3] L'opérateur de mise-à-jour ainsi obtenu a été proposé indépendamment par Doherty et al. [61] sous le nom de $MPMA$ et par Herzig et Rifi [126] sous le nom $WSS \downarrow$.

telle que $x \notin Dep(A)$, ω et ω' donnent la même valeur de vérité à x – on dit que $\diamond_{Dep}$ est l'opérateur DFV-mise à jour induit par *dep*. Les opérateurs de DFV-mise à jour peuvent être calculés grâce à la notion d'oubli de variable (voir en Section 2.3.1) : $A \diamond_{Dep} \alpha \equiv (\exists Dep(\alpha).A) \wedge \alpha$.

Si ces opérateurs de DFV-mise à jour sont satisfaisants en bien des points (en particulier en ce qui concerne leur complexité), ils présentent un inconvénient important : oublier toutes les informations relatives aux *variables* concernées par la mise à jour mène souvent à *oublier trop de choses*. Considérons par exemple un robot à qui l'on demande d'aller dans une pièce dans laquelle il y a deux portes (1 et 2), et de faire en sorte que l'une des deux portes soit rouge (éventuellement, par l'action de peindre l'une des portes), ce qui peut être représenté par une mise à jour par $\texttt{red1} \vee \texttt{red2}$. Supposons maintenant que les deux portes soient initialement rouges : $B = \texttt{red1} \wedge \texttt{red2}$. Alors, en utilisant n'importe quelle relation de dépendance formule-variable *Dep* telle que $\texttt{\{red1, red2\}} \subseteq Dep(\texttt{red1} \vee \texttt{red2})$, on obtient $B \diamond_{Dep} (\texttt{red1} \vee \texttt{red2}) = \texttt{red1} \vee \texttt{red2}$. Ainsi, on a oublié que les deux portes étaient initialement rouges. Ce n'est évidemment pas ce que l'on attendait : intuitivement, on n'aurait pas dû oublier que les portes 1 et 2 étaient rouges, parce qu'une mise à jour par $\texttt{red1} \vee \texttt{red2}$ n'a pas d'influence *négative* sur red1 ni sur red2. Par conséquent, seules les occurrences *négatives* de red1 et red2 devraient être oubliées avant l'ajout de la nouvelle formule, pas les occurrences positives.

Puisque nous avions proposé, avec Pierre Marquis, un raffinement de la dépendance formule-variable à la dépendance *formule-littéral* (voir [155], [151] et la section 2.3.1), nous nous sommes alors demandé s'il serait pertinent d'exploiter cette dépendance pour la mise à jour . Nous avons ainsi défini, dans l'article [124] (écrit avec Andreas Herzig, Pierre Marquis et Thomas Polacsek) la mise à jour à base de dépendance formule-littéral (ou DFL-mise à jour), par

$$B \diamond_{Dep} A \equiv A \wedge ForgetLit(B, NEG(Dep(A)))$$

où Dep est une fonction de $PROP_{PS}$ dans 2^{LIT}.

Les résultats que nous avons obtenus ont été très encourageants (voir la Section 3 de l'article [124]). L'utilisation de la dépendance formule-littéral (au lieu de la dépendance formule-variable) pour la mise à jour à base de dépendance permet non seulement d'éviter le problème de l'oubli excessif

mentionné plus haut, mais permet aussi de tenir compte des littéraux persistants, c'est-à-dire des littéraux qui restent vrais en toute circonstance : ainsi, $Dep(A) = DepLit(A) \cap NEG(Pers)$ où $Pers$ est un ensemble de littéraux persistants, est une fonction de dépendance intéressante. Enfin, la mise à jour à base de dépendance formule-littéral généralise la mise à jour à base de dépendance formule-variable (puisque le premier type de dépendance généralise le second, comme on l'a vu en Section 2.3.1), et cette généralisation ne conduit pas à un saut de complexité [124] : étant données trois formules A, B et C, décider si $B \diamond_{Dep} A \models C$ est "seulement" coNP-complet (soit un problème moins complexe qu'avec la mise à jour à base de minimisation, qui donne en général des problèmes au second niveau de la hiérarchie polynomiale).

3.1.3 Mises-à-jour conditionnelles, non-déterministes et concurrentes

Si la DFL-mise à jour permet un meilleur traitement de la persistance (et donc du problème du décor) que les opérateurs de mise à jour usuels, elle est encore loin de pouvoir représenter des effets complexes d'action, pour plusieurs raisons : (i) elle ne permet pas la prise en compte d'effets conditionnels ; (ii) elle permet certes la représentation d'une certaine forme de non-déterminisme, par la prise en compte des disjonctions, mais d'autres formes de non-déterminisme sont mal traitées, parce qu'elles ne peuvent pas se ramener simplement à une disjonction ; (iii) elle ne permet pas la prise en compte d'actions concurrentes.

Ces défauts nous ont amenés à étendre les opérateurs de mise-à-jour en considérant que la "formule d'input" n'est plus une simple formule propositionnelle, mais des constructions plus complexes que l'on appelle "input étendus", définis inductivement comme suit : toute formule A de $\mathcal{L}_{VAR}$ est un input étendu et si Φ, Ψ sont deux inputs étendus et $C \in \mathcal{L}_{VAR}$ alors $\Phi \cup \Psi$ (nondéterminisme), $\Phi || \Psi$ (concurrence) et `if` C `then` Φ `else` Ψ (conditionnel) sont des inputs étendus.

Mises-à-jour non-déterministes

$\omega \diamond (\Phi \cup \Psi)$ est une mise à jour non-déterministe : ω est "non-déterministiquement" mise-à-jour par Φ ou par Ψ, c'est-à-dire , formellement :

$$\omega \diamond (\Phi \cup \Psi) = (\omega \diamond \Phi) \cup (\omega \diamond \Psi)$$

Cette définition a été donnée pour la première fois par Brewka et Hertzberg [35], qui ont observé que $\omega \diamond (A \cup B)$ est en général différent de $\omega \diamond (A \vee B)$. Dans le cas où $\Phi = A$ et $\Psi = B$ sont de simples formules propositionnelles, la mise à jour de ω par $A \cup B$ signifie intuitivement qu'une action a été exécutée, dont l'effet possible est soit A soit B, et que (je cite [35]) "le planificateur n'a aucun moyen de savoir si l'une des postconditions est déjà satisfaite, et choisit de façon non-déterministe de mettre-à-jour par A ou par B". Un exemple simple est l'action `jeter-une-pièce`, représentable par l'input étendu `pile` $\cup \neg$ `pile`. [4]

Mises-à-jour conditionnelles

$$\omega \diamond \ (\texttt{if } C \texttt{ then } \Phi \texttt{ else } \Psi) = \begin{cases} \omega \diamond \Phi & \text{if } \omega \models C \\ \omega \diamond \Psi & \text{if } \omega \models \neg C \end{cases}$$

est une mise à jour conditionnelle : le modèle initial est mis-à-jour par Φ ou par Ψ, selon que ω satisfait la condition C ou non.

Les mises-à-jour conditionnelles sont indispensables pour les actions à effets conditionnels, tels que la célèbre action de tirer sur une dinde, qui correspond à une mise à jour par `if` loaded `then` $(\neg$ alive $\wedge \neg$ loaded$)$ `else` $\top$.

Mise-à-jours concurrentes

$(\Phi || \Psi)$ est une mise à jour concurrente. Nous n'en donnons pas la définition (qui est complexe – voir [124]). Une mise à jour concurrente opère en rassemblant conjonctivement les combinaisons de résultats possibles, et en exécutant ensuite les mise à jour correspondantes séparément ; en particulier,

$$\begin{aligned} & B \diamond ((A_1 \cup A_2) || (B_1 \cup B_2)) \\ \equiv \ & B \diamond (A_1 \wedge B_1) \vee B \diamond (A_1 \wedge B_2) \vee B \diamond (A_2 \wedge B_1) \vee B \diamond (A_2 \wedge B_2) \end{aligned}$$

[4] Un autre exemple (adapté de [35]) : on reprend les portes rouges dont on a parlé plus haut ; si on a dit au robot de peindre la porte 1 ou la porte 2 en rouge mais qu'il n'a aucun moyen de savoir si l'une des deux portes est initialement rouge (parce qu'il n'a pas de capteurs), alors l'action devrait être modélisée par `red1` $\cup$ `red2`. À l'opposé, l'action correspondant à la mise à jour par `red1` $\vee$ `red2` correspond à un contexte où le robot peut vérifier si l'une des portes est initialement rouge (et laisse le monde inchangé si c'est le cas). L'introduction de ces deux types d'actions non-déterministes nécessite à la fois la disjonction et $\cup$.

Les mises-à-jour concurrentes peuvent servir à représenter les effets indépendants d'une même action de façon compacte : considérons par exemple l'action de charger une arme à feu en présence d'une dinde[5], qui a deux effets indépendants : (1) l'arme est chargée ; (2) la dinde va se cacher si elle n'est pas sourde. Ceci correspond à la mise à jour concurrente $\alpha =$ (loaded $||$ `if` $\neg$ deaf `then` hidden `else` $\top$). La concurrence peut bien entendu servir aussi aux actions exécutées en parallèle. En particulier, dans l'article [94] (avec H. Fargier et P. Marquis), une action complexe consiste en l'instanciation de chacune d'un ensemble de *variables de décision* : l'instanciation d'une variable de décision peut être vue comme une micro-action élémentaire (voir Section 6.3), et les intanciations complètes comme des combinaisons en parallèle d'actions élémentaires (concurrentes). Dans cet article on examine également les cas particuliers issus de restrictions syntaxiques sur les effets des actions (à un extrême, on se ramène à une représentation de type STRIPS).

Les définitions précédentes sont indépendantes du choix d'un opérateur de mise à jour particulier. Dans le cas d'un opérateur à base de dépendance, le problème de prédiction reste coNP-complet.

3.1.4 Perspectives

Il se passe quelque chose d'intéressant lorsqu'on considère la restriction des opérateurs de DLF-mise à jour au cas où l'on requiert que la formule par laquelle on met à jour soit toujours un *littéral*. D'abord, la dépendance formule-littéral ne pose guère problème sous cette hypothèse : sauf en présence de dépendances explicites, on a $Dep(l) = \{l\}$. On a alors $\omega \diamond_{Dep} l = force(\omega, l)$, où $force(\omega, l) = \omega$ si $\omega \models l$, et $force(\omega, l)$ coïncide avec ω partout sauf en l sinon. Considérons alors l'input étendu $\Phi =$ (if C_1 then l_1) $||$ ... $||$ (if C_1 then l_n), et soit $\Gamma^+(l) = \bigvee\{C_i | l_i = l\}$. On a alors

$$\forall x \in VAR,\ \omega \diamond_{Dep} \Phi \models x \text{ ssi } \omega \models (x \wedge \neg\Gamma^+(\neg x)) \vee \Gamma^+(x)$$

et l'on retrouve là une expression bien connue dans la littérature du raisonnement sur l'action : celle d'un "successor state axiom". Ce résultat, qui ne nous est apparu qu'après la publication de [124], montre bien les positions, atouts et défauts respectifs de la mise à jour étendue et des langages

[5] Je reprends des exemples "bateau" de la littérature ; pour ceux qui n'en sont pas familiers, je conçois bien que de tels exemples sentent le ridicule.

d'action (calcul des situations, langage A et successeurs, causalité de McCain et Turner) : les deux modèles *coïncident* lorsque :

- la mise à jour étendue est restreinte aux inputs littéraux (mais autorise des inputs conditionnels et concurrents) ;
- les langages d'action sont restreints aux règles d'effets d'action (pas de règles causales statiques).

L'intérêt d'avoir mis ce rapprochement en évidence est clair : il s'agit maintenant d'unir la potentialité des approches "langages d'action" (prise en compte des règles causales statiques pour la ramification) et celles de la mise à jour (prise en compte des disjonctions et plus généralement du non-déterminisme). Cette perspective est d'autant plus prometteuse que les langages d'action souffrent justement de ce qu'ils traitent peu, ou mal, le non-déterminisme. Elle est l'objet d'un article en préparation avec Fangzhen Lin et Pierre Marquis [153], qui contient en outre des résultats de complexité et des caractérisations avec des formules de complétion.

3.1.5 Mise-à-jour d'états de croyance

Les opérateurs de mise-à-jour précédemment évoqués supposent que les informations sont de simples formules propositionnelles. Or, pour plusieurs raisons, il apparait plus pertinent d'opérer sur des informations plus complexes que de simples formules, à savoir des états de croyance. En d'autres termes, on voudrait définir des opérateurs de mise à jour $\diamond : \mathcal{B} \times \mathcal{B} \rightarrow \mathcal{B}$ au lieu de $\diamond : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$.

Cette nécessité d'opérer sur des états de croyances a été mise en évidence en ce qui concerne la révision des croyances, lorsque le processus de révision est destiné à être itéré. La raison en est simple : le processus de révision en tant qu'opérations sur des ensembles de croyances ("belief sets") *n'est pas markovien* : en effet, réviser un ensemble de croyances par une information ne donnera pas le même résultat selon que cet ensemble de croyances provient d'une séquence de révisions ou d'une autre : ainsi, réviser $a \leftrightarrow b$ par $\neg a$ donne l'ensemble de croyances $\neg a \wedge \neg b$, réviser $a \wedge \neg b$ par $\neg a$ donne aussi $\neg a \wedge \neg b$; pourtant, une révision ultérieure donne (intuitivement) deux résultats différents : $((a \leftrightarrow b) * \neg a) * a) = a \wedge b$ (ou éventuellement a) alors que $(a \wedge \neg b) * \neg a) * a) = a \wedge \neg b$.

Ce problème peut être résolu en mémorisant l'histoire des révisions [133, 165] ou en représentant les croyances par des états de croyance, ce qui re-

vient en fin de compte au même type d'information : on mémorise non seulement les croyances "pures" (celles au sujet desquelles l'agent est prêt à parier) mais aussi la façon dont elles doivent être révisées ; alors, la révision opérant sur des états de croyances ou sur des séquences d'ensembles de formules peut, elle, être considérée sans encombre comme markovienne. Le fait que la révision (sur des formules) ne soit pas markovienne n'est pas surprenant, puisque la révision concerne en principe un monde statique et que les soi-disant "vieilles" croyances doivent en tout état de cause jouer un rôle encore important puisqu'elles portent en définitive sur le *même* état du monde (vieilles croyances sur un monde tout neuf).

En quoi est-il pertinent de mettre-à-jour par un état de croyance ? Rappelons d'abord que l' "input" d'une mise-à-jour est la projection (les effets escomptés) d'une action ou d'un événement dont l'occurrence a été identifiée. Les actions et les événements pouvant fort bien avoir des effets avec des niveaux de plausibilité distincts, la mise-à-jour par états de croyance permet d'en tenir compte. On pourra ainsi, par exemple, mettre-à-jour par la projection de l'action `shoot` : **normally** ($\neg$ alive $\wedge$ loaded) $\wedge$ **always** loaded.

Un travail en collaboration avec Pierre Marquis et Mary-Anne Williams [160] a porté sur une généralisation des opérateurs de mise-à-jour à base de dépendance, opérant sur des états de croyance représentés par des bases de croyances possibilistes. Le calcul de $B \diamond B'$ est effectué en 3 étapes :

- déterminer les variables dont dépend B' (ce qui nécessite une généralisation de la relation de dépendance formule-variable aux états de croyance) ;
- les oublier dans B (ce qui nécessite une généralisation similaire) ;
- étendre le résultat obtenu par B'.

3.2 Temps, révision et mise à jour

À la lumière des résultats de la section précédente, on peut affirmer que, modulo quelques extensions d'un côté et quelques restrictions de l'autre, la mise à jour (ou, en tout cas, certains opérateurs de mise à jour) représente (dans une bonne mesure) les effets d'actions. De façon identique, elle permet de représenter les effets d'événements, puisque la distinction entre événements et actions n'est pas liée à la nature de leurs effets mais à leur

origine (exogène pour les événements, endogène pour les actions). Plus synthétiquement : la mise à jour permet de calculer la *progression* d'un état de croyances par un changement explicite (action, ou événement dont l'occurrence est connue a priori ou a été déterminée au préalable).

Dans cette section, nous allons nous intéresser au raisonnement sur le changement lorsque les connaissances ne sont plus des effets d'action mais (i) des observations et (ii) des croyances a priori sur l'évolution naturelle du système. Voici les hypothèses que nous faisons :

- *l'échelle de temps est finie.* Elle est a fortiori discrète, donc pas de changement continu.
- *pas d'actions (ni ontiques, ni épistémiques).* On considère donc un agent purement délibératif (équipé seulement d'actions délibératives).
- *les événements sont instantanés et sans effet à retardement.*
- *les observations sont instantanées et statiques.* Une observation instantanée et statique est une propriété du monde à un instant fixé (en quelque sorte une *photographie* du monde), ce qui exclut l'observation *directe* d'un évènement en train de se produire[6].
- *les observations sont fiables.*

et, parfois, l'hypothèse supplémentaire

- *le système tend à l'inertie.* Les événements (ou les actions exogènes) sont donc rares. Dans la terminologie de [202], ils sont appelés *surprises.*

Les hypothèses précédentes (inertie non comprise) nous permettent de poser sans perte de généralité $\mathcal{H} = \{1, ..., N\}$ et de supposer que (i) les observations ont lieu aux instants 1,..., N (l'absence d'observation se traduisant par l'observation de $\top$ – appelée observation vide) ; (ii) les événements (instantanés) ont lieu *entre deux instants consécutifs* t et $t + 1$. Le problème est donc assez simplifié. L'agent dispose de deux données :

- une séquence Σ d'observations $obs(t)$ pour chaque $t \in \mathcal{H}$, que nous appelons, comme chez Sandewall, un *scénario*[7]. Puisque les observations sont fiables, chaque $obs(t)$ est consistant.

[6] Attention : cela n'empêche pas l'agent d'*inférer* à partir des observations qu'un évènement s'est produit. Ainsi, on exclut *(j'observe qu') un individu malveillant est en train de faire feu sur une pauvre dinde*, mais on peut inférer à partir des observations *un individu tient un pistolet chargé et dirigé sur une dinde*, et, à l'instant suivant, *le pistolet est fumant et la dinde morte*, qu'un événement sinistre vient de se produire.

[7] Attention à la terminologie : un scénario est également un sous-ensemble cohérent de formules (aux chapitres 2 et 6). Dans la version finale je changerai cela.

– un modèle d'évolution du monde (parfois, tendant à l'inertie). Ce modèle consiste en la donnée d'un ensemble d'événements possibles (de leurs effets et de leur plausibilité[8] – pouvant l'un comme l'autre dépendre du contexte).

À partir de ces données, l'agent va raisonner sur les "événements" qui ont pu se produire, ce qui va permettre de compléter ses croyances aux différents instants. J'ai contribué au développement de plusieurs cadres pour le type de raisonnement précédent, qui diffèrent entre eux essentiellement quant à la nature de l'incertitude liée aux événements et à la nature de l'output.

3.2.1 Persistance graduelle

Dans [66] et [82], une hypothèse simplificatrice supplémentaire est faite : les événements sont *atomiques*. Un événement atomique est le changement de la valeur de vérité d'un unique fluent propositionnel, c'est-à-dire le passage de faux à vrai, ou de vrai à faux, du fluent f entre t et $t+1$[9]. Dans ces deux travaux on s'intéresse essentiellement à la persistance, qui est vue ici comme une notion *graduelle*, en vertu du principe simple selon lequel *plus t est proche de t' – dans le futur ou dans le passé – plus il est plausible que $obs(t)$ soit vérifié à t'*. L'article [66] reprend pour l'essentiel la démarche suivie par Dean et Kanazawa [59] et en donne deux critiques. La première porte sur la non-adéquation, dans certains cas (voir [66], page 470) d'un traitement probabiliste de la persistance, qui colle mal au principe d'*ignorance croissante* à mesure que l'on s'éloigne de l'instant où l'observation a été faite (une distribution de probabilité n'étant pas capable à elle seule de représenter l'ignorance). La seconde critique porte sur l'absence de traitement dans [59] de la persistance "vers le passé" (étant donné une observation $obs(t)$, il est – de façon générale – tout aussi pertinent de supposer que $obs(t)$ était déjà vérifiée peu avant t que de supposer qu'elle l'est encore peu après). L'article [66] propose un traitement possibiliste de la persistance répondant à ces deux critiques, et discute avant tout des propriétés souhaitables des fonctions de persistance en distinguant les propriétés purement ordinales, les propriétés dites semi-quantitatives qui ne nécessitent pas que l'échelle d'incertitude soit numérique, mais qui par contre nécessitent au moins une *métrique* sur l'échelle temporelle, et les propriétés qui nécessitent que les deux échelles soient numériques. Cet article, par contre, ne traite pas de l'application de

[8] au sens le plus général du terme.

[9] Ainsi, en raison de ses nombreux effets, instantanés et retardés, sur de multiples fluents, la catastrophe de Tchernobyl n'est pas un événement atomique.

fonctions de persistance à des observations disjonctives.

La seconde partie (pages 229 à 233) de l'article [82] est une approche cette fois probabiliste de la persistance, qui étend celle de Dean et Kanazawa en calculant les explications (ensembles de changements) plausibles en fonction de deux facteurs : la durée entre les observations et la tendance à l'inertie de chaque fluent. On discute des propriétés relatives à la persistance que les fluents peuvent ou non vérifier (indépendance mutuelle, markovianité, stationnarité) et on étudie la façon dont le calcul des probabilités des explications les plus plausibles varie avec ces différentes hypothèses, et plus particulièrement lorsque les probabilités des changements sont très proches de 0 (fluents fortement persistants), cette dernière hypothèse justifiant la minimisation des changements.

3.2.2 Extrapolation de croyances

Quel opérateur de changement des croyances est pertinent pour la prise en compte, dans un monde dynamique, d'observations à différents instants ? A priori ni la révision (pertinente pour la prise en compte d'une nouvelle information sur un monde statique), ni la mise à jour (pertinente pour la projection, dans un monde dynamique, du résultat attendu d'une action) – nous reviendrons plus en détail là-dessus en Section 3.2.3.

C'est pourquoi nous avons travaillé, avec Florence Dupin de Saint-Cyr, à la définition d'un nouvel opérateur de changement des croyances, que nous appelons *extrapolation de croyances*, qui permet de compléter des croyances initiales (consistant en des observations fiables en différents instants) par des hypothèses sur l'évolution naturelle du système.

Soit $\Sigma = \langle obs(1), ..., obs(N)\rangle$ un *scénario* cohérent (c'est-à-dire tel que chaque observation φ_i soit cohérente). On dit qu'une trajectoire $\tau = \langle \tau(1), .., \tau(N)\rangle$ de longueur N satisfait $\Sigma \in \mathcal{S}_N$ (noté $\tau \models \Sigma$) si et seulement si $\forall t \in 1..N$ on a $\tau(t) \models obs(t)$, et on note $Traj(\Sigma) = \{\tau \in TRAJ_{|\Sigma|} | \tau \models \Sigma\}$. τ est *statique* ssi $\tau(1) = ... = \tau(N)$.

L'extrapolation des croyances consiste à compléter Σ en utilisant des hypothèses sur l'évolution naturelle du système : les trajectoires du scénario complété (ou extrapolé) sont les trajectoires, parmi celles qui sont en accord avec le scénario, qui sont les plus en accord avec les croyances que l'agent possède sur l'évolution du système, ou, plus formellement, celles qui

sont préférées pour une relation de préférence représentant les croyances sur l'évolution naturelle du système (en particulier, si le système tend à l'inertie, l'extrapolation des croyances préfère les trajectoires statiques).

Une *relation de préférence* $\preceq$ est une relation réflexive et transitive sur $TRAJ_N$. τ est *minimale* pour $\preceq$ dans $X \subseteq TRAJ_N$ si et seulement si il n'y a pas de $\tau' \in X$ telle que $\tau' \prec \tau$. L'opérateur d'extrapolation $\updownarrow_{\preceq} : \mathcal{S} \rightarrow \mathcal{S}$ induit par $\preceq$ est défini par

$$Mod(\Sigma\updownarrow_{\preceq}(t)) = \{\tau(t) \mid \tau \in Min(Traj(\Sigma), \preceq)\}$$

Dans l'article [83], nous donnons plusieurs exemples d'opérateurs d'extrapolation, étudions les propriétés qu'ils vérifient, déterminons leur complexité, et proposons une méthode de calcul abductive pour un opérateur d'extrapolation particulier. Nous donnons aussi une axiomatique de l'extrapolation des croyances, et nous étudions la position de l'extrapolation par rapport à la révision et à la mise à jour .

3.2.3 Discussion

Revenons à la distinction entre révision et mise à jour et discutons du rôle des différents opérateurs de changement des croyances existant dans la littérature ainsi que des liens qu'ils entretiennent entre eux.

La distinction entre révision et mise-à-jour [131] est généralement exprimée en mettant en avant l'opposition entre monde statique et monde dynamique : la révision incorpore de nouvelles croyances à d'anciennes, toutes ces croyances portant sur un même monde (statique), tandis que la mise-à-jour modifie les croyances sur l'état du monde *en réponse à un changement du système, "notifié" par une nouvelle formule* φ .

Chacune de ces formulations recèle de l'ambiguïté.

En premier lieu, la révision peut être pertinente même lorsque l'état de croyances initial et la nouvelle formule ne réfèrent pas au même instant, pourvu qu'une distinction syntaxique (via une réification) soit faite entre un fluent à un instant et le même fluent à un autre. Ce constat n'est pas nouveau (comme on peut le lire dans [106], "ce qui est important pour la révision n'est pas que le monde soit statique, mais que les propositions utilisées pour décrire le monde soient statiques"). Ceci explique pourquoi la réification temporelle permet de capturer un processus dynamique dans un cadre

de révision des croyances : ainsi, on montre dans [83] qu'un opérateur d'extrapolation peut être exprimé à partir d'un opérateur de révision : le scénario extrapolé est équivalent à *la révision des croyances a priori sur l'évolution du système (comprenant notamment les lois de persistance des fluents) par les observations aux différents instants.*

En second lieu, dans la définition informelle de la mise-à-jour, il y a une ambiguïté dans la nature exacte de la formule par laquelle on met à jour ("input formula") α. Que veut dire exactement une "notification de changement ?" Cette formulation ne permet pas de distinguer entre une observation faite à l'instant suivant l'évolution du système et la conséquence attendue d'une action réalisée depuis la dernière observation du système (voir l'exemple illustratif en Section 1 de [83], qui montre que la non-distinction de ces deux types d'information mène à des résultats contre-intuitifs).

Pour voir plus clair, comparons formellement la mise à jour à l'extrapolation sur des scénarios de longueur 2 ; lorsque $N = 2$, la projection de l'extrapolation de Σ à l'instant 2 peut être réexprimée ainsi : à partir d'un ensemble de croyances $K = \Sigma(1)$ à l'instant 1 et d'une observation $\alpha = \Sigma(2)$ à l'instant 2, on calcule un ensemble de croyances complété à l'instant 2. Cela peut paraître à première vue similaire à la mise-à-jour de croyances, puisque dans la littérature on lit habituellement que la mise-à-jour "intègre deux informations concernant deux instants différents"[10].

Or, cette interprétation de la mise-à-jour est ambiguë, puisque la 2-extrapolation concerne des observations à différents instants avec des hypothèses de changement minimal et pourtant, ce n'est en général pas de la mise-à-jour (voir [83]).

Ce résultat simple signifie que la mise-à-jour n'est pas pertinente pour raisonner avec des observations datées sur un monde évolutif. Le point clé est le postulat U8 de Katsuno et Mendelzon [131]

$$\text{(U8) } (K_1 \vee K_2) \diamond \varphi \equiv (K_1 \diamond \varphi) \vee (K_2 \diamond \varphi)$$

[10]Par exemple : "si la nouvelle formule reflète un changement du monde réel, alors l'état de croyances doit être mis-à-jour" [126] ;" "la mise-à-jour est censée capturer des changements des croyances en réponse à un monde changeant"[28] ; "mise-à-jour : intégration de deux informations, toutes deux complètement fiables, se référant à deux instant différents (des changements pouvant avoir eu lieu entre temps)"[106].

qui, en requiérant que tous les modèles de l'ensemble initial de croyances soient mis-à-jour séparément, *nous empêche d'inférer des nouvelles croyances sur le passé à partir d'observations plus récentes* (ce qu'on nomme couramment *postdiction*)[11].

Par conséquent, l'impossibilité, pour la mise-à-jour, d'inférer de nouvelles croyances à l'instant 1 à partir de croyances à l'instant 2 signifie que l'on n'*observe rien* à l'instant 2, ou du moins, rien qui ne soit *complètement prédictible* à partir de ce que l'on savait à l'instant 1 et des croyances a priori sur l'évolution du monde. Comme elle n'est pas une observation, la nouvelle formule α ne peut être qu'une *prédiction*, plus précisément, la projection des effets attendus d'un changement "explicite". Par *changement explicite*, nous voulons dire que quelque chose se produit, *et l'agent sait que cela se produit*, dont les résultats vont peut-être changer l'état du monde.

On voit donc que les questions essentielles sont *l'observabilité* (qu'observe-t-on du monde réel et à quels instants ?) et la *prédictabilité des changements*. L'extrapolation est adaptée aux observations et aux changements imprévus, tandis que la mise-à-jour est adapt́ee aux changements attendus sans observations. Dans la classification de Sandewall [202], l'extrapolation est adéquate pour la sous-classe sans actions de K-**IS** (connaissances correctes, inertie et surprises) tandis que la mise-à-jour est adéquate pour la classe $\mathrm{K_p}$-**IA** (pas d'observations après l'instant initial, inertie, actions à résultats alternatifs). L'extrapolation et la mise-à-jour se complètent mutuellement, et si l'on veut être capable de raisonner à la fois avec des changements explicites et implicites, il faut alors intégrer les deux ce qui est relativement aisé. Dans [83] nous évoquons comment cette intégration peut être faite.

Nous terminons cette Section en évoquant quelques autres classes d'opérateurs de changement des croyances en les positionnant par rapport aux précédents.

La *mise à jour généralisée* [27] pourrait être considérée comme la restriction de l'extrapolation à des scénarios à deux instants et à une relation de préférence induite par des fonctions conditionnelles ordinales, au détail

[11] Cette inadéquation de la mise-à-jour n'est pas nouvelle (voir [28], [48] [163]) ; cependant, elle est rarement évoquée dans les articles sur la mise-à-jour, et même dans des articles récents on lit fréquemment l'idée – incorrecte – que la mise-à-jour de KB par α aurait comme but d'incorporer l'observation α (provenant par exemple de capteurs) reflètant un changement du monde.

près que la mise à jour généralisée permet la prise en compte d'événements non-déterministes avec différents niveaux de plausibilité – ce qui pourrait être intégré à l'extrapolation.

La position de la *révision itérée* est bien plus problématique. La différence intuitive entre extrapolation et révision itérée apparait clairement sur l'exemple suivant : soit $\Sigma = \langle a \rightarrow b, a, \neg a \rangle$; pour tout opérateur d'extrapolation raisonnable on a $\Sigma\updownarrow = \langle a \wedge b, a \wedge b, \neg a \wedge b \rangle$ tandis que la plupart des opérateurs de révision itérée (en prenant comme état épistémique initial l'état d'ignorance totale) conduisent à $[a \rightarrow b, a, \neg a] = \neg a$. La raison qui est au coeur de cette différence est que la révision itérée concerne des informations, certes obtenues à différents instants, mais concernant *un monde statique*[12]. Il me semble que la révision itérée se rapproche davantage de la fusion de croyances avec des sources d'inégales fiabilité que de n'importe quelle autre classe d'opérateurs de changement des croyances.

J'ai essayé de réunir les conclusions de la discussion qui précède sur un tableau récapitulatif.

	actions sans feedback (pas d'observations)	observations fiables événements ; pas d'actions	obs. fiables év. + actions
1 instant	–	RÉVISION (**)	–
2 instants	MISE-À-JOUR	MISE-À-JOUR GÉNÉRALISÉE[13]	? [14]
N instants	MISE-À-JOUR ITÉRÉE	EXTRAPOLATION	? [15]

(**) Détail pour la situation "pas d'actions + un seul instant" :

[12] Ceci explique pourquoi, une fois que la nouvelle information $\neg a$ a "annihilé" la précédente en raison du principe de la primauté forte de la nouvelle information [16] [133], on a supprimé de ce fait les raisons de croire en b.

[15] En présence de l'hypothèse d'inertie, l'opérateur qui convient est la *double révision* : si $\Sigma = \langle obs(1), obs(2) \rangle$ alors le scénario extrapolé est $\Sigma\updownarrow = \langle obs(2) * obs(1), obs(1) * obs(2) \rangle$ où $*$ est un opérateur de révision.

[15] La mise-à-jour généralisée permet la prise en compte d'un événement (non-identifiée) *ou* d'une action (identifiée) mais pas les deux à la fois – elle permet donc de raisonner sur l'action *ou* sur les changements imprévus (événements) mais pas les deux simultanément.

[15] Voir la fin de la section 5 de [83].

	observations fiables	observations non fiables
une seule observation	RÉVISION AGM [16]	RÉVISION NON-PRIORITAIRE [116]
plusieurs observations [17]	RÉVISION AGM [18]	FUSION ([137], [171]), RÉVISION ITÉRÉE[19]

3.3 Raisonnement sur l'espace et le changement

On a vu que le raisonnement sur le temps et l'action consiste à inférer des croyances sur l'état du monde en certains instants, compte tenu de croyances initiales sur l'état du monde (provenant par exemple d'observations) en certains instants et de croyances sur l'évolution du monde.

Ce principe est suffisamment général pour pouvoir s'appliquer à d'autres modes de raisonnement, où la structure temporelle est remplacée par une autre structure qui partage avec elle quelques propriétés de base (relations d'adjacence et/ou de proximité, éventuellement une distance, métrique ou ordinale). Appelons une telle structure un *ensemble d'étiquettes*, noté $Etiq$. Le *raisonnement sur le changement* consiste alors à inférer de nouvelles croyances sur l'état du monde en certaines étiquettes, compte tenu de croyances initiales sur l'état du monde en diverses étiquettes, et de croyances sur les liens (persistance, causalité etc.) entre les états du monde en différentes étiquettes. Voici des exemples :

- *raisonnement sur le temps et le changement* : $Etiq = T$ est un ensemble d'instants et $Evol$ contient des croyances sur l'évolution du monde ;
- *raisonnement sur l'espace et le changement* : $Etiq = S$ est un ensemble de points ou de régions de l'espace et $Evol$ contient des croyances sur la façon dont les propriétés du monde persistent ou changent lorsqu'on passe d'un point ou d'une région à l'autre ;
- *raisonnement sur le mouvement* : $Etiq = S \times T$ est un ensemble de couples composés d'un point de l'espace et d'un instant et $Evol$ contient des croyances sur la façon dont les propriétés du monde changent en fonction du temps et de l'espace (des croyances sur le mouvement d'objets, par exemple).
- *raisonnement à partir de cas, par analogie* : $Etiq$ est un ensemble de cas (ou encore, de situations), muni d'une distance (au moins ordinale)

[19] Les croyances initiales sont révisées par l'observation.
[19] éventuellement contradictoires
[19] Les croyances initiales sont révisées par la conjonction des observations.
[19] Avec l'hypothèse que les croyances sont ordonnées par un ordre (strict) de fiabilité.

entre situations, et *Evol* contient des croyances sur la façon dont les propriétés du monde changent d'une situation à l'autre.
- *raisonnement taxonomique* : similaire au raisonnement par analogie mais *Etiq* est cette fois un ensemble de classes (d'objets ou de situations) plutôt que de cas individuels.

On peut imaginer facilement d'autres contextes de raisonnement, mais je pense toutefois que ceux que je viens de mentionner sont ceux qui apparaissent le plus fréquemment dans la littérature.

À présent, examinons ce que deviennent les notions jusqu'ici étudiées dans le contexte temporel, lorsque l'espace des étiquettes varie. Je me concentrerai sur le contexte spatial, à la fois par souci de concision et parce que c'est celui auquel j'ai accordé le plus d'attention jusqu'à présent.

Deux remarques importantes, d'abord :

1. le raisonnement sur l'espace et le changement suppose que les variations de l'état du monde sont *uniquement* spatiales (et donc pas temporelles) : le monde est *temporellement statique*[20]. En conséquence, la distinction entre action (endogène) et événement (exogène) n'est pas pertinente, puisque les frontières entre régions *sont* – et il n'y a pas lieu de se préoccuper de *comment* elles sont arrivées là où elles sont, puisque le monde est temporellement statique : il suffit de raisonner comme si elles avaient toujours été là et le seront toujours (ceci n'est évidemment plus valable si l'on fait du raisonnement spatio-temporel).
2. il n'y a pas (a priori) de *directionnalité* lorsqu'on raisonne sur l'espace et le changement, alors qu'en ce qui concerne le raisonnement sur le temps et le changement la distinction entre passé et futur est cruciale. Il n'y a donc pas non plus de distinction entre prédiction et postdiction.

Examinons maintenant les analogues, en raisonnement sur l'espace, des objets de base et des problèmes les plus significatifs du raisonnement sur le temps et l'action.

- un *fluent spatial* est une propriété qui peut varier selon le point ou la région de l'espace considérée ;

[20] Güsgen et Hertzberg [114] illustrent cette hypothèse par l'image du "château de la Belle au Bois Dormant", où rien ne se passe et où l'on se contente d'observer et de raisonner sur l'état du monde dans les différentes pièces.

– puisque dans un contexte temporel, les actions sont destinées à changer certains fluents, c'est-à-dire à provoquer certains changements entre un instant et son/ses successeur(s), dans un contexte spatial ce rôle sera joué par les *frontières*. L'analogue du problème de la description des effets d'une action est donc la description des changements susceptibles d'intervenir lors du passage d'une frontière donnée ; en voici des exemples : (i) passage d'une pièce à l'autre dans le contexte de la navigation d'un robot de surveillance qui inspecte les diffèrentes pièces d'un immeuble ; (ii) franchissement d'une frontière physique ou administrative dans le contexte d'un système d'informations géographiques.
– problème du *décor spatial* [150] : il s'agit de spécifier seulement les fluents spatiaux susceptibles de changer de part et d'autre d'une frontière. Dans les exemples ci-dessus, le passage d'une pièce à l'autre influe (éventuellement) sur la température et la luminosité mais pas sur la pression ni sur les fluents spatiaux de plus grande granularité (le numéro de l'étage, de l'immeuble, l'adresse) tandis que le franchisssement de la frontière d'un état de la zone Euro à un autre laisse inchangées les propriétés relatives à la monnaie ou aux réglementations de l'UE.
– le problème de la *ramification* (c'est-à-dire des changements indirects, des relations de causalité entre fluents spatiaux) est tout aussi pertinente que dans le contexte temporel[21].
– l'analogue des *actions non-déterministes* consiste en les frontières dont les effets sur certains fluents spatiaux sont partiellement ou totalement imprévisibles : on ne sait pas ce qu'il y a de l'autre côté : par exemple, dans les problèmes de surveillance automatisée [89, 219], le fluent spatial *incendie* peut changer ou non de valeur quand on passe d'une cellule à l'autre. La distinction entre effets normaux ou exceptionnels reste elle aussi pertinente.
– *effets dépendants du contexte* : par exemple, lors du passage de la frontière franco-belge, le fluent spatial $francophone$ reste vrai si le point de franchissement est à l'est de Lille mais devient faux sinon.
– *concurrence* : plusieurs frontières passant au même endroit, dont il faut déterminer les effets cumulés.
– *inertie* : par défaut (c'est-à-dire tant que rien ne contredit cette hypothèse), les fluents spatiaux persistent d'un point à son voisin ; cette conclusion peut être remise en cause par des observations en d'autres

[21] Ce n'est par contre plus le cas pour ce qui est de la qualification, liée aux conditions d'exécutabilité d'une action : une frontière est toujours "exécutable" même si lon ne peut pas la franchir, l'autre côté existe bel et bien ; une frontière n'échoue pas, elle *est là*.

points qui l'invalideraient (non-monotonie).
- *persistance graduelle* : en général, la certitude qu'un fluent spatial persiste d'un point ou d'une région à un(e) autre est d'autant plus forte que la distance entre ces points ou régions est faible.

Je donne maintenant quelques détails sur les approches que j'ai développées dans le contexte du raisonnement sur l'espace (ou plus généralement du raisonnement dans des ensembles d'étiquettes).

Non-monotonie et distances

Dans l'article [3], que j'ai écrit avec Nicholas Asher, on montre comment des relations d'inférence non-monotones apparaissent très naturellement dans des contextes temporels, spatiaux ou plus généralement dans des contextes où les informations sont attachées à des étiquettes, l'espace des étiquettes étant muni d'une distance (ou, du moins, une structure pseudo-métrique ordinale[22].).

Soit un ensemble *fini* d'étiquettes $Etiq$ muni d'une pseudo-distance $d : Etiq \times Etiq \rightarrow I\!N$. Appelons *scénario* un ensemble $\Sigma \subseteq Etiq \times \mathcal{L}_{VAR}$, c'est-à-dire un ensemble d'observations étiquetées $x : \varphi$, et soit $\Sigma(x)$ la conjonction de tous les $x : \varphi$ contenus dans Σ ($\top$ s'il n'en existe pas). On étudie alors les deux relations d'inférence non-monotones suivantes (nous n'en donnons ici que les définitions informelles) :

- *inférence par projection* : $\Sigma \mid\!\sim_{proj} x : \varphi$ si et seulement si φ est vraie en tous les points les plus proches de x en lesquels la valeur de φ est déterminée par Σ.
- *inférence radiale* : $\Sigma \mid\!\sim_{rad} x : \varphi$ si et seulement si l'accumulation des informations en toutes les étiquettes situées dans le plus gros "disque" consistant autour de x permettent de déduire φ.

[22] On peut en fait se passer de pseudo-distances et travailler seulement avec des relations de proximité (donc une notion de distance purement ordinale) en utilisant, au lieu de d, une relation de proximité $<\subseteq Etiq^3$, où $z <_x y$ signifie que x est plus proche de y que de z. Il est équivalent de travailler avec d ou avec $<$; dans un sens, c'est immédiat : $z <_x y$ si et seulement si $d(x, z) < d(x, y)$; dans l'autre sens, le passage est possible grâce à l'hypothèse de finitude de $Etiq$. On choisit l'expression faisant usage de pseudo-distances, même si elle est moins "pure" puisqu'elle requiert une structure quantitative non indispensable, pour des raisons de commodité.

Ce cadre de raisonnement fonctionne avec tout espace de labels muni d'une pseudo-distance, en particulier le raisonnement spatial et le raisonnemnt par analogie, mais l'article [3] n'explore pas les spécificités de ces contextes de raisonnement.

On développe plus avant cette approche précédente dans l'article [150], cette fois dans le contexte spécifique du raisonnement sur l'espace. On raisonne sur les changements spatiaux à partir d'un scénario, d'un ensemble de *frontières* connues a priori et d'une relation de *dépendance* $Dep \subseteq FR \times Var$ entre frontières et fluents spatiaux, où $Dep(b, f)$ signifie que la franchissement de la frontière b "bloque" la persistance du fluent spatial f (c'est-à-dire que la connaissance de la valeur de f d'un côté de la frontière ne donne aucune indication sur sa valeur de l'autre côté).

Raisonnement plausible à partir d'observations spatiales : une approche fondée sur les fonctions de croyance

Les approches précédentes ont comme output des relations d'inférence (non-monotones, certes) et se montrent insuffisantes lorsqu'il s'agit de quantifier l'intensité des croyances en différents points de l'espace. Or, cet aspect quantitatif est pertinent dans de nombreux contextes, notamment lorsqu'il s'agit de décider de mesures "de terrain" à effectuer dans le but de rendre une carte plus fiable et/ou plus précise.

Le travail que Philippe Muller et moi entreprenons depuis deux ans vise à donner un modèle de ce processus de quantification de l'intensité des croyances induites par des observations en certains points (ou régions) et des hypothèses quantitatives de persistance spatiale des différents fluents spatiaux. L'information disponible, consistant en un ensemble d'observations ponctuelles, est extrapolée aux points voisins. Le principe de l'approche est le suivant : pour simplifier, nous ne nous intéressons qu'à une seule propriété de l'état du monde, dont le domaine est un ensemble fini S supposé purement qualitatif, c'est-à-dire que S n'est pas la discrétisation d'un ensemble de valeurs numériques (par exemple, S est un ensemble de types de sol ou de climat). Un ensemble d'observations est, comme dans ce qui précède, un ensemble de couples $\langle x, obs(x)\rangle$ où $x \in X$ et $obs(x)$ est une partie non vide de S.

Soit x un point de X (*point focus*), en lequel on s'intéresse aux valeurs des fluents spatiaux. Pour cela, on examine ce qui a été observé en d'autres points y, et on suppose que $obs(y)$ persiste de façon connexe de y à x avec

une probabilité $p_{x,y,f}$ qui décroît avec la distance de x à y, et induit en x une fonction de masse $m_{y \hookrightarrow x}$, de type "fonction de support simple, définie par

$$\begin{cases} m_{y \hookrightarrow x}(obs(y)) & = f(obs(y), d(x,y)) \\ m_{y \hookrightarrow x}(S) & = 1 - f(obs(y), d(x,y)) \end{cases}$$

où f est une fonction de $(2^S \setminus \emptyset) \times I\!R^+$ dans $[0,1]$, non-décroissante en son second argument, et vérifiant $f(obs, \alpha) = 1$ si et seulement si $\alpha = 0$, ainsi que $f(obs, \alpha) \rightarrow_{\alpha \rightarrow +\infty} 0$.

La lecture intuitive de $m_{y \hookrightarrow x}$ est la suivante : le fait que $obs(y)$ est observé en y supporte la croyance que $obs(y)$ est également vraie en x, à un degré qui est d'autant plus fort que y est proche de x. En particulier, si $x = y$ alors $obs(x) = obs(y)$ a un impact maximal (et absolu) sur x tandis que cet impact devient nul (en général) lorsque y devient trop loin de x. $f(obs, \alpha)$ peut être vue comme une probabilité de *persistance pertinente*, ou encore, la probabilité que la vérité de $obs(y)$ en y soit une bonne raison de croire que $obs(y)$ soit également vraie en x[23].

Dans un cadre purement probabiliste, ce degré d'impact, ou encore cette probabilité de persistence pertinente, ne peut pas être distinguée d'une simple probabilité[24]. On peut en tirer une première conclusion : *une modèlisation purement probabiliste de la persistance spatiale requiert non seulement des connaissances sur la façon dont les propriétés persistent dans l'espace mais aussi une probabilité a priori que les propriétés soient vérifiées en un point donné ; cette dernière, qui peut être difficile à obtenir, n'est pas indispensable avec l'approche à base de fonctions de croyance.*

Le second défaut d'une approche purement probabiliste de la persistance spatiale est l'absence de distinction entre ignorance et conflit. Considérons

[23] Il se peut très bien, cependant, que $obs(y)$ soit vrai en x même si $f(obs(y), \alpha) = 0$: supposons par exemple qu'il pleuve en y ; cela n'est pas du tout une raison de croire qu'il pleut aussi en z où $d(x,z) = 8000\ km$ (l'impact de x sur z en ce qui concerne la pluie est presque nul). Mais *ceci ne veut pas dire que la probabilité qu'il pleuve en z soit proche de zéro.* Il se peut très bien qu'il pleuve x ; mais dans ce cas, la fait qu'il pleuve en z est (presque certainement) *non corrélé* au fait qu'il pleuve en x.

[24] Si l'on veut exprimer des probabilités de persistance dans un cadre purement probabiliste, on a besoin, par exemple pour le fluent spatial *pluie*, d'une fonction $g_{pluie} : E^2 \rightarrow [0,1]$ telle que $P([x]pluie|[y]pluie)) = g_{pluie}(d(x,y))$. Cette fonction g est différente de f. Plus précisément, on a $g \geq f$, et g et f sont d'autant plus proches de f que $d(x,y)$ est de plus en plus petite ; quand $d(x,y)$ devient grande, l'impact f tend vers 0 tandis que g tend vers la *probabilité a priori* qu'il pleuve en x.

les deux situations suivantes : (1) absence totale d'information en x ; (2) on a observé qu'il pleut en y, qu'il ne pleut pas en z, où y et z sont deux points *très proches de* x. Les fonctions de croyance permettent de distinguer clairement ces deux situations : pour (1) on aura $Bel(x : pluie) = Bel(x : \neg pluie) = 0$ tandis que pour (2) on aura, par exemple, $Bel(x : pluie) = Bel(x : \neg pluie) = \frac{1}{2}$ (si les fonctions de persistance de *pluie* et de $\neg$*pluie* sont égales et si x est aussi proche de y que de z). D'où la seconde conclusion : *une modélisation purement probabiliste de la persistance spatiale ne permet pas de distinguer entre information conflictuelle et manque d'information, alors que les fonctions de croyance le permettent.*

Une fois que chaque observation est traduite par une fonction de support simple $m_{y \hookrightarrow x}$, la croyance en la valeur de la variable V est calculée en combinant toutes les fonctions de masse $m_{y \hookrightarrow x}$ pour $y \in R(O)$. La combinaison de Dempster vient tout de suite à l'esprit ($m_x = \bigoplus_{y \in E} m_{y \hookrightarrow x}$), mais elle pose des problèmes de redondances entre observations en des points proches. Pour pallier ce problème, on introduit un *facteur d'atténuation* qui croît avec la dépendance entre sources, c'est-à-dire avec les proximité entre points où les observations ont été faites. On généralise alors la combinaison de Dempster en une nouvelle méthode de combinaison, qui ressemble fort à l'intégrale de Choquet (plus précisément, elle est à l'intégrale de Choquet ce que la combinaison de Dempster est à la moyenne arithmétique).

Nous avons procédé à de nombreuses expériences, qui nous ont permis de vérifier empiriquement le comportement satisfaisant de ce nouvel opérateur de combinaison.

Chapitre 4

Représentation logique de préférences

4.1 Introduction

On a dit que la spécification d'un problème de décision exigeait l'expression des préférences de l'agent sur les états du système. On a déjà vu, au chapitre 1, qu'il existe plusieurs classes de modèles possibles pour les préférences ; cependant, quel que soit le modèle choisi, celui-ci ne dit pas comment le passage de l'expression des préférences par l'agent à la structure préférentielle est effectué. On va dans ce chapitre examiner de plus près des langages de représentation de préférences qui permettent d'exprimer des structures préférentielles de manière compacte, tout comme les formalismes dont on a parlé aux chapitres précédents visent à représenter de manière compacte des *croyances*.

4.1.1 Hypothèses

Dans la suite du chapitre on ne s'intéressera qu'aux préférences sur l'espace des conséquences X, en ignorant délibérément toute considération relative à l'incertitude ou au non-déterminisme. Formellement, on peut formuler un problème de décision *purement préférentiel* comme suit :

– l'ensemble des actions $\mathcal{A} = \{a_1, ..., a_p\}$ et l'ensemble des conséquences possibles des actions $\mathcal{X} = \{x_1, ..., x_p\}$ sont deux ensembles finis de même cardinalité p. Toutes les actions sont déterministes et pour tout $a_i \in \mathcal{A}$, le résultat de a_i dans l'état initial est x_i *pourvu que* a_i *soit disponible. On peut alors sans encombre identifier l'espace des actions*

et celui des conséquences : dorénavant on utilisera seulement $\mathcal{X}$, et une conséquence $x \in$X désignera indifféremment l'action permettant d'obtenir la conséquence x de façon déterministe, ou cette conséquence elle-même. Et, pour embrouiller encore un peu plus le tout, on appellera les éléments de $\mathcal{X}$ des *alternatives* (au sens anglais) plutôt que des actions ou des conséquences (ceci par souci d'homogénéité avec la littérature des sciences de la décision, en partie de la théorie du choix social).

– par contre, les actions ne sont pas nécessairement toutes exécutables (donc les conséquences ne sont pas toutes réalisables) ; l'ensemble X_R des conséquences réalisables est, selon le cas, connu hors ligne ou en ligne :
  – lorsque X_R est connu hors ligne, ce n'est pas la peine de calculer toute la relation de préférence mais il suffit de calculer une, ou éventuellement plusieurs, décisions optimales.
  – lorsque X_R n'est connu qu'en ligne, le problème est un peu plus complexe puisqu'il faut, si possible "compiler" la structure de préférence afin d'être en mesure, une fois X_R connu, de déterminer une décision optimale réalisable. Un cas particulier intéressant est celui où X_R ne contient que deux éléments (connus seulement en ligne), c'est-à-dire X= $\{x, y\}$; il suffira alors, en ligne, de *comparer* x et y.

4.1.2 Représentations explicites et quasi-explicites

On a vu au chapitre 1 qu'une *structure préférentielle* est, selon le cas, une fonction d'utilité $u : \mathcal{X} \to Val$, où $Val \subseteq \mathbb{R}$, ou une relation de préordre R (partielle ou totale).

La *représentation explicite* d'une fonction d'utilité u consiste en la donnée de la liste $\{(x, u(x)) | x \in \mathcal{X}\}$. Dans de nombreux cas, il existe une valeur "neutre" q^* (par exemple 0 dans une échelle numérique bipolaire) prise par défaut. En présence d'une valeur neutre, la représentation *quasi-explicite* d'une structure préférentielle évaluationnelle est l'ensemble de couples $\{\langle x, u(x)\rangle, x \in \mathcal{X}, u(x) \neq q^*\}$, avec la convention $u(x) = q^*$ pour tous les x n'apparaissant pas dans la liste précédente. La représentation explicite ou semi-explicite d'une fonction d'utilité a, dans le cas général, une complexité spatiale en $\mathcal{O}(|X|)$.

La représentation explicite d'une structure préférentielle relationnelle R est l'ensemble des couples $\{\langle x, y\rangle, (x, y) \in \mathcal{X}^2 \& R(x, y)\}$. En général, il suffit

d'expliciter un sous-ensemble de R, le reste s'en déduisant par fermeture transitive (et éventuellement réflexive[1]). Formellement, une représentation semi-explicite d'une structure préférentielle relationnelle R est un ensemble des couples $R' \subseteq \mathcal{X}^2$ telle que la fermeture transitive [et réflexive] de R' soit égale à R. Il n'y a pas unicité de la représentation semi-explicite, cependant la représentation semi-explicite la plus compacte (c'est-à-dire minimisant le nombre de couples explicités) d'une relation R est assez souvent évidente. Dans le meilleur des cas, une représentation semi-explicite a une complexité spatiale en $\mathcal{O}(|\mathcal{X}|)$ (au lieu de $\mathcal{O}(|\mathcal{X}|^2)$), ce qui ne permet pas d'échapper à l'explosion combinatoire.

4.1.3 Représentations compactes

La complexité spatiale des représentations [quasi-]explicites est au moins en $\mathcal{O}(|\mathcal{X}|)$, ce qui est raisonnable lorsque $\mathcal{X}$ est un ensemble d'individus ou d'objets, mais cela n'est plus lorsque $|\mathcal{X}|$ est un ensemble de conséquences ayant une structure combinatoire forte (des combinaisons d'objets, par exemple). Dans ce dernier cas, $\mathcal{X}$ est un produit cartésien de domaines $\mathcal{X} = D_1 \times ... \times D_n$, et sa taille est exponentiellement plus grande que la taille des données de bases (les variables et leurs domaines). Et bien souvent, l'agent exprime ses préférences non pas de façon explicite mais de façon structurée, en décrivant des préférences localement, sur des variable isolées ou sur des groupes de quelques variables corrélées.

L'objectif de ce chapitre est de passer en revue quelques approches, développées dans la communauté de l'intelligence artificielle ou de la recherche opérationnelle, permettant de représenter des structures de préférence de façon compacte et intuitivement satisfaisante.

Comme dans ce mémoire on ne s'intéresse qu'aux structures finies, je parlerai essentiellement des langages de représentation logiques, ainsi que (plus brièvement) des extensions des langages de contraintes sur des domaines finis[2]. Pour chacun de ces langages nous faisons une brève étude de complexité. Puisqu'il ne s'agit pas ici de *raisonner sur des connaissances* mais de *décider à partir de préférences*, les problèmes importants ne sont pas tout-à-fait les mêmes que pour les langages logiques pour le raisonnement. En particulier,

[1] Par exemple, la relation d'ordre total $x_1 > x_2 > x_3 > x_4$ sera explicitée par $\{\langle x_1, x_2\rangle, \langle x_2, x_3\rangle, \langle x_3, x_4\rangle\}$ plutôt que $\{\langle x_1, x_2\rangle, \langle x_2, x_3\rangle, \langle x_3, x_4\rangle, \langle x_1, x_3\rangle, \langle x_1, x_4\rangle, \langle x_2, x_4\rangle\}$.

[2] Ce chapitre ne prétend pas à l'exhaustivité.

l'inférence joue un bien moindre rôle ; les problèmes qui sont plus particulièrement pertinents sont la comparaison de deux alternatives (à savoir : étant donné une spécification logique $\mathcal{G}$ des préférences de l'agent et deux alternatives x et x', déterminer si $x \geq_{\mathcal{G}} x'$, la non-dominance (déterminer si une alternative x est non-dominée pour $\geq_{\mathcal{G}}$) et dans une moindre mesure le problème de déterminer s'il existe une alternative non-dominée satisfaisant une formule ψ. On appellera ces problèmes respectivement COMPARAISON, NON-DOMINANCE et CAND-OPT-SAT.

Dans tout ce chapitre, sauf en ce qui concerne les langages à base de contraintes, l'ensemble des alternatives sera l'ensemble des modèles d'une formule propositionnelle K restreignant l'ensemble des alternatives (physiquement) réalisables.

Une manière courante pour un agent d'exprimer des préférences consiste à énumérer un ensemble de *buts* (ou encore *désirs*, ou *préférences élémentaires*), chacun d'entre eux pouvant être représenté par une formule propositionnelle, accompagné éventuellement d'informations supplémentaires telles que des poids, priorités, contextes ou distances. Dans la suite du chapitre, G_i et C_i dénotent des formules propositionnelles et les α_i des valuations numériques. $\mathcal{G}$ est une "base de buts" (par analogie avec une "base de connaissances" et $u_{\mathcal{G}}$ (resp. $R_{\mathcal{G}}$) dénote la fonction d'utilité (resp. la relation de préférence) induite par $\mathcal{G}$.

4.1.4 Représentation propositionnelle brute

La représentation logique des préférences la plus simple, "prototypique", appelée "représentation propositionnelle brute" et notée R_{brute}, consiste tout simplement, pour l'agent, à exprimer des préférences à l'aide d'une formule propositionnelle G (ou de façon équivalente, un ensemble fini de formules propositionnelles, interprété conjonctivement). G représente le but à satisfaire. L'espace d'états-conséquences $\mathcal{X}$ est identifié à l'ensemble des mondes 2^{VAR}. Ce langage des plus rudimentaires nous permet de distinguer deux types d'états-conséquences : x est *favorable* si $x \models G$ et *défavorable* sinon. La relation de préférence induite par G est le préordre total défini par $x \geq_G x'$ ssi ($x \models G$ ou $x' \models \neg G$). De manière équivalente, la fonction d'utilité "canonique" induite par G est définie par $u_G(x) = 1$ si $x \models G$ et $u_G(x) = 0$ si $x \models \neg G$. Cette représentation, très grossière puisqu'elle ne permet qu'une distinction entre états buts et état non buts (utilité "binaire"), est de peu d'intérêt en pratique mais nous y ferons référence pour aider à la bonne

compréhension des différentes notions introduites – et elle nous donne en outre des bornes inférieures de complexité.

La logique propositionnelle est donc certes un langage permettant de représenter des préférences de manière compacte, mais il est peu expressif car il n'est adapté qu'à des structures préférentielles très rudimentaires. Dans la suite de ca chapitre on s'intéresse à des familles de langages plus sophistiquées. On donnera dans un premier temps une famille destinée à la représentation compacte de préférences évaluationnelles numériques : les *logiques à pondération*, ainsi que leurs contreparties en termes de contraintes. Dans un second temps on parlera de familles destinées à la représentation compacte de préférences relationnelles à base de *préférences contextuelles*. Enfin on abordera quelques problèmes liés à l'utilisation de tels langages de représentation dans le contexte de la décision de groupe.

4.2 Logiques et contraintes à pondérations ou à priorités

4.2.1 Langages logiques à pondérations

Le raffinement le plus simple de la représentation brute est la représentation R_{card}, dont les inputs sont des *multi-ensembles* de buts : $\mathcal{G} = \{\{\varphi_1, ..., \varphi_m\}\}$, et

$$u_{GB}(x) = |\{i \in 1..m | w \models G_i\}|$$

Ce raffinement de R_{brute} considère des buts tout-ou-rien mais indépendants, et permet donc des compensations, mais il ne permet pas la prise en compte de l'importance relative des buts élémentaires ; pour aller plus loin, on considère toujours des multi-ensembles de buts, cette fois attachés à une valuation.

Un *langage logique à pondérations pour la représentation de préférences* (LLP) permet la représentation d'une fonction d'utilité $u : \mathcal{X} \to Val$ à l'aide d'items $\langle \varphi, \alpha \rangle$ où les φ sont des formules de $\mathcal{L}_{VAR}$ et les α sont des valuations de Val. Dans le cas le plus général, la signification intuitive de $\langle \varphi, \alpha \rangle$ est l'existence d'un but à satisfaire, dont la satisfaction ou la violation a un effet sur la fonction d'utilité u. Une *base de préférences* $\mathcal{G}$ est un *multi-ensemble* fini de tels couples : $\mathcal{G} = \{\{\langle \varphi_i, \alpha_i \rangle, i = 1 \dots m\}\}$.

L'utilité $u_{\mathcal{G}}(x)$ d'une conséquence est par définition une fonction des poids des buts satisfaits par x et fonction des poids des buts violés par x :

$$u_{GB}(x) = F_1(F_2(\{\{\alpha_i | x \models \varphi_i\}\}, F_3(\{\{\alpha_j | x \models \neg\varphi_j\}\})))$$

où F_1, F_2 et F_3 sont des fonctions d'agrégation. Une logique pondérée est donc définie par son triplet $\langle F_1, F_2, F_3\rangle$ de fonctions d'agrégation.

Un langage logique à pondérations est dit *à polarité négative* [217] [140] [142] (resp. *positive*) si $u(x)$ ne dépend que des formules φ_i *non satisfaites* (resp. *satisfaites*) de $\mathcal{G}$. Quand seules les utilités relatives (c'est-à-dire les différences d'utilités entre mondes) comptent (ce qui est en général le cas), on peut faire l'une ou l'autre de ces hypothèses (mais évidemment pas les deux !). L'hypothèse de représentation négative conduit à l'expression suivante :

$$u_{\mathcal{G}}(x) = F(\{\{\alpha_i | x \models \neg\varphi_i\}\})$$

F doit évidemment satisfaire certaines propriétés (voir [141]). Deux choix classiques de LLP à représentation négative sont

- $Val = I\!R$, $F(w_1, \ldots, w_n) = -\sum_{i=1}^{n} w_i$; on retrouve alors la logique des *pénalités* (voir Section 2.1.1) mais dans un autre usage ;
- $Val = [0, 1]$, $F(w_1, \ldots, w_n) = 1 - \max_{i=1}^{n} w_i$; on retrouve cette fois la *logique possibiliste* [75], là encore dans un usage différent.

Les langages de représentation R_{brute} et R_{card} sont bien évidemment des cas particuliers de langages logiques à pondérations – dégénérés, puisqu'ils ne tiennent pas compte des pondérations.

L'article que j'ai écrit avec Céline Lafage [141], et, avec plus de détails, la thèse de Céline Lafage [140], discute des propriétés que doit satisfaire la fonction d'agrégation F. Plus formellement, dans le cas où $Val = I\!R^+$, nous énonçons des postulats portant sur la fonction Ξ qui associe une fonction d'utilité à une base de buts, puis nous établissons un premier théorème de représentation qui exprime que les postulats précédents sont vérifiés si et seulement si il existe une fonction $F : I\!R^+ \times I\!R^+ \to I\!R^+$ commutative, associative[3], non décroissante, ayant 0 comme élément neutre, telle que

$$u_{\mathcal{G}}(x) = - * \{\{\alpha_i | x \models \neg\varphi_i\}\}$$

Des variantes de ce résultat, obtenus en faisant varier les postulats et donc les propriétés des fonctions d'agrégations (pour des langages logiques à polarité positive, négative ou mixte) figurent dans [140] et dans un article en

[3] L'associativité est cependant plus discutable ; voir [140]

préparation.

Des résultats de complexité en ce qui concerne les problèmes COMPARAISON, NON-DOMINANCE et CAND-OPT-SAT pour les langages à logiques pondérées peuvent être facilement dérivés de résultats de complexité de la fusion à base de distances figurant dans [134].

Une dernière remarque. Nous employons la terminologie "logiques à pondérations" tout comme en Section 2.2.1, alors que les cadres diffèrent (notamment, l'input est dans un cas un *ensemble* de couples $\langle \varphi, \alpha \rangle$ et dans l'autre un *multi-ensemble* de tels couples.

Langages logiques à priorités

Commençons par définir la représentation $R_{\subseteq}$: étant donné $\mathcal{G} = \{\varphi_1, ..., \varphi_n\}$, soit $sat_{\mathcal{G}}(x) = \{i | x \models \varphi_i\}$ et $nonsat_{\mathcal{G}}(x) = \{1, ..., n\} \setminus sat_{\mathcal{G}}(x)$ La relation de préférence induite par $\mathcal{G}$ est : $x \geq_{\mathcal{G}} x'$ si et seulement si $sat_{\mathcal{G}}(x) \supseteq sat_{\mathcal{G}}(x')$.

Ce préordre partiel sur les mondes n'est autre que l'ordre de Pareto : en effet, un candidat x est non-dominé pour $R_{\subseteq}$ si et seulement si il n'existe pas de candidat $x' \in Mod(K)$ satisfaisant tous les buts satisfaits par x et au moins un autre.

Les langages logiques à pondérations sont des langages de représentation compacte de fonctions d'utilité. Leur contrepartie ordinale est la famille des langages logiques à priorités. Une *base de préférences avec priorités* est un couple formé d'un ensemble de formules indexées et d'une relation de priorité : $\mathcal{G} = \langle \{1 : \varphi_1, ..., n : \varphi_n\}, \succeq \rangle$ où $\succeq$ est un préordre sur $\{1, ..., n\}$[4].

$\geq_{GB}$ est alors calculé de manière à étendre R_{brute} ou R_{card} en prenant en considération cette relation de priorité entre buts. Voici des choix possibles :

1. $x \geq^1_{\mathcal{G}} x'$ si et seulement si $\forall i \in nonsat_{\mathcal{G}}(x)$ $\exists j \in nonsat_{\mathcal{G}}(x')$ tel que $j \succ i$ (voir [40]) ;
2. $x \geq^2_{\mathcal{G}} x'$ si et seulement si $\forall i \in nonsat_{\mathcal{G}}(x) \backslash nonsat_{\mathcal{G}}(x')$ $\exists j \in nonsat_{\mathcal{G}}(x') \backslash nonsat_{\mathcal{G}}(x)$ tel que $j \succ i$ (voir [40]) ;
3. $x \geq^3_{\mathcal{G}} x'$ si et seulement si il existe une bijection σ d'un sous-ensemble A de $nonsat_{\mathcal{G}}(x)$ dans un sous-ensemble B de $nonsat_{\mathcal{G}}(x')$ telle que (a) pour tout $i \in A$, $i \sim \sigma(i)$ et (b) $\forall i \in nonsat_{\mathcal{G}}(x) \setminus A$ $\exists j \in nonsat_{\mathcal{G}}(x') \setminus B$ tel que $j \succ i$.

[4]Si l'on se donne comme relation de préférence un ordre strict plutôt qu'un préordre, on peut alors reformuler les définitions 1 et 2 qui suivent, mais pas la définition 3.

4. $x \geq^4_{\mathcal{G}} x'$ si et seulement si pour tout $i \in 1 \ldots n$ on a $|\{j \in nonsat_{\mathcal{G}}(x'), j \succeq i\}| \geq |\{k \in nonsat_{\mathcal{G}}(x), k \succeq i\}|$.

Lorsque $\succeq$ est un préordre total, 1., 2. et 3. coïncident respectivement avec les ordres *best-out*, *discrimin* et *leximin*. Ces quatre façons d'induire une relation de préférence à partir d'un ensemble de buts muni d'une relation de priorité est la contre-partie ordinale des logiques à pondérations "à polarité négative", puisqu'elles se focalisent sur les buts non satisfaits ; on peut de manière similaire construire des relations de préférence à polarité positive ou à polarité mixte. Il n'y a sans doute qu'un nombre assez limité de constructions pertinentes. En faire une étude systématique est une perspective de recherche intéressante.

On constate là encore (et peut-être de façon encore plus évidente qu'en ce qui concerne les logiques pondérées) que des outils initialement développés pour la représentation de *connaissances* sont tout aussi pertinents pour la représentation de *préférences*. La complexité des problèmes COMPARAISON, NON-DOMINANCE et CAND-OPT-SAT pour les langages logiques à priorités peuvent être dérivés de résultats de complexité obtenus en raisonnement non-monotone [87], révision [184] et raisonnement tolérant à l'incohérence [42].

Les problèmes de satisfaction de contraintes comme un langage de représentation de préférences

Commençons par rappeler quelques définitions de base en ce qui concerne les *problèmes de satisfaction de contraintes à domaines finis*[5] (CSP). Un CSP est constitué d'un ensemble de *variables* $V = \{v_1, \ldots, v_n\}$, chacune ayant un domaine fini de valeurs D_i, et d'un ensemble de *contraintes* $\{C_1, \ldots, C_p\}$, où une contrainte C_i porte sur un ensemble de variables $V(C_i) \subseteq V$ et restreint les combinaisons de valeurs possibles des variables de $V(C_i)$: plus formellement, si $V(C_i) = \{v_{i_1}, \ldots, v_{i_q}\}$ (q est l'arité de C_i), alors C_i est un ensemble de q-uplets de $D_{i_1} \times \ldots \times D_{i_q}$. Traduite en logique propositionnelle, une combinaison de valeurs est une conjonction de littéraux, une contrainte est une disjonction de combinaisons de valeurs, donc une DNF, et enfin, un ensemble (conjonctif) de contraintes est une conjonction de DNF (CDNF). Voir à ce sujet [38]. Une *solution* d'un CSP est une instanciation de ses variables satisfaisant toutes ses contraintes. La communauté de recherche qui travaille

[5] Je laisse de côté le cas des domaines infinis qui nous emmèneraient trop loin.

sur les CSP, qui se situe à la frontière de l'intelligence artificielle et de la recherche opérationnelle, a développé de puissants outils permettant la recherche d'une solution ou d'un ensemble de solutions, ou l'adaptation d'une solution à la modification d'un problème (ajout ou retrait de contraintes), entre autres choses.

Une contrainte est généralement vue comme un but à satisfaire[6] (souvent avec une connotation de polarité négative, à moins que ce ne soit simplement l'usage du terme "contrainte" qui ne le suggère). Les CSP constituent donc un langage de représentation de préférences, dont la version de base équivaut, à la syntaxe près, à la représentation propositionnelle brute. Motivée par le besoin d'introduire de la flexibilité dans l'expression et la résolution des contraintes, la communauté a développé des extensions du modèle de base et de ses outils algorithmiques, d'abord en maximisant le nombre de contraintes satisfaites (CSP "partiels" [104]), puis en introduisant des valuations ou des priorités, soit sur les combinaisons de valeurs dans les contraintes ("contraintes flexibles") soit sur les contraintes elles-mêmes ("contraintes à priorités"). C'est le sujet de la thèse d'Hélène Fargier [92] ainsi que de plusieurs travaux en collaboration entre le CERT, l'INRA et l'IRIT (voir notamment [207]) auxquels j'ai un peu participé : dans l'article [98], nous allons au-delà des CSP flous et possibilistes [206, 71] : nous proposons des raffinements de type "discrimin" et "leximin", qui permettent une meilleure discrimination entre solutions et nous discutons de l'impact de ces raffinements sur les algorithmes de résolution.

En Section 6.4 je parlerai d'une autre contribution (plus importante en ce qui concerne ma participation) à l'étude des CSP pour la décision, à savoir les CSP mixtes et probabilistes pour la décision dans l'incertain.

4.3 Logiques des conditionnels et représentation de préférences

Dans cette section et la suivante on considère que les préférences élémentaires sont des *préférences contextuelles*, à savoir : $\mathcal{G}$ est un ensemble de buts, chacun d'entre eux étant associé à un *contexte*, c'est-à-dire $GB = \{C_1 \ : \varphi_1, ..., C_n \ : \varphi_n\}$. C_i est le *contexte* de φ_i.

[6] Mais pas toujours : on utilise aussi les CSP pour résoudre des problèmes de raisonnement pur ; voir la Section 6.4.

L'idée d'utiliser des conditionnels pour représenter des préférences est due originellement à Boutilier [26]. Il utilise la logique CO^* : là encore, un formalisme logique est exporté de la représentation de connaissances (puisque c'est dans cette optique qu'a d'abord été développée la logique CO^*) à la représentation de préférences.

Un modèle de CO^*est un couple $M = \langle W, \geq \rangle$ où W est un ensemble de mondes associés de manière univoque (et identifiés) à des interprétations de Ω_{VAR}[7], et $\geq$ est un *préordre total* exprimant la préférence entre mondes. Le langage de CO^*contient une modalité dyadique D [8].

Un modèle $M = \langle W, \geq \rangle$ de CO^*satisfait un conditionnel $D(\varphi|\psi)$ si et seulement si

$$Max(\geq, Mod(\psi)) \subseteq Mod(\varphi)$$

c'est-à-dire : *dans les mondes préférés où ψ est vrai, φ doit être vrai*, ou *idéalement φ si ψ*.

Il reste à décider quel principe doit être adopté pour générer une relation de préférence $\geq_{GB}$ à partir d'un ensemble de désirs conditionnels $\mathcal{D} = \{D(\varphi_1|\psi_1), \ldots, D(\varphi_p|\psi_p)\}$ et d'un ensemble de contraintes physiques K. Ce choix serait simple si $\mathcal{D}$ était satisfait par un modèle unique, ce qui n'est bien entendu pas le cas en général.

La façon la plus immédiate (qu'on nommera (S)tandard) de générer $\geq_{GB}$ à partir de $\mathcal{D}$ et d'un ensemble de contraintes physiques K consiste à :

1. imposer $W = Mod(K)$ – on a donc $M = \langle Mod(K), \geq_M \rangle$; et
2. tenir compte de tous les modèles satisfaisant $\mathcal{D}$ – c'est-à-dire à utiliser la déduction : pour tous $x, x' \in Mod(K)$,

   $x \geq_{GB}^{cond,S} x'$ si et seulement si pour tout $M \models GB$ on a $x \geq_M x'$ [9]

   Une formulation équivalente est : $x \geq_{GB}^{cond,S} x'$ si et seulement si $\mathcal{D} \models D(for(x)|for(x) \vee for(x'))$. La relation de préférence ainsi construite

[7] La raison pour laquelle on peut faire cette identification, et par là se passer de faire figurer une fonction de valuation dans le modèle, est que la relation de conséquence de CO^*est inchangée selon qu'on impose ou non que deux mondes différents de W aient des valuations différentes.

[8] J'utilise la notation D (comme "désir", pas comme "dyadique" !) ; ailleurs, $D(\varphi|\psi)$ est notée $\psi \Rightarrow \varphi$ ou, dans [26], $I(\varphi|\psi)$ – I comme *idéalement* ; je préfère éviter d'utiliser I pour une raison qui va être bientôt claire.

[9] Rappelons qu'on identifie les valuations aux mondes.

$\geq_{GB}^{ded}$ est un préordre partiel.

Or, sauf dans des cas rares, l'un des deux problèmes survient toujours :

- GB a trop de modèles ;
- GB n'a pas de modèle.

Le second problème (incohérence de GB) est probablement plus marginal (il donnera lieu à un développement rapide à la fin de cette Section) que le premier, qui est la norme : le trop grand nombre de modèles de GB implique que $\geq_{GB}^{cond,S}$ est en général trop faible (elle ne permet pas assez de comparaisons). Il suffit pour s'en convaincre de considérer l'exemple suivant. Soit $GB = \{D(a|\top)\}$; pour tous $x, y \in \{(a,b),(a,\neg b),(\neg a,b),(\neg a,\neg b)\}$, on a $x \geq_{GB}^{cond,S} y$ si et seulement si $x = y$ (et par conséquent $x >_{GB}^{cond,S} \vec{y}$ n'est jamais vérifié : toutes les alternatives sont non-dominées).

Trop de modèles

Une façon de remédier au problème que l'on vient de mettre en lumière consiste à sélectionner *un* modèle satisfaisant GB : celui qui maximise la préférence monde par monde – ou de manière équivalente, la relation de préférence obtenue par Z-complétion de GB ([26], page 79). La relation de préférence $\geq_{GB}^{cond,Z}$ ainsi obtenue est bien plus discriminante que $\geq_{GB}^{cond,S}$, mais son inconvénient est que, comme pour $\geq_{GB^{cond,S}}$, un effet de "noyade" se produit fréquemment (certains buts conditionnels deviennent complètement ignorés).

Pour remédier à ce problème de noyade, j'ai proposé une définition alternative aux deux précédentes, qui évite les écueils de l'une et de l'autre. La relation de préférence $\geq_{GB}^{cond,U}$ a est construite à partir de K et de GB en ajoutant aux contraintes d'idéalité des contraintes supplémentaires exprimant que la violation d'un désir conditionnel induit explicitement une perte d'utilité [146] ; l'introduction d'utilités dans un cadre ordinal peut sembler incongrue, mais elle ne l'est pas car les fonctions d'utilité ne sont utilisées ici que comme des objets intermédiaires (la même construction pourrait être faite en se passant de fonctions d'utilité ; elle serait tout simplement moins commode). L'article [146] étudie les propriétés de cette construction et la met à l'épreuve des exemples.

Cependant, la relation de préférence $\geq_{GB}^{cond,U}$ induite est un ordre partiel, comme $\geq_{GB}^{cond,S}$. Dans certains cas, on peut la rendre plus spécifique par un

procédé de complétion en un préordre total – ou de façon équivalente une fonction d'utilité ; ce procédé est décrit dans un article écrit avec Leendert van der Torre et Emil Weydert [161]. Toutefois, cette complétion est souvent arbitraire et elle souffre, à mon avis, d'être *trop* spécifique (c'est-à-dire de permettre plus de comparaisons que nécessaire) – il me semble que $\geq_{GB}^{cond,U}$ est un meilleur compromis.

Enfin, dans l'article [161] nous allons plus loin dans l'expressivité des désirs conditionnels en introduisant deux paramètres quantitatifs : la *force* (quantitative) d'un désir, et sa *polarité*, qui exprime la proportion entre le gain induit par la satisfaction de $D(\varphi|\psi)$ et la perte induite par sa violation[10]. La polarité n'a d'intérêt que si l'objectif est de construire une fonction d'utilité u_{GB} à partir de GB – mais pas si l'on ne s'intéresse qu'à la relation de préférence $\geq_{GB}$.

La représentation de préférences par des désirs conditionnels est proche, par certains côtés, de la représentation d'obligations et de permissions (et en particulier d'obligations conflictuelles comme les "obligations contraires aux normes"). On trouvera une discussion sur ce sujet dans [146], [161] ainsi que dans l'article de Cholvy et Garion [44] qui discutent eux aussi de la pertinence de la logique CO^*pour la représentation de préférences et d'obligations. Le tableau suivant montre schématiquement trois interprétations différentes des conditionnels (voir [178, 216] pour une discussion poussée sur les diverses interprétations de relations d'ordre sur un ensemble de mondes).

	$\geq$	$D(\varphi\|\psi)$
interprétation épistémique	normalité	normalement φ si ψ
interprétation préférentielle	préférence	idéalement φ si ψ
interprétation déontique	permission	φ obligatoire si ψ

Pas de modèle (désirs conditionnels conflictuels)

Lorsque $\mathcal{D}$ est incohérent (dans CO^*), on a affaire, par définition à un ensemble de désirs conflictuels. Dans un article écrit avec Leendert van der Torre et Emil Weydert [161], nous avons examiné la signification des ensembles de désirs conflictuels, en avons ébauché une taxonomie et avons

[10]Un désir conditionnel, en tant qu'objet conditionnel [78], a trois "valeurs de vérité" possible dans un monde : $D(\varphi|\psi)$ est satisfait dans w si $w \models \psi \wedge \varphi$, violé si $w \models \psi \wedge \neg\varphi$ et inapplicable si $w \models \neg\psi$.

proposé des ébauches de solution (mais ce travail reste cependant préliminaire et essentiellement prospectif).

Avant d'aller plus loin dans les différentes significations possibles dun ensemble de désirs conflictuels, remarquons d'abord que les deux situations suivantes *ne sont pas conflictuelles* (en tout cas au sens de la logique des conditionnels CO^*) :

- l'existence de deux (ou plusieurs) désirs dont les conclusions sont contradictoires mais où le contexte de l'un est *plus spécifique* que celui de l'autre (des autres) ;
- les désirs équivalents d'un point de vue logique à des obligations "contrary-to-duty" (c'est-à-dire des désirs dont le contexte contredit la conclusion d'un autre désir).

Nous avons ensuite ébauché une classification des désirs conflictuels en deux classes :

- les *ensembles de désirs utopiques* sont des ensembles de désirs conditionnels, correspondant par exemple à plusieurs critères, dont la conjonction est incohérente en raison de présence de contraintes d'intégrité (des contraintes physiques, par exemple) qui rend leur conjonction concevable dans l'abstrait, mais inaccessibles dans la réalité. Les solutions techniques (que nous n'avons pas explorées) consisteraient à adapter aux logiques des conditionnels les méthodes de résolution d'incohérence dont nous avons parlé au chapitre 2, ou encore d'ajouter à l'ensemble des mondes des mondes fictifs (irréalisables mais concevables) qui auront la préférence maximale.
- les *désirs avec incertitude ou normalité cachée* sont des ensembles de désirs conditionnels conflictuels, dont l'incohérence ne peut pas être expliquée seulement en termes de préférence, mais doit considérer que sous-jacente à l'expression des désirs existe des présupposés concernant le caractère plus ou moins normal des mondes. Une sémantique permettant de traiter correctement des désirs conflictuels de ce type doit alors, comme celle de [26], faire intervenir une relation de normalité sur les mondes en sus de la relation de préférence. Cela dit, nous critiquons l'approche proposée par [26] et en proposons un raffinement, bien plus satisfaisant, dans laquelle $D(B|A)$ est satisfait si et seulement si les mondes préférés parmi les $A \wedge B$-mondes les plus normaux sont strictement préférés aux mondes préférés parmi les $A \wedge \neg B$-mondes les plus normaux. Voir [162].

4.3.1 Préférences "ceteris paribus"

Il existe une seconde interprétation des préférences contextuelles, tout-à-fait différente de la précédente : les préférences dites *ceteris paribus*. Contrairement aux logiques des conditionnels, ce langage de représentation a été construit dans le but spécifique de représenter des préférences. Les deux articles fondateurs datent de 1991 [64] [65]. Les préférences *ceteris paribus* interprètent un désir contextuel $C\ : G$ ainsi : *toutes choses non pertinentes étant égales par ailleurs, un monde où* $C \wedge G$ *est satisfait est préféré à un monde où* $C \wedge \neg G$ *est satisfaite*. Le point délicat de cette définition est l'interprétation à donner aux "choses non pertinentes". Preuve que cela n'est pas simple, les définitions proposées dans les articles sur le sujet ([64, 65, 213, 30, 63, 62]) diffèrent sur ce point. Certaines considèrent que ces "choses non pertinentes" sont toutes les variables propositionnelles sauf celles qui apparaissent dans G – c'est évidemment très simple lorsque G est toujours un littéral, comme dans [30] –; d'autres font excluent aussi les variables apparaissant dans C; d'autres préfèrent n'exclure que les variables dans $DepVar(G)$.

La définition que je donne ici généralise les définitions variées, sans pour autant compliquer le formalisme (ni faire monter sa complexité computationnelle) : il suffit de considérer une donnée supplémentaire dans l'expression d'un désir *ceteris paribus*, à savoir l'ensemble V des variables pertinentes pour ce désir. Pour chaque i, en plus des formules propositionnelles C_i, G_i et G'_i, on considère V_i un sous-ensemble de VAR tel que $Var(G_i) \cup Var(G'_i) \subseteq V_i$, Voici maintenant la définition de R_{CP} :
Soit $GB = \langle K, \mathcal{D} \rangle$ où $\mathcal{D} = \{C_1\ : G_1 > G'_1[V_1], \ldots, C_m\ : G_m > G'_m[V_m]\}$ et deux alternatives $x, y \in \mathcal{X}$. On dit que x *domine* y pour $D_i = (C_i\ : G_i > G'_i[V_i])$, ce que l'on note $\vec{x} >_{D_i} \vec{y}$, si et seulement si les conditions suivantes sont vérifiées :

1. $x \models K \wedge C_i \wedge G_i \wedge \neg G'_i$;
2. $y \models K \wedge C_i \wedge \neg G_i \wedge G'_i$;
3. x et y coïncident sur toutes les variables qui ne sont pas dans V_i.

Maintenant, l'ordre strict $>_{GB}$ est défini à partir des relations de dominance qui précèdent par fermeture transitive : $x >^{cp}_{GB} y$ si et seulement si il existe une suite finie $x_0 = x, x_1, \ldots, x_{q-1}, x_q = y$ d'alternatives telles que pour tout $j \in \{0, \ldots, q-1\}$ il existe un $i \in \{1, \ldots, m\}$ tel que $x_j >_{D_i} x_{j+1}$. Enfin, $\geq_{GB}$ est défini par $x \geq_{GB} y$ if and only if $x >_{GB} y$ ou $x = y$.

J'ai étudié dans [149] la complexité des problèmes COMPARAISON et

NON-DOMINANCE pour les désirs *ceteris paribus* : de façon surprenante, le problème COMPARAISON est PSPACE-complet alors que NON-DOMINANCE est seulement coNP-complet[149]. La preuve de PSPACE-complétude vient d'une réduction du problème de planification déterministe avec représentation STRIPS. Il faut noter que pour la restriction des "réseaux ceteris paribus" [63] [62], la complexité du problème COMPARAISON est encore ouverte[11]

Le langage des préférences *ceteris paribus* a été récemment étendu de façon à introduire des préférences quantitatives [29] : encore un exemple de la réintroduction du quantitatif dans un formalisme ordinal.

4.4 Langages logiques à base de distances

Revenons aux préférences quantitatives. Le principe des langages de représentation de préférences à base de distances est que, lorsque l'agent spécifie un but G, il exprime implicitement que, si le but n'est pas satisfait, alors plus la solution x est "loin" de G, plus elle est mauvaise. Formellement, dans un langage logique de représentation de préférences à base de distances ([141], [142], [134]), une base de préférences est un couple $GB = \langle \{G_1, \ldots, G_n\}, d \rangle$ où d est une pseudo-distance sur $Mod(K)$. La fonction d'utilité u_{GB} induite par GB est définie d'abord dans le cas où GB ne contient qu'un seul but par $u_{GB}(x) = -d(x, G)$, puis dans le cas général par $u_{GB}(x) = -F(d(x, G_1), ..., d(x, G_n))$, où F est une fonction d'agrégation commutative et non-décroissante[12].

Il reste à savoir comment cette distance est spécifiée. L'est-elle par l'agent ou est-elle fixée ? Si elle est spécifiée par l'agent, il se pose naturellement un problème d'explosion combinatoire qui conduit à un besoin de représentation compacte. Si elle est fixée, comment la choisir ? Dans la plupart des articles en IA qui font usage de distances (notamment, les articles sur la révision, la mise à jour et la fusion de croyances), ce problème est négligé puisqu'on

[11] Les contraintes syntaxiques supplémentaires sont (a) les buts sont des littéraux, et surtout (b) les variables sont structurées en un graphe acyclique $\Gamma - v' \in \Gamma(v)$ si les préférences sur v dépendent de v' – et pour toute variable v et pour toute effectation complète γ des variables de $\Gamma(v)$ il existe un désir ceteris paribus $\gamma : v > \neg v$ ou $\gamma : \neg v > v$ dans GB. À la suite de discussions avec Carmel Domshlak, nous sommes restés sur l'impression que le problème de comparaison reste PSPACE-complet. On peut trouver des classes traitables pour ce problème dans [63].

[12] Plus généralement, on peut considèrer que la pseudo-distance varie avec les buts : $GB = \{\langle G_1, d_1 \rangle, ..., \langle G_n, d_n \rangle\}$, et $u_{GB}(x) = F(d_1(x, G_1), ..., d_n(x, G_n))$. Je ne vois cependant pas la pertinence, en pratique, d'une telle construction.

suppose que la distance utilisée est simplissime (en général, la distance de Hamming d_H – où $d_H(x, x')$ est définie comme le nombre de variables propositionnelles affectées différemment par x et x'). Dans l'article [142] nous proposons des familles de distances plus sophistiquées.

Dans [141], [142], on montre que les langages logiques à base de distances et les langages logiques à base de pondérations sont conceptuellement proches ; ceci dit, computationnellement parlant, la différence est plus marquée : on peut transformer une base de préférences à base de pondérations GB en une base de base de préférences à base de distances GB' d'espace polynomial (en la taille de GB) qui lui soit équivalente[13] L'existence d'une transformation dans l'autre sens est probablement impossible (voir [51]), mais il existe pourtant une transformation naturelle, dans le sens distances vers pondérations, qui génère un output de taille exponentielle dans le cas général mais qui reste raisonnable dans de nombreux cas, et qui est intéressante d'une point de vue représentationnel [141]. Elle passe par la définition d'un *disque de préférence* autour d'une formule : étant donné une pseudo-distance d, on définit $D(\varphi, k)$ comme une formule (unique à l'équivalence près) dont les modèles sont $\{\omega \mid d(\omega, \varphi) \leq k\}$. On donne dans [141] un exemple de situation où ce disque est de taille polynomiale : lorsque $\varphi = D_1 \vee \ldots \vee D_n$ avec $D_i = l_1 \wedge \ldots \wedge l_{|D_i|}$ est en DNF et d est la distance de Hamming, on a

$$D(\varphi, k) = \bigvee_{i=1}^{n} (k : l_1, \ldots, l_{|D_i|})$$

(où $k : l_1, \ldots, l_{|D_i|}$ est une formule de cardinalité, qui, on le rappelle, est un raccourci syntaxique exprimable en espace polynomial à partir des connecteurs habituels). Dans une perspective de fusion des croyances, [130] propose des transformations analogues.

Discussion

Comparer des langages de représentation de préférences est une tâche malaisée, parce que plusieurs critères pertinents existent.

D'un point de vue "représentationnel", la *pertinence cognitive*, ou plus exactement la proximité d'un langage de représentation avec la façon dont

[13] Cette définition d'équivalence n'est pas triviale, mais je ne souhaite pas entrer dans les détails ; ceux-ci se trouvent dans [37, 169].

les individus appréhendent leurs préférences et les expriment en langage naturel, est un facteur important en ce qui concerne la facilité d'obtenir les préférences au cours de la phase d'élicitation (on en reparlera en conclusion de ce chapitre).

D'un point de vue computationnel, la complexité des problèmes de comparaison et de non-dominance pour les différents langages de représentation dont on a parlé dans ce chapitre est synthétisée sur le tableau suivant (voir [149] pour plus de détails).

	COMPARAISON	NON-DOMINANCE	CAND-OPT-SAT
R_{brute}	P	coNP-complet	NP(3)-complet
R_{wg}	P	coNP-complet	jusqu'à Δ_2^p-complet
$R_{\succ}$	P	coNP-complet	jusqu'à Σ_2^p-complet
$R_{cond,S}$	coNP-complete	(BH_2-dur, dans Σ_2^p)	(BH_2-dur, dans Σ_2^p)
$R_{cond,Z}$	Θ_2^p-complet	Θ_2^p-complet	Θ_2^p-complet
R_{cp}	PSPACE-complet	coNP-complet	Σ_2^p-complet
R_d	jusqu'à Δ_2^p-complet	jusqu'à Δ_2^p-complet	jusqu'à Δ_2^p-complet

L'origine de ces résultats est diverse. La plupart d'entre eux peut être obtenue plus ou moins facilement à partir de résultats de complexité pour des problèmes d'*inférence* dans des langages analogues mais développés pour la représentation de *connaissances*, et en particulier, de [134] pour R_d et R_{wg}, de [42, 186] pour $R_{\succ}$, de [105] pour $R_{cond,S}$, de [88] pour $R_{cond,Z}$.

Ces résultats ont une valeur en eux-mêmes, puisqu'ils permettent une première comparaison des langages de représentation de préférences d'un point de vue computationnel. Cependant, il manque encore une étude de leur *complexité représentationnelle* [37], ou, en d'autres termes, de leur "pouvoir de concision" : c'est le sujet sur lequel nous sommes en train de travailler avec Sylvie Coste-Marquis, Paolo Liberatore et Pierre Marquis, dans le cadre d'un projet franco-italien Galilée [51].

4.5 De la représentation de préférences à la décision de groupe

Depuis quelques années se développent, au sein de la communauté de l'intelligence artificielle, des thématiques de recherche à la frontière de l'économie, qui traitent d'interaction, de coopération ou de négociation dans une

société d'agents. J'essaie ici d'établir une classification très succinte des différents types de problèmes d'intelligence artificielle concernant des communautés d'agents[14].

1. *décision distribuée (environnements dits "multi-agents")* :
   (a) *coopération et résolution distribuée de problèmes* : ce thème concerne la résolution de problèmes complexes par un ensemble d'agents ayant chacun ses propres capacités (dans certains cas, les agents sont totalement autonomes, dans d'autres ils sont plus ou moins contrôlés par une entité centrale) ;
   (b) *jeux à plusieurs joueurs* : ce thème concerne la construction de stratégies pour des agents qui n'ont pas nécessairement les mêmes objectifs ;
2. *communication et dialogue* : on étudie les effets des actes de communication (par exemple lors d'un dialogue) sur les croyances (individuelles, mutuelles et communes) des agents, ainsi que sur leurs intentions. (À noter que le dialogue peut être vu comme un jeu – on revient alors au thème précédent).
3. *choix social* : il s'agit ici de problèmes de décision de groupe, où la décision est centralisée (qu'elle soit prise par la communauté d'agents ou par une entité coordinatrice). Il s'agit, dans ce type de problèmes, de prendre une décision commune en fonction des préférences, éventuellement conflictuelles, que les agents ont sur les conséquences de la décision. Selon que la nature de la décision à prendre, on distingue deux sous-classes de problèmes :
   (a) *problèmes de partage* : la décision concerne ici la répartition d'un ensemble de biens entre des agents, étant données les préférences de ceux-ci sur les ressources qui peuvent leut être allouées. Plusieurs hypothèses peuvent être distinguées, ce qui permet d'établir une classification plus fine des problèmes : ainsi, pour les problèmes d'*enchères combinatoires*, il s'agit, pour une entité centrale (qui joue le rôle d'un commissaire-priseur) de déterminer une attribution optimale de chacun d'un ensemble de biens à l'un

[14]Cette brève synthèse se fonde sur les articles des conférences généralistes d'intelligence artificielle (IJCAI, AAAI, ECAI) des cinq dernières années – il est évident que pour une synthèse plus poussée il faudrait examiner les actes des conférences plus spécialisées, mais ce n'est pas notre objectif – nous partons du principe que les conférences généralistes donnent une vue à peu près correcte du domaine.

des agents, sachant que chaque agent a au préalable exprimé la somme d'argent qu'ils est prêt à payer pour chaque combinaison possible de biens (voir par exemple [203]) ; pour les problèmes de *partage équitable*, en revanche, l'optimalité est fonction de l'équité du partage et non plus des gains du commissaire-priseur (voir par exemple [166]) ;

(b) *vote* : un ensemble d'agents (appelés *votants*) exprime des préférences, éventuellement conflictuelles, sur un choix à faire parmi un ensemble de solutions possibles (appelées *candidats*) concernant un choix collectif ne mettant pas en jeu de biens consommables et/ou partageables : ici, tous les agents sont en principe concernés par tous les aspects de la décision – alors qu'en ce qui concerne le partage de ressources, les préférences d'un agent portent exclusivement sur la part qui lui est attribuée. Il s'agit alors de combiner ou d'agréger (éventuellement après des phases d'identification des conflits et de négociation) ces préférences de manière à dégager un bon compromis ; la décision finale, une fois déterminée, est alors prise de manière centralisée (par le groupe de votants).

Pour chacun de ces problèmes, dès que l'espace des alternatives possibles, ou plus précisément sa projection sur le *champ de pertinence de chaque agent*[15], a une forte combinatoire, il faut s'intéresser à des langages de représentation compacte des structures préférentielles concernées.

Jusqu'à une époque très récente, mes contributions n'ont pas concerné la décision distribuée. Ce qui suit concerne donc les deux dernières classes de problèmes (partage de ressources et vote).

4.5.1 Vote combinatoire

Dans ce qui suit, on considère un ensemble d'agents (appelés *votants*) exprimant leurs préférences sur un ensemble *fini* d'alternatives $\mathcal{X}$ (appelées plus simplement *candidats*). On appelle *profil de préférences* la donnée d'une structure de préférences (fonction d'utilité ou relation de préordre) pour chacun des votants.

[15] Par exemple, dans un problème d'affectation de ressources, on fait en général l'hypothèse que la satisfaction de l'agent ne dépend que des biens qui lui sont attribués : les ensembles de "paniers de biens" qui peuvent être alloués à un agent constituent son champ de pertinence.

La théorie du vote (voir [183] pour une excellente synthèse), qui est une branche de la théorie du choix social, définit une règle de vote comme une fonction associant à tout profil de préférences P un candidat unique. Le problème des règles de vote est leur impossibilité à traiter de façon satisfaisante les ex-aequo[16] ; pour cette raison, on travaille plutôt sur des *correspondances de vote* ("voting correspondence"), notées C, définies comme des fonctions $select_C$ associant à tout profil de préférences P un sous-ensemble de candidats, qu'on notera $Select_C(P)$.

Les différentes propriétés des règles et correspondances de vote ont été bien étudiées en théorie du choix social ; cependant, on considère généralement que les candidats sont listés explicitement (ce sont, typiquement, des personnes ou des listes de personnes, comme lors d'élections politiques), et de ce fait relativement peu nombreux. Lorsque l'espace des candidats a une structure combinatoire forte, les langages de repŕesentation de préférences dont on a parlé dans ce chapitre sont donc particulièrement pertinents pour l'expression des données du problème.

On appellera *vote combinatoire* le problème de la définition, de l'étude et de la mise en oeuvre algorithmique de correspondances de vote lorsque les préférences sont exprimées dans un langage de représentation compacte. Un problème de vote combinatoire est défini par deux paramètres :
- la langage de représentation des préférences, noté R ;
- la correspondance de vote, notée C.

Pour tout langage de représentation R, on définit un *R-profil à p votants* comme une collection $B = \langle GB_1, ..., GB_p \rangle$ de bases de buts (une pour chaque votant), exprimée dans le langage R, générant un profil $P = Induce_R(B)$, c'est-à-dire , selon le cas :
$Induce_R(B) = \{u_{GB_1}, ..., u_{GB_p}\}$ (préférences évaluationnelles) ou
$Induce_R(B) = \{\geq_{GB_1}, ..., \geq_{GB_p}\}$ (préférences relationnelles).

L'article [149] examine divers langages de représentation de préférences et diverses familles de correspondances sous l'angle du vote combinatoire ; plus précisément :

[16] Il existe, pour pallier cette difficulté, deux échappatoires possibles : (1) l'abandon de la neutralité entre candidats (on définit un ordre *a priori* sur les candidats et on sélectionne le candidat préféré – au sens de cet ordre – dans $Select_C(P)$) ; (2) le recours à un tirage au sort entre les candidats de $Select_C(P)$.

– la *pertinence* des diverses familles de correspondances de vote, lorsque l'espace des candidats a une structure combinatoire et est exprimé dans un langage de représentation compacte ;
– la *complexité algorithmique*. Pour une correspondance de vote C et un langage de représentation R fixés, on est amené à considèrer les problèmes suivants : (a) étant donnés $B = \langle GB_1, ..., GB_p \rangle$ et un candidat x, déterminer si $x \in Select_C(Induce_R(B))$; (b) étant données $B = \langle GB_1, ..., GB_p \rangle$ et une formule ψ, déterminer s'il existe un candidat dans $Select_C(Induce_R(B))$ satisfaisant ψ.

La complexité théorique de ces problèmes dépend avant tout de ce que les problèmes de comparaison et de non-dominance dans le langage R est polynomial ou non.

Perspectives

Vote électronique La perspective d'application immédiate du vote combinatoire est le vote électronique concernant des décisions concernant plusieurs variables dépendantes à prendre au sein d'organisations de petite taille (entreprises, laboratoires, commissions de spécialistes etc.). Quand la décision à prendre n'a pas de combinatoire particulière (par exemple, choisir *un* candidat à recruter sur un poste), tout ceci peut être fait manuellement, mais lorsque l'espace des décisions explose combinatoirement, cela n'est plus possible[17].

Un problème clé non abordé dans cet article, indispensable à la réalisation de logiciels pour le vote combinatoire électronique, est l'élaboration de correspondances automatisées d'*élicitation des préférences* (c'est-à-dire , comment interagir avec le votant pour construire sa relation de préférence). Pour ce faire, l'identification d'independances préférentielles entre variables

[17] On peut le voir sur l'exemple suivant, concernant le vote en commission de spécialistes : lorsque non plus un seul, mais k candidats (parmi n), doivent être recrutés, l'espace des décisions possibles a une forte combinatoire et ne peut être résolu manuellement qu'en ignorant les dépendances entre variables : ainsi, lors d'un vote de commission de spécialistes, on n'exprime jamais sur le bulletin de vote les *corrélations* entre candidats, par exemple : "mon choix me porte en priorité sur A, puis B en second lieu, puis C en troisième, mais, A et B travaillant sur des thèmes voisins alors que C leur est complémentaire, le recrutement conjoint de A et de C, voire de B et C, me plait davantage que le recrutement de A et B". Permettre l'expression de telles préférences est difficilement envisageable en pratique dans un contexte purement manuel.

pour un votant donné (voir [4]) sera d'une grande utilité.

Vote partiel et négociation Il est des situations où il n'est pas nécessaire de déterminer automatiquement une décision complète, c'est-à-dire une affectation de *toutes* les variables. On peut alors s'intéresser à l'identification (mesure et localisation) des conflits, puis à élaboration d'une décision partielle (une affectation d'un sous-ensemble des variables) qui soit à la fois la moins conflictuelle possible et la plus complète possible ; cette décision partielle est ensuite communiquée aux votants ; ces derniers passent ensuite dans une phase de discussion et de négociation (non automatisée) à l'issue de laquelle ils redéfinisent leurs profils de préférences sur les affectations des variables restantes. Dans certains cas, le nouveau profil de préférences d'un agent sera juste la restriction de son ancien profil sur les variables restantes, mais ce ne sera pas toujours le cas : il se peut que la discussion ait mis en lumière des informations dont certains agents ne disposaient pas, que certains agents aient été plus ou moins convaincus par d'autres, ou que des négociations (par exemple des promessses de compensations) aient eu lieu entre agents. À partir du nouveau profil de chacun des agents, on peut alors procéder soit au calcul d'une affectation optimale complète sur les variables restantes, soit à une nouvelle étape d'identification des conflits (sur les variables restantes) et de négociation. Le processus est itéré jusqu'à ce que toutes les variables aient été affectées.

Ce problème nécessite une adaptation des correspondances de vote combinatoire à des affectations partielles, ce qui constitue une piste de recherche intéressante.

4.5.2 Problèmes de partage

Je travaille depuis environ un an avec Hélène Fargier, Michel Lemaître et Gérard Verfaillie, en collaboration avec une équipe du CNES, sur un problème d'allocations de prises de vues pour un satellite d'observation. C'est un problème de partage de ressources : le satellite d'observation a été financé (à parts inégales) par plusieurs entités (pays) ; ces entités déposent chaque jour des demandes (c'est-à-dire des prises de vues d'un endroit spécifique) ; les demandes déposées par les différentes entités sont en général conjointement incompatibles, à cause de contraintes physiques limitant la capacité de rotation du satellite. Il s'agit, pour chaque période de temps, de décider des prises de vues à allouer aux différentes entités de façon égalitaire (modulo

les différences d'investissement initial).

L'article en soumission [93] examine les problèmes de partage (de ressources indivisibles et finies) de façon plus générale et d'un point de vue informatique. On y propose un langage de représentation compacte pour l'expression des demandes, qui est un langage logique à pondérations ou à priorités avec deux niveaux d'agrégation (un niveau intra-agent et un niveau inter-agent) ; on discute des propriétés souhaitables des fonctions d'agrégation ; on examine la complexité des problèmes de partage lorsque différents paramètres varient (caractère préemptif ou non de l'allocation, existence de contraintes autres que les contraintes de préemption, nature égalitariste ou utilitariste des fonctions d'agrégation), et on propose un algorithme de calcul de partages optimaux.

4.5.3 Manipulation de règles de vote

On sait depuis longtemps en théorie du choix social que dès que le nombre de candidats est supérieur ou égal à 3, il n'existe pas de règle de vote qui soit à la fois non-dictatoriale et non-manipulable (théorème de Gibbard-Sattherthwhaite). Les règles dictatoriales étant de façon évidente à rejeter, on en arrive à la conclusion apparemment très négative que quelle que soit la règle choisie, les votants seront incités à exprimer leurs préférences de façon stratégique plutôt que sincère (par exemple, un votant prétendra préférer un candidat à un autre alors que ce n'est pas le cas).

Quels sont les moyens possibles, en pratique, de "faire avec" ce résultat négatif ? Une première façon de faire consiste à imposer des restrictions sur les profils de préférence que les agents peuvent exprimer ; cette solution n'est raisonnable que dans des domaines très spécifiques. Une seconde façon d'y remédier partiellement est de s'assurer que la manipulation est *suffisamment difficile à calculer* pour qu'il soit en pratique peu plausible que les votants aient les ressources computationnelles suffisantes pour trouver une bonne manipulation, et qu'ils y renoncent pour finalement exprimer leurs préférences sincères. Il faut noter ici quelque chose d'assez inhabituel : la forte complexité computationnelle est ici *souhaitable* (tout comme en cryptographie).

Or, la complexité de la manipulation selon la règle de vote adoptée a été peu étudiée et il reste des problèmes ouverts. [10, 9] donnent des résultats, en supposant que le nombre de candidats n'est pas borné, ce qui ne permet

pas d'évaluer la difficulté du problème lorsque le nombre de candidats est fixé à un petit entier. [45] se sont attaqués au problème en étudiant l'impact du nombre de candidats sur la complexité du problème de l'existence d'une manipulation, mais plusieurs problèmes restaient ouverts. Dans un article écrit en collaboration avec Vincent Conitzer et Tuomas Sandholm (Carnegie Mellon University), nous avons identifé précisément, pour une dizaine de règles de vote différentes, le nombre de candidats à partir duquel ce problème "saute de P à NP-complet" (de façon assez surprenante, pour plusieurs protocoles classiques, le saut de complexité se situe non pas entre 2 et 3, mais entre 3 et 4).

Ce travail est cependant encore assez préliminaire, puisque la définition d'une manipulation est très restrictive (elle suppose que la coalition de votants qui recherche une manipulation a une connaissance complète des votes des votants à l'extérieur de la coalition) ; nous sommes justement en train de poursuivre nos travaux de façon à prendre en considération des notions de manipulation plus faibles.

Chapitre 5

Processus de prise de décision purement épistémiques

Dans les deux derniers chapitres de ce mémoire nous nous intéressons à des agents actifs et aux politiques d'action qu'ils entreprennent pour réaliser leurs buts. Pour des raisons qui tiennent avant tout à l'équilibre des chapitres, je choisis de traiter dans deux chapitres différents les problèmes purement épistémiques et les problèmes de planification plus généraux (qui font intervenir à la fois des actions épistémiques et des actions effectives).

Dans ce chapitre on s'intéresse donc à tout ce qui touche aux actions épistémiques : comment les représenter et dans quels langages ? comment définir simplement et résoudre des problèmes de planification purement épistémiques en logique propositionnelle ? quelle est la complexité de ces problèmes ? on discute enfin de quelques exemples de problèmes de planification purement épistémique en intelligence artificielle.

5.1 Introduction

Les hypothèses que nous faisons tout au long de ce chapitre sont les suivantes :

1. l'agent est épistémiquement volitif : il a du goût pour la connaissance, ce qu'on peut exprimer simplement par l'axiome suivant, qui dit que l'utilité de l'agent croît avec l'information dont il dispose.

   monotonie informative Pour tous états de croyance b, b', si b est

plus informatif que b' alors b' n'est pas préféré à b.[1]

Cette définition suppose que l'espace des états de croyance B est muni d'une relation d'informativité, à savoir un préordre $\sqsupseteq$ ("au moins aussi spécifique/informatif que")[2].

2. l'agent est indifférent à l'état du monde (il n'est intéressé que par l'acquisition de connaissances sur ce dernier).

   indifférence au monde

   pour tous états de croyance complets $\{s\}$, $\{s'\}$, on a $\{s\} \sim \{s'\}$ (où $\sim$ est la relation d'indifférence associée à $\sqsupseteq$).

3. l'agent a à sa disposition un ensemble d'actions épistémiques ACT_E – mais pas d'actions effectives.

La structure de préférence épistémique peut aussi être modélisée quantitativement par une *fonction d'utilité épistémique* $u_E : \mathcal{B} \to \mathbb{R}$. Attention cependant à la signification de $u_E(b)$: *ce n'est pas l'utilité espérée sachant* b. Il faut bien distinguer l'utilité "ontique" $u : \mathcal{S} \to \mathbb{R}$ de l'utilité epistémique u_E *qui n'est pas déterminée par* u – et qui n'a d'ailleurs en général aucun lien avec u : il suffit de remarquer qu'un agent [purement] épistémiquement volitif est indifférent au monde et a donc une utilité ontique u identiquement nulle (l'utilité espérée $\bar{u}_b$ sachant un état de croyance b étant donc nulle aussi), alors que u_E ne l'est en principe pas. Voici quelques exemples de fonctions d'utilité épistémiques (on en verra d'autres au cours de ce chapitre) :

- soit une formule φ dont on s'intéresse à la valeur de vérité (voir la Section 5.3) ; si les états de croyance sont tout-ou-rien : $u_E(b) = +1$ si $b \models \varphi$ ou $b \models \neg\varphi$, $u_E(b) = +0$ sinon ; si les états de croyance sont probabilistes, une fonction d'utilité épistémique pertinente est $u_E(p) = |P(\varphi) - P(\neg\varphi)|$ (qui est égale à la probabilité d'erreur lorsqu'on parie sur la valeur de vérité de φ) ;
- supposons cette fois qu'on s'intéresse à la connaissance sur le système tout entier. Lorsque les états de croyance sont tout-ou-rien, voici des fonctions d'utilité épistémique :

[1] On retrouve ici l'aversion pour l'ignorance [79] mais dans un tout autre contexte.

[2] Selon la nature de B , le choix de $\sqsupseteq$ est plus ou moins évident : ainsi, dans le modèle possibiliste, on peut poser $\pi \sqsupseteq \pi'$ si et seulement si $\pi \leq \pi'$; en en particulier dans le modèle tout-ou rien, $b \sqsupseteq b'$ si et seulement si $b \subseteq b'$; un raffinement total de cette relation est la cardinalité [floue] : $\pi \sqsupseteq \pi'$ si et seulement si $|\pi| \leq |\pi'|$, où $|\pi| = \sum_{s \in \mathcal{S}} \pi(s)$. Dans le modèle probabiliste, le choix est moins facile ; une façon de faire – assez discutable, cependant – est d'utiliser l'entropie : $p \sqsupseteq p'$ si et seulement si $H(p) \leq H(p')$, où $H(p) = -\sum_{s \in \mathcal{S}} p(s).\ln p(s)$.

– $u_E^1(b) = |\{v \in Var | b \models v \text{ ou } \ b \models \neg v\}|$;
– $u_E^2(b) = |Var| - \ln_2 |Mod(b)|$ est une version "tout-ou-rien" de l'entropie qui tient compte des croyances sur les dépendances entre variables[3], au contraire de $u_E^1(b)$; on a $u_E^2(b) \geq u_E^1(b)$, et les deux coïncident lorsque $\forall v, v' \in Var$, v et v' sont fortement conditionnellement indépendantes selon b (voir Section 2.3).

Avec des états de croyance probabilistes, deux fonctions d'utilité épistémique qui jouent des rôles analogues à u_E^1 et u_E^2 sont $u_E^1(p) = \sum_{v \in Var} |P(v) - P(\neg v)|$ et $u_E^2(p) = |Var| - \frac{H(p)}{\ln 2}$.

On peut alors donner un modèle très général d'un processus de planification purement épistémique, indépendant du modèle choisi pour la modélisation de l'incertitude et des préférences.

Un problème de planification purement épistémique est un processus de prise de décision (cf. section 1.1) où $\mathcal{A}$ est un ensemble d'actions épistémiques et $\mathcal{G}$ est une structure de préférence d'agent épistémiquement volitif.

En particulier, la fonction *evol* laisse inchangé l'état du monde, mais pas l'état de croyance de l'agent : *evol* spécifie les résultats épistémiques escomptés d'une action, en projetant l'état de croyance actuel sur les états de croyance suivants. Le problème de la représentation de cette fonction *evol*, c'est-à-dire le problème de la *représentation des effets des actions épistémiques*, a été moins souvent traité qu'en ce qui concerne les actions effectives[4].

5.2 Une formalisation des actions épistémiques en logique modale

On s'intéresse ici à des états de croyance propositionnels et à la description d'action épistémiques.

Décrire une action épistémique consiste à spécifier ses effets sur l'état de croyance de l'agent. Pour ce faire, il est indispensable de pouvoir distinguer formellement entre propriétés objectives du monde et croyances/connaissances

[3] $-\ln_2 |Mod(b)|$ est appelée *entropie de Hartley*.

[4] Plus exactement, il l'a été seulement dans des cas simples (en particulier dans le cadre du calcul des situations : voir [205] et [167]).

de l'agent (comment, sinon, exprimer qu'un agent qui ne connaît pas la valeur de vérité de φ la connaîtra après avoir effectué l'action de tester φ ?) et il est tout aussi indispensable de pouvoir représenter les interactions complexes entre actions et connaissances/croyances, qui sont de deux natures distinctes :

- l'exécution d'une action épistémique influe sur les croyances de l'agent ;
- le choix d'une action à exécuter dépend de l'état des croyances actuel de l'agent.

Un langage adéquat pour la représentation d'actions épistémiques doit donc être en mesure de représenter des *actions* et des *connaissances* (ou des *croyances*) et leurs interactions.

Pour cela nous développons depuis plusieurs années (avec Andreas Herzig, Dominique Longin et Thomas Polacsek, et plus récemment Philippe Balbiani et Noël Laverny) une famille de logiques multimodales intégrant des modalités dynamiques (pour les actions) et épistémiques/doxastiques (pour les connaissances/croyances). Cette famille de logiques n'est pas seulement dédiée aux problèmes purement épistémiques mais plus généralement aux problèmes de raisonnement et de planification en environnement partiellement observable ; nous y reviendrons donc au chapitre 6, cependant je choisis (pour des raisons liées au plan de ce document – en outre, cette présentation en deux étapes offre plus de lisibilité) de commencer par la restriction de ces logiques aux seules actions épistémiques, et de bâtir ensuite sur cette dernière pour prendre en compte les actions effectives.

Un mot sur le choix entre connaissances et croyances. L'élément crucial qui permet de trancher entre l'utilisation d'une logique épistémique (connaissances) ou doxastique (croyances) est la fiabilité des croyances initiales et acquises (observations). Comme, d'une part, l'étude d'une logique épistémico-dynamique et celle d'une logique doxasto-dynamique présentant peu de différences, et que d'autre part nous ferons la plupart du temps cette hypothèse de fiabilité, on ne parlera à partir de maintenant (sauf mention contraire) que de la famille épistémico-dynamique.

La logique épistémico-dynamique de base, appelée EDL, a été définie dans un article écrit avec Andreas Herzig, Dominique Longin et Thomas Polacsek [121]. Sa restriction aux seules actions épistémiques (et des extensions de cette restriction) a été étudiée plus en profondeur dans un autre article,

écrit avec Andreas Herzig et Thomas Polacsek [125]. Enfin, ses aspects de décidabilité et de complexité ont fait l'objet du sujet de DEA (2001-2002) de Noël Laverny [164], encadré essentiellement par Philippe Balbiani.

Ce qui manque à la logique dynamique pour qu'elle puisse représenter des plans sous connaissance incomplète est, d'une part, la possibilité pour les actions d'avoir des effets épistémiques et d'autre part, la possibilité, pour les plans, de "brancher"[5] sur des conditions épistémiques. Ce second point mérite de plus amples commentaires : en effet, la logique dynamique propositionnelle (PDL) permet certes d'exprimer certaines actions conditionnelles, de la forme `if` φ `then` α `else` β, mais ces dernières ne sont pas adéquates pour exprimer des plans conditionnels, parce que tandis qu'on peut souvent supposer qu'un programme connaît à chaque instant la valeur de chaque variable[6], ce n'est plus vrai pour un agent qui exécute un plan dans un environnement partiellement observable.

La logique EDL [121] combine un fragment de PDL et de la logique épistémique S5. Son langage est construit à partir d'un ensemble fini de variables propositionnelles Var, d'un ensemble fini d'actions atomiques $\mathcal{A}$, des connecteurs usuels, des constructeurs dynamiques "$\emptyset$", ";" et "`if then else` ", des modalités "[.]" et de la modalité épistémique "$\mathbf{K}$". On définit successivement plusieurs classes de formules.

- une *formule objective* est une formule sans modalité ;
- une *formule statique* est une formule de S5 (donc sans modalité dynamique) ;
- un *atome épistémique* est une formule statique de la forme $\mathbf{K}A$ (par exemple, $\mathbf{K}(a \vee \neg\mathbf{K}(a \rightarrow [\alpha]b))$) ;
- Une *formule épistémiquement interprétable* est une formule construite à partir d'atomes épistémiques et des connecteurs usuels. Par exemple, $\mathbf{K}(a \vee \neg\mathbf{K}(a \rightarrow [\alpha]b)) \vee \neg(\mathbf{K}[\beta]c \wedge \mathbf{K}d)$ est une formule épistémique, mais pas $a \vee \neg\mathbf{K}(b \wedge \mathbf{K}c)$.

On note les formules objectives par des lettres grecques minuscules (φ, ψ etc.) et les autres formules de EDL par des lettres grecques majuscules (Φ, Ψ etc.).

On définit maintenant les *plans interprétables* inductivement :

- l'action vide λ est un plan interprétable ;

[5] Encore un anglicisme que je ne sais pas traduire élégamment en français.

[6] Il y a cependant de nombreuses exceptions à cette règle, notamment en informatique répartie – c'était d'ailleurs la motivation initiale des constructions logiques épistémico-temporelles de [91].

– toute action atomique $\alpha \in \mathcal{A}$ est un plan interprétable ;
– si π et π' sont des plans interprétables alors $[\pi; \pi']$ est un plan interprétable ;
– si π et π' sont des plans interprétables et Φ est une formule épistémiquement interprétable alors [`if` Φ `then` π `else` π'] est un plan interprétable.

Un plan est donc interprétable si ses conditions de branchement sont épistemiquement interprétables : l'agent peut décider s'il *sait* qu'une formule donnée est vraie (alors qu'il n'est pas nécessairement capable de décider si une formule donnée est *vraie* dans le monde réel).

Enfin, le langage de EDL est défini inductivement comme suit :
– si $x \in Var$ alors x est une formule de EDL ;
– si Φ et Ψ sont des formules de EDL et π est un plan interprétable, alors $\Phi \Rightarrow \Psi, \Phi \wedge \Psi, \Phi \vee \Psi, \neg\Phi$ et $[\pi]\Phi$ sont des formules de EDL ;

La sémantique de EDL est définie en termes de modèles de Kripke qui contiennent une relation d'accessibilité épistémique $R_{\mathbf{K}}$ (relation d'équivalence) et une relation d'accessibilité dynamique R_α pour chaque $\alpha \in \mathcal{A}$, avec la contrainte suivante entre $R_{\mathbf{K}}$ et les R_α :

$$R_\alpha \circ R_{\mathbf{K}} \subseteq R_{\mathbf{K}} \circ R_\alpha$$

qui est la traduction sémantique de l'axiome

$$\mathbf{K}[\alpha]\Phi \rightarrow [\alpha]\mathbf{K}\Phi$$

Cet axiome est une version dynamique de l'axiome de non-oubli chez [91] ; sa lecture intuitive est : si l'agent sait que s'il exécute l'action α, Φ sera vrai (en d'autres termes, s'il sait prédire Φ), alors après avoir réellement exécuté α, il sait que Φ est vrai.

Lorsqu'on s'intéresse uniquement aux actions (purement) épistémiques, on ajoute l'axiome de *staticité* :

staticité $\varphi \rightarrow [\alpha]\varphi$ pour toute formule *objective* φ

On peut montrer que les actions purement informatives ne font jamais perdre de connaissances à l'agent :

non-oubli $\mathbf{K}\Phi \rightarrow [\alpha]\mathbf{K}\Phi$

Au-delà de la staticité et du non-oubli, EDL permet d'exprimer des propriétés (souhaitables ou non) des actions épistémiques – que ne pourraient pas faire PDL seule ni S5 seule. C'est l'objet principal de l'article [125]. Voici les plus significatives :

déterminisme $\langle\alpha\rangle\Phi \rightarrow [\alpha]\Phi$

exécutabilité $[\alpha]\Phi \rightarrow \langle\alpha\rangle\Phi$

idempotence $[\alpha][\alpha]\Phi \leftrightarrow [\alpha]\Phi$

commutativité (ou indépendance) $[\alpha][\beta]\Phi \leftrightarrow [\beta][\alpha]\Phi$

La fiabilité des actions, quant à elle, est implicite avec l'usage de la modalité de connaissance[7].

Voici maintenant quelques classes concrètes d'actions épistémiques .

– une classe particulièrement intéressante d'actions purement épistémique est celle *tests binaires*, ou encore *tests de vérité* : si φ est une formule objective, on définit l'action $t(\varphi)$ (test de la valeur de φ) par les axiomes de staticité, d'exécutabilité et les deux axiomes suivants :

  informativité $[t(\varphi)]\mathbf{K}\varphi \vee \mathbf{K}\neg\varphi$. [8]

  préservation des ignorances (en-dehors de φ et $\neg\varphi$)
  $\neg\mathbf{K}(\varphi \rightarrow \psi) \wedge \neg\mathbf{K}(\neg\varphi \rightarrow \psi) \rightarrow [t(\varphi)]\neg\mathbf{K}\psi$

  On montre que ces axiomes impliquent que les tests de vérité sont déterministes, idempotents et respectent le non-oubli.

– *observations pures* Les actions d'observation pures $o(\varphi)$ (où φ est une formule objective) forment une classe d'actions épistémiques dégénérées, définies par la staticité et les axiomes suivants :

  demi-informativité $[o(\varphi)]\mathbf{K}\varphi$

  préservation des ignorances $\neg\mathbf{K}(\varphi \rightarrow \psi) \rightarrow [o(\varphi)]\neg\mathbf{K}\psi$

[7] Rendre possible l'existence d'actions épistémiques non fiables nécessite non seulement le remplacement de **K** par une modalité doxastique **Bel** mais aussi d'une modalité supplémentaire d'observation **O** (pour laquelle la règle de nécessitation n'est pas requise). La fiabilité d'une action α s'exprime alors ainsi :

fiabilité $[\alpha](\mathbf{O}\varphi \rightarrow \varphi)$

et la confiance en α, ou encore l'adoption des croyances observées (l'agent décide de croire l'observation donnée par α) par

confiance $[\alpha](\mathbf{O}\varphi \rightarrow \mathbf{Bel}\ \varphi)$

[8] Cet axiome se décompose (de manière équivalente) en
(1) $\varphi \rightarrow [t(\varphi)]\mathbf{K}\varphi$
(2) $\neg\varphi \rightarrow [t(\varphi)]\mathbf{K}\neg\varphi$

exécutabilité restreinte $\varphi \leftrightarrow \langle o(\varphi)\rangle\top$

Ces actions peuvent paraître étranges, puisque l'axiome de demi-informativité suggère que *leur résultat est prédictible !* et on pourrait croire qu'elles ne servent à rien ; ce paradoxe s'explique par le fait quelles ne sont exécutables que lorsque φ est vraie (voir [125]).

- les *tests de valeur* (ou *mesures*) généralisent les tests de vérité à un ensemble de résultats mutuellement exclusifs $\{x = v_1, \ldots x = v_n\}$, où $x = v_i$ est le domaine (fini) des valeurs possibles d'une variable x (attention au terme "variable" : x n'est pas une variable propositionnelle tandis que $x = v_1, \ldots, x = v_n$ le sont). L'axiome caractéristique de l'action "mesurer x" est $[tv(x)]\mathbf{K}(x = x_1) \vee \ldots \vee \mathbf{K}(x = x_n)$.
- plus généralement, une action épistémique peut avoir un nombre quelconque de résultats non nécessairement exclusifs ; on note $ae(r_1, \ldots, r_n)$ l'action épistémique dont les résultats possibles sont $r_1, \ldots, r_n$ et qui se traduit en EDL par $[ae(r_1, \ldots, r_n)]\mathbf{K}(r_1) \vee \ldots \vee \mathbf{K}(r_n)$. Ceci dit, $ae(r_1, \ldots, r_n)$ peut toujours se réécrire en utilisant un nombre logarithmique de tests binaires $t(\varphi)$.

Quelques mots sur l'exécutabilité. Certains articles sur la formalisation logique des tests (en particulier [182, 181]) insistent sur les *conditions d'applicabilité* des actions épistémiques. Toujours est-il que ce qui a été dit en Section 1.2.4 est particulièrement vrai ici : la plupart du temps, une condition d'applicabilité C peut être réécrite sous l'une de ces deux formes :

- en exprimant que si C n'est pas satisfaite alors l'action $ae(r_1, \ldots, r_n)$ ne produit pas de résultat (ou, en tout cas, pas de résultat fiable), ce qui peut être réécrit, selon le cas, en $ae(C \rightarrow r_1, \ldots, C \rightarrow r_n)$ ou en $ae(r_1, \ldots, r_n, \neg C)$;
- en exprimant que C est une précondition dont la violation peut rendre l'action intrusive et conduire à un résultat catastrophique (voir l'exemple de [182] sur la radioactivité) ; dans ce cas, l'action (qui n'est pas purement épistémique) peut être écrite en tenant compte de ce résultat catastrophique dans la description de l'action ainsi que dans la structure préférentielle de l'agent.

Dans [125], nous étendons EDL en considérant des tests plus généraux $t(\Phi)$ où Φ est une formule quelconque de EDL (non nécessairement objective) – comme c'est le cas pour les tests de PDL. Ce problème a un intérêt avant tout théorique (qui va au-delà du champ de la planification sous connaissance incomplète) et je ne donne pas plus de détails ici.

Un problème de planification épistémique $\mathcal{P}$ s'exprime simplement dans EDL : il consiste en la donnée d'un ensemble d'actions épistémiques $\mathcal{A}$ et de la description de leurs effets par un ensemble Σ de formules de EDL, d'un état de croyance initial b_{init} consistant en une formule Φ_{init} de S5 de la forme $\mathbf{K}\,\varphi$ et un but représenté par une formule Γ de S5 consistant en une combinaison booléenne d'atomes épistémiques positifs (typiquement, $\mathbf{K}\,\varphi \vee \mathbf{K}\,\neg\varphi$ si seule la valeur de vérité de φ a un intérêt pour l'agent). Un plan π est alors une solution du problème $\mathcal{P}$ si et seulement si

$$\Sigma \models_{EDL} \Phi_{init} \rightarrow [\pi]\Gamma$$

La validité d'un plan revient donc à un test de conséquence logique dans EDL. Ceci ne répond cependant pas (ou du moins très partiellement) aux trois questions suivantes (toutes étant l'objet de travaux en cours) :

- *comment représenter au mieux les effets des actions épistémiques complexes ?* Certes, ce qui précède apporte une réponse dans la cas d'actions simples (tests de vérité ou de valeur, par exemple), mais ne permet pas de représenter des actions plus complexes comme celles du type *MasterMind*, où une action consiste en la combinaison de micro-actions (chaque micro-action correspondant à l'affectation d'une valeur à une variable). Par contre, certains problèmes qui se posent pour la représentation d'actions effectives (notamment les problèmes du décor et de la ramification) ne se posent pas de manière aussi aiguë pour les actions épistémiques.
- *quelle est la complexité du problème d'existence de plan en environnement purement épistémique ?* Répondre à cette question n'est simple que dans le cas où les actions n'ont pas de coût : il suffit de considérer le plan qui consiste à exécuter dans n'importe quel ordre *toutes* les actions épistémiques et à vérifier que le plan obtenu est valide ; le problème est alors (seulement) coNP-complet. La question est bien plus complexe lorsqu'on recherche des plans de taille bornée – elle sera abordée en Section 6.2.
- *comment construire un plan (vérifiant certaines conditions de minimalité) pour un processus purement épistémique ?* Un travail en cours explore la génération de plans en chaînage arrière, en calculant la *régression* d'un but épistémique par une action épistémique (voir [123]).

5.3 Définissabilité

La définissabilité est la faculté qu'un ensemble de variables propositionnelles a de pouvoir déterminer la valeur d'une autre variable étant donné une base de croyances.

En logique propositionnelle, la définissabilité s'exprime formellement comme suit : soient Σ une base de connaissances propositionnelle sur un ensemble Var de variables propositionnelles, $V \subseteq Var$ et $\psi \in \mathcal{L}_{VAR}$. On dit que V définit ψ selon Σ si et seulement si pour toute affectation $\vec{v} \in 2^V$ des variables de V on a $\vec{v} \wedge \Sigma \models \psi$ ou $\vec{v} \wedge \Sigma \models \neg\psi$.

L'étude de cette notion se place naturellement dans la continuation de la section précédente, puisque déterminer si V définit ψ selon Σ est un problème de vérification de plan épistémique très simple, obtenu en faisant les hypothèses simplificatrices suivantes :

- les actions épistémiques sont des tests de vérité non-intrusifs, fiables et toujours exécutables (le fait que, de surcroît, les formules testables sont atomiques n'est absolument pas une perte de généralité), à savoir $ACT_E = \{t(v)|v \in V\}$;
- le seul but de l'agent est de déterminer la valeur de vérité de ψ , ce qui correspond à la fonction d'utilité épistémique $u(\varphi) = +1$ si et seulement si $\varphi \models \psi$ ou $\varphi \models \neg\psi$ et $u(\varphi) = +0$ sinon.

Il est alors évident que V définit ψ selon Σ si et seulement si le plan consistant à exécuter (dans n'importe quel ordre, voire en parallèle) toutes les actions de ACT_E est un plan valide pour le problème défini précédemment.

L'intérêt d'étudier une telle notion de base dans le cadre élémentaire de la logique propositionnelle est, là encore, d'obtenir des caractérisations élémentaires ainsi que des bornes inférieures de complexité. La définissabilité est l'objet de plusieurs travaux réalisés avec Pierre Marquis [155, 156].

Voici deux caractérisations importantes (équivalentes à sa définition) de la définissabilité. La première est une contrepartie syntaxique de la définition (sémantique) : V définit ψ selon Σ si et seulement si il existe une formule $def_V(\psi) \in \mathcal{L}_V$ telle que $\Sigma \models \psi \leftrightarrow def_V(\psi)$. Cette définition n'est pas constructive (et il faut noter que $def_V(\psi)$ n'est en général pas de taille

polynomiale).

La seconde est constructive puisqu'elle donne une méthode abductive de calcul de $def_V(\psi)$: V définit ψ selon Σ si et seulement si $\Sigma \models PI^V_\Sigma(\psi) \vee PI^V_\Sigma(\neg\psi)$, où $PI^V_\Sigma(\psi)$ est l'ensemble des impliquants premiers de ψ étant donné Σ sur le sous-langage généré par V. Si cela est vérifié, $PI^V_\Sigma(\psi)$ est alors équivalent à $def_V(\psi)$.

Nos travaux ont premis d'identifier la complexité de la définissabilité (coNP-complétude dans le cas général, identification de classes polynomiales), des liens avec les notions d'oubli de variable et d'indépendance et des notions dérivées. Illustrons l'intérêt de la définissabilité en raisonnement et décision sur quelques exemples :

- la plus immédiate est la recherche de plans de tests pour la *discriminabilité d'hypothèses* : on y revient en détail en Section 5.4 ;
- en raisonnement sur l'action, le *déterminisme* d'une théorie d'actions se ramène à un problème de définissabilité (des variables à l'instant $t+1$ en fonction des variables à l'instant t – les définitions des variables en $t+1$ étant alors appelées "axiomes des états successeurs") ;
- toujours en raisonnement sur l'action, la vérification que certains fluents (dits secondaires) sont définissables (ou *dérivés*) à partir d'autres fluents (primaires) est essentiel pour traiter la ramification ;
- la vérification que la description fonctionnelle d'un système est complète est également un problème de définissabilité (l'output doit être définissable à partir de l'input selon cette description).

On a aussi étudié (plus brièvement) un affaiblissement de la définissabilité : V discrimine potentiellement ψ selon Σ si et seulement si dans certains cas (mais pas nécessairement tous), la connaissance des valeurs de vérité des variables de V permet de connaître la valeur de vérité de ψ . Cet affaiblissement (définissabilité potentielle) a une complexité plus forte que la définissabilité (il est Σ^p_2-complet [146, 157, 159]). On montre enfin dans [157, 159] comment la définissabilité permet de construire des tests complexes à partir de tests élémentaires.

5.4 Discrimination d'hypothèses ; application au diagnostic et à la résolution d'incohérences

5.4.1 Problème général

On s'intéresse maintenant à la faculté qu'un ensemble de tests de vérité $\{t(v_1), \ldots, t(v_n)\}$ a de discriminer un ensemble d'hypothèses $\{H_1, \ldots, H_p\} \subseteq \mathcal{L}_{VAR}$ selon Σ , où les H_i sont des formules propositionnelles (pas nécessairement mutuellement exclusives, et ne couvrant pas nécessairement Σ) : on dit que $\{t(v_1), \ldots, t(v_n)\}$ discrimine $\{H_1, \ldots, H_p\}$ si et seulement si pour tout $\vec{v} \in 2^{\{v_1,\ldots,v_n\}}$ il existe un i tel que $\vec{v} \wedge \Sigma \models H_i$ ([156]). (La fonction d'utilité épistémique s'écrit ici : $u_E(\varphi) = +1$ si il existe un i tel que $\varphi \models H_i$, $u_E(\varphi) = 0$ sinon). La discriminabilité d'hypothèses, qui généralise la définissabilité, est également un problème coNP-complet et les caractérisations (notamment abductives) de la définissabilité s'étendent à la discriminabilité d'hypothèses. Cette notion apparaît dans de nombreux problèmes de raisonnement et décision :

– *décision en environnement partiellement observable* : supposons qu'une politique d'action π , de la forme suivante :

```
Case
    Cond_1   :  α_1
    Cond_2   :  α_2
    ...
    Cond_p   :  α_p
end case
```

(donc ne faisant pas intervenir d'actions épistémiques), ait été calculée pour un problème donné de décision en environnement partiellement observable (voir Section 6.3) et que l'agent ait à sa disposition un ensemble de tests $\{t(v_1), \ldots, t(v_n)\}$. Afin de pouvoir exécuter π, il faut d'abord discriminer l'ensemble d'hypothèses $\{Cond_1, \ldots, Cond_n\}$ pour $\{t(v_1), \ldots, t(v_n)\}$ selon la base de connaissances Σ .

– *classification* : soit H un ensemble de classes et soit Σ une base de croyances liant des propriétés d'un objet à sa classe ; peut-on identifier la classe d'un objet en observant un ensemble P de propriétés de l'objet ?

– *identification de pannes* : voir la Section 5.4.2 ;

– *résolution d'incohérences* : voir la Section 5.4.3.

5.4.2 Discrimination de pannes en diagnostic à base de modèles

Un problème d'identification de pannes consiste en la donnée d'un problème de diagnostic à base de modèles $\langle COMPS, \Sigma, OBS\rangle$ et d'un ensemble de tests[9] $ACT_E = \{\alpha_1, \ldots, \alpha_n\}$. $COMPS$ est l'ensemble fini de composants du système à diagnostiquer, Σ est un *modèle* du système (d'où la terminologie "à base de modèles") consistant en un ensemble de formules propositionnelles décrivant le fonctionnement normal (et éventuellement certains modes de dysfonctionnement) de chacun des composants et OBS est un ensemble d'observations (formules propositionnelles).

La première étape, purement délibérative, consiste à déterminer les diagnostics préférés[10] ; elle est l'objet d'une littérature abondante en diagnostic à base de modèles, où on trouve plusieurs notions de ce que peut être un diagnostic préféré, selon

– certaines hypothèses de minimalité de l'ensemble des pannes, ou encore *parcimonie*, qui repose sur l'hypothèse par défaut de bon fonctionnement des composants ;
– la "force" des diagnostics voulue, liée à la façon d'utiliser l'inférence logique (par ordre de force : diagnostics à base de cohérence, abductifs, déductifs)[11]

[9] Ou plus généralement d'actions épistémiques quelconques, notamment des actions de mesure – pour simplifier la présentation nous supposerons cependant que ce sont des tests de vérité.

[10] Cette terminologie "diagnostic préféré" est quelque peu malheureuse, mais consacrée. Il s'agit ici, comme souvent dans la littérature de l'intelligence artificielle, des diagnostics les plus plausibles même si, par pure coïncidence, l'hypothèse de parcimonie implique que les diagnostics les plus plausibles minimisent l'ensemble de pannes, et sont, justement (mais par hasard) ceux qui seraient sans doute préférés par l'agent, au sens de la théorie de la décision ...

[11] Même si ce n'est pas le sujet principal de ce chapitre, je ne résiste pas à l'envie d'en donner les définitions. Dans tous les cas, une configuration de pannes (cdp) δ est une conjonction de littéraux $ab(i)$ ou $\neg ab(i)$ pour $i \in COMPS$.

– *diagnostic à base de cohérence* : une cdp δ est un diagnostic si et seulement si $\delta \wedge \Sigma \wedge OBS$ est cohérent – les diagnostics préférés sont ceux qui minimisent l'ensemble de pannes $\{i, ab(i) \in \delta\}$ (ou sa cardinalité, ou un critère plus complexe) ;
– *diagnostic abductif* : une cdp δ est un diagnostic si et seulement si $\delta \wedge \Sigma$ est consistant

Soit $DIAG(COMPS, \Sigma, OBS)$ l'ensemble des diagnostics préférés ainsi déterminés (supposé non vide).

La deuxième étape (active) de l'identification de pannes consiste à discriminer $DIAG(COMPS, \Sigma, OBS)$ par ACT_E selon $\Sigma \wedge OBS$. La validité d'une telle démarche repose sur une hypothèse forte (sur laquelle nous allons revenir en Section 5.3.3) : le *vrai* diagnostic (celui qui correspond au monde réel) fait partie des diagnostics préférés[12].

L'article [147] ébauche une méthode de recherche de plans d'identification de pannes à l'aide de tests et étudie quelques problèmes dérivés. On pourrait aller plus loin en introduisant des actions de réparation (le problème n'étant alors plus purement épistémique) et des probabilités de panne.

5.4.3 Discrimination d'extensions et résolution d'incohérences

On a vu au chapitre 2 des stratégies de raisonnement en présence d'incohérence. Le point commun de ces stratégies était le caractère purement délibératif de l'agent – qui se contentait donc de *faire avec* l'incohérence. On va parler maintenant d'une stratégie complètement différente : l'agent, cette fois épistémiquement actif, va en quelque sorte *agir sur* l'incohérence en en identifiant ses causes autant que possible.

Avant de formaliser le propos, fixons un cadre de représentation pour faciliter la discussion. On considère une théorie de défauts normaux sans prérequis (appelée aussi "théorie de défauts supernormaux" [34] ou "théorie d'hypothèses" [193] en divers endroits de la littérature) : $KB = \langle F, \Delta \rangle$ où $\Delta = \{\delta_1, \ldots, \delta_n\}$[13]. L'ensemble F contient toutes les formules représentant

et $\delta \wedge \Sigma \models OBS$ – là encore, les diagnostics préférés sont là encore ceux qui minimisent les pannes ;

– *diagnostic déductif* : une cdp δ est un diagnostic si et seulement si $\Sigma \wedge OBS \models \delta$ – cette fois les diagnostics préférés sont les plus spécifiques (δ est préféré s'il n'existe pas de δ' tel que $\delta \models \delta'$ tel que δ' soit un diagnostic).

[12] La vraisemblance d'une telle hypothèse dépend naturellement de la notion de diagnostic préféré choisie : elle est ainsi peu vraisemblable pour le diagnostic abductif et plus encore pour le diagnostic déductif (puisqu'il arrive alors que l'ensemble des diagnostics préférés soit vide !) ; elle semble raisonnable pour le diagnostic à base de cohérence avec des hypothèses de minimalité pertinentes, et elle est évidemment valide lorsque *tous* les diagnostics possibles (minimaux ou non) sont préférés.

[13] Les articles sur la logique des défauts notent d'habitude W au lieu de F, mais je trouve

des connaissances *certaines* et Δ contient des croyances pouvant être remises en cause. Une *extension* E est la clôture logique $Cn(D \wedge F)$ où $D \subseteq \Delta$, appelé ensemble de défauts générateur, est un sous-ensemble de Δ maximalement (pour une relation de préférence donnée – à l'origine, l'inclusion) cohérent avec F.

Avant d'aller plus loin, on peut s'interroger sur la véritable signification (la plupart du temps implicite) de la notion d'extension. Cette question est d'autant plus pertinente qu'elle est presque toujours escamotée dans les articles parlant de logique des défauts, où la technique l'emporte vite sur la pertinence des définitions postulées.

La discussion qui suit est reprise de l'introduction de l'article [157][14]

Dans un cadre logique de représentation des connaissances où les informations élémentaires sont représentées par des formules propositionnelles sur le langage $\mathcal{L}_{VAR}$, le monde réel correspond à une interprétation particulière de Ω_{VAR}, notée I.

Quelles sont les *hypothèses de représentation* qui sont usuellement faites dans un tel cadre ? Autrement dit, quels sont les liens logiques supposés entre les connaissances dérivables de KB et celles qui sont vraies dans I ? La réponse dépend typiquement de la nature des connaissances représentées dans KB. Si on suppose que KB ne contient que des connaissances certaines (i.e., vraies dans I) – ce qui revient à supposer que $\Delta = \emptyset$ – l'hypothèse de représentation s'exprime formellement par : I est un modèle de KB. Sous cette hypothèse, la fermeture déductive $Cn(KB)$ de KB, c'est-à-dire l'ensemble des formules déductibles de KB est un sous-ensemble de l'ensemble $Cn(I)$ des formules vraies dans I : $Cn(KB) \subseteq Cn(I)$.

Que se passe-t-il maintenant si toutes les connaissances disponibles ne sont pas certaines ? Si on ne fait aucune hypothèse de représentation sur la nature des croyances représentées dans Δ et que l'on ne souhaite dériver de KB que des formules pour lesquelles on est sûr qu'elles sont vraies dans I, la seule relation de dérivation raisonnable que l'on peut construire à partir de KB est la déduction à partir de W – et dans ce cas de figure, les for-

cette notation est ambiguë : ce n'est pas un ensemble de mondes.

[14]Et comme de nombreux autres choses dans ce document, elle doit beaucoup à Pierre Marquis.

mules de Δ ne peuvent jouer aucun rôle. Si on suppose maintenant que les formules de Δ représentent des *défauts*, c'est-à-dire des formules que l'on est prêt à considérer comme vraies dans I *en l'absence d'information contraire*, on va pouvoir définir une relation de dérivabilité plus riche que la déduction à partir de W (formellement, une relation qui inclut cette dernière) et donc mieux approcher l'ensemble $Cn(I)$ des formules vraies dans I en exploitant les informations représentées dans Δ.

Maintenant, comment modéliser formellement l'hypothèse de représentation "les formules de Δ sont vraies dans I par défaut" ? Intuitivement, les extensions représentent les états de croyance cohérents les plus grand possibles (c'est-à-dire pour lesquels l'ensemble de formules déduites est le plus grand possible, au sens de l'inclusion) que l'on peut construire à partir de KB. Pour capturer formellement le fait que les formules de Δ représentent des défauts, on fait l'hypothèse de représentation suivante :

(D) Il existe une extension E_I dans $Ext(KB)$ telle que $I \models E_I$.

Puisque deux extensions distinctes sont conjointement incohérentes, E_I est forcément unique.

On peut donner une interpétation probabiliste de (D). Comme en Section 2.1.1, si on suppose que les sources qui ont fourni les défauts $\delta_1, \ldots, \delta_n$ sont indépendantes et ont une probabilité de fiabilité $1 - \varepsilon_i$, alors la probabilité de la disjonction des extensions tend vers 1 lorsque $(\varepsilon_1, \ldots, \varepsilon_n)$ tend vers $(0, \ldots, 0)$ [15].

Quel est maintenant l'ensemble des formules vraies dans I qui sont dérivables de KB sous une telle hypothèse de représentation ? On aimerait pouvoir répondre E_I puisque on sait que $Cn(W) \subseteq E_I \subseteq Cn(I)$. Malheureusement, en règle générale, KB possède plusieurs extensions et l'hypothèse qui est posée ne dit pas laquelle est la bonne. Aussi, si on ne veut pas prendre de risque (i.e., ne dériver que des formules vraies dans I depuis KB), on doit recourir à une relation de dérivabilité prudente : les seules formules dérivables de KB dont on puisse garantir la vérité dans I sont celles de $\{\psi | \forall E \in Ext(KB), \psi \in E\}$. Quoique $\{\psi | \forall E \in Ext(KB), \psi \in E\}$ soit seulement inclus strictement dans E_I dans le cas général, l'hypothèse de représentation a permis de progresser vers l'identification des formules

[15] Si, de plus, on suppose $\varepsilon_i = \varepsilon_j$ pour tous i, j, il suffit de s'en tenir aux extensions générées par les ensembles générateurs de cardinalité maximale.

vraies dans I par la prise en compte des défauts puisque $Cn(W)$ est seulement inclus strictement dans $\{\psi | \forall E \in Ext(KB), \psi \in E\}$ dans le cas général.

Et si on veut aller plus loin, i.e., capturer plus de formules de E_I sans prendre de risque ? Plusieurs voies sont possibles. L'une d'entre elles consiste à renforcer l'hypothèse de représentation en supposant par exemple que certains défauts sont préférés à d'autres et leur sont donc prioritaires (voir la Section 2.1.1). Dans tous les cas de figure, l'idée est de sélectionner un sous-ensemble de $Ext(KB)$ correspondant aux *extensions préférées* de KB. L'hypothèse de représentation réalisée consiste à supposer que E_I est parmi ces extensions. Sous cette hypothèse, en ayant recours à la relation de dérivabilité prudente, on peut effectivement atteindre l'objectif (plus on supprime d'extensions plus l'ensemble des formules appartenant à toutes les extensions grossit).

Dans l'article [157], on se place dans une approche radicalement différente mais complémentaire. Nous supposons que l'agent a à sa disposition des actions épistémiques de vérification des informations ou des sources qui les ont fournies ; ces deux types d'action doivent être distinguées :

- *vérification d'une information*, c'est-à-dire d'un défaut δ_i : l'information est vérifiée directement, par des moyens indépendants de la source.
- *vérification d'une source* i (qui a fourni le défaut δ_i) : c'est la fiabilité de la source qui est vérifiée, pas directement la véracité de l'information ; par conséquent, si la source est validée, on considèrera que les défauts qu'elle a fournis sont vrais dans I ; si par contre elle n'est pas validée, on n'a aucune garantie que δ_i soit vraie dans I (elle peut néanmoins être vraie quand même – mais c'est alors une coïncidence).

Pour un défaut i, il y a donc trois types de résultats possibles d'une action de vérification : $ok(i)$ (validation de l'information *ou* de la source), $bad(i)$ (réfutation de l'information) et $forget(i)$ (invalidation de la source, plus faible que la réfutation de l'information). La révision de KB par le résultat r d'une action de vérification dépend évidemment du type de résultat :

- $KB * ok(i) = \langle F \cup \{\delta_i\}, \Delta \setminus \{\delta_i\}\rangle$;
- $KB * bad(i) = \langle F \cup \{\neg\delta_i\}, \Delta \setminus \{\delta_i\}\rangle$;
- $KB * forget(i) = \langle F, \Delta \setminus \{\delta_i\}\rangle$.

Cette notion de révision s'étend inductivement à un ensemble R de résultats d'actions de vérification. On étudie dans [157] les effets de ces trois types de révision sur les conflits et les extensions de KB : la validation (qui

se traduit par une expansion par δ_i) réduit l'ensemble des extensions ; l'invalidation de la source (qui se traduit par un rejet de δ_i – plus faible qu'une contraction – affaiblit certaines extensions mais n'en crèe pas de "réellement nouvelles" ; enfin, la réfutation de l'information (qui se traduit par une expansion par ϕ_i) crèe en général de nouvelles extensions.

Intéressons-nous maintenant au problème consistant si possible à identifier, ou sinon à approcher au mieux *la* "bonne extension" (celle qui correspond aux sources pertinentes, une fois mises de côté les informations venant des sources erronées) : il s'agit de chercher des plans qui discriminent entre toutes les extensions, afin de trouver celle (E_I) qui correspond à la réalité, ou de l'approcher au mieux. Ceci passe par la notion de *purification* d'une base de connaissances par un *plan de vérification*. On donne dans [157] deux définitions différentes de la purification d'une théorie de défauts :

- le plan de vérification π *N-purifie* KB si et seulement si dans toute exécution possible de πassociée à un ensemble de résultats R, $Ext(KB*R)$ est un singleton ($KB * R$ a une seule extension) ;
- le plan de vérification π *C-purifie* KB si et seulement si dans toute exécution possible de πassociée à un ensemble de résultats R, $Ext(KB * R) \cap Ext(KB)$ est un singleton.

Bien entendu, une action de vérification a un coût, et, par ailleurs, toutes les sources ne sont pas forcément vérifiables. En effectuant un minimum de vérifications, notre but est de cerner au mieux E_I voire de discriminer E_I parmi toutes les extensions possibles. Alors que, pour un agent intelligent, raisonner sert typiquement à agir au mieux selon les buts à atteindre, dans notre cadre, l'agent intelligent agit, c'est-à-dire effectue des actions de prises d'information en vérifiant les sources, de façon à mieux raisonner.

Dans l'article [157] on étudie en outre la complexité du problème de "purifiabilité" d'une théorie de défauts, et on montre comment calculer un plan (parallèle) de tests de coût minimal dans les hypothèses où le coût d'un ensemble d'actions est la somme (respectivement le maximum) des coûts des actions élémentaires ; dans le cas de la somme, la recherche d'un plan de moindre coût à partir de la donnée des conflits minimaux de KB revient à résoudre un problème de programmation linéaire.

Chapitre 6

Décision et planification en environnement partiellement observable

Dans ce chapitre on lève la restriction du chapitre précédent (actions épistémiques seulement) et on fait les hypothèses suivantes :

- les agents sont volitifs et actifs ; ils ont à leur disposition des actions effectives et éventuellement des actions épistémiques ;
- $\mathcal{T} = \mathbb{N}$ ou $\mathcal{T} = \{0, 1\}$;
- à l'exception de la dernière section, les préférences sont binaires et représentées par un but G (cette restriction n'induit cependant pas de perte de généralité, et la plupart des résultats de ce chapitre restent valides avec des préférences ordinales ou quantitatives) ;
- l'incertitude sur l'état initial du monde et les effets des actions est, selon les travaux considérés, binaire, probabiliste ou possibiliste ;
- la représentation des effets des actions, des connaissances sur l'état courant, des observations et des buts est explicite en section 6.5.3, compacte partout ailleurs (logique classique propositionnelle en section 6.3, modale propositionnelle en section 6.1, contraintes en section 6.4, opérateurs STRIPS non-déterministes ou possibilistes en section 6.5.2).

Le paramètre-clé de ce chapitre est, on va le voir, l'observabilité : le choix d'une hypothèse ou d'une autre concernant l'observabilité a un impact très important sur le problème de planification, sa complexité, ses méthodes de

résolution.

Le plan du chapitre est subordonné à la division thématique des travaux auxquels j'ai contribué. La première section évoque la représentation d'actions et de plans dans la logique épistémique dynamique dont nous avons parlé en 5.1, ainsi qu'une ébauche de formalisation logique des hypothèses courantes d'observabilité. En Section 2 on étudie la complexité des problèmes de vérification et d'existence de plan selon (i) les hypothèses sur l'observabilité de l'environnement, (ii) la nature de l'incertitude sur les effets des actions, (iii) l'horizon et (iv) le type de représentation choisi. En Section 3 on se focalise sur un problème plus spécifique : comme dans les chapitres précédents, on choisit d'abord le langage de représentation le plus simple, à savoir la logique propositionnelle ; on exprime et, pour une fois, on *résoud* des problèmes de décision en insistant particulièrement sur le cas simple (et pourtant déjà bien complexe) où il n'y a qu'une seule étape de décision. On en vient à donner une importance particulière à la résolution de *formules booléennes quantifiées* (problème QBF), puisqu'il joue le rôle, pour la planification en environnement non-déterministe, que SAT joue pour la planification classique ; et on conclut naturellement la section par des questions et des éléments de réponse concernant le résolution de certains problèmes QBF. En Section 4 on s'intéresse à un autre langage de représentation compacte : celui des problèmes de satisfaction de contraintes. Bien que proche de la logique propositionnelle, il diffère quand même suffisamment de celle-ci (à cause de ses hypothèses sur la syntaxe des contraintes) pour que les méthodes de résolution envisageables soient assez différentes. On considère des problèmes à horizon 1 ; la nature de l'incertitude est binaire ou probabiliste. En Section 5 on s'intéresse à la décision dynamique en environnement incertain, lorsque l'incertitude sur l'état initial et les effets des actions est possibiliste. On discute de la signification de ce choix et on évoque deux lignes de travail : l'une concerne des problèmes de planification en environnement non-observable exprimés dans un langage compact (opérateurs STRIPS probabilistes) ; pour l'autre, il s'agit d'une alternative possibiliste aux processus décisionnels markoviens.

6.1 Description en logique modale d'actions et de plans

La logique EDL permet de représenter aussi bien des actions effectives que des actions épistémiques (il suffit de se placer dans le cadre plus géné-

ral où l'on ne requiert pas l'axiome de staticité), et par là, de problèmes de planification en environnement partiellement observable.

La faculté de EDL à représenter les interactions entre action et connaissance permet de traduire logiquement certaines hypothèses courantes d'observabilité , à un détail important près : les caractéristiques d'observabilité sont d'habitude une propriété de l'environnement global (pas des actions individuelles) ; EDL, par contre, va les associer aux actions, ce qui est un gain de généralité, comme on va le voir plus loin.

Un environnement de planification est constitué d'un ensemble d'états de croyance initiaux possibles $\mathcal{B}_{init}$, d'un ensemble d'actions $\mathcal{A}$ et de croyances sur l'évolution du monde (naturelle ou en réponse à des actions). Ainsi, un environnement est non-observable si et seulement si chacune des actions qu'il contient est *non-informative*, ou encore *sans feedback*, ce que l'on exprime par

$$\langle\alpha\rangle\mathbf{K}\ \varphi \rightarrow \mathbf{K}\ [\alpha]\varphi \text{ pour tout } \varphi \text{ objectif} \qquad (\mathbf{NI}(\alpha))$$

ce qui signifie, informellement : s'il est possible que l'agent sache φ après avoir exécuté α, alors, nécessairement, il sait (prévoir) que φ sera vrai après α (ou, en d'autres termes, on sait avant de l'exécuter que l'action α n'apportera pas de feedback). Un environnement est dit *non-observable* si et seulement si chacune de ses actions α satisfait $\mathrm{NI}(\alpha)$.

L'observabilité totale est plus délicate à exprimer. Une action α est "totalement informante" si et seulement si elle vérifie

$$[\alpha]\mathbf{K}\ \varphi \vee \mathbf{K}\ \neg\varphi \text{ pour tout } \varphi \text{ objectif} \qquad (\mathbf{TI}\ (\alpha))$$

mais $\mathrm{TI}(\alpha)$ est trop fort pour être réaliste (sauf dans des domaines restreints) : il suffit de l'existence d'un fluent qui ne soit pas automatiquement observable et qui n'ait rien à voir avec α pour qu'il ne puisse pas être vérifié. C'est pourquoi on s'intéresse plutôt aux actions "informantes des changements", qui vérifient

$$\varphi \rightarrow [\alpha](\neg\varphi \rightarrow \mathbf{K}\ \neg\varphi) \text{ pour tout } \varphi \text{ objectif} \qquad (\mathbf{CI}\ (\alpha))$$

Un environnement est totalement observable si (i) l'état initial (éventuellement inconnu hors-ligne et observé en ligne) est complètement connu et (ii) chaque action est informante des changements.

On peut bien entendu exprimer toutes sortes d'hypothèses d'observabilité intermédiaires (comme l'observabilité totale sur une partie du monde et la non-observabilité sur une autre, etc.).

Il faut noter que EDL, si elle offre la possibilité d'exprimer des formes variées d'interaction entre action et connaissance, ne propose pas de solution *a priori* au problème du décor (ni à la ramification), mais les différentes approches pour celui-ci sont en quelque sorte "superposables" à EDL (voir [121]).

EDL est avant tout un langage d'expression de *plans* en environnement partiellement observable (tout comme Golog, basé sur le calcul des situations). Rappelons (voir Section 5.1) qu'un plan π est soit le plan vide λ , soit une action atomique $\alpha \in ACT_0$, soit une séquence $\pi;\pi'$ de plans, soit un conditionnel `if` Φ `then` π `else` π', où Φ est épistémiquement interprétable et π , π ' sont des plans. C'est là l'intérêt le plus manifeste de EDL que de pouvoir exprimer des conditions de branchement *épistémiques* qui seront évaluées *en ligne*, c'est-à-dire lors de l'exécution du plan. L'évaluation de ces conditions étant un problème coNP-complet (validité dans S5), l'exécution risque fort de requérir un certain temps passé seulement à des actions délibératives, mais d'un autre côté, on peut ainsi représenter en très peu d'espace des plans qui, s'ils devaient se passer de conditions épistémiques, seraient exponentiellement plus grands (voir [121], Section 1). Ce compromis entre temps "en ligne" et espace "hors-ligne" permet de répartir à la guise du programmeur la charge computationnelle entre la phase hors-ligne (phase de "compilation", ou "pré-traitement") et la phase d'exécution du plan, les deux choix extrêmes étant

- *compilation complète* (attitude prévoyante) : les conditions de branchement sont interprétables en temps polynomial, donc l'exécution est presque instanée (pourvu que les effets des actions soient immédiats), mais les plans sont en général de taille prohibitive au sortir de la phase de pré-traitement ;
- *aucun pré-traitement* (attitude paresseuse) : il n'y a pas de calcul de politique (on attend d'en savoir plus avant de se lancer dans les calculs). Il n'y a alors pas de problème d'espace, mais l'exécution devra être entrelacée de *lourdes* phases de calcul, ce qui est raisonnable seulement si le temps en ligne ne coûte pas trop cher (c'est-à-dire dans un contexte de prise de décision en situation non urgente).

Si cette notion de plan "qui raisonne / délibère pendant son exécution" a reçu une certaine attention dans le domaine des "agents délibératifs" sous le terme de *reconsidération d'intention* (voir par exemple [209]), elle a cependant été relativement peu étudiée d'un point de vue formel, à l'exception de [91], et dans une moindre mesure dans [194]. Le développement d'outils formels pour la "programmation épistémique" ouvre probablement de nombreuses voies de recherche, notamment en développant des logiques qui seront aux programmes d'agents ce que PDL est aux programmes d'ordinateurs, et des algorithmes pour la vérification et surtout la *génération* de programmes d'agents en utilisant des techniques de "model checking", dans la veine des travaux de Cimatti et al. [17]. Enfin, il serait pertinent de proposer une extension *probabiliste* (ou utilisant d'autres théories de l'incertain) de la programmation épistémique, où le choix d'une action sera fait en fonction d'un état de croyance probabiliste[1]. On pourra ainsi représenter des plans comme celui-ci :

> `répéter` effectuer des sondages de terrain
> `jusqu'à` Prob(pétrole) < 0.01 ou Prob(pétrole) > 0.75 ;
> `si` Prob(pétrole) > 0.75 `alors` forer `sinon` ne rien faire.

6.2 Complexité de la planification en environnement non-déterministe

Cette Section contient à la fois (a) une synthèse bibliographique sur la complexité de l'existence de plan en environnement non-déterministe et (b) de nouveaux résultats – certains publiés, d'autres non. Le but de cette étude est d'y voir un peu plus clair dans les raisons de la difficulté de la planification selon les hypothèses qui sont faites sur la nature du problème et sa représentation.

La nature de l'observabilité conditionne la définition d'un plan solution pour un problème de planification, et au-delà, a une influence prépondérante sur la complexité de la vérification et de l'existence de plan. On va dans cette section passer rapidement en revue les principaux résultats de complexité de la vérification et de l'existence de plan en fonction de (i) la nature des actions

[1] Il existe des versions probabilistes des logiques épistémiques – voir par exemple [119] – auxquelles on peut songer à adjoindre PDL.

↓ horizon représentation ⇒	explicite	compacte [2]
$T = 1$	P	P
polynomialement borné	P	NP-complet
non borné	P	PSPACE-complet

TAB. 6.1 – Complexité de l'existence de plans en environnement déterministe

et les hypothèses d'observabilité, (ii) l'horizon et (iii) le type de représentation (explicite ou compacte) ; par contre on se restreint, pour simplifier, à des préférences et des incertitudes binaires (on donnera des pointeurs bibliographiques sur le reste de temps en temps) ; enfin, on supposera les actions toujours exécutables (en donnant là aussi des pointeurs bibliographiques sur ce qui se passe lorsqu'on ne fait plus cette hypothèse).

On va voir que les éléments-clé sont

1. la *profondeur* du plan solution le plus court pour un problème ;
2. la *taille* du plan le plus petit (la taille d'un plan dépend non seulement de la profondeur mais aussi de sa largeur – le plan le plus petit n'est pas forcément le plus court) ;
3. le nombre de plans concevables ;
4. la complexité de la vérification qu'une transition (s, α, s) est possible – cependant, pour simplifier, *nous supposerons que ce dernier problème est toujours polynomial* ; de plus, lorsque le problème est déterministe, nous supposons que l'état successeur $next(a, s)$ peut être calculé en temps polynomial.

6.2.1 Planification en environnement déterministe

On commence par rappeler la complexité de l'existence de plan pour les problèmes de planification "classique" (c'est important pour bien situer le saut de difficulté quand on passera à la planification non-déterministe).

Rappelons qu'en planification classique, l'état initial est connu et les actions sont déterministes. On sait que dans ce contexte, il suffit de considérer des plans linéaires (suites inconditionnelles d'actions) : il n'y a pas besoin de branchements). Pour les représentations compactes on connsidère des opérateurs STRIPS – ce qui laisse un niveau suffisant de généralité [6].

[2] opérateurs STRIPS sans perte de généralité [6].

La vérification de plan, quant à elle, est polynomiale partout.

Commentaires : le seul résultat délicat à obtenir est la PSPACE-complétude de la vérification de plan dans le cas général, dont la preuve est due à Bylander [36]. Intuitivement, ce qui "empêche" le problème dêtre plus bas que PSPACE est le fait que la longueur du plan le plus court est exponentielle (elle est bornée par $2^n - 1$, et cette borne est atteinte – n est ici le nombre de variables propositionnelles).

Des chutes de complexité peuvent être obtenues en faisant des hypothèses syntaxiques sur les effets des actions [36, 5].

6.2.2 Planification en environnement non-déterministe non-observable

La planification en environnement non-observable est parfois qualifiée de *conformante* (je ne sais pas pourquoi). Puisque les actions ne sont plus nécessairement déterministes, il existe en général plusieurs trajectoires possibles pour un état de croyance initial et un plan donné, et la définition d'un plan solution doit être revue. La plus couramment utilisée est la plus forte, où un plan solution est un plan dont *toutes* les exécutions possibles dans tout état initial possible mènent à un état but. (Des définitions moins fortes peuvent naturellement être posées, et les résultats de complexité seront différents).

Il est évident que l'hypothèse de non-observabilité préserve l'absence du besoin de branchement dans les plans solutions (s'il existe un plan solution pour un problème donné, il en existe forcément un qui soit inconditionnel). La longueur du plan minimal pour un problème n'est par contre plus bornée par $|\mathcal{S}|-1$ comme dans le cas déterministe , mais seulement par le nombre d'*états de croyance* possibles (moins un), c'est-à-dire par $2^{|S|} - 2$ (donc par une fonction *doublement exponentielle* de la taille de l'input si la représentation est compacte) ; cette borne est atteinte quel que soit le nombre d'états de $\mathcal{S}$ (preuve dans [154]). Ce qui ne manque pas d'avoir des conséquences sur la complexité de l'existence de plan.

Les résultats dans le cas de la représentation compacte sont dans une bonne mesure indépendants du langage de représentation choisi[3], pourvu que la complexité de la vérification qu'une transition $\langle s, \alpha, s' \rangle$ est possible

[3]Voir [174]. Certes, les résultats valent dans le cas totalement observable, mais ils peuvent être facilement généralisés au cas non-observable.

↓ horizon représentation ⇒	explicite	compacte
$T = 1$	P	coNP-complet
polynomialement borné	NP-complet	Σ_2^p-complet
non borné	PSPACE-complet	EXPSPACE-complet

TAB. 6.2 – Complexité de l'existence de plans en environnement non observable

soit polynomial (ce que nous avons supposé au début de cette Section)[4].

Le résultat de EXPSPACE-complétude est dû à [117]. Le résultat de Σ_2^p-complétude est dans [54] (pour des actions représentées en STRIPS non-déterministe), [7] (actions représentées dans une extension non-déterministe du langage $\mathcal{A}$). Ces résultats supposent en outre que l'exécutabilité est totale. Sans cette hypothèse, la complexité monte (voir [215]). Dans le cas probabiliste, l'existence de plan dont la probabilité de réussite soit supérieur à un seuil donné n'est pas décidable dans le cas général ([177] ; dans les cas où on borne la longueur du plan, des résultats de complexité sont dans [175].

Une conséquence de ces résultats de complexité est que la planification en environnement non-observable ne peut pas être résolue comme une instance de SAT comme l'est la planification classique à horizon borné (ou alors, l'instance de SAT générée est exponentiellement longue), sauf dans des cas particuliers, par exemple lorsque le problème appartient à une "classe fine" [122] : seulement dans ces cas (rares), l'existence de plan de profondeur polynomialement bornée est NP-complète. Dans tous les autres cas, l'existence de plan de profondeur polynomialement bornée s'exprime et se résoud comme un problème d'*abduction* [122].

6.2.3 Planification en environnement non-déterministe totalement observable

Un plan est désormais une *politique conditionnelle*, contenant des branchements conditionnels (donc une structure de graphe, parfois d'arbre). Puisque l'observabilité est totale, les états de croyance sont des singletons, et les branchements se font donc sur l'*état courant*.

[4]Ce n'est pas le cas, par exemple, pour les opérateur s de mise-à-jour à base de minimalité comme la PMA – avec ces langages-là, les résultats de complexité en-dessous de PSPACE monteraient d'un niveau dans la hiérarchie polynomiale.

↓ plan représentation ⇒	explicite	compacte
$T = 1$	P	de P à coNP-complet (1)
prof. polynomialement bornée	P	PSPACE-complet (2)
taille polynomialement bornée	P	Σ_2^p-complet (3)
non borné	P	EXPTIME-complet (4)

TAB. 6.3 – Complexité de l'existence de plans en environnement totalement observable

On montre facilement que s'il existe un plan solution pour un problème donné, il en existe un de profondeur bornée par $|\mathcal{S}|-1$, où la profondeur d'un plan est définie comme celle de sa branche la plus longue. Le problème, désormais, est la *largeur* des plans, qui est généralement en $o(|\mathcal{S}|)$; la *taille* d'un plan (nombre de noeuds du graphe) est en $\mathcal{O}(|\mathcal{S}|)$ à condition que le plan soit codé comme un graphe (si on exige qu'il soit codé comme un arbre, elle est alors en $\mathcal{O}(2^{|\mathcal{S}|})$). Lorsque la représentation est compacte, le fait qu'un plan soit de profondeur polynomialement bornée n'implique pas qu'il le soit de largeur (et donc de taille) polynomialement bornée ; en plus des plans à profondeur polynomialement bornée (plans "courts") , on considère donc aussi des plans à *largeur polynomialement bornée* (plans "fins") et à *taille polynomialement bornée* (plans "petits").

Dans le cas mono-étape (1), la complexité du problème d'existence dépend pour une fois du langage d'action utilisé, ainsi que de la réponse à la question suivante : l'état initial est-il observé hors-ligne ou en-ligne ? S'il est observé hors ligne et que $Next(s, a)$ est de taille polynomiale et calculable en temps polynomial, le problème est dans P ; avec la présence d'effets concurrents (comme dans [124]), cette dernière condition n'est pas satisfaite et le problème devient coNP-complet. Si l'état initial est observé en-ligne, la politique solution doit tenir compte de tous les états initiaux possibles et le problème est également coNP-complet(cette fois, indépendamment du fait que la seconde condition soit vérifiée ou non).

Les résultats (2) et (4) viennent de [174] – ils sont certes formulés et prouvés dans le cadre de la planification avec effets d'actions probabilistes, mais, comme il le fait remarquer en conclusion de son article, dans le cas totalement observable c'est un détail et les preuves s'adaptent immédiatement au cadre non-déterministe tout-ou-rien. La preuve de (3) est dans le

document [154].

6.2.4 Observabilité partielle (cas général)

Un plan est toujours une politique conditionnelle, mais les conditions de branchement sont cette fois des états de croyance quelconques (c'est-à-dire n'importe quel sous-ensemble non-vide de $\mathcal{S}$). Mais il reste un point délicat à éclaircir : *comment ces conditions de branchement sont-elles exprimées ?* Il est évident qu'il ne s'agit pas d'énumérer explicitement les états, en nombre en général exponentiel, contenus dans un état de croyance. Deux hypothèses plus raisonnables peuvent être faites :

- *les conditions de branchement sont des observations* : en chaque noeud de la politique issu d'une action apportant une observation, la condition de branchement porte directement sur cette observation.
- *les conditions de branchement sont des formules épistémiques* : en chaque noeud de la politique, la condition de branchement est une formule épistémique (voir la Section 6.1).

Dans le second cas (qui est plus général que le premier), les plans seront souvent bien plus courts, mais ils seront plus difficiles à exécuter, puisqu'il faudra évaluer les conditions de branchements : alors que la vérification de plan dans le cas des "branchements sur observations" est un problème coNP-complet, il devient Π_2^p-complet dans le cas des "branchements sur formules épistémiques" [121].

Examinons maintenant le complexité de l'existence de plan, en distinguant, pour les représentations compactes, le cas des branchements sur observations (bso) de celui des branchements sur formules épistémiques (bsfe).

La plupart des résultats de difficulté sont des corollaires de résultats obtenus dans les cas non-observable et totalement observable, sauf la 2-EXPTIME-difficulté dans le cas général [197]. Les résultats d'appartenance ne posent pas de difficultés particulières.

Voici enfin quelques perspectives de recherche :

1. quelle est la complexité de l'existence de plan pour les problèmes *purement épistémiques* ? L'article [154] contient quelques résultats ;

↓ plan représentation ⇒	explicite	compacte
$T = 1$	P	coNP-complet
taille polynomiale, bso	NP-complet	Σ_2^p-complet
taille polynomiale, bsfe	NP-complet	dans Σ_3^p
profondeur polynomiale	dans PSPACE	PSPACE-complet
non borné	PSPACE-complet	2-EXPTIME-complet

TAB. 6.4 – Complexité de l'existence de plans en environnement partiellement observable

2. peut-on obtenir des sauts de complexité en faisant des restrictions (notamment syntaxiques) sur l'expression des effets des actions, comme cela a été fait dans le cas de la planification déterministe dans [36] et [6] ?
3. quelles autres hypothèses d'observabilité, autres que les formes extrêmes de non-observabilité et d'observabilité totale, est-il pertinent de considérer et quelle est dans ce cas la complexité de l'existence de plan ?
4. quel est l'impact sur les résultats de complexité de l'introduction de macro-actions effectives (comme en Section suivante) ou épistémiques (voir Section 5) ?
5. même si l'introduction d'actions à effets stochastiques rend le problème d'existence indécidable, peut-on trouver des hypothèses assez simples où cela n'est plus le cas ?

6.3 Planification non-déterministe et logique propositionnelle : le rôle de l'abduction et des problèmes QBF

Comment exprimer simplement et représenter des problèmes de planification non-déterministe (et les résoudre) en logique propositionnelle ?

Commençons par les problèmes à une seule étape de décision. La manière la plus simple d'exprimer un problème est la *contrôlabilité*, introduite dans [26], réétudiée plus en détail dans [156]) et généralisée en contrôlabilité conditionnelle : étant donnée une base de connaissance Σ, un but G et une partition de $Var(\Sigma) \cup Var(G)$ en trois classes $ObsVar$ (variables non contrôlables

observables), $DecVar$ (variables de décision, contrôlables) et $UnobsVar$ (variables non contrôlables non observables), on dit que G est conditionnellement contrôlable selon $\langle \Sigma, ObsVar, DecVar, UnobsVar \rangle$ si et seulement si la formule boolénne quantifiée $\forall ObsVar \exists DecVar \forall UnobsVar(\Sigma \rightarrow G)$ est valide. Cette instance de $\text{QBF}_{3,\exists}$, qui est équivalente à

$$\forall \vec{o} \in 2^{ObsVar}(\vec{o} \wedge \Sigma \text{ cohérent} \Rightarrow \exists \vec{d} \in 2^{DecVar} \forall \vec{u} \in 2^{UnobsVar}(\vec{o}, \vec{d}, \vec{u}) \models G)$$

se lit intuitivement comme suit : "quelle que soit l'observation faite (c'est-à-dire les valeurs des variables observables, connues *en ligne*), il existe une affectation des variables de décision qui réalise certainement le but". Ce problème de décision diffère quelque peu de ceux que nous avons vus jusqu'à présent au chapitre 6, en raison de la structure *combinatoire* de l'espace des actions (voir chapitres 1 et 4). Cette structure combinatoire correspond à des actions complexes (ou macro-actions) composées de micro-actions (une micro-action est ici l'affectation d'une valeur de vérité à une variable de décision), qu'on peut considérer comme exécutées concurremment ; on rencontre ce type de structure assez souvent (problèmes de conception, d'affectation notamment)[5][6].

Cela dit, cette notion de contrôlabilité ne permet de représenter que des actions très frustes. Peut-on mieux faire sans pour autant faire monter la complexité ? La réponse est oui : dans [94] on propose un modèle plus général d'expression de problèmes de décision à une étape, à espace d'actions à structure combinatoire, où les effets d'actions sont calculés par des mise à jour conditionnelles ; en vertu des résultats du chapitre 3, on montre alors que, lorsqu'on choisit un opérateur de mise à jour à base de dépendance, la complexité du problème d'existence de politique solution reste Π_3^p-complet dans le cas général , Π_2^p (resp. Σ_2^p) en environnement totalement (resp. non-)

[5]Par ailleurs, on peut coder des séquences inconditionnelles d'action à longueur bornée comme des macro-actions en dupliquant chaque variable autant de fois qu'il y a d'instants – comme on le fait dans le formalisme SATPLAN – et exprimer ainsi des problèmes de planification à horizon borné.

[6]Lorsque l'ensemble des actions n'a plus de structure combinatoire, le nombre de variables existentiellement quantifiées est bornée par une fonction en $o(log(n))$, ce qui fait baisser la complexité du problème , comme on peut le voir sur le tableau ci-dessous :

	$\lvert A\rvert \leq o(log(n))$	cas général
totalement observable	coNP-complet	Π_2^p-complet
non-observable	coNP-complet	Σ_2^p-complet
cas général	coNP-complet	Π_3^p-complet

observable et coNP-complet si $|A|$ est logarithmiquement borné.

On code alors le problème comme un problème $\text{QBF}_{3,\exists}$, $\text{QBF}_{2,\forall}$ ou $\text{QBF}_{2,\exists}$; la résolution du problème de calcul de politique correspond-il alors à la résolution du problème QBF associé ? Pas tout-à-fait ; plus exactement, ce que l'on peut dire est que les problèmes de décision (QBF d'un côté, existence de politique solution de l'autre) sont équivalents ; mais, lorsqu'on passe au problème de recherche, on se heurte à quelques problèmes.

Premièrement, la version "problème-fonction" de QBF n'a reçu que peu d'attention de la part de la communauté IA (alors que la version "problème de décision" en a reçu beaucoup, particulièrement ces dernières années). Dans [94] et surtout dans [50] (où l'on généralise le problème à des instances de $\text{QBF}_{k,q}$ avec k quelconque), on définit formellement la *solution* d'une instance de QBF comme une politique solution du problème de planification associé. Cette solution est un arbre, de taille hélas exponentielle dès que $(q = \exists$ et $k \geq 3)$ ou $(q = \forall$ et $k \geq 2)$; elle peut jouer le rôle d'un *certificat* pour le problème QBF qu'elle satisfait (et la définition d'un certificat pour QBF est une question importante au sein de cette communauté, en raison du besoin d'évaluation des performances des solveurs QBF), de la même façon qu'un modèle de φ est un certificat de la satisfaisabilité de φ . Mais on n'est pas encore sorti de l'auberge : un certificat, certes, mais un certificat exponentiellement grand, est-ce raisonnable ? Certes non. Heureusement, une politique solution peut souvent être représentée avec bien plus de concision qu'avec sa représentation explicite $\{\langle \vec{o}, \pi(\vec{o})\rangle | \vec{o}$ observation possible$\}$. Une première façon de faire consiste à représenter π par une liste de couples $\sigma = \{\langle \varphi_i, \vec{d_i}\rangle, i = 1 \ldots p\}$, où φ_i est une formule de $\mathcal{L}_{ObsVar}$, $\vec{d_i} \in 2^{DecVar}$; les φ_i ne sont pas nécessairement disjoints ; la politique π induite par σ est définie par $\pi(\vec{o}) = \vec{d_i}(\vec{o})$ où $i(\vec{o}) = \min\{j | \vec{o} \models \varphi_j\}$ [94]. Il n'y a certes pas de garantie dans le cas général qu'une telle représentation permette de gagner de l'espace : dans le pire des cas, la taille de la représentation par liste la plus courte d'une politique solution est $2^{|ObsVar|}$ (mais dans le meilleur des cas, elle est en $o(|DecVar|)$).

Un autre problème est que la version habituelle de QBF est insuffisamment adaptée à la représentation d'un problème de décision dans la mesure où on ne se préoccupe pas de chercher des politiques "approximativement solutions" pour des instances *négatives* de QBF. Pourtant, il serait impensable de renoncer à exprimer une solution partielle, c'est-à-dire qui spécifie uen action à entreprendre dans les situations où il en existe une, sous prétexte

qu'il existe au moins une situation où il n'en existe pas. Voici un exemple :

$$\forall a_1 a_2 \exists b_1 b_2 (b_1 \wedge (a_1 \rightarrow (b_1 \oplus b_2)) \wedge (a_2 \rightarrow b_2))$$

On représente sur le tableau qui suit toutes les affectations de $\{b_1, b_2\}$ satisfaisantes, selon les valeurs de $\{a_1, a_2\}$ observées.

observation	décisions solutions
(a_1, a_2)	$\emptyset$
$(a_1, \neg a_2)$	$\{(b_1, \neg b_2)\}$
$(\neg a_1, a_2)$	$\{(b_1, b_2)\}$
$(\neg a_1, \neg b_2)$	$\{(b_1, \neg b_2), (b_1, b_2)\}$

Cette instance de $\text{QBF}_{2,\forall}$ est négative ; or, il existe pourtant une décision adéquate à prendre pour 3 des 4 observations possibles. Nous avons ainsi été amenés à définir des *politiques maximalement saines* comme des politiques qui affectent une "bonne" décision à chaque fois que c'est possible (ce que l'on appelle "second problème fonction pour QBF dans [50]). Bien entendu, les considérations qui précèdent sur la représentation compacte de politiques solutions s'applique encore à de telles politiques "pseudo"-solution. Pour l'exemple précédent, une politique maximalement saine est celle qui est induite par la liste $\sigma = \{\langle \neg a_1, (b_1, b_2)\rangle, \langle \neg a_2, (b_1, \neg b_2)\rangle\}$.

Enfin, le point qui reste, le plus important sans aucun doute, concerne la résolution pratique d'instances de QBF (où "résolution" signifie production d'une politique – partielle ou totale, si possible saine et maximale).

Plutôt que de s'attaquer d'emblée à QBF dans toute sa généralité, il nous a semblé raisonnable et pertinent de porter notre attention, dans un premier temps, sur les problèmes "d'en bas", spécifiquement $\text{QBF}_{2,\exists}$, $\text{QBF}_{2,\forall}$ et $\text{QBF}_{3,\exists}$. Le point commun de ces trois classes d'instances (ainsi que de SAT = $\text{QBF}_{1,\exists}$) et qu'elles ne comportent qu'une seule série de variables quantifiées existentiellement, donc une seule étape de décision (ou éventuellement une seule série de décisions sans feedback, ce qui revient formellement au même). S'intéresser d'abord à ces problèmes est d'autant plus pertinent qu'un grand nombre de problèmes de raisonnement et décision en intelligence artificielle se situent au second ou au troisième niveau de la hiérarchie polynomiale ; par ailleurs on peut utiliser QBF comme un langage d'expression de problèmes (voir [86] où cela est fait et expérimenté avec succès), ce qui renforce l'intérêt du développement d'algorithmes spécifiques pour QBFk, q avec $q \in \{2, 3\}$. On

reviendra là-dessus en Conclusion.

Les algorithmes auxquels nous nous intéressons exploitent le rôle que l'abduction joue pour la résolution de QBF. Ce rôle est évident pour $\text{QBF}_{2,\exists}$ puisqu'une solution pour $\exists\vec{a}\,\forall\vec{b}\,\varphi$ n'est rien d'autre qu'une explication pour φ sur le langage des $\{a_i\}$. Pour $\text{QBF}_{2,\forall}$ et $\text{QBF}_{3,\exists}$, nous avons étudié les propriétés de plusieurs algorithmes dont le point commun est de calculer itérativement des décisions $\vec{d}$ "couvrant" au moins une partie de l'espace d'observations non encore couvert, de déterminer (abductivement) l'ensemble des observations possibles $\Phi_{\vec{d}}$ couvert par $\vec{d}$ (et non encore couvertes par les décisions précédemment calculées) et d'ajouter $\langle\Phi_{\vec{d}}, \vec{d}\rangle$ à la liste de décisions. L'output est donc une politique exprimée sous forme compacte ; il tend vers une politique saine maximale si on laisse l'algorithme se dérouler jusqu'au bout. On propose plusieurs variations de cet algorithme générique, qui répartissent de diverses façons la charge computationnelle entre la phase hors-ligne et la phase en-ligne.

Enfin, dans l'article [122], on va un peu plus loin et on montre comment résoudre des problèmes de planification en utilisant des techniques d'abduction (plus précisément, par la génération d'impliquants premiers sur des sous-langages de variables observables) en environnement totalement observable. La base de l'algorithme est le calcul de la *régression* d'un ensemble d'états de connaissances par une action ontique. Il reste, pour pouvoir enfin développer un générateur de plans en environnement partiellement observable, à déterminer la régression d'un ensemble d'états de connaissances par une action épistémique (ce qui est l'objet d'un article en préparation). Des expérimentations (dans le cadre de la thèse de Thomas Polacsek) sont en cours de réalisation.

6.4 Décision dans l'incertain et satisfaction de contraintes

Au Chapitre 4, on a (brièvement) vu que le cadre (de représentation et de résolution) des CSP peut être enrichi (via l'introduction de valuations) de façon à représenter des préférences et à calculer des décisions optimales en l'absence d'incertitude. On va voir ici qu'on peut également l'enrichir de manière à pouvoir représenter et résoudre des problèmes de décision sous incertitude.

Posons d'abord la question suivante : pour quel type de problèmes utilise-

t-on généralement le cadre classique des CSP ? En examinant les exemples et applications dans la littérature (récente ou pas), on constate qu'il existe deux classes bien distinctes de problèmes pour lesquels on utilise les CSPs :

- lorsque les variables du CSP sont contrôlables, on a affaire à un CSP "purement orienté décision". C'est l'interprétation la plus courante (bien qu'implicite) dans la littérature sur les CSP (par exemple, les problèmes de conception ou d'ordonnancement...) . Les contraintes représentent des préférences (certes binaires, dans la version standard non valuée), une solution étant alors une instanciation des variables satisfaisant chaque "granule de préférence", exactement (à la syntaxe près) comme le formalisme R_{brute} pour la logique propositionnelle.
- lorsque les variables du CSP sont incontrôlables, on a affaire à un CSP "purement orienté raisonnement" : les contraintes représentent alors des *connaissances* sur les valeurs réelles des variables. La résolution du CSP ne consiste alors pas nécessairement à trouver une affectation consistante, mais plutôt à trouver la *véritable valeur* de chaque variable *dans le monde réel*. Cette interprétation est plus rare dans la littérature, mais pas inexistante. C'est par exemple le cas pour les problèmes d'étiquetage d'arêtes (en vision) de Waltz ou différents problèmes de reconstruction de données à partir d'informations indirectes (on peut aussi citer – quoique plus anecdotiquement – les puzzles logiques comme "trouver qui habite dans quelle maison et boit quelle boisson ...").

Enfin, notons que les requêtes adressées aux deux types de problèmes semblent sensiblement différentes. Dans un problème de décision, la principale question est de déterminer si le problème est consistant, et de lui trouver une solution. Même s'il est intéressant, dans le cadre d'un problème de raisonnement, de déterminer la consistance du CSP (ce qui signifie vérifier que les connaissances ne sont pas contradictoires), les requêtes dans ce type de problème concernent plutôt la preuve d'une propriété (comme c'est le cas dans les approches logiques de représentation des connaissances). En termes de CSP, cette dernière requête doit pouvoir être assimilée à un type particulier d'inférence de contraintes.

Plus généralement, les problèmes de décision sous incertitude non dégénérés impliquent les deux types de variables – tout comme en Section 6.3 pour la logique propositionnelle : c'est l'objet des travaux que j'ai effectués avec Hélène Fargier, Roger Martin-Clouaire et Thomas Schiex. Nous parlerons de problèmes mixtes, utilisant le terme de contrainte mixte pour désigner une

contrainte faisant dépendre les valeurs acceptables des variables de décisions des valeurs effectives des variables contingentes[7]. Quand l'incertitude sur les valeurs effectives des variables contingentes est représentée qualitativement, par un ensemble de contraintes-connaissances, on parlera de *CSP mixtes*; quand elle sera représentée plus quantitativement par une distribution de probabilité, on parlera de *CSP probabilistes*. Ces deux extensions du modèle CSP, assez similaires conceptuellement, sont l'objet des articles [95] (CSP probabilistes) [99] (CSP mixtes) et [96] (article de synthèse).

Dans un *CSP mixte* [99] l'incertitude est exprimée de façon qualitative, alors que dans les *CSP probabilistes* [95], l'incertitude sur les valeurs des variables contingentes est représentée par une distribution de probabilité. Dans chacun de ces articles nous considérons les deux situations extrêmes de *non-observabilité* (pas de nouvelles connaissances on-line sur les valeurs des variables contingentes) et d'*observabilité totale* (l'état du monde est complètement connu en ligne).

Un *CSP mixte* est un sextuplet $\mathcal{P} = \langle \Lambda, W, X, D, \mathcal{C}, \mathcal{K} \rangle$ où $\Lambda = \{\lambda_1, \ldots, \lambda_p\}$ est un ensemble de variables contingentes; $W = W_1 \times \cdots \times W_p$, où W_i est le domaine de λ_i; $X = \{x_1, \ldots, x_n\}$ est un ensemble de variables de décision; $D = D_1 \times \cdots \times D_n$, où D_i est le domaine de x_i; $\mathcal{C}$ est un ensemble de contraintes, chacune mettant en jeu au moins une variable de décision. K est un ensemble de contraintes ne portant que sur des variables contingentes. La distinction entre K et $\mathcal{C}$ est à peu près la même qu'en Section 6.3 entre Σ et G. L'ensemble des solutions du CSP $\langle \Lambda, W, K \rangle$, désigne l'ensemble des mondes possibles.

Un *CSP probabiliste* est un sextuplet $\mathcal{P} = \langle \Lambda, W, X, D, \mathcal{C}, pr \rangle$ où Λ, W, X, D et $\mathcal{C}$ sont comme pour les CSP mixtes, et $pr : W \rightarrow [0, 1]$ est une distribution de probabilité sur W. [8]

L'output d'un CSP mixte ou probabiliste est, sans surprise, une simple affectation inconditionnelle des variables dans le cas non-observable et une politique conditionnelle dans le cas totalement observable.

[7] Ces deux classes de contraintes (et les deux classes de variables) sont également présentes, sous une forme syntaxique différents, dans les *diagrammes d'influence* [127].

[8] Il n'est en général pas raisonnable de supposer que pr est spécifiée explicitement. Dans le cas le plus simple, les variables contingentes sont mutuellement indépendantes (pr est donnée par des distributions de probabilité individuelles pr_{λ_i} associées à chaque λ_i). On peut aussi considérer un cas plus complexe où pr est spécifiée à l'aide d'un réseau bayésien.

On a, dans ces deux cadres de représentation et de résolution, défini deux formes de cohérence (forte et faible) et élaboré des algorithmes anytime dont l'output donne, lorsqu'on laisse l'algorithme se dérouler jusqu'au bout, une affectation optimale, c'est-à-dire : dans le cas totalement observable, une affectation conditionnelle couvrant tous les mondes possibles qui peuvent l'être ; dans le cas non-observable mixte, une affectation conditionnelle couvrant un ensemble maximal (au sens de l'inclusion) de mondes ; dans le cas non-observable probabiliste, une affectation conditionnelle maximisant la probabilité d'être solution.

6.5 Planification et incertitude possibiliste

6.5.1 Considérations générales

Jusqu'à présent (dans ce chapitre) on a considéré uniquement deux types d'incertitude sur les effets des actions (et sur l'état initial du monde), à savoir le non-déterminisme pur et le modèle probabiliste classique. On va maintenant s'intéresser à ce que devient la planification en environnement non-déterministe avec un autre type d'incertitude, à savoir le modèle possibiliste.

Pourquoi ce modèle ? D'abord parce qu'il est, comme on l'a dit au chapitre 1, un bon intermédiaire entre le non-déterminisme pur (qu'il généralise) et les probabilités ; ensuite, pour une autre raison (sans doute liée à la précédente), spécifique au raisonnement sur l'action : en représentation des connaissances, l'incertitude sur les effets des actions est souvent représenté par une "fonction de rangement" (ranking function) ou un préordre, structures qui sont dans une large mesure équivalentes au modèle possibiliste (voir par exemple [14] et, plus spécifiquement au raisonnement sur l'action, [68]).

Le modèle possibiliste convient particulièrement aux actions à effets normaux et exceptionnels [68] et [110]. $\pi(s'|s,a)$ est la possibilité d'obtenir l'état s' quand on a exécuté l'action a dans l'état s, et représente, si l'on préfère, le *niveau de normalité (ou de "non-exceptionnalité"* du résultat s' dans le contexte (s,a) ; en particulier :

- $\pi(s'|s,a) = 1$ signifie que s' est un résultat normal – ou *le* résultat normal, quand il existe un unique s' tel que $\pi(s'|s,a) = 1$ (lorsque cela est vrai pour tous a et s, on dit que l'action est *pseudo-déterministe*) ;
- $\pi(s'|s,a) = 0$ signifie que s' est totalement exclu.

Toutes les valeurs intermédiaires correspondent à des niveaux d'exceptionnalité différents (N niveaux de possibilité distincts correspondent à $N-2$ niveaux d'exceptionnalité – il en suffit souvent de quelques-uns). Les degrés d'exceptionnalité associés aux différents résultats peuvent être déterminés automatiquement à partir de règles par défaut, en utilisant (entre autres) un critère de spécificité (voir [68], [110] et [49]).

Le choix du modèle possibiliste impose que la possibilité d'obtenir un état s après avoir exécuté un plan P dans un état s_0 est la possibilité de la trajectoire la plus plausible dont le dernier état est s ; formellement :

$$\pi(s|s_0, P) = \max\{\pi(\tau|s_0, P)|\tau \in TRAJ, last(\tau) = s\}$$

Si $G \subseteq S$ est un ensemble d'états buts, la qualité du plan P peut se mesurer par la possibilité et la nécessité qu'il a d'atteindre un état but, à savoir

$$\Pi(G|s_0, P) = \max\{\pi(s|s_0, P)|s \in G\} = \max\{\pi(\tau)|\tau \in TRAJ, last(\tau) \in G\}$$

$$N(G|s_0, P) = 1 - \Pi(\bar{G}|s_0, P)$$

C'est tout ce que le modèle possibiliste impose. Restent trois paramètres dont le choix va influer de manière significative sur la complexité et les méthodes de résolution du problème :

- comment $\pi(\tau|s_0, P)$ est déterminée en fonction des possibilités de transitions élémentaires ;
- comment l'utilité d'une trajectoire est déterminée en fonction des utilités instantanées ;
- comment on définit l'optimalité d'un plan.

Pour le premier point, l'hypothèse de markovianité impose que l'incertitude sur les trajectoires soit temporellement décomposable, c'est-à-dire qu'il existe une fonction $\bigotimes$ telle que

$$\pi(\tau|s_0, P) = \bigotimes\{\pi(\tau(t+1)|\tau(t), \alpha(t))|t = 0 \ldots |\tau| - 1\}$$

où α_t est l'action entreprise en t selon P. Plusieurs choix sont pertinents pour $\bigotimes$. Il est entendu que $\bigotimes$ doit vérifier

1. $\bigotimes(1, \ldots, 1) = 1$, puisqu'une trajectoire dont toutes les transitions sont possibles doit être possible ;

2. $\bigotimes(\vec{x}) = 0$ dès que $\vec{x}$ comprend une composante x_t nulle ;
3. monotonie : $\bigotimes(\vec{x}) \geq \bigotimes(\vec{y})$ dès que $\vec{x} \geq \vec{y}$ composante à composante.

et on a aussi envie de requérir (4) la commutativité de $\bigotimes$ (peu importe l'ordre dans lequel les transitions exceptionnelles arrivent pour déterminer le degré d'exceptionnalité de la trajectoire) et (5) son associativité (pour des raisons cette fois computationnelles – on veut éviter d'avoir à mémoriser les valeurs rencontrées le long de la trajectoire). Il ne manque alors plus grand chose pour que $\bigotimes$ soit une t-norme[9]. De fait, les deux choix les plus pertinents sont min et le produit[10].

Nous passons plus rapidement sur le second point. Il y a moins de propriétés à requérir : seulement la décomposabilité temporelle de l'utilité, ce qui signifie qu'il existe une fonction $\bigoplus$ telle que

$$u(\tau) = \bigoplus\{u(\tau(t), \alpha(t)) | t = 0 \dots |\tau| - 1\}$$

, plus la monotonie de $\bigoplus$ et probablement sa commutativité (et là encore, ne pas requérir l'associativité pose de gros problèmes computationnels). Dans ce qui suit, on ne se posera plus de problèmes avec l'utilité des trajectoires : on fera l'hypothèse de les états intermédiaires et les actions ne comptent pas et donc que l'utilité d'une trajectoire est égale à l'utilité de son dernier état.

Le dernier point dépend des paramètres épistémo-préférentiels du modèle. Une hypothèse sympathique est la commensurabilité incertitude-utilité (qui suppose que les utilités des trajectoires sont à valeurs dans $[0, 1]$), qui permet d'évaluer les différentes politiques selon leur utilité optimiste et leur utilité pessimiste.

6.5.2 Planification possibiliste conformante

La thèse de Célia Da Costa Pereira, que j'ai encadrée à 50 %, portait sur la représentation et la résolution de problèmes de planification en environnement non-observable (ou encore "planification conformante") avec incertitude

[9] Il manque le fait que 1 soit élément neutre – propriété qui peut être considérée elle aussi comme assez naturelle.

[10] Lorsque $\bigotimes$ = produit, le calcul de $\pi(\tau)$ à partir des $\pi(\tau(t+1)|\tau(t), \alpha(t))$ est identique au calcul de $pr(\tau)$ à partir des $pr(\tau(t+1)|\tau(t), \alpha(t))$ lorsque les actions sont stochastiques ; ce qui fait alors que le modèle *n'est pas probabiliste* est le passage des valeurs attachées aux singletons aux valeurs attachées aux sous-ensembles de S.

possibiliste. La restriction à des environnements non-observables peut certes paraître très limitative ; il faut dire qu'à l'époque où cette thèse a commencé (1994), peu d'articles se préoccupaient à la fois de planification dans l'incertain et de représentations compactes, et l'un des seuls qui était dans ce cas était [139] qui donnait un modèle et des algorithmes pour la planification probabiliste en environnement non-observable (le planificateur BURIDAN), et nous voulions démontrer que le passage à des modèles moins quantitatifs de l'incertain permettait un fort gain computationnel[11]. Ceci dit, une bonne partie des résultats que nous avions obtenus a un intérêt qui ne se limite pas à la planification conformante.

Une action à effets possibilistes est définie par une fonction de transition possibiliste (voir Section). La représentation de telles actions est faite à l'aide d'opérateurs STRIPS possibilistes qui sont à quelques détails près des transpositions des opérateurs STRIPS possibilistes de [139]. Un plan P est optimal si ca certitude d'aboutir à un état but $N(G|s_0, P)$ est maximale. Dans [54] on montre qu'en environnement possibiliste non-observable, l'existence d'un plan P de longueur polynomialement bornée tel que $N(G|s_0, P)$ soit supérieur à un certain seuil est Σ_2^p-complet, c'est-à-dire pas plus complexe qu'en environnement non-déterministe tout-ou-rien[12].

Les algorithmes de recherche d'un plan valide en environnement non-observable tout-ou-rien, et d'un plan de certitude maximale en environnement non-observable possibiliste sont, comme BURIDAN, des généralisations de SNLP [180] permettant la prise en compte d'actions non-déterministes.

6.5.3 Processus décisionnels pseudo-markoviens et incertitude ordinale

Dans l'article [200] (et sa version courte [97]), on se place cette fois sous l'hypothèse d'*observabilité totale* et on développe une contrepartie possibiliste des processus décisionnels markoviens (avec, pour une fois, une représentation explicite des états et des actions).

[11] En effet, dès que l'environnement n'est plus totalement observable, la planification en environnement probabiliste devient un vrai calvaire computationnel : le problème d'existence de plan dont la probabilité de succès (ou plus généralement l'utilité espérée) dépasse un seuil donné est même indécidable [177].

[12] Les résultats en ce qui concerne la planification possibiliste sont en fait des corollaires des résultats – simples – dans le cas non-déterministe tout-ou-rien, qui figurent eux aussi dans l'article [54] ; à l'époque ils étaient nouveaux.

On fixe le choix $\bigotimes = \min$ pour le calcul de la possibilité d'une trajectoire en fonction des possibilités de transition[13], et on montre que le calcul d'une politique optimale (au sens de la maximisation de $N(G|s_0, P)$) peut être fait par induction à rebours, tout comme avec le critère classique de l'utilité espérée. Ce résultat est dû à une propriété sympathique qui exprime qu'une politique d'action pour les instants t à N est optimale si chacune de ses sous-politiques de $t-1$ à N est optimale[14].

À l'opposé de celui de la section précédente, ce travail ne considère pas des représentations compactes ; voir Guéré et Alami [113] pour un planificateur en environnement possibiliste totalement observable et représentation à la STRIPS.

Enfin, les travaux exposés dans [200] ont été étendus aux environnements partiellement observables par Sabbadin [199] ; le point-clé est le choix d'un opérateur de conditionnement – sachant que plusieurs définitions sont acceptables [77].

[13] Ce qui n'est en fait pas une si grande perte de généralité ; ce qu'on va dire par la suite tiendrait avec n'importe quelle autre norme triangulaire.

[14] Attention : à cause de l'utilisation de max, la condition d'optimalité des sous-trajectoires est ici suffisante mais pas nécessaire – ce qui ne pose d'ailleurs aucun problème.

Chapitre 7

Conclusion

J'ai choisi de structurer ce document en fonction des *problèmes* à résoudre (raisonnement plausible, raisonnement sur l'action, raisonnement sur le temps et l'espace, représentation de préférences, décision de groupe, planification épistémique, diagnostic, planification en environnement partiellement observable) et non des outils utilisés. Cependant, un plan suivant les outils et formalismes mis en œuvre aurait eu une certaine pertinence, puisque certains d'entre eux reviennent régulièrement au cours du document. Je me propose ici de revenir brièvement sur ces formalismes et outils, et j'illustrerai leur généricité en citant pour chacun deux quelques problèmes de raisonnement et décision pour lesquels ils sont pertinents. Là encore, je ne prétends à aucune exhaustivité, et la liste que j'élabore est avant tout fondée sur les travaux auquels j'ai contribué (d'où l'existence inévitable d'un biais).

7.1 Raisonnement, décision et logique propositionnelle

Pourquoi la logique ?

Elle est omniprésente dans ce document. Ce n'est pas qu'un choix de ma part : une grande partie des travaux d'IA en font usage, en construisant des formalismes centrés sur la satisfaisabilité, la déduction (en logique propositionnelle classique). C'est d'ailleurs pourquoi la résolution du problème SAT revêt autant d'importance dans la communauté – et c'est aussi pour cela que l'on considère que c'est *aussi* de l'IA, et non seulement de la re-

cherche opérationnelle ou de la déduction automatique[1]. Je ne souhaite pas m'étendre sur la place de la logique en intelligence artificielle (on peut trouver cela partout et bien mieux argumenté que je ne saurais le faire). Dire que la logique est pertinente pour la formalisation du raisonnement est presque une trivialité : c'est l'un des buts dans lesquels elle a été développée depuis ses origines – tant la logique classique que ses variantes non-classiques (intuitionniste, de la relevance, du vague) et ses extensions (modales en particulier). Son rôle en décision est (un peu) plus récent. Derrière cette prééminence de la logique en IA il y a, à mon avis, deux motivations assez différentes (ce qui ne veut pas dire qu'il n'y a pas de travaux qui adhèrent aux deux démarches à la fois) :

1. *la logique comme modèle formel* : il s'agit de décrire dans un langage formel des activités mentales ayant trait au raisonnement et à la prise de décision : c'est le point de vue qui prédomine, par exemple, dans les articles présentés aux congrès TARK ou LOFT, à cheval entre économie mathématique et intelligence artificielle, et pour une partie des articles des congrès KR. Ces travaux recherchent en général une expressivité riche et tendent à développer des formalismes logiques souvent d'une grande complexité (et parfois d'une grande généralité).
2. *la logique comme outil de résolution* : il s'agit ici d'exprimer des problèmes en logique dans le but de les résoudre en utilisant des "prouveurs" (au sens large du terme) : la logique est vue ici comme une boîte noire ou un langage de programmation. Ici, le compromis efficacité-expressivité est une préoccupation importante et les formalismes utilisés sont souvent plus élémentaires que ceux qui apparaissent dans les travaux relevant du point précédent. La logique propositionnelle, enfin, est *le langage prototypique de représentation compacte* (on y revient plus loin).

Pourquoi la logique propositionnelle ?

Quand on examine les revues ou actes de congrès d'IA de ces dernières années, on constate que le "véritable" premier ordre (avec symboles de fonction) y tient une place de plus en plus limitée – en tout cas plus limitée que dans les années 70 et 80. (Ceci ne vaut pas pour les congrès et revues

[1]Il est également vrai que les frontières entre domaines sont non seulement floues mais arbitraires la seule façon objective que je connais de répondre à la question "est-ce de l'IA ?" est de définir l'IA par le contenu des conférences et revues généralistes IJCAI, AAAI, ECAI, AIJ, JAIR : l'IA est ce que font les chercheurs qui publient dans des revues d'IA.

spécialisés en déduction automatique ou en programmation logique. J'hésite à avancer des explications. Une raison plausible est que les formalismes développés sont décidables et se marient assez facilement aux méthodes de résolution et d'optimisation combinatoire de la recherche opérationnelle ; une autre serait que, tout simplement, la logique propositionnelle (ou le premier ordre sans symboles de fonction) et ses extensions toujours propositionnelles suffisent à représenter et à résoudre la plupart des problèmes visés en intelligence artificielle.

Quand je dis "logique propositionnelle", je n'entends pas seulement la logique propositionnelle classique, mais l'ensemble des logiques dont le langage est propositionnel. Il est évident que les logiques non-classiques jouent un rôle important en IA. Encore fait-il s'accorder sur ce que signifie cette terminologie "non-classique" : les logiques substructurelles (logique intuitionniste, logiques multivaluées, logiques de la relevance, logiques paraconsistantes, logique linéaire etc.), qui sont véritablement non-classiques, sont relativement peu présentes en IA – à l'exception, dans une certaine mesure, des logiques paraconsistantes. Les "formalismes logiques non-classiques" jouant un fort rôle en IA sont en fait souvent des constructions englobant la logique propositionnelle classique, et donc plus des *extensions de la logique classique* que de véritables logiques non-classiques (en vrac : logiques modales de la croyance et de l'action ; logique des défauts ; logique probabiliste, possibiliste ; logiques de l'action et du changement ; logiques de description ; etc.).

Logique propositionnelle, connaissances et préférences

Il est intéressant de revenir sur l'usage qui est fait de la logique propositionnelle "de base" (langage et relation d'inférence classiques). Si on y regarde de plus près (dans le document comme plus généralement en IA), on utilise la logique propositionnelle pour coder, à l'aide de formules propositionnelles :

- des *connaissances* (ou croyances – en logique classique la distinction ne peut être faite formellement) ; le rôle de l'inférence (déduction, abduction, "théorèmicité") est ici important pour permettre de déterminer si une conclusion suit un ensemble de prémisses, si une explication est satisfaisante) ;
- des *préférences* : le rôle, ici, de la "modèlicité" – donc de la recherche de modèles – dépasse ici celui de l'inférence au sens premier (le modèle est une solution satisfaisante – idem en CSP).

Cependant, il y a un danger à représenter conjointement des connaissances et des préférences dans un même formalisme (et pourtant, on trouve régulièrement, dans la littérature de l'IA, des formalismes qui confondent allégrement les deux). L'un des effets aberrants en est le "wishful thinking' (le fait de prendre ses désirs pour des réalités) [214]. Il est en revanche assez simple de s'en sortir à moindre coût, en introduisant une distinction formelle (un *typage*) entre formules traduisant des croyances/connaissances et formules traduisant des préférences (voir [94]).

Incohérence

Il est intéressant de remarquer le rôle important que joue le traitement non-trivial de l'incohérence (dans ce document comme plus généralement en IA). On a vu en effet plusieurs problèmes où ce besoin apparaît :

1. le raisonnement à partir d'informations provenant de sources multiples conflictuelles (chapitres 2 et 3) ;
2. le conflit entre le résultat escompté d'une action et une observation (révision), ou entre des effets concurrents et conflictuels d'une action (chapitre 3) ;
3. la représentation de préférences : préférences conflictuelles d'un agent isolé (chapitre 4) ;
4. la décision de groupe : préférences de plusieurs agents, individuellement cohérentes mais conjointement conflictuelles (chapitre 4) ;
5. le diagnostic : conflits entre les hypothèses de bon fonctionnement des composants et les observations (chapitres 2 et 5).

Et on a aussi vu une panoplie d'outils (essentiellement au chapitre 2) : logiques paraconsistantes, révision et fusion, approches dites "syntaxiques" à base de sélections de sous-ensembles maximaux consistants, conditionnels, argumentation, oubli de variables, résolution active par discrimination... Il resterait sans doute à aller plus loin afin de savoir dire quels outils sont pertinents pour quels types de problèmes, mais je ne m'y aventurerai pas.

Structures et dépendances

L'un des avantages de la logique propositionnelle est qu'elle se prête bien à l'identification (puis l'exploitation) de structures sous-jacentes aux bases de croyance ou de préférences. On a vu dans ce document plusieurs types de dépendances (entre variables, entre variables et formules, entre formules...) qui permettent de structurer ces bases de croyance ou de préférences (il est

d'ailleurs intéressant de remarquer que certaines de ces structures sont tout aussi pertinentes pour le raisonnement que pour la prise de décision) : *indépendance formule-variable* et *formule-littéral* (et leur fort lien avec l'*oubli* de variables et de littéraux), *indépendance conditionnelle*, *définissabilité*, *contrôlabilité*. Il existe de multiples contextes (et on en a vu plusieurs dans ce document – presque à chaque chapitre) où il est pertinent de travailler avec de telles structures : raisonnement sur l'action, révision et mise-à-jour des croyances, représentation de préférences, diagnostic, décision à une étape, déduction automatique. Il me semble en particulier que la résolution de problèmes QBF, et *a fortiori* des problèmes qui peuvent facilement être codés comme des problèmes QBF (voir la Section 6.3), gagnera beaucoup à l'identification de structures pendant la phase hors-ligne.

7.2 Logiques, pondérations, priorités, distances

Pondérations, priorités et logique propositionnelle

Le besoin de représenter de façon quantitative (ou du moins, graduelle) de l'incertitude (relative à des croyances) ou des préférences, tout en conservant le caractère compact et structuré de la logique propositionnelle, nous a conduit à développer des formalismes associant des formules propositionnelles à des poids ou des priorités (on en a vu des exemples aux chapitres 2 et 4). Ce qui soulève quelques questions : *y a-t-il derrière ces constructions logiques un cadre général simple* ? ces constructions peuvent-elles être les mêmes pour représenter des croyances et des préférences ? Si oui (alors que l'on n'a pas cessé de souligner le hiatus entre croyances et préférences), pourquoi est-ce possible ?

La réponse à la première question n'est pas simple : il n'y a pas *un*, mais *plusieurs* modèles généraux (en tout cas au moins deux) – j'omets ici les détails techniques de ces constructions générales. La réponse à la seconde question est moins technique. Il peut effectivement paraître surprenant à première vue qu'on puisse utiliser des mêmes langages pour représenter des connaissances et des préférences, alors qu'on n'a pas cessé de répéter que c'étaient deux types d'information radicalement différents. Pourtant, ce n'est pas si surprenant si l'on pense que les logiques à pondération et à priorités sont avant tout des langages de représentation compactes de structures élémentaires : préordres sur 2^{VAR}, fonctions de 2^{VAR} dans $[0,1]$ dans $\mathbb{N}$. Que ces préordres représentent une incertitude comparative ou une préférence, que ces fonctions associant des nombres aux interprétations soient des me-

sures d'incertitude (probabilités, possibilités, "fonctions kappa" ...) ou des utilités, n'est pas fondamental dès lors qu'il ne s'agit que de les représenter de façon compacte. Le même argument prévalait d'ailleurs pour la logique propositionnelle pure et simple (voir la discussion en Section 7.1), et elle prévaudra encore en Section 7.3 lorsqu'on parlera de logiques des conditionnels et de logiques non-monotones.

Distances et logique propositionnelle

Et l'argument que l'on vient juste d'évoquer au-dessus vaut aussi pour le mariage des (pseudo-)distances avec la logique propositionnelle : il joue en effet un rôle important dans de nombreux – et variés – problèmes de raisonnement et décision en intelligence artificielle, concernant des croyances tout autant que des préférences :

- *distances et croyances* : la révision ([132]), la mise-à-jour ([131]), la fusion de croyances (voir par exemple [136, 134]) mais aussi le raisonnement basé sur la similarité [69] et à partir de cas, font appel à des pseudo-distances, où en tout cas à des relations de proximité comparative ou de similarité qui sont leur contrepartie ordinale.
- *distances et préférences* : les structures préférentielles sont pertinentes tant pour la représentation compacte de préférences (voir [141, 142, 130] que pour leur élicitation (voir [115] – notons toutefois que les notions de distance utilisées dans ces deux contextes sont différentes).

Or, étrangement, peu d'efforts ont été faits en ce qui concerne la proposition de distances pertinentes, alors qu'un usage quelque peu abusif est fait de la distance de Hamming (dont la pertinence est limitée du fait qu'elle dépend étroitement du choix du langage). Un objectif de recherche intéressant consisterait à proposer des familles de distances pertinentes (ce à quoi j'ai contribué dans [142], en proposant des familles de distances assez générales, permettant de traduire des dépendances entre variables propositionnelles, ce que ne permet pas la distance de Hamming), ainsi que des langages pour les exprimer de façon compacte.

Je voudrais ici citer un dernier travail auquel j'ai contribué, qui se situe plus ou moins dans la lignée de ce qui précède. Dans [25], Isabelle Bloch (ENST, Paris) et moi avons regardé comment intégrer des concepts issus de la morphologie mathématique en logique propositionnelle ; nous avons ainsi défini plusieurs types d'opérateurs de morphologie mathématique sur des formules propositionnelles (dilatation, érosion, érosion ultime, squelette, ou-

verture, fermeture), étudié certaines de leurs propriétés et montré comment les calculer dans certains cas simples.

7.3 "Logiques" non-monotones, révision, conditionnels, et ordinalité

"Logiques" non-monotones, révision, logiques des conditionnels : ces formalismes apparaissent à maintes reprises dans le document, que ce soit pour représenter des croyances (révisables) ou des préférences. Ce qui soulève encore une fois une question : pourquoi des outils à l'origine développés pour raisonner sur des *croyances* peuvent-ils être également pertinents pour traiter des *préférences* ? Et, par là, que signifient la non-monotonie ou les conditionnels lorsqu'il s'agit de préférences ?

Une raison possible en est que ces notions, là encore, sont plus universelles qu'elles ne le paraissent à première vue : les structures sémantiques qui leur sont sous-jacentes sont identiques, et, de fait, élémentaires : ce sont des relations de préordre (en tout cas tant que $\mid\!\sim$ est une relation d'inférence préférentielle au sens de [138], que $*$ est un opérateur de révision au sens de AGM [1] et que $\Rightarrow$ est un opérateur conditionnel au sens de Lewis [168] dans une logique vérifiant la propriété de *absoluteness*). Le fait que l'inférence à partir de bases de connaissances possibilistes se ramène aux versions "totalement ordonnées" de ces formalismes – voir [76, 103] – n'est pas plus surprenant, puisque, dès lors qu'il ne s'agit que d'inférence, la logique possibiliste et sa contrepartie ordinale sont équivalentes).

On pourrait donc reformuler la première phrase de cette Section en disant que ce sont les *structures ordinales* qui sont omniprésentes dans le document (et, plus généralement, dans la littérature de la représentation des connaissances et de la formalisation du raisonnement). Le rôle de ces structures ordinales, et des processus de minimisation (qui dit ordinalité dit minimisation...) est significatif dans une grande variété de domaines, et les relations de préordre peuvent en conséquence revêtir plusieurs significations (voir en particulier [178, 216]) : l'incertitude (ou les notions proches de normalité, de spécificité) ; la préférence ; la permission ; la similarité.

Il faut noter à ce sujet que c'est bien souvent le processus de minimisation, associé à l'utilisation de la logique (et donc à des requêtes SAT ou UNSAT) qui fait monter la complexité des problèmes au second niveau de la

hiérarchie polynomiale.

Pourquoi les structures ordinales sont-elles si intéressantes en IA ? J'avancerai deux raisons :

- l'ordinalité représente un bon compromis entre le qualitatif pur ("tout-ou-rien") qui est trop pauvre, trop peu expressif, et le quantitatif, qui est parfois trop riche – une trop grande expressivité pouvant conduire à une modélisation peu pertinente parce que contenant trop d'arbitraire ;
- la complexité computationnelle des problèmes lorsque les structures sont ordinales est souvent moindre qu'avec des structures numériques et à peine plus élevée qu'avec des structures qualitatives pures[2]. C'est flagrant, par exemple, pour la planification en environnement partiellement observable, où le choix du critère de l'utilité espérée conduit à des problèmes indécidables [177].

Le choix de modèles ordinaux n'est cependant pas sans poser de problèmes. Premièrement, le fait de représenter, dans un cadre purement ordinal, incertitude et préférence par des structures de même nature amène parfois à des confusions (qui ne peuvent pas se produire dans un modèle quantitatif : une fonction d'utilité ne peut pas être confondue avec une distribution de probabilité) ; en témoigne le grand flou qui règne autour du terme "préférence" dans la littérature de l'IA où, en définitive, il n'est souvent rien de plus qu'un synonyme de "préordre" et n'a souvent que très peu de rapport avec la notion de préférence au sens que lui donne la théorie de la décision. Cette confusion mène d'ailleurs à des paradoxes comme celui du "wishful thinking" dont on a parlé plus haut.

Ensuite, les structures purement ordinales restent finalement assez pauvres (et donc peu expressives), ce qui pose des problèmes dès qu'il s'agit d'agréger incertitude et préférence (à cause de l'absence de commensurabilité, voir [70]), sans oublier les divers effets de "noyade" évoqués aux chapitres 2 et 4 de ce document.

Je termine cette Section en remarquant le rôle important que joue la révision, et plus généralement le changement des croyances, dans de multiples domaines : philosophie, intelligence artificielle, et économie (en particulier en théorie des jeux [222]).

[2]Attention ! je n'affirme pas que le passage du qualitatif au quantitatif s'accompagne *toujours* d'une hausse de complexité

7.4 Affronter la complexité : représentations compactes, prétraitement, résolution approchée

Un problème revient constamment tout au long du document (et, de façon plus générale, dans la sous-communauté de l'IA qui s'occupe de représentation des connaissances et de formalisation du raisonnement et de la prise de décision) : la haute complexité computationnelle des problèmes posés. Reviennent tout aussi souvent deux sujets, fortement liés au précédent :

1. le besoin de *représentations compactes* des données des problèmes considérés ;
2. le rôle du *prétraitement*, et donc de la séparation entre phase en-ligne et phase hors-ligne, dans la résolution des problèmes.

Commençons par les langages de représentation compacte. L'IA, par opposition à la recherche opérationnelle, a privilégié le développement de formalismes *génériques* et à forte expressivité. Or, il est bien connu que l'augmentation de l'expressivité et de la généricité s'accompagne inévitablement d'une hausse de complexité.

Une constante de la représentation des connaissances et de la formalisation du raisonnement en intelligence artificielle est, on l'a dit, le besoin de langages de représentation compacte pour la spécification des données d'un processus. Ce besoin est en partie une conséquence de la recherche de langages génériques et expressifs : la recherche opérationnelle (en tout cas son sous-ensemble "optimisation combinatoire") traite elle aussi de problèmes combinatoires complexes mais contourne le besoin de langages de représentation sophistiqués en renonçant à la généricité et à l'expressivité, c'est-à-dire en fixant une fois pour toutes le format du problème, et en développant des méthodes de résolution *spécifiques* pour ce format : ainsi, la *rigidité*, la *spécificité* et la *faible expressivité* facilitent la résolution. La communauté de l'IA, en choisissant l'expressivité plutôt que la spécificité, se heurte à des problèmes d'une complexité souvent rédhibitoire.

Certes, la haute complexité computationnelle des problèmes est en partie due au fait que l'input est exprimé dans un langage compact (comme on l'a bien vu au chapitre 6 en ce qui concerne les problèmes de planification) : les représentations compactes étant, au mieux, exponentiellement plus succintes que les représentations explicites, les problèmes associés sont, au pire, exponentiellement plus complexes (voir par exemple le chapitre 20 de [188]). Peut-on dire pour autant que les fortes complexités identifiées

sont artificielles, arbitraires ? Certainement pas, parce que la façon naturelle, pour l'agent humain, d'exprimer les données d'un problème (croyances et préférences) est naturellement compacte (et exprimée dans un langage non explicite) : d'où l'intérêt de l'utilisation de logiques, qui, par leur expressivité, sont certainement plus proches du langage naturel et nécessitent un moindre effort de traduction (que les langages plus rigides de la recherche opérationnelle). Il reste cependant à évaluer plus précisément la pertinence cognitive des langages de représentation compacte et à développer des techniques élaborées d'élicitation des préférences (et, dans une moindre mesure, des croyances) permettant de traduire dans ces langages cibles ce que souhaite (ou ce que croit) l'agent.

Parlons à présent de résolution. Les problèmes de représentation des connaissances et de formalisation du raisonnement , dont l'input est (presque toujours) exprimé dans des langages de représentation compacte, sont dans des classes de complexité élevées, très souvent au-delà de NP et de coNP : les problèmes de raisonnement non-monotone, de révision, de mise-à-jour, de programmation logique avec négation par échec, de causalité, de diagnostic, se situent souvent au second niveau de la hiérarchie polynomiale (parfois au troisième) ; c'est aussi le cas pour certains problèmes de planification. Ce phénomène (l'importance du second niveau) semble assez spécifique à l'IA[3]. Les premiers problèmes Σ_2^p-complets et Π_2^p-complets en IA ont été identifiés au début des années 90 ([111, 87]) et depuis, les problèmes (pertinents) d'IA complets pour ces classes se comptent par dizaines. Pourquoi cette importance du second niveau ? Sans doute à cause des définitions complexes faisant appel à des tests de cohérence, associés aux processus de minimisation et (dans une moindre mesure) des définitions non-constructives à base de points fixes. D'autres problèmes (moins nombreux, toutefois), se situent plus haut, souvent au niveau de la PSPACE-complétude (en particulier en planification – encore plus haut parfois, voir la section 6.2).

Puisque le second niveau de la hiérarchie polynomiale semble si central en représentation des connaissances , le rôle des problèmes canoniques des classes correspondantes, à savoir $\text{QBF}_{2,\exists}$ et $\text{QBF}_{2,\forall}$, n'en est que plus évident. Pourtant, peu d'attention a été consacrée *spécifiquement* à ces problèmes : en effet, même si depuis plusieurs annnées, le nombre d'articles consacrés à la résolution de QBF s'accroît rapidement, peu sont consacrés à $\text{QBF}_{2,\exists}$ et

[3]En logique, par exemple, on saute vite de NP ou coNP à PSPACE ; en recherche opérationnelle,, on reste souvent au premier niveau de la hiérarchie polynomiale ; etc.

$\text{QBF}_{2,\forall}$; or, il paraît au moins aussi pertinent de développer des outils pour les problèmes Π_2^p- et Σ_2^p-complets que pour les problèmes PSPACE-complets, moins fréquents en IA.

Il y a fort à parier que. vu l'importance de ces classes de complexité en IA, la résolution de problèmes via des "prouveurs QBF" va prendre de l'importance (tout comme SAT pour les problèmes de premier niveau). Une première plateforme de résolution (QUIP) a été développée [86] et s'est montrée efficace en ce qui concerne des problèmes de raisonnement non-monotone ou d'abduction. Il reste maintenant à se concentrer sur le type de tâches qu'on veut faire résoudre à des solveurs QBF (voir Section 6.3).

On a coutume, surtout en recherche opérationnelle et en théorie des graphes, de qualifier les problèmes NP-difficiles ou coNP-difficiles d'"intraitables". Si tel est le cas, comment, alors peut-on imaginer vouloir résoudre des problèmes bien plus complexes ? (Il semble en effet que les problèmes du premier niveau – jusqu'à Δ_2^{P} – sont considérés comme "faciles" par une partie de la communauté, ce qui ne manque pas de faire hurler une autre partie, celle des "sateux". Ceci dit, ces problèmes existent bel et bien, ce ne sont pas de simples élucubrations théoriques de chercheurs en mal de problèmes représentatifs. Il faut donc bien songer à les résoudre. Puisqu'on sait que ce ne sera pas simple (euphémisme s'il en est), il faut pallier cette complexité d'une façon ou d'une autre. Je vois trois grandes types d'approches :

- la *simplification* du problème : on ignore des paramètres, on hiérarchise (en ignorant certaines variables), voire, on retourne plus ou moins vers un format figé (du type de ceux qu'on rencontre en recherche opérationnelle) et on résoud alors quelque chose qui peut être assez loin du véritable problème.
- le *prétraitement* (ou compilation) du problème : on réécrit la base de croyances ou de préférences initiales de façon à minimiser les ressources computationnelles en-ligne.
- la *résolution approchée* : on abandonne l'optimalité, et l'on de tourne vers algorithmes incomplets itératifs et/ou randomisés et/ou anytime, convergeant ou non vers une solution optimale.

7.5 Raisonnement et décision : au-delà de l'IA

Une des limites de l'ensemble des travaux auxquels j'ai contribué est qu'il sort peu de l'intelligence artificielle. Je ne voudrais cependant pas lais-

ser l'impression que je considère que le raisonnement et la décision sont de son ressort exclusif, et pour cette raison je mentionne ici très brièvement exclusif d'autres disciplines, qui sont souvent en-dehors de mon champ de compétences (malheureusement). J'exprime en fait un léger regret de ne pas m'être davantage investi plus tôt dans quelques-unes de ces disciplines (mais un vif enthousiasme à la perspective de m'y plonger).

Recherche opérationnelle Elle offre une panoplie de méthodes de résolution algorithmique de certains des problèmes posés en sciences du raisonnement et de la décision ; notamment, le domaine de la *programmation à base de contraintes* permet de représenter des problèmes combinatoires avec des données à la fois symboliques et numériques dans un langage déclaratif, et de les résoudre au moyen d'algorithmes d'optimisation.

Informatique théorique Bien des sujets relevant de l'informatique théorique ont beaucoup à voir avec le raisonnement et la décision : la *complexité computationnelle* bien entendu (je n'y reviens pas – voir la section 7.4), mais aussi la *théorie de la programmation* : un programme n'est-il avant tout pas une politique d'action ? Cette remarque suggère qu'il est pertinent de se pencher sur des notions qui sont à la conception d'agents intelligents et à la robotique cognitive ce que l'informatique théorique est à la programmation classique. Par exemple, la notion de calculabilité mène naturellement à celle de *réalisabilité* ("achievability") – voir [172], les machines de Turing sont augmentées de capacités d'interaction avec le monde grâce à des capteurs et des effecteurs, pour devenir des robots abstraits ; par ailleurs, des travaux sont en cours pour donner une sémantique opérationnelle au langage *Golog*.

Automatique (qui, dans son sens large, inclut la commande de systèmes, la productique et la robotique). Comme l'intelligence artificielle, l'automatique s'intéresse au contrôle d'un agent sur un système (et d'ailleurs, je me réfère souvent dans ce document à des notions qui sont classiques en automatique : contrôlabilité, observabilité, feedback, politique d'action...). Toutefois, l'automatique s'intéresse surtout á modéliser le système lui-même, plutôt que l'agent.

Économie S'il est un domaine où la prise de décision est un sujet fondamental, c'est bien l'économie mathématique : la *théorie de la décision* s'occupe avant tout de construire des systèmes axiomatiques caractérisant le comportement rationnel d'un agent individuel, la *théorie des jeux* formalise le comportement d'un agent rationnel en présence d'interactions stratégiques (de coopération, de concurrence, de commu-

nication ...) avec d'autres agents rationnels ; la *théorie du choix social* formalise les processus de décision collective (choix d'investissements publics, problèmes de partage, théorie du vote ...). Une branche récente de l'économie, à savoir l'*économie cognitive* (voir [221]) s'occupe en outre de raisonnement, en particulier de révision des croyances et d'apprentissage. L'économie expérimentale s'occupe quant à elle de tester sur des sujets humains le bien-fondé des constructions théoriques développées par les économistes formels.

Logique Comme l'écrit à intervalles réguliers un assez célèbre logicien français, ce n'est pas parce que les chercheurs en IA utilisent des notions et outils issus de la logique mathématique qu'ils sont des logiciens. Je ne sais pas si c'est vrai, mais toujours est-il que l'IA a su importer de la logique mathématique à peu près tout ce qu'il était pertinent d'importer, à l'exception, peut-être, des logiques des ressources consommables (à l'instar de la logique linéaire), qui sont peu présentes dans la littérature de l'IA, alors que les ressources consommables sont dans presque tous les problèmes réels.

Psychologie cognitive Là encore, le raisonnement et la prise de décision sont des sujets centraux ; comme en économie cognitive, on s'intéresse ici à des sujets humains (ou parfois animaux – est-ce alors encore de la psychologie ?). Les processus de raisonnement et de choix sont étudiés de façon plus "microscopique" que ne le fait, par exemple, l'économie expérimentale. Peut-on aller encore plus en amont dans l'étude du raisonnement et de la décision ? (Sans doute, avec les neurosciences).

Linguistique La sémantique des langues naturelles recèle bien des informations sur les aspects profonds des fonctions de raisonnement et de prise de décision. (Peut-on raisonner sans langage ? Le processus de raisonnement est-il dépendant du langage ? Autant de sujets délicats et difficiles sur lesquels je ne me souhaite pas m'aventurer). Le projet *Langue, Raisonnement et Calcul*, mené pendant de nombreuses années par M. Borillo en collaboration avec une équipe de linguistes, avait justement comme objectif (entre autres) d'identifier les processus de raisonnement implicites sous-jacents à la production d'énoncés en langage naturel.

7.6 Quelques perspectives (en vrac)

Vers des langages d'action vraiment expressifs

La représentation des actions est, on l'a dit plusieurs fois, un thème de recherche important en intelligence artificielle (qui devient crucial lorsqu'il s'agit de représenter des processus de décision de manière concise – et de les résoudre). Jusqu'à présent, ces langages se sont essentiellement attachés à traiter le problème du décor, la causalité, la concurrence et certaines formes de non-déterminisme. Il leur manque encore cependant bien des choses pour qu'ils soient *vraiment* expressifs et facilement utilisables pour représenter des problèmes concrets.

- *effets incertains* : si le non-déterminisme "tout-ou-rien" est plus ou moins bien traité dans les langages d'action, les effets à incertitude graduelle (probabilistes par exemple) ont reçu bien moins d'attention. Les réseaux bayésiens temporisés (2TBN) [59] et les opérateurs STRIPS probabilistes (PSOs), ainsi que l'article récent [8] visant à intégrer des effets probabilistes dans la calcul des situations, me semblent insuffisants. J'attends bien plus d'expressivité des développements futurs concernant la révision et la mise-à-jour des croyances opérant sur des représentations compactes de fonctions de transition (probabilistes ou non).
- *ressources consommables* : il est pour l'instant malaisé de les représenter de façon efficace et intuitive. Pourtant, développer une logique propositionnelle pour les ressources consommables, associée à des opérateurs de révision et surtout de mise-à-jour, ne semble pas si difficile. Une logique propositionnelle dont la sémantique serait décrite en termes de valuations $V : VAR \rightarrow I\!N$ associant à chaque variable propositionnelles son *nombre d'occurrences* ferait probablement l'affaire. Développer de telles logiques consiste probablement à faire un pas vers la logique linéaire [109] mais en se restreignant à des fragments suffisamment simples pour éviter de tomber dans des complications excessives. Le langage pour les enchères combinatoires esquissé dans [33] peut être vu comme un premier pas dans cette direction.
- *effets épistémiques* : bien qu'ils aient fait l'objet de quelque attention, notamment dans le cadre du calcul des situations (et de nos travaux sur la logique dynamico-épistémique, voir le chapitre 5), il reste beaucoup à faire (en particulier si l'on considère des processus à plusieurs agents – voir à ce sujet le point 7.6 de cette Section.

Il resterait enfin à rassembler les morceaux et à faire une étude comparative des différents formalismes, notamment sur le plan de leur complexité (dont une première étude figure dans [153]) et de leur efficacité spatiale.

Programmation à base de croyances

Cette problématique, qui apparaît pour la première fois dans [91], n'a pas reçu toute l'attention qu'elle devrait. Il est fondamental que les politiques d'action d'agents rationnels soient fonction des croyances que ces agents entretiennent (voir le chapitre 6) et que le choix d'une action à entreprendre puisse donc être le fruit d'une "réflexion" en cours de processus (réflexion qui peut prendre la forme de la résolution d'un problème NP-difficile ou coNP-difficile). L'importance de cette problèmatique est claire dès que l'on remarque que de telles politiques d'action jouent le rôle pour la "robotique cognitive" (ou encore les "agents intelligents autonomes", selon la communauté dont on fait partie) que les programmes classiques jouent pour les ordinateurs classiques. Ce qui ne manque pas de soulever plusieurs questions fort intéressantes et à peine défrichées. Notamment, pourquoi ne pas développer une théorie de la programmation d'agents qui serait à la robotique cognitive et aux agents autonomes intelligents ce que l'informatique théorique est à la programmation classique ? C'est certes un programme de recherche très ambitieux. Des premiers pas ont très récemment été fait dans cette direction, notamment [172] qui définissent une notion de réalisabilité effective qui est aux robots cognitifs ce que la décidabilité est aux ordinateurs, et [176] qui propose une sémantique à la Hoare pour le langage Golog. Parmi les sujets d'étude que cette thématique fait naître, on peut citer une future théorie de la complexité "réalisationnelle" qui mesure non seulement les ressources computationnelles (temps et espace) nécessaires à la résolution d'un problème mais également le temps de *réalisation* de la tâche. Le rôle du prétraitement et des compromis entre phase hors-ligne et phase en-ligne est ici crucial. Par exemple, certains problèmes peuvent être résolus en un nombre polynomial d'actions effectives mais la taille de la politique est alors exponentiellement grande (à cause de la "largeur" de cette dernière) ; par contre, si l'on autorise la résolution de tâches computationnellement complexes (NP- ou coNP-difficiles) en ligne, alors la taille de la politique est en général bien inférieure (elle peut redevenir polynomiale) mais au prix d'un nombre de tâches exponentiel à réaliser en ligne.

Le rôle de la logique épistémico-dynamique EDL que nous développons depuis plusieurs années pourrait alors jouer pour la programmation d'agents

intelligents celui que la logique dynamique PDL joue pour la programmation classique.

Changement des croyances en environnement à plusieurs agents

On a déjà dit que la révision des croyances était cruciale dans un environnement où les agents communiquent et interagissent (donc particulièrement en théorie des jeux [222]). Le changement des croyances en environnement multi-agents est un vaste champ de recherche. Plusieurs problèmes différents se posent :

- la prise en compte d'un *message émis par un agent* conduit à une révision des croyances mutuelles (voir [212] pour le cas des messages émis publiquement[4]).
- la prise en compte d'une *action* effectuée par un agent devrait conduire cette fois à une mise-à-jour des croyances mutuelles.
- la prise en compte d'une *question* posée par un agent A à un agent B (ou à la collectivité) renseigne B (partiellement) sur les *buts épistémiques* de A.
- l'observation du comportement d'un agent *sans interaction* (donc a fortiori sans communication) est un problème plus simple, pour lequel on peut songer à utiliser des critères de *pertinence dans l'interprétation d'actions*, qui permettent à l'agent d'inférer (de façon non-monotone) de nouvelles croyances sur les croyances, les buts et les intentions de l'agent observé (voir [143] pour un travail préliminaire).

Les applications de tels sujets de recherche vont du dialogue homme-machine aux protocoles cryptographiques, en passant par la reconnaissance de plans.

Choix social computationnel

Il y a là aussi de nombreux sujets de recherche prometteurs.

D'abord, le besoin de prendre des décisions collectives portant sur plusieurs variables intercorrélées, déjà mis en évidence par les chercheurs en théorie du choix social ("décisions sur des domaines cartésiens"), met en avant des difficultés computationnelles auxquelles l'intelligence artificielle et la recherche opérationnelle sont en mesure d'apporter quelques réponses, notamment grâce aux langages de représentation compacte et aux algorithmes

[4]La prise en compte de messages privés dans ce cadre ne devrait pas d'ailleurs pas poser de problème.

d'optimisation combinatoire. Le projet "Galileo" dans lequel je suis impliqué (avec Sylvie Coste-Marquis, Paolo Liberatore, Pierre Marquis et Marco Schaerf) porte justement sur l'étude de l'efficacité spatiale des langages de représentation de préférences [51, 52].

Le point qui précède suppose que toutes les variables doivent être affectées à une valeur simultanément, sans que les agents puissent intervenir entre le moment où ils ont exprimé leurs préférences et le moment où la décision collective est entièrement déterminée. Relâcher cette hypothèse conduit à définir des procédures de vote "locales" : on peut par exemple choisir d'affecter dans un premier temps certaines variables, soit en raison de leur importance, de leur pertinence (on ne décide pas du contenu des valises avant de s'être entendu sur un lieu de vacances, même si chaque agent possède bel et bien une relation de préférence statique portant sur la totalité des variables concernées), et laisser ensuite les agents réexprimer (avec ou sans négociation préalable) leurs préférences sur les variables restant à affecter. Une autre façon d'aider à la négociation consiste à identifier, localiser et quantifier les conflits entre agents et à les faire négocier entre eux en leur expliquant au mieux les tenants et les aboutissants des conflits. Ce dernier point soulève la question de la mesure de la contradiction d'une base de préférence. Je contribue à deux travaux assez distincts en cours ayant trait à cette question des conflits :

- avec Pierre Marquis et Sébastien Konieczny, nous travaillons sur la définition de degrés d'ignorance et de contradiction de bases de croyances ou de préférences (mono-agent ou multi-agents) ; ces notions assez générales sont relativement indépendantes de la logique dans laquelle ces croyances ou ces préférences sont représentées [135].
- avec Leendert van der Torre et Emil Weydert, nous travaillons sur l'élaboration d'une *typologie* des conflits entre les multiples désirs d'un agent isolé. En particulier, certains conflits sont dûs à une inadéquation du monde réel avec les désirs de l'agent (*utopie*), alors que d'autres sont des conflits apparents dûs à la présence implicite de considération de normalité, d'*incertitude cachée* (l'agent exprimant ses désirs en considérant les mondes les plus normaux possibles) [162].

En amont des problèmes précédents se pose le problème difficile de l'*élicitation* des préférences des agents, à savoir leur obtention à l'aide par exemple d'un questionnaire interactif, d'un processus d'apprentissage automatique ou de classification. Il y a besoin, pour ce faire, de compétences multiples, avant tout en économie et psychologie cognitives, et dans une moindre mesure en

interaction homme-machine et en recherche opérationnelle[5].

[5]Je souhaite en particulier me mettre à l'étude de la *théorie des questionnaires* [191] ; cette théorie, un peu oubliée, est fondée sur la théorie des graphes et vise à la conception de plans optimaux de prise d'information.

Bibliographie

[1] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change : partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50 :510–530, 1985.

[2] L. Amgoud. *Contribution à l'étude des préférences dans le raisonnement argumentatif.* PhD thesis, Université Paul Sabatier, 1999.

[3] N. Asher and J. Lang. When nonmonotonicity comes from distances. In B. Nebel and L. Dreschler-Fischer, editors, *KI-94 : Advances in Artificial Intelligence, 18th German annual conference on Artificial intelligence*, Saarbrücken, Germany, September 1994. Springer-Verlag.

[4] F. Bacchus and A. Grove. Utility independence in a qualitative decision theory. In *Proceedings of KR'96*, pages 542–552, 1996.

[5] Ch. Bäckström. *Computational complexity of reasoning about plans.* PhD thesis, Universitet i Linköping, Suède, 1992.

[6] Ch. Bäckström. Expressive equivalence of planning formalisms. *Artificial Intelligence*, 76(1–2) :17–34, 1995.

[7] C. Baral, V. Kreinovich, and R. Trejo. Computational complexity of planning and approximate planning in presence of incompleteness. *Artificial Intelligence Journal*, 122 :241–267, 2000.

[8] C. Baral, N. Tran, and L.-C. Tuan. Reasoning about actions in a probabilistic setting. In *Proceedings of AAAI-02*, pages 507–512, 2002.

[9] J.J. Bartholdi and J.B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4) :341–354, 1991.

[10] J.J. Bartholdi, C.A. Tovey, and M.A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3) :227–241, 1989.

[11] N. Belnap. *A useful four-valued logic*, pages 8–37. Reidel, 1977.

[12] N. Ben Amor, S. Benferhat, D. Dubois, H. Geffner, and H. Prade. Independence in qualitative uncertainty frameworks. In *Proceedings of KR2000*, pages 235–246, 2000.

[13] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax- based entailment. In Ruzena Bajcsy, editor, *Proceedings of the* 13^{th} *International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 640–645. Morgan-Kaufmann, 1993.

[14] S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In W. Swartout B. Nebel, C. Rich, editor, *Proceedings of KR'92*, pages 673–684, October 1992.

[15] S. Benferhat, D. Dubois, and H. Prade. How to infer from inconsistent beliefs without revising ?. In *Proceedings of IJCAI-95*, pages 1449–1455, 1995.

[16] S. Benferhat, S. Konieczny, O. Papini, and R. Pino-Pérez. Révision itérée basée sur la primauté forte de la nouvelle information. In *Journées Nationales sur les Modèles de Raisonnement (JNMR'99)*, http ://www.irit.fr/GDR-i3-ModRais/articlesJNMR.html, 1999.

[17] P. Bertoli, A. Cimatti, M . Roveri, and P. Traverso. Planning in nondeterministic domains under partial observability via symbolic model checking. In *Proceedings of IJCAI-01*, pages 473–478, 2001.

[18] Ph. Besnard and A. Hunter. Quasi-classical logic : non-trivializable classical reasoning from inconsistent information. In *Proceedings of ECSQARU'95*, pages 44–51, 1995.

[19] Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128 :203–235, 2001.

[20] Ph. Besnard and J. Lang. Possibility and necessity functions over non-classical logics. In *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'94)*, pages 69–76, 1994.

[21] Ph. Besnard and J. Lang. *Frontiers of Paraconsistent Logic*, chapter Graded paraconsistency – Reasoning with inconsistent and uncertain knowledge (C. Mortenson, G. Priest, J.P. Van Bendegem, eds.). King's College Publications, 2000. Preliminary version in the Proceedings of the First World Congress on Paraconsistency, Gent, Belgium, 1997.

[22] Ph. Besnard and T. Schaub. Signed systems for paraconsistent reasoning. *Journal of Automated Reasoning*, 20 :191–213, 1998.

[23] Philippe Besnard and Els Laenens. A knowledge representation perspective : logics for paraconsistent reasoning. *Int. J. of Intelligent Systems*, 9 :153–168, 1994.

[24] I. Bloch and A. Hunter, editors. *Fusion : General Concepts and Characteristics*, volume 16 (10) of *International Journal of Intelligent Systems*. Wiley, 2001. Special Issue on Data and Knowledge Fusion.

[25] I. Bloch and J. Lang. Towards mathematical "morpho-logics". In *Technologies for Contructing Intelligent Systems*, volume 2, pages 367–380. Springer-Verlag, 2000. Version courte dans "Proceedings of the 8th International Conf. on Information Processing and the Management of Uncertainty in Knowledge-Based Systems (IPMU'2000).

[26] C. Boutilier. Toward a logic for qualitative decision theory. In *KR-94*, pages 75–86, 1994.

[27] C. Boutilier. Generalized update : belief change in dynamic settings. In *Proceedings of the* 14^{th} *International Joint Conference on Artificial Intelligence (IJCAI'95)*, volume 2, pages 1550–1556, 1995.

[28] C. Boutilier. A unified model of qualitative belief change : a dynamical systems perspective. *Artificial Intelligence*, 1-2 :281–316, 1998.

[29] C. Boutilier, F. Bacchus, and R. Brafman. UCP-networks : a directed graphical representation of conditional utilities. In *Proceedings of the 17th Conf. on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 56–64, 2001.

[30] C. Boutilier, R. Brafman, H. Hoos, and D. Poole. Reasoning with conditional *ceteris paribus* statements. In *Proceedings of the 15th Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, pages 71–80, 1999.

[31] C. Boutilier, Th. Dean, and S. Hanks. Decision-theoretic planning : structural assumptions and computational leverage. *Journal of Artificial Intelligence research*, 11 :1–94, 1999.

[32] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proceedings of IJCAI-95*, 1995.

[33] C. Boutilier and H. Hoos. Bidding languages for combinatorial auctions. In *Proceedings of IJCAI-2001*, pages 1211–1217, 2001.

[34] G. Brewka. Preferred subtheories : an extended logical framework for default reasoning. In *Proceedings of the* 11^{th} *International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1043–1048, Detroit (MI), 1989.

[35] G. Brewka and J. Hertzberg. How to do things with worlds : on formalizing actions and plans. *J. of Logic and Computation*, 3(5) :517–532, 1993.

[36] T. Bylander. The computational complexity of propositional STRIPSplanning. *Artificial Intelligence*, 69 :161–204, 1994.

[37] M. Cadoli, F.M. Donini, M. Schaerf, and R. Silvestri. On compact representations of propositional circumscription. *Theoretical Computer Science*, 182 :183–202, 1997.

[38] Th. Castell and H. Fargier. Between SAT and CSP : Propositional satisfaction problems and clausal csps. In *Proceedings of ECAI-98*, pages 214–218, 1998.

[39] M. A. Castilho, O. Gasquet, and A. Herzig. Formalizing action and change in modal logic I : the frame problem. *J. of Logic and Computation*, 9(5), 1999.

[40] C. Cayrol. Un modèle logique pour le raisonnement révisable. *Revue d'Intelligence Artificielle*, 6 :255–284, 1992.

[41] C. Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In *Proceedings of IJCAI-95*, pages 1443–1448, 1995.

[42] C. Cayrol, M. Lagasquie-Schiex, and Th. Schiex. Nonmonotonic reasoning : from complexity to algorithms. *Annals of Mathematics and Artificial Intelligence*, 22(3-4) :207–236, 1998.

[43] C. Cayrol and M.C. Lagasquie-Schiex. Classification de relations d'inference non-monotone : la prudence et les propriétés de déduction. Technical report, Institut de Recherche en Informatique de Toulouse, 1994.

[44] L. Cholvy and Ch. Garion. An attempt to adapt a logic of conditional preferences for reasoning with contrary-to-duties. In *Proceedings of the 5th International Workshop on Deontic Logic In Computer Science (DEON'00)*, pages 125–145, 2000.

[45] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of AAAI-02*, pages 314–319. Morgan Kaufmann, 2002.

[46] M.-O. Cordier and J. Lang. Linking transition-based update and base revision. In *Proceedings of ECSQARU'95*, pages 133–141, 1995.

[47] M.-O. Cordier and P. Siegel. Prioritized transitions for updates. In *Proceedings of ECSQARU'95*, pages 142–150, 1995.

[48] C. Cossart and C. Tessier. Filtering vs. revision and update : let us debate ! In *Proceedings of ECSQARU'99*, pages 116–127, 1999.

[49] C. Da Costa. *Planification d'actions en environnement incertain : une approche basée sur la théorie des possibilités*. PhD thesis, Université Paul Sabatier, 1998.

[50] S. Coste-Marquis, H. Fargier, J. Lang, D. Le Berre, and P. Marquis. Formules booléennes quantifiées : problèmes et algorithmes. In *Actes de RFIA-2002*, pages 289–298, 2002.

[51] S. Coste-Marquis, J. Lang, P. Liberatore, and P. Marquis. Expressive power and succinctness of propositional languages for preference representation. Manuscript, 2003.

[52] S. Coste-Marquis, J. Lang, P. Liberatore, P. Marquis, and M. Schaerf. Complexity results for preference languages. Manuscript, 2003.

[53] S. Coste-Marquis and P. Marquis. Complexity results for paraconsistent inference relations. In *Proceedings of KR2002*, pages 61–71, 2002.

[54] C. da Costa, F. Garcia, J. Lang, and R. Martin-Clouaire. Possibilistic planning : Representation and complexity. In *Proceedings of ECP'97*, Lectures Notes in Artificial Intelligence 1348, pages 143–155. Springer-Verlag, 1997.

[55] N.C.A. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15 :497–510, 1974.

[56] A. Darwiche. A logical notion of conditional independence : properties and applications. *Artificial Intelligence*, 97(1 & 2) :45–82, 1997.

[57] A. Darwiche and J. Pearl. Symbolic causal networks. In *Proceedings of AAAI'94*, pages 238–244, 1994.

[58] B. de Finetti. La prévision : ses lois logiques et ses sources subjectives. *Ann. Inst. H. Poincaré*, 7 :1–68, 1937.

[59] T. Dean and K. Kanazawa. Persistence and probabilistic projection. In *Proceedings of IEEE Trans. on Systems, Man and Cybernetics*, volume 19(3), pages 574–585, 1989.

[60] A. del Val. On the relation between the coherence and the foundation theories of belief revision. In *Proceedings of AAAI-94*, pages 909–914, 1994.

[61] P. Doherty, W. Lukascewicz, and E. Madalinska-Bugaj. The PMA and relativizing change for action update. In *Proceedings of KR'98*, pages 259–269, 1998.

[62] C. Domshlak. *Modelling and reasoning about preferences with CP-nets*. PhD thesis, Ben-Gurion University, 2002.

[63] C. Domshlak and R. Brafman. Cp-nets : reasoning and consistency testing. In *Proceedings of KR2002*, pages 121–132, 2002.

[64] J. Doyle and M. P. Wellman. Preferential semantics for goals. In *Proceedings of AAAI-91*, pages 698–703, 1991.

[65] J. Doyle, M. P. Wellman, and Y. Shoham. A logic of relative desire (preliminary report). In *Proceedings of the 6th Intern. Symposium on Methodologies for Intelligent Systems*, pages 16–31, 1991.

[66] D. Driankov and J. Lang. Possibilistic decreasing persistence. In *Proceedings of UAI'93*, 1993.

[67] D. Dubois. Belief structures, possibility theory and decomposable confidence measures on finite sets. *Computers and Artificial Intelligence*, 5(5) :403–416, 1986.

[68] D. Dubois, F. Dupin de Saint-Cyr, and H. Prade. Updating, transition constraints and possibilistic Markov chains. In *Proceedings of IPMU'94*, pages 826–831, 1994.

[69] D. Dubois, F. Esteva, P. Garcia, L. Godo, and H. Prade. A logical approach to interpolation based on similarity relations. *International Journal on Approximate Reasoning*, 16 :235–260, 1997.

[70] D. Dubois, H. Fargier, and P. Perny. On the limits of ordinality in decision making. In *Proceedings of KR-02*, pages 133–144, 2002.

[71] D. Dubois, H. Fargier, and H. Prade. *Fuzzy Sets, Neural Networks and Soft Computing*, chapter Propagation and propagation of flexible constraints. Kluwer Academic Publishers, 1993.

[72] D. Dubois, H. Fargier, and H. Prade. Decision making under ordinal preference and comparative uncertainty. In *Proceedings of UAI-97*, pages 157–164, 1997.

[73] D. Dubois, H. Fargier, and H. Prade. *The ordered weighted averaging operators – Theory and applications*, chapter Beyond aggregation in multicriteria decision : (ordered) weighted min, discri-min, leximin. Kluwer Academic Publishers, 1997.

[74] D. Dubois, J. Lang, and H. Prade. Inconsistency in possibilistic knowledge bases - to live or not live with it. *Fuzzy logic for the management of uncertainty*, pages 335–351, 1992.

[75] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of logic in Artificial*

Intelligence and logic programming, volume 3, pages 439–513. Clarendon Press - Oxford, 1994.

[76] D. Dubois and H. Prade. Epistemic entrenchment and possibilistic logic. *Artificial Intelligence*, pages 223–239, 1991.

[77] D. Dubois and H. Prade. A survey of belief revision and updating rules in various uncertainty models. *International journal of Intelligent Systems*, 9 :61–100, 1994.

[78] D. Dubois and H. Prade. *Conditionals : from Philosophy to Computer Science (G. Crocco, L. Fariñas del Cerro, A. Herzig, eds.)*, chapter Conditional objects, possibility theory and default rules, pages 301–336. Oxford University Press, 1995.

[79] D. Dubois and H. Prade. Possibility theory as a basis for qualitative decision theory. In *Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, pages 1924–1930, 1995.

[80] D. Dubois, H. Prade, and R. Sabbadin. Qualitative decision theory with sugeno integrals. In *Proceedings of UAI'98*, pages 121–128, 1998.

[81] F. Dupin de Saint-Cyr. Rapport de DEA. Université Paul Sabatier, 1993.

[82] F. Dupin de Saint-Cyr and J. Lang. Reasoning about unpredicted change and explicit time. In Proceedings of ECSQARU'97, 1997.

[83] F. Dupin de Saint-Cyr and J. Lang. Belief extrapolation (or how to reason about observations and unpredicted change). In *Proceedings of KR2002*, pages 97–508, 2002.

[84] F. Dupin de Saint-Cyr, J. Lang, and T. Schiex. Gestion de l'inconsistance dans les bases de connaissances : une approche syntaxique basée sur la logique des pénalités. In *Actes de RFIA'94*, 1994.

[85] F. Dupin de Saint-Cyr, J. Lang, and T. Schiex. Penalty logic and its link with Dempster-Shafer theory. In *Proceedings of UAI'94*, pages 204–211. Morgan Kaufmann, 1994.

[86] U. Egly, T. Eiter, H. Tompits, and S. Woltran. Solving advanced reasoning tasks using quantified boolean formulas. In *Proceedings of AAAI-2000*, pages 417–422, 2000.

[87] T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. *Artificial Intelligence*, 57 :227–270, 1992.

[88] T. Eiter and T. Lukasiewicz. Complexity results for default reasoning from conditional knowledge bases. In *Proceedings of KR-2000*, pages 62–73, 2000.

[89] P.J. Fabiani. A new approach in temporal representation of belief for autonomous observation and surveillance systems. In *Proceedings of the* 11^{th} *European Conference on Artificial Intelligence (ECAI'94)*, pages 391–395, 1994.

[90] R. Fagin and J. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34 :39–76, 1988.

[91] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[92] H. Fargier. *Problèmes de satisfaction de contraintes flexibles ; application à l'ordonnancement de production*. PhD thesis, Université Paul Sabatier, 1994.

[93] H. Fargier, J. Lang, M. Lemaitre, and G. Verfaillie. Partage équitable de ressources communes – un modèle général et son application au partage de ressources satellitaires. En soumission, 2002.

[94] H. Fargier, J. Lang, and P. Marquis. Propositional logic and one-stage decision making. In *Proceedings of KR'2000*, pages 445–456, 2000.

[95] H. Fargier, J. Lang, R. Martin-Clouaire, and Th. Schiex. A constraint satisfaction framework for decision under uncertainty. In Morgan Kaufmann, editor, *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, pages 167–174, 1995.

[96] H. Fargier, J. Lang, R. Martin-Clouaire, and Th. Schiex. Traitement de problèmes de décision sous incertitude par des problèmes de satisfaction de contraintes. *Revue d'Intelligence Artificielle*, 11(3) :375–378, 1997.

[97] H. Fargier, J. Lang, and R. Sabbadin. Toward qualitative approaches to multi-stage decision making. In *Proceedings of the 6th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, 1996.

[98] H. Fargier, J. Lang, and Th. Schiex. Selecting preferred solutions in fuzzy constraint satisfaction problems. In *Proceedings of the 1st European Conference on Fuzzy Information Technologies*, pages 1128–1134, 1993.

[99] H. Fargier, J. Lang, and Th. Schiex. Mixed constraint satisfaction : a framework for decision under incomplete knowledge. In *Proceedings of AAAI'96*, pages 175–180, 1996.

[100] H. Fargier and P. Perny. Qualitative decision models under uncertainty without the commensurability assumption. In *Proceedings of UAI'99*, pages 188–195, 1999.

[101] L. Fariñas del Cerro and A. Herzig. Belief change and dependence. In *Proceedings of the* 6^{th} *Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'96)*, pages 147–161, 1996.

[102] L. Fariñas del Cerro, A. Herzig, and J. Lang. From ordering-based nonmonotonic reasoning to conditional logics. In *Proceedings of ECAI'92*, pages 38–42, 1992.

[103] L. Fariñas del Cerro, A. Herzig, and J. Lang. Ordering-based nonmonotonic reasoning and conditional logics. *Artificial Intelligence*, 66 :375–393, 1994.

[104] E. Freuder and R. J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, pages 21–71, 1992.

[105] N. Friedman and J.Y. Halpern. On the complexity of conditional logics. In *Proceedings of the* 4^{th} *International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 202–213, 1994.

[106] N. Friedman and J.Y. Halpern. Modeling beliefs in dynamic systems. part 2 : revision and update. *JAIR*, 10 :117–167, 1999.

[107] P. Gärdenfors and D. Makinson. Nonmonotonic inference based on expectations. *Artificial Intelligence*, 65 :197–245, 1994.

[108] I. Gilboa. Expected utility with purely subjective non-additive probabilities. *Journal of Mathematical Economics*, 16 :65–88, 1987.

[109] J.Y. Girard. Linear logic. *TCS 50*, 1987.

[110] M. Goldszmidt and J. Pearl. Rank-based systems : A simple approach to belief revision, belief update, and reasoning about evidence and actions. In *Proceedings of KR'92*, pages 661–672, 1992.

[111] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2 :397–425, 1992.

[112] E. Guéré. PhD thesis, LAAS, 2002.

[113] E. Guéré and R. Alami. A possibilistic planner that deals with nondeterminism and contingency. In *Proceedings of IJCAI-99*, pages 996–1001, 1999.

[114] H. W. Güsgen and J. Hertzberg. Spatial persistence. In *Proceedings of the Ijcai Workshop on Temporal and Spatial Reasoning*, 1993.

[115] V. Ha and P. Haddawy. Towards case-based preference elicitation : similarity measures on preference structures. In *Proceedings of UAI'98*, pages 193–201, 1998.

[116] S.-O. Hansson. A survey on non-prioritized belief revision. *Erkenntnis*, 50(2/3) :413–427, 1999.

[117] P. Haslum and P. Jonsson. Some results on the complexity of planning with incomplete information. In *Proceedings ECP'99,* 308-318, 1999.

[118] S. Hegner. Specification and implementation of programs for updating incomplete information databases. In *Proceedings of the 6th Symposium on Principles of Databases Systems*, pages 146–158, 1987.

[119] A. Heifetz and Ph. Mongin. Probability logic for type spaces. *Games and Economic Behavior*, 35(1/2) :31–53, 2001.

[120] A. Herzig. The PMA revisited. In *Proceedings of KR'96*, pages 40–50, 1996.

[121] A. Herzig, J. Lang, D. Longin, and T. Polacsek. A logic for planning under partial observability. In *Proceedings of AAAI'2000*, pages 768–773, 2000.

[122] A. Herzig, J. Lang, and P. Marquis. Planning as abduction. In *Proceedings of the Workshop on Planning under Incomplete Information, held during Ijcai-2001*, 2001.

[123] A. Herzig, J. Lang, and P. Marquis. Action representation and partially observable planning in epistemic logic. In *Proceedings of IJCAI03*, pages 1067–1072, 2003.

[124] A. Herzig, J. Lang, P. Marquis, and T. Polacsek. Actions, updates, and planning. In *Proceedings of IJCAI'2001*, pages 119–124, 2001.

[125] A. Herzig, J. Lang, and T. Polacsek. A modal logic for epistemic tests. In *Proceedings of ECAI'2000*, pages 553–557, 2001.

[126] A. Herzig and O. Rifi. Propositional belief base update and minimal change. *Artificial Intelligence*, 1999.

[127] R.A. Howard and J.E. Matheson. Influence diagrams. *The Principles and Applications of Decision Analysis*, 2 :720–761, 1984.

[128] J.-Y. Jaffray. Linear utility for belief functions. *Operation Research Letters*, 8 :107–112, 1989.

[129] J.-Y. Jaffray and P. Wakker. Decision making with belief functions : compatibility and incompatibility with the sure thing principle. *Journal of Risk and Uncertainty*, 8, 1994.

[130] S. Kaci. *Fusion de connaissances et de préférences en logique possibiliste.* PhD thesis, Université Paul Sabatier, 2002.

[131] H. Katsuno and A.O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings of KR'91*, pages 387–394, 1991.

[132] H. Katsuno and A.O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52 :263–294, 1991.

[133] S. Konieczny. *Sur la logique du changement : révision et fusion de bases de connaissances.* PhD thesis, Université de Lille I, 1999.

[134] S. Konieczny, J. Lang, and P. Marquis. Distance-based merging : a general framework and some complexity results. In *Proceedings of KR2002*, pages 97–108, 2002.

[135] S. Konieczny, J. Lang, and P. Marquis. Quantifying information and contradiction in propositional logic through test actions. In *Proceedings of IJCAI03*, pages 106–111, 2003.

[136] S. Konieczny and R. Pino-Pérez. On the logic of merging. In *Proc. of KR'98*, pages 488–498, 1998.

[137] S. Konieczny and R. Pino-Pérez. Merging with integrity constraints. In *Proc. of ECSQARU'99*, LNAI 1638, pages 233–244, 1999.

[138] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44 :167–207, 1990.

[139] N. Kushmerick, S. Hanks, and D. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76 :239–286, 1995.

[140] C. Lafage. *Représentation logique de préférences et application à la décision de groupe.* PhD thesis, Université Paul Sabatier, 2001.

[141] C. Lafage and J. Lang. Logical representation of preferences for group decision making. In *Proceedings of KR2000*, pages 457–468, 2000.

[142] C. Lafage and J. Lang. Propositional distances and preference representation. In *Proceedings of ECSQARU-2001*, pages 48–59, 2001.

[143] J. Lang. La pertinence dans le choix et l'interprétation d'actions. En soumission.

[144] J. Lang. *Logique possibiliste : aspects formels, déduction automatique et applications.* PhD thesis, Université Paul Sabatier, 1991.

[145] J. Lang. Syntax-based default reasoning as probabilistic model-based diagnosis. In *Proceedings of UAI'94*, pages 391–398, 1994.

[146] J. Lang. Conditional desires and utilities - an alternative logical approach to qualitative decision theory. In *Procedings of ECAI'96*, pages 318–322, 1996.

[147] J. Lang. Planning to discriminate diagnoses. In *Proceedings DX'97*, pages 135–139, Le Mont St Michel, 1997.

[148] J. Lang. *Handbook of Defeasible Reasoning and Uncertainty Management Systems (D. Gabbay and Ph. Smets, eds.)*, volume 5, chapter Possibilistic logic : complexity and algorithms, pages 179–220. Kluwer Academic Publishers, 2000.

[149] J. Lang. From preference representation to combinatorial vote. In *Proceedings of KR2002*, pages 277–288, 2002.

[150] J. Lang and N. Asher. A nonmonotonic approach to spatial reasoning based on distances and boundaries. In P. Amsili, M. Borillo, and L. Vieu, editors, *Time, Space and Movement (Meaning and Knowledge in the Sensible World)*, Bonas, June 1995. IRIT, Université Paul Sabatier.

[151] J. Lang, P. Liberatore, and P. Marquis. Conditional independence in propositional logic. *Artificial Intelligence*, 2002.

[152] J. Lang, P. Liberatore, and P. Marquis. Propositional independence : Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18 :391–443, 2003.

[153] J. Lang, F. Lin, and P. Marquis. Causal theories of action : a computational core. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1073–1078, 2003.

[154] J. Lang and P. Marquis. Complexity results for planning in partially observable environments : a synthesis, plus some new results. Draft.

[155] J. Lang and P. Marquis. Complexity results for independence and definability in propositional logic. In *Proceedings of the* 6^{th} *International Conference on Knowledge Representation and Reasoning (KR'98)*, Trento, 1998. 356–367.

[156] J. Lang and P. Marquis. Two notions of dependence in propositional logic : controllability and definability. In *Proceedings of AAAI'98*, 1998. 268-273.

[157] J. Lang and P. Marquis. In search of the right extension. In *Proceedings of the* 7^{th} *International Conference on Knowledge Representation and Reasoning (KR'00)*, pages 625–636, Breckenridge (CO), 2000.

[158] J. Lang and P. Marquis. Resolving inconsistencies by variable forgetting. In *Proceedings of KR2002*, pages 239–250, 2000.

[159] J. Lang and P. Marquis. Removing inconsistencies in assumption-based theories through knowledge-gathering actions. *Studia Logica*, 67 :179–214, 2001.

[160] J. Lang, P. Marquis, and M.-A. Williams. Updating epistemic states. In Springer-Verlag, editor, *Lectures Notes in Artificial Intelligence 2256 (Proceedings of 14th Australian Joint Conference on Artificial Intelligence)*, pages 297–308, 2001.

[161] J. Lang, L. van der Torre, and E. Weydert. Utilitarian desires. *International Journal on Autonomous Agents and Multi-Agent Systems*, 5 :329–363, 2002.

[162] J. Lang, L. van der Torre, and E. Weydert. Hidden uncertainty in the logical representation of desires. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 685–690, 2003.

[163] C. Largouët and M.-O. Cordier. Combining observations and expectations : application to the refinement of an image sequence classification. In *Workshop UAI : Fusion of domain knowledge with data for decision support*, Standford, CA, USA, 2000.

[164] N. Laverny. Rapport de DEA, 2002.

[165] D. Lehmann. Belief revision, revised. In *Proceedings of IJCAI'95*, pages 1534–1540, 1995.

[166] M. Lemaîitre, G. Verfaillie, and N. Bataille. Exploiting a common property resource under a fairness constraint : a case study. In *Proceedings of IJCAI-99*, pages 206–211, 1999.

[167] H. Levesque. What is planning in the presence of sensing ? In *Proceedings AAAI'96*, pages 1139–1146, 1996.

[168] D. Lewis. *Counterfactuals*. Harvard University Press, 1973.

[169] P. Liberatore. *Proprocessing of untractable problems – Applications in Artificial Intelligence*. PhD thesis, Università di Roma – La Sapienza, 1998.

[170] P. Liberatore and M. Schaerf. Arbitration (or how to merge knowledge bases). *IEEE Transactions on Knowledge and Data Engineering*, 10(1) :76–90, 1998.

[171] P. Liberatore and M. Schaerf. Brels : a system for the integration of knowledge bases. In *Proceedings of KR2000*, pages 145–152, 2002.

[172] F. Lin and H. Levesque. What robots can do : robot programs and effective achievability. *Artificial Intelligence*, 101 :201–226, 1998.

[173] F. Lin and R. Reiter. Forget it ! In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 154–159, New Orleans, 1994.

[174] M. L. Littman. Probabilistic propositional planning : Representations and complexity. In *Proceedings of AAAI'97*, pages 748–754, 1997.

[175] M. L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9 :1–36, 1998.

[176] Y. Liu. A hoare-style proof system for robot programs. In *Proceedings of AAAI-02*, pages 74–79, 2002.

[177] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially-observable markov decision processes. In *Proceedings of AAAI-99*, pages 541–548, 1999.

[178] D. Makinson. Five faces of minimality. *Studia Logica*, 52 :339–379, 1993.

[179] P. Marquis and N. Porquet. The complexity of entailment in quasi-classical logic. *Journal of Applied Non-classical Logics*, 2002.

[180] D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proceedings of AAAI-91*, pages 634–639, 1991.

[181] S. McIlraith. Generating tests using abduction. In *Proceedings of KR'94*, pages 449–460, 1994.

[182] S. McIlraith and R. Reiter. *Readings in model-based diagnosis*, chapter On tests for hypothetical reasoning. Morgan Kaufman, 1992.

[183] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[184] B. Nebel. *Belief Revision*, chapter Syntax-based approaches to belief revision, pages 52–88. Number 29 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1992.

[185] B. Nebel. Base revision operations and schemes : Semantics, representation, and complexity. In *Proceedings of ECAI'94*, pages 341–345, 1994.

[186] B. Nebel. *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, chapter How hard is it to revise a knowledge base ? Kluwer Academics, 1998.

[187] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28 :71–87, 1986.

[188] Ch. H. Papadimitriou. *Computational complexity*. Addison–Wesley, 1994.

[189] S. Parsons, M. Schut, and M. Wooldridge. Reasoning about intentions in uncertain domains. In *Proceedings of ECSQARU-2001*, pages 84–95, 2001.

[190] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[191] C.-F. Picard. *Graphes et Questionnaires*. 1972.

[192] G. Pinkas. Propositional nonmonotonic reasoning and inconsistency in symmetric neural networks. In *Proceedings of IJCAI'91*, pages 525–530. Morgan-Kaufmann, 1991.

[193] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36 :27–47, 1988.

[194] R. Reiter. *Knowledge in Action : Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press,, 2001.

[195] N. Rescher and R. Manor. On inference from inconsistent premises. *Theory and Decision*, 1 :179–219, 1970.

[196] P.Z. Revesz. On the semantics of arbitration. *Int, Journal of Algebra and Computation*, pages 133–160, 1997.

[197] J. Rintanen. Computational complexity of plan and controller synthesis under partial observability. Manuscript (available on the author's web page), 2003.

[198] R. Sabbadin. *Une approche ordinale de la décision dans l'incertain : axiomatisation, représentation logique et application à la décision séquentielle*. PhD thesis, Université Paul Sabatier, 1998.

[199] R. Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *Proceedings of UAI'99*, pages 567–574, 1999.

[200] R. Sabbadin, H. Fargier, and J. Lang. Toward qualitative approaches to multi-stage decision making. *International Journal on Approximate Reasoning*, 19 :441–471, 1998.

[201] A. Saffiotti. A belief function logic. In *Proceedings of AAAI'92*, pages 642–647, 1992.

[202] E. Sandewall. *Features and Fluents*. Oxford University Press, 1994.

[203] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions,. In *Proceedings of IJCAI'99*, pages 542–547, 1999.

[204] L. Savage. *The Foundations of Statistics*. J. Wiley and Sons, 1954.

[205] R. B. Scherl and H. J. Levesque. The frame problem and knowledge-producing actions. In *Proceedings of AAAI'93*, pages 698–695, 1993.

[206] T. Schiex. Possibilistic constraint satisfaction problems or how to handle soft constraints. In *Proceedings of UAI'92*, pages 268–275, 1992.

[207] T. Schiex, H. Fargier, and G. Verfaillie. Valued constraints satisfaction problems : hard and easy problems. In *Proceedings of IJCAI-95*, pages 631–637, 1995.

[208] D. Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57 :571–587, 1989.

[209] M.C. Schut and M. Wooldridge. Principles of intention reconsideration. In *Proceedings of the Fourth International Conference in Autonomous Agents (Agents 2001)*, 2001.

[210] P. Smets. Probability of deductibility and belief functions. Technical Report 93-5.2, IRIDIA, Bruxelles, BELGIUM, 1993.

[211] W. Spohn. Ordinal conditional functions : a dynamic theory of epistemic states. In William L. Harper and Brian Skyrms, editors, *Causation in Decision, Belief Change and Statistics*, volume 2, pages 105–134. Kluwer Academic Pub., 1988.

[212] J.-M. Tallon, J.-C. Vergnaud, and S. Zamir. Communication and belief revision in mutual belief systems. Personal communication, 2002.

[213] S.W. Tan and J. Pearl. Specification and evaluation of preferences for planning under uncertainty. In *Proceedings of KR-94*, 1994.

[214] R. Thomason. Desires and defaults : a framework for planning with inferred goals. In *Proceedings of KR-2000*, pages 702–713, 2000.

[215] H. Turner. Polynomial-length planning spans the polynomial hierarchy. In *Logics in Artificial Intelligence (Proceedings of JELIA 2002)*, volume 2424 of *Lecture Notes in Artificial Intelligence*, pages 111–124. Springer Verlag, 2002.

[216] L. van der Torre. *Reasoning about obligations*. PhD thesis, Erasmus University Rotterdam, 1997.

[217] L. van der Torre and E. Weydert. Parameters for utilitarian desires in a qualitative decision theory. *Applied Intelligence*, 14(3) :285–302, 2001.

[218] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavious*. Pinceton University Press, 1944.

[219] F. Voorbraak and N. Massios. Decision-theoretic planning for autonomous robot surveillance. *Applied Intelligence*, 3(14) :253–262, 2001.

[220] A. Wald. *Statistical Decision Functions*. Wiley and Sons, 1950.

[221] B. Walliser. *L'économie cognitive*. Odile Jacob, 2001.

[222] B. Walliser. Révision des croyances et théorie des jeux. In B. Livet, editor, *La révision des croyances*, pages 129–144. Hermès, 2002.

[223] A. Weber. Updating propositional formulas. In *Proceedings of the First Conference on Expert Database Systems*, pages 487–500, 1986.

[224] M. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7^{th} National Conference on Artificial Intelligence*, pages 89–93, St. Paul, 1988.

Annexe 1 : Curriculum Vitae

Données personnelles

Né le 25/01/1965
Fonction : Chargé de Recherche au Centre National de la Recherche Scientifique depuis Octobre 1991 ; laboratoire d'affectation : Institut de Recherche en Informatique de Toulouse (UMR 5505).

Titres universitaires

Thèse de l'Université Paul Sabatier (Informatique), 1991 ;
Diplôme d'ingénieur ENSEEIHT filière informatique et maths appliquées, 1987.
DEA (informatique), Université Paul Sabatier, 1987.

Thèmes de recherche

Représentation des connaissances en intelligence artificielle : raisonnement dans l'incertain, raisonnement non-monotone, révision des croyances, raisonnement sur l'action et le changement, diagnostic à base de modèles ;
Décision et intelligence artificielle : logiques pour la représentation de préférences ; décision de groupe, vote combinatoire, partage équitable de ressources ; planification en environnement partiellement observable ;
Complexité et déduction automatique : satisfaisabilité en logique propositionnelle, satisfaction de contraintes, formules booléennes quantifiées, complexité algorithmique.

Tâches administratives

Membre du comité de programme de plusieurs congrès nationaux et internationaux, dont : *International Conference on Uncertainty in Artificial Intelligence (UAI)*, *Reconnaisance des Formes et Intelligence Artificielle (RFIA)*, *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)* ; éditeur d'un numéro spécial de la revue "Applied Intelligence" ; responsable de l'organisation d'un congrès national (RJCIA'98, Toulouse, septembre 1998) ; co-responsable de l'organisation d'un congrès international (KR2002, Toulouse, avril 2002) ; co-responsable de l'organisation d'une école d'été (EURO summer school on "Decision Analysis and Artificial Intelligence", Toulouse, septembre 2001) ; co-responsable d'un groupe de travail au sein du PRC-GDR IA puis au sein du GDR I3 jusqu'en 2001 ; directeur adjoint du GDR I3 depuis novembre 2001 ; co-responsable du comité éditorial de la revue électronique francophone "JEDAI" jusqu'en décembre 2002.

Enseignement

Je donne en moyenne une vingtaine d'heures de cours par an sur des domaines variés (intelligence artificielle, raisonnement dans l'incertain, probabilités, théorie de la décision) à différents niveaux (du premier cycle au DEA).

Collaborations et visites

- Università di Roma "La Sapienza", Roma, Italie (8 jours, octobre-novembre 2002) dans le cadre d'un projet *Galileo* ;
- Hong Kong University of Science and Technology, Hong Kong (2 mois, novembre-décembre 2001) dans le cadre d'un projet *Procore* ;
- Potsdam Universität, Potsdam, Allemagne (2 semaines, octobre 2000 ; 3 mois prévus à l'automne 2003) ;
- University of Newcastle, Australie (2 mois, fevrier-avril 2000) ;
- LADSEB, Centro Nazionale delle Ricerche, Padova, Italie (2 semaines en 1996, un mois en 1999) ;
- Universidad Simon Bolívar, Caracas, Venezuela (2 semaines en 1996 et 2 semaines en 1999) ;
- University of British Columbia, Vancouver, Canada (6 mois en 1994) ;
- Universitetet i Linköping, Suède (3 mois en 1992) ;

– Participation à plusieurs projets européens (DRUMS, DRUMS-II, FUSION).

Langues étrangères

Anglais et (dans une moindre mesure) allemand, italien.

Encadrement de thèses et de stages de DEA

1. **Thèses** :
   – Célia da Costa Pereira (1994-1998) ; encadrementà 50 % ; thèse soutenue en juin 1998 ; plusieurs articles cosignés (cf. liste de publications) ; Célia da Costa Pereira fait maintenant de la recherche appliquée (application des algorithmes génétiques à la prévision et à la décision dans le domaine bancaire) à Milan dans une société "start-up" qui travaille pour une banque (je ne sais plus laquelle !) ;
   – Céline Lafage (1997-2001) ; encadrement à 100 % ; thèse soutenue en juin 2001 ; plusieurs articles cosignés (cf. liste de publications) ; Céline Lafage a obtenu (et refusé) un poste de maître de conférences à Lyon ; son poste d'ATER vient juste d'arriver à terme, elle est depuis lors en recherche d'emploi ;
   – Thomas Polacsek (2000-2003) ; encadrement à 50% ; soutenance prévue pour mai 2003 ; plusieurs articles cosignés (cf. liste de publications) ;
   – Noël Laverny (thèse commencée en 2002) : encadrement à 50%.
2. **Stages de DEA** (DEA "Représentation des Connaissances et Formalisation du Raisonnement") :
   – Florence Dupin de Saint-Cyr (1992-1993) ; encadrement à 50% ; a obtenu son DEA (major) et une allocation de recherche et a poursuivi en thèse avec Didier Dubois et Henri Prade (et a continué à travailler avec moi, mais pas sous ma direction) ; plusieurs publications communes ;
   – Céline Lafage (1996-1997) ; encadrement à 100% ; a obtenu son DEA et une allocation de recherche et a poursuivi en thèse sous ma direction (voir ci-desssus) ;
   – Marc Boyer (1997-1998) ; encadrement à 100% ; a obtenu son DEA mais pas d'allocation de recherche ; a poursuivi en thèse au CERT sous la direction de Laurent Chaudron ;

– Thomas Polacsek (1999-2000) ; encadrement à 50% ; a obtenu son DEA et une allocation de recherche et a poursuivi en thèse sous la direction conjointe d'Andreas Herzig et de moi-même (voir ci-dessus) ;
– Vianney Darmaillacq (2000-2001) ; encadrement à 50% ; n'a pas obtenu son DEA ;
– Noël Laverny (2001-2002) ; encadrement à 50% ; a obtenu son DEA (major à l'écrit) mais ne peut pas avoir d'allocation de recherche en raison de son âge ; continue en thèse sous la direction conjointe de Philippe Balbiani, Andreas Herzig et moi-même.

Annexe 2 : Publications

Je fais une présentation par ordre chronologique, en faisant précéder la publication d'un signe indiquant son type :

- RI : article dans une revue internationale avec comité de lecture ;
- CI : article dans les actes d'un congrès international avec sélection et actes publiés ;
- $CI+$: comme CI mais avec taux de sélection inférieur ou égal à 35 %.
- WI : article dans les actes d'un workshop international avec sélection ou dans les actes d'une conférence internationale avec sélection et actes non publiés ;
- CL : chapitre d'ouvrage collectif ;
- RF : article dans une revue française avec comité de lecture ;
- CF : article dans les actes d'un congrès francophone avec sélection et actes ;
- WF : article dans les actes d'un congrès francophone avec sélection et actes non publiés ;
- M : mémoire ayant permis de soutenir un diplôme ;
- D : divers

1987

CI+ Didier Dubois, Jérôme Lang and Henri Prade, "Theorem proving under uncertainty : a possibility-theory-based approach", *Proc. of the 10th Inter. Joint Conf. on Artificial Intelligence (IJCAI'87)*, Milano, Aug. 87. 984-986.

1988

WI Didier Dubois, Jérôme Lang and Henri Prade, "Advances in automated reasoning using possibilistic logic", *Preprints of the European Workshop on Logical Methods for Artificial Intelligence (JELIA'88)*, Roscoff, France, June 88, 95-99.

1989

WI Didier Dubois, Jérôme Lang and Henri Prade, "Automated reasoning using possibilistic logic : semantics, belief revision and variable certainty weights", *Proc. of the 5th Int. Workshop on Uncertainty in Artificial Intelligence*, Windsor, Ontario, 81-87.

1990

CO Vincent Andrès, Didier Dubois, Jérôme Lang and Henri Prade, " Logique possibiliste et logique floue – Application au raisonnement automatisé et aux systèmes d'information". In *Méthodes Logiques et Systèmes d'Intelligence Artificielle* (L. Iturrioz, A. Duchaussoy, eds.), Hermès, 163-181.

CF Didier Dubois, Jérôme Lang and Henri Prade, "Gestion d'hypothèses en logique possibiliste", *Actes des 10èmes Journées Internationales sur les Systèmes Experts et leurs Applications*, Avignon, May 1990, 299-313.

WI Didier Dubois, Jérôme Lang and Henri Prade, "POSLOG, an inference system based on possibilistic logic", *Proc. of the 1990 North American Fuzzy Association Congress (NAFIPS'90)*, Toronto, 177-180.

CI Didier Dubois, Jérôme Lang and Henri Prade, "Handling uncertain knowledge in an ATMS using possibilistic logic", *Proc. of the 5th Inter. Symposium on Methodologies for Intelligent Systems (ISMIS'90)*, Knoxville, Tennessee, Oct. 1990, North-Holland, pp.252-259.

CI Jérôme Lang, "Semantic evaluation in possibilistic logic", *Proc. of the 3rd Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'90)*, Paris, July 1990, pp. 51-55.

RF Jérôme Lang, "Evaluation sémantique en logique possibiliste", *Revue d'Intelligence Artificielle* 4(4), 1990, Hermès, pp.27-48.

1991

RI Didier Dubois, Jérôme Lang and Henri Prade, "Fuzzy Sets in Approximate Reasoning – Part 2 : Logical Approaches", *Fuzzy Sets and Systems*, 40, 1991, 203-244.

RI Didier Dubois, Jérôme Lang and Henri Prade, "Timed possibilistic logic", *Fundamenta Informaticae, Special Issue on Logics for Artificial Intelligence* (Z. Ras, ed.), XV, 1991, 53-57.

CI+ Didier Dubois, Jérôme Lang and Henri Prade, "Towards possibilistic logic programming", *Proc. Inter. Conf. on Logic Programming* (ICLP'91), Paris, June 1991, MIT Press, 581-595.

M Jérôme Lang, "Logique possibiliste : aspects formels, déduction automatique, et applications". Thèse de Doctorat, Université Paul Sabatier, Toulouse, Janvier 1991.

CI Jérôme Lang, "Possibilistic logic as a logical framework for min-max discrete optimization problems and prioritized constraints", *Fundamentals of Artificial Intelligence Research (FAIR'91)*, Lecture Notes in Computer Science, Vol. 535, Springer Verlag, 1991, 112-116.

CI Jérôme Lang, Didier Dubois and Henri Prade, "A logic of graded possibility and certainty coping with partial inconsistency", *Proc. of the 7th Inter. Conf on Uncertainty in Artificial Intelligence (UAI'91)*, Morgan Kaufmann, 188-196.

1992

CF Salem Benferhat and Jérôme Lang, "Logique possibiliste et application au raisonnement automatisé", *Actes des Deuxièmes Journées Nationales sur les Ensembles Flous et Leurs Applications*, Nîmes, Nov. 1992, 57-64.

CO Didier Dubois, Jérôme Lang and Henri Prade, "Inconsistency in possibilistic knowledge bases : to live or not live with", in *Fuzzy Logic for the Management of Uncertainty* (L. Zadeh, J. Kacprzyk, eds.), Wiley, 1992, 335-352.

CI+ Didier Dubois, Jérôme Lang and Henri Prade, "Dealing with multi-source information in possibilistic logic", *Proc. of the 10th European Conf. on Artificial Intelligence (ECAI'92)*, Vienna, Aug. 1992, 38-42.

CI+ Luis Fariñas del Cerro, Andreas Herzig and Jérôme Lang, "From ordering-based nonmonotonic reasoning to conditional logics", *Proc. of the 10th European Conf. on Artificial Intelligence (ECAI'92)*, Vienna, Aug. 1992, 314-318.

1993

CI+ Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang and Henri Prade, "Inconsistency management and prioritized syntax-based entailment", *Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, Chambéry, Aug. 1993, 640-645.

CI Dimiter Driankov and Jérôme Lang, Possibilistic decreasing persistence, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (UAI'93)*, Washington, Juillet 1993, 469-476.

CI Hélène Fargier and Jérôme Lang, "Uncertainty in constraint satisfaction problems : a probabilistic approach", *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, Granada, Nov. 1993, LNCS vol. 747 (Springer Verlag), 97-104.

CI Hélène Fargier, Jérôme Lang and Thomas Schiex, "Selecting preferred solutions in fuzzy constraint satisfaction problems". *Proceedings of the 1st European Conference on Fuzzy Information Technologies*, Aachen, September 1993.

1994

CI Nicholas Asher and Jérôme Lang, "When nonmonotonicity comes from distances", *Proc. of KI'94* (B. Nebel, L. Fischer-Dressler, eds.), Saarbrücken, Sept. 94, Springer Verlag, *Lectures Notes in Artificial Intelligence*, Vol. 861, 308-318.

CI Philippe Besnard and Jérôme Lang, "Possibility and necessity functions over non-classical logics", *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, July 94 (Morgan Kaufman), 69-76.

CO Didier Dubois, Jérôme Lang and Henri Prade, "Possibilistic logic", in *Handbook of Logic for Artificial Intelligence and Logic Programming*, D.M. Gabbay, C.J. Hopper and J.A. Robinson eds., Clarendon Press, Oxford, Vol.3, 439-513, 1994.

CF Florence Dupin de Saint-Cyr, Jérôme Lang et Thomas Schiex, "Gestion de l'inconsistance dans les bases de connaissances : une approche basée sur la logique des pénalités", *Actes de RFIA'94*.

CI Florence Dupin de Saint-Cyr, Jérôme Lang and Thomas Schiex, "Penalty logic and its links with Dempster-Shafer theory", *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'94)*, Seattle, July 94 (Morgan Kaufmann), 204-211.

RI Luis Fariñas del Cerro, Andreas Herzig and Jérôme Lang, "Ordering-Based Nonmonotonic Reasoning and Conditional Logics", *Artificial Intelligence Journal* 66, 375-393, 1994.

CI Jérôme Lang, "Syntax-based default reasoning as probabilistic model-based diagnosis", *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'94*, Seattle, July 94 (Morgan Kaufman), 391-398.

1995

CI Marie-Odile Cordier and Jérôme Lang, "Linking transition-based update and base revision", in *Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (Proceedings of ECSQARU'95, Fribourg, July 95), Lectures Notes in Artificial Intelligence 946, Springer Verlag, 133-141.

CI Hélène Fargier, Jérôme Lang, Roger Martin-Clouaire and Thomas Schiex, "A constraint satisfaction framework for decision under uncertainty", *Proceedings of the 10th Int. Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, Montréal, Aug. 95, Morgan Kaufmann, 167-174.

WI Jérôme Lang and Nicholas Asher, "A nonmonotonic approach to spatial persistence based on distances and boundaries", *Workshop Notes of the 5th International Workshop on Time, Space and Movement* (P. Amsili, M. Borillo, L. Vieu, eds.), Bonas, June 1995.

1996

WI Célia da Costa Pereira, Frédérick Garcia, Jérôme Lang and Roger Martin-Clouaire, "A possibilistic approach to planning", *Proceedings of the 4th European Congress on Intelligent Techniques and Soft Computing (EUFIT'96)*, Aachen, Sept. 96.

CF Célia da Costa Pereira, Frédérick Garcia, Jérôme Lang and Roger Martin-Clouaire, "Une approche possibiliste de la planification dans l'incertain", *Actes des Quatrièmes Journées sur les Ensembles Flous et leurs applications (LFA'96)*, Nancy, 1996.

CI+ Hélène Fargier, Jérôme Lang and Thomas Schiex, "Mixed constraint satisfaction : a framework for decision under incomplete knowledge", *Proceedings of AAAI'96*, Portland (Oregon), Aug. 96., 175-180.

CI Hélène Fargier, Jérôme Lang and R. Sabbadin, "Toward qualitative approaches to multi-stage decision making", *Proceedings of the 6th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96)*, Granada, July 96.

CF Jérôme Lang, "A propos des tests – une ébauche de classification et de formalisation", Actes du Dixième Congrès *Reconnaissance des formes et intelligence artificielle (RFIA'96)*, Rennes, Janvier 96, 477-356.

CI+ Jérôme Lang, "Conditional desires and utilities – An alternative logical framework for qualitative decision theory", *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96)*, Budapest, Aug. 96, 318-322.

1997

CI+ Célia da Costa Pereira, Frédéric Garcia, Jérôme Lang and Roger Martin-Clouaire, "Possibilistic planning : representation and complexity", in *Recent Advances in Planning* (Sam Steel, Rachid Alami, eds.), Lectures Notes in Artificial Intelligence, Springer Verlag, 1997, 143-155.

RI Célia da Costa Pereira, Frédéric Garcia, Jérôme Lang and Roger Martin-Clouaire, "Planning with graded nondeterministic actions : a possibilistic approach", *International Journal of Intelligent Systems*, 12 (11/12), 1997, 935-962.

CI Florence Dupin de Saint-Cyr and Jérôme Lang, "Reasoning about unpredicted change and explicit time", in *Qualitative and Quantitative Practical Reasoning*, Lectures Notes in Artificial Intelligence 1244, Springer-Verlag, 1997, 223-236.

RF Hélène Fargier, Jérôme Lang, Roger Martin-Clouaire and Thomas Schiex, "Traitement de problèmes de décision sous incertitude par des problèmes de satisfaction de contraintes", *Revue d'Intelligence Artificielle*, Vol. 11, numéro 3, Septembre 1997, Hermès, 375-398.

WI Jérôme Lang, "Planning to discriminate diagnoses", *Proceedings of DX'97*, Le Mont St Michel, Sept. 1997, 135-139.

1998

CI+ Salem Benferhat, Didier Dubois, Jérôme Lang, Henri Prade, Alessandro Saffiotti and P. Smets, "A general approach for inconsistency handling and merging information in prioritized knowledge bases", *Proc. of KR'98*, Trento, June 1998, Morgan Kaufmann, 466-477.

CI+ Jérôme Lang and Pierre Marquis, "Complexity results for independence and definability in propositional logic", *Proc. of KR'98*, Trento, June 1998, Morgan Kaufmann, 356-367.

CI+ Jérôme Lang and Pierre Marquis, "Two forms of dependence in propositional logic : controllability and definability", *Proc. of AAAI'98*, Madison, Aug. 1998, 268-273.

RI Régis Sabbadin, Hélène Fargier and Jérôme Lang, "Toward qualitative approaches to multi-stage decision making", *International Journal on Approximate Reasoning* 19 (1998), 441-471. Preliminary version in *Proceedings of IPMU'96*.

1999

CO Céline Lafage, Jérôme Lang and R. Sabbadin, "A logic of supporters", in *Information, Uncertainty and Fusion* (B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh, eds.), 381-392, Kluwer Academic Publishers, 1999. Preliminary version in *Proceedings of IPMU'98*.

WF Hélène Fargier, Jérôme Lang and Pierre Marquis, "Décision en environnement partiellement observable et résolution de formules booléennes quantifiées", *Actes des 3èmes Journées Nationales sur la Résolution Pratique de Problèmes NP-complets (JNPC'99)*, Lyon, June 1999.

WI Andreas Herzig, Jérôme Lang and Thomas Polacsek, "Knowledge, actions and tests". In Bell J., ed., *IJCAI'99 Workshop on Practical Reasoning and Rationality*.

2000

CO Ph. Besnard and Jérôme Lang, "Graded paraconsistency – Reasoning with inconsistent and uncertain knowledge", in *Frontiers of Paraconsistent Logic*, King's College Publications (co-editors : C. Mortenson, G. Priest, J.P. Van Bendegem), 2000.

CI Isabelle Bloch and Jérôme Lang, "Towards mathematical "morpho-logics" ", *Proceedings of the 8th International Conf. on Information Processing and the Management of Uncertainty in Knowledge-Based Systems (IPMU'2000)*, Madrid, Juillet 2000. Version révisée dans "Technologies for Contructing Intelligent Systems" , Vol. 2, 367-380, Springer-Verlag, 2002.

CI+ Hélène Fargier, Jérôme Lang and Pierre Marquis, "Propositional logic and one-stage decision making", *Proceedings of the 7th International Conference on Knowledge Representation and Reasoning (KR2000)*, 445-456, Morgan Kaufmann.

CF Frédérick Garcia, Jérôme Lang et Abdel-Illah Mouaddib, "Processus décisionnels de Markov", École d'été *Évolutif et Incertain* du GDR I3, Cépaduès.

CI+ Andreas Herzig, Jérôme Lang, Dominique Longin and Thomas Polacsek, "A logic for planning under partial observability", *Proceedings of AAAI'2000*, Austin, Texas, Juillet 2000, 768-773.

CI+ Andreas Herzig, Jérôme Lang and Thomas Polacsek, "A modal logic for epistemic tests", *Proceedings of ECAI'2000*, Berlin, Août 2000, 553-557.

CF Céline Lafage and Jérôme Lang, "Représentation logique de préférences pour la décision de groupe", *Actes du 12ème Congrès sur la reconnaissance des Formes et l'Intelligence Artificielle (RFIA'00)*, Paris, Février 2000, Vol. 3, 267-276

CI+ Céline Lafage and Jérôme Lang, "Logical representation of preferences for group decision making", *Proceedings of the 7th International Conference on Knowledge Representation and Reasoning (KR2000)*, Breckenbridge, Colorado, Avril 2000, 457-468, Morgan Kaufmann.

CO Jérôme Lang, "Possibilistic logic : complexity and algorithms", *Handbook of Defeasible Reasoning and Uncertainty Management Systems* (D. Gabbay and Ph. Smets, eds.), Vol. 5, 179-220, 2000, Kluwer Academic Publishers.

CF Jérôme Lang and Pierre Marquis, "Boutons les incohérences hors de nos bases de connaissances", *Actes du 12ème Congrès sur la reconnaissance des Formes et l'Intelligence Artificielle (RFIA'00)*, Paris, Février 2000, Vol. 1, 185-194.

CI+ Jérôme Lang and Pierre Marquis, "In search of the right extension", *Proceedings of the 7th International Conference on Knowledge Representation and Reasoning (KR2000)*, Beckenbridge, Colorado, Avril 2000, 625-636, Morgan Kaufmann.

2001

RI Jérôme Lang and Pierre Marquis, "Removing inconsistencies in assumption-based theories through knowledge-gathering actions", *Studia Logica* 67 : 179-214, 2001, Kluwer Academic Publishers.

CI+ Andreas Herzig, Jérôme Lang, Pierre Marquis and Th. Polacsek, "Updates, actions and planning", *Proceedings of IJCAI'2001*, 119-124.

WI Andreas Herzig, Jérôme Lang and Pierre Marquis, "Planning as abduction", *Proceedings of the Workshop on Planning in Nondeterministic Domains* held during Ijcai-2001.

CI Céline Lafage and Jérôme Lang, "Propositional distances and preference representation", *Lectures Notes in Artificial Intelligence* 2143 (Proceedings of ECSQARU-2001), 48-59, 2001, Springer-Verlag.

CI+ Jérôme Lang and Philippe Muller, "Plausible reasoning from plausible observations", *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence* (UAI-01), 285-292, 2001, Morgan Kaufmann.

LMW01 Jérôme Lang, Pierre Marquis and Mary-Anne Williams, "Updating epistemic states", *Lectures Notes in Artificial Intelligence* 2256

2002

RI Jérôme Lang, Paolo Liberatore and Pierre Marquis, "Conditional independence in propositional logic". *Artificial Intelligence Journal* 141(1-2) :79-121, 2002, Elsevier.

RI Jérôme Lang, Leendert van der Torre and Emil Weydert, "Utilitarian desires", *International Journal of Autonomous Agents and Multi-Agent Systems*, 5, 329-363, 2002, Kluwer Academic Publishers.

CF S. Coste-Marquis, Hélène Fargier, Jérôme Lang, D. Le Berre et Pierre Marquis, "Formules booléennes quantifiées : problèmes et algorithmes", *Actes du 13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et d'Intelligence Artificielle (RFIA-2002)*, 2002, Vol. 1, 289-298 (prix du meilleur article).

CI+ Florence Dupin de Saint-Cyr and Jérôme Lang, "Belief extrapolation (or how to reason about observations and unpredicted change)", *Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002)*, 497-508, 2002, Morgan Kaufmann. Version française dans les *Actes de RFIA-2002*, Vol. 2, 705-714.

CI+ S. Konieczny, Jérôme Lang and Pierre Marquis, "Distance-based merging : a general framework and some complexity results", *Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002)*, 97-108, 2002, Morgan Kaufmann. Version française dans les *Actes de RFIA-2002*, Vol. 2, 695-704.

CI+ Jérôme Lang, "From preference representation to combinatorial vote", *Proceedings of the 8th International Conference on Knowledge Repre-*

sentation and Reasoning (KR2002), 277-288, 2002, Morgan Kaufmann. Version française dans les *Actes de RFIA-2002*, Vol. 3, 945-954.

CI+ Jérôme Lang and Pierre Marquis, "Resolving inconsistencies by variable forgetting", *Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR2002)*, 239-250, 2002, Morgan Kaufmann.

2003

RI Propositional Independence - Formula-Variable Independence and Forgetting J. Lang, P. Liberatore and P. Marquis *Journal of Artificial Intelligence Research* 18 :391-443, 2003.

CI+ Vincent Conitzer, Jérôme Lang and Tuomas Sandholm, "How many candidates are needed to make an election hard to manipulate ?", *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-03)*, 201-214.

CI+ A. Herzig, J. Lang and P. Marquis, "Action representation and partially observable planning in epistemic logic"*Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, 1067-1072.

CI+ S. Konieczny, J. Lang and P. Marquis, "Quantifying information and contradiction in propositional logic through test actions ", *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, 106-111.

CI+ J. Lang, F. Lin and P. Marquis, "Causal theories of action : a computational core", *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, 1073-1078.

CI+ J. Lang, L. van der Torre and Emil Weydert, "Hidden uncertainty in the logical representation of desires", *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, 685-690.

À paraître

RI J. Lang, "Logical preference representation and combinatorial vote", À paraître dans *Annals of Mathematics and Artificial Intelligence (Special Issue on the Computational Properties of Multi-Agent Systems)*, 2003.

RI Céline Lafage and Jérôme Lang, "Preference representation : an integrated framework for weighted and distance-graded goals". À paraître dans *European Journal of Operation Research*, 2003.

RF J. Lang. "Vote électronique : problèmes algorithmiques". À paraître dans *Sciences de la Société*, 2003.

Table des matières