Designing RNA Secondary Structures is Hard

Édouard Bonnet, Paweł Rzążewski, and Florian Sikora

ENS Lyon, LIP

RECOMB 2018, April 24th, Paris

RNA folding

- Predicting a likely secondary structure for an RNA sequence
- Probably an easy computational task: Nature computes it...
- ...not a sound argument; NP-hard with pseudoknots.

Pseudoknot-free RNA folding

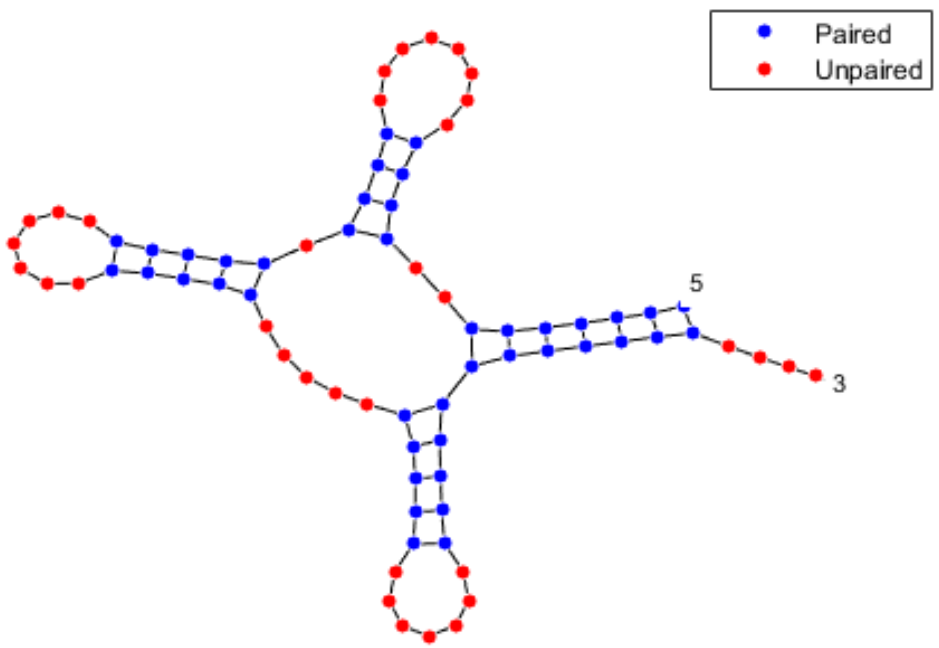
GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGA
UCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA

$\downarrow$?

Pseudoknot-free RNA folding

GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGA
UCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA

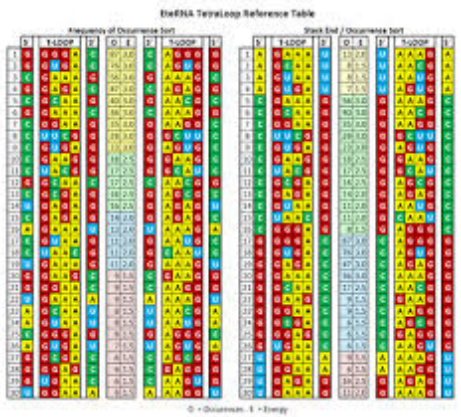
↓



(((((((..((((........))))).(((((.......)))))).....(((((.......)))))))))).....

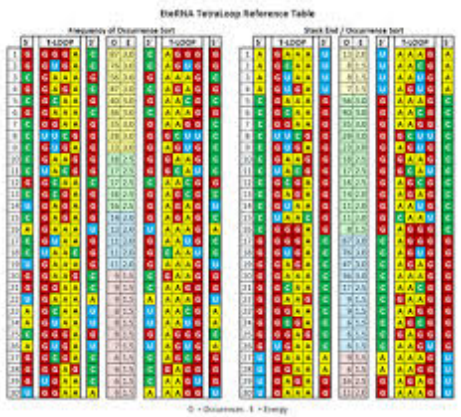
Energy models

- Watson-Crick: maximize the number of AU and GC pairs
- Nussinov-Jacobson: maximize a linear combination of AU, GC and GU pairs
- ...
- Turner: much more realistic model



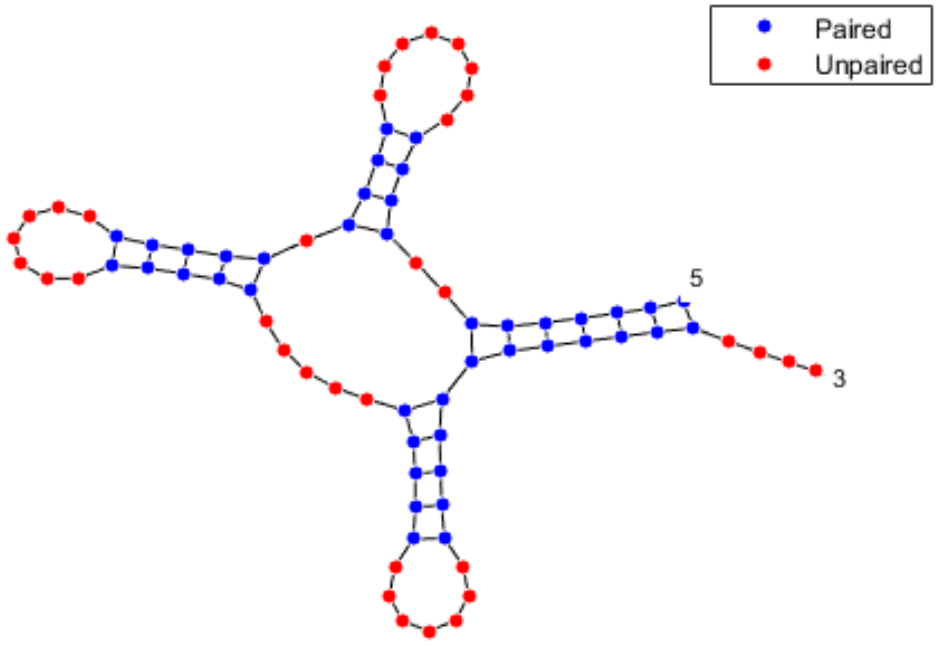
Energy models

- Watson-Crick: maximize the number of AU and GC pairs
- Nussinov-Jacobson: maximize a linear combination of AU, GC and GU pairs
- ...
- Turner: much more realistic model



Dynamic programming $O(n^3)$-algorithm for the first two, recently improved to $O(n^{2.861})$.

RNA design

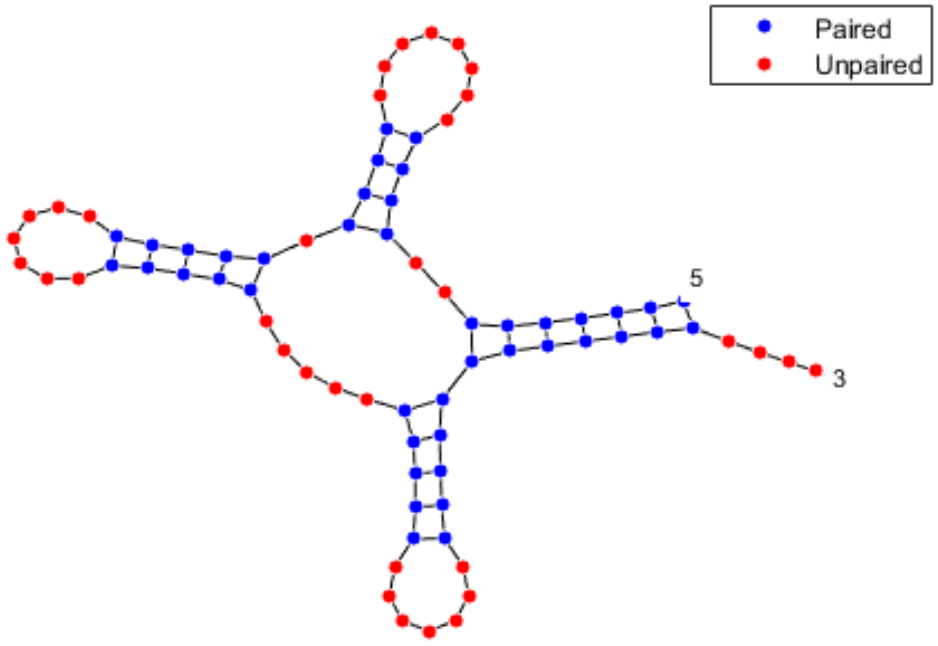


(((((((..((((........))))..(((((.......)))))).....(((((.......)))))))))))....

↓

???????????????????????????????????
???????????????????????????????????

RNA design

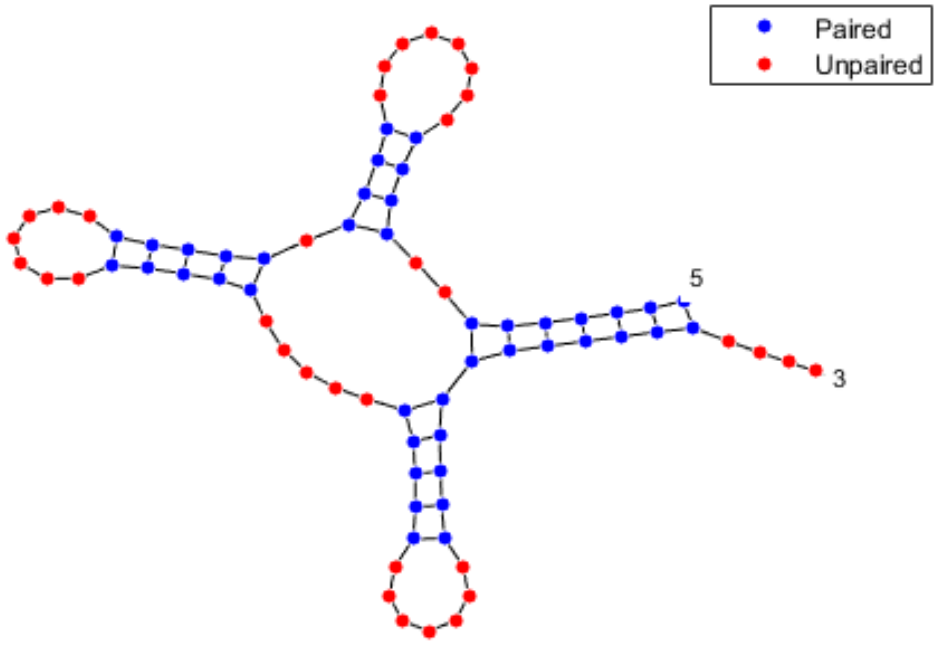


(((((((..((((........))))..(((((.......)))))).....(((((.......)))))))))))....

↓

?C???UUUA?CUCAGU??GG??????CCA??CUGA?GA
?????AGGUCC??????CG?UC?????????CG????A

RNA design



(((((((..((((........))))..(((((.......)))))).....(((((.......)))))))))))....

↓

GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGA
UCUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCACCA

Our result

Theorem

Given a pseudoknot-free secondary structure S with imposed nucleotides at some places, finding a complete RNA sequence that folds uniquely into S in the Watson-Crick model is NP-hard.

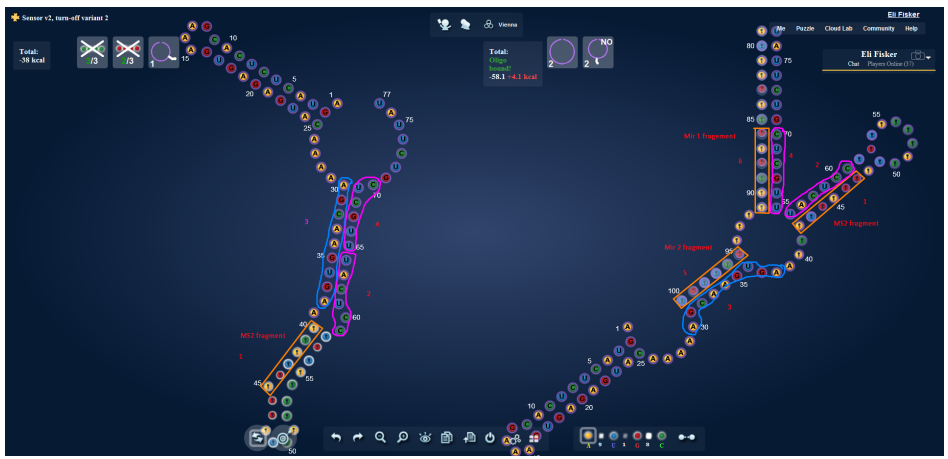
Such a sequence is called a design.

Our result

Theorem

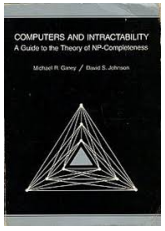
Given a pseudoknot-free secondary structure S with imposed nucleotides at some places, finding a complete RNA sequence that folds uniquely into S in the Watson-Crick model is NP-hard.

Such a sequence is called a design.



The problem most likely remains hard with more realistic models.

3-SAT reduction



Given a 3-SAT formula ϕ with clauses $\{C_j\}_{1 \leqslant j \leqslant m}$ on variables $\{x_i\}_{1 \leqslant i \leqslant n}$, we build a structure S with pre-assigned nucleotides such that:

$$\phi \text{ is satisfiable} \Leftrightarrow S \text{ admits a design.}$$

Formula ϕ is said satisfiable if we can attribute T/F to each variable x_i so that each disjunction $C_j = \ell_a \vee \ell_b \vee \ell_c$ is T.

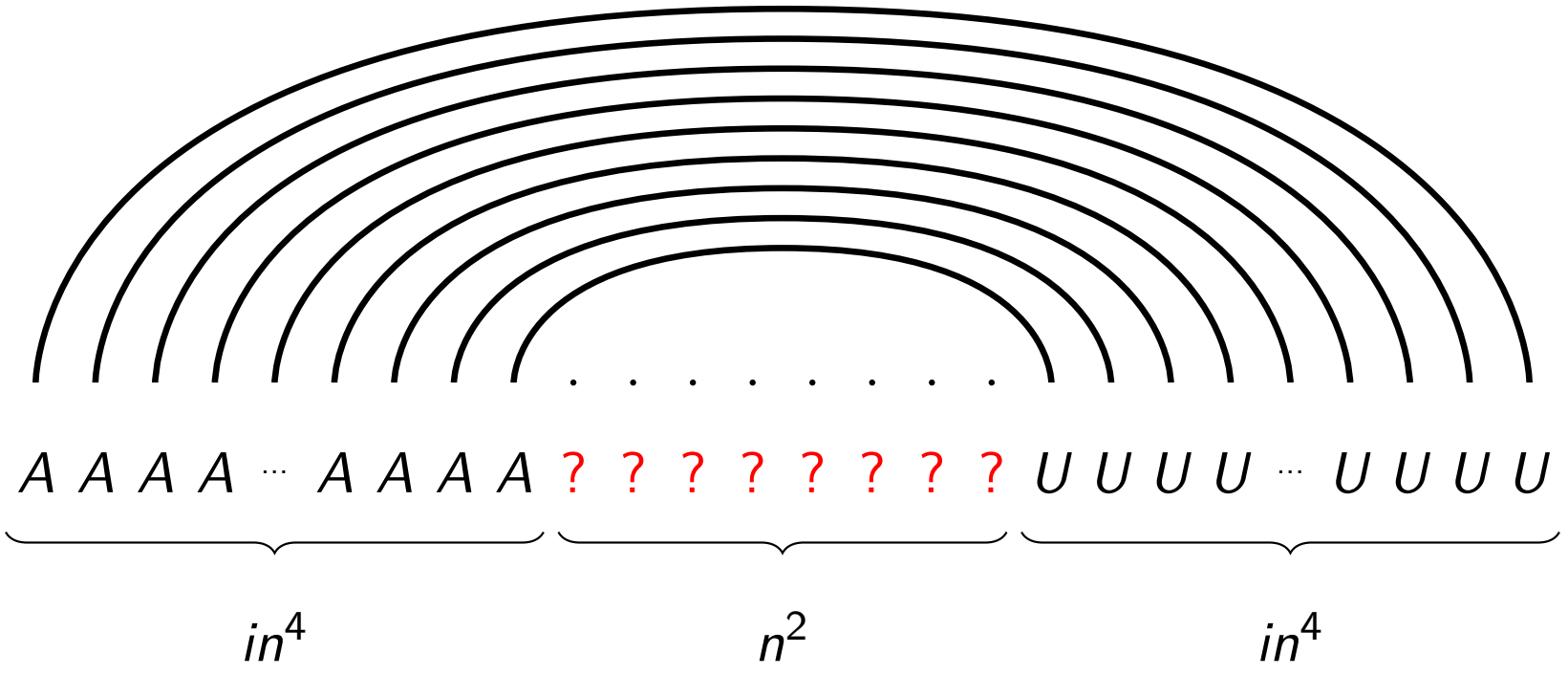
A way to see designs

A labeling extension such that there is no rematching.

What we want is:

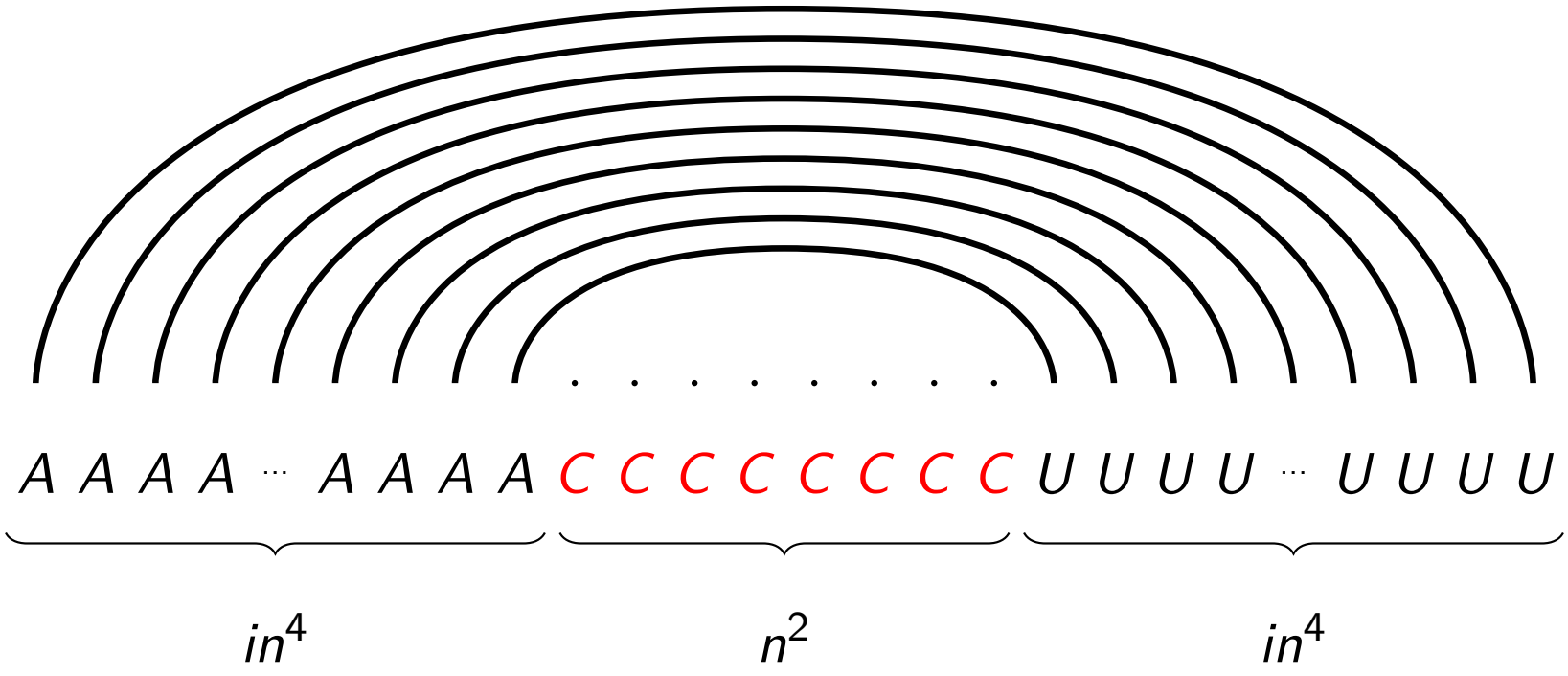
- ϕ satisfiable $\Rightarrow$ there is one extension without rematching.
- ϕ not satisfiable $\Rightarrow$ every extension admits a rematching.

Variables: hairpin loops with increasing arches



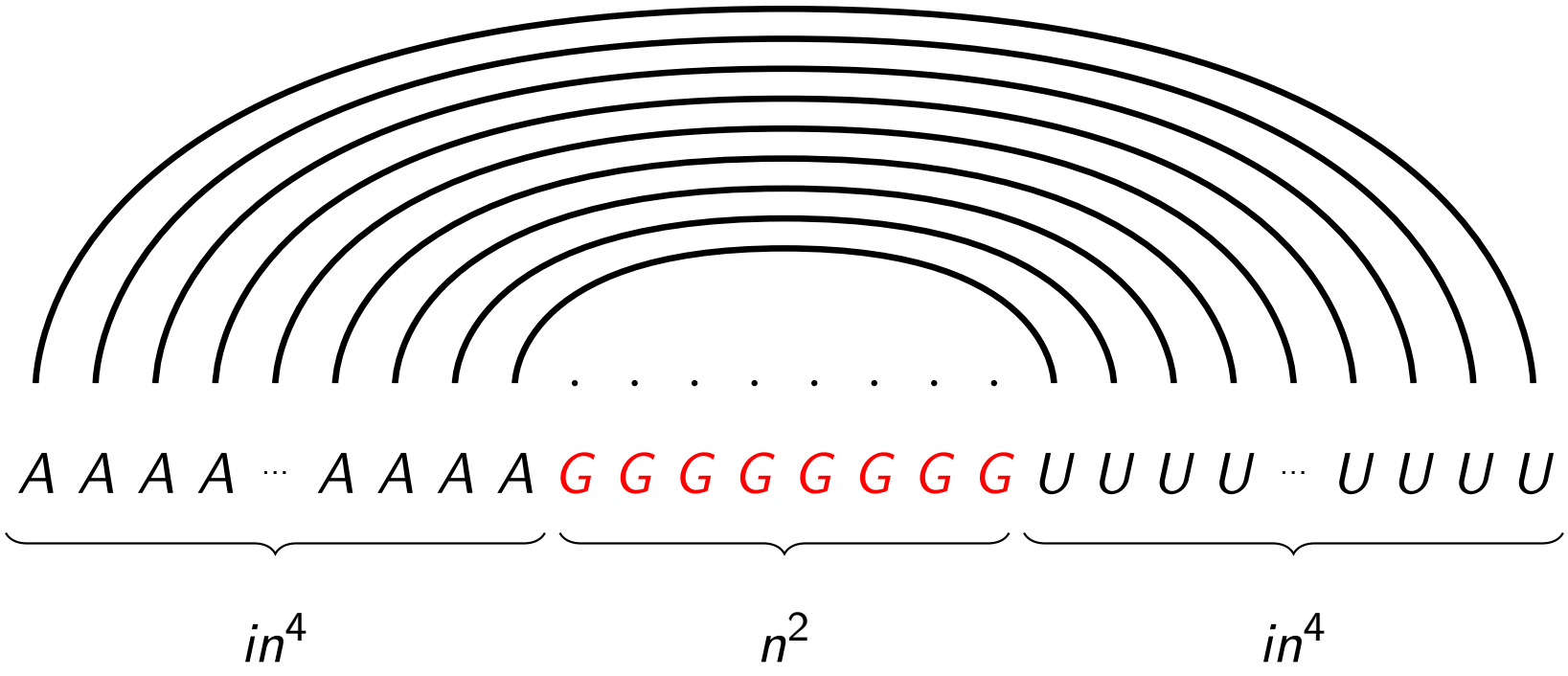
$V\langle x_i \rangle$: encoding of x_i

Variables: hairpin loops with increasing arches



$L\langle x_i \rangle$: Setting x_i to T $\leftrightarrow$ labeling the dots by C

Variables: hairpin loops with increasing arches



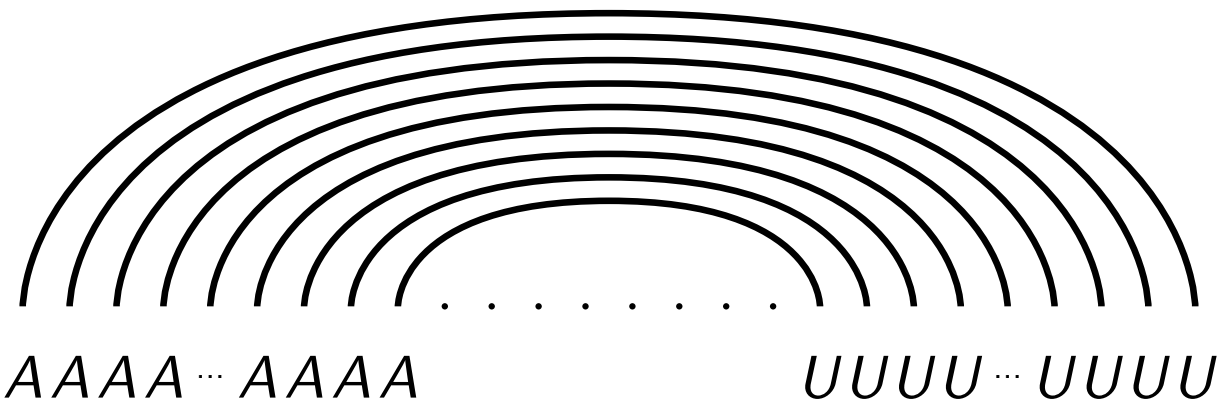
$L\langle \neg x_i \rangle$: Setting x_i to F $\leftrightarrow$ labeling the dots by G

Locality

Seeing S as a tree, the variable gadgets are subtrees of S.

Fact: **a design should be a design for each subtree**

So, a design *has to set each variable to T or F*

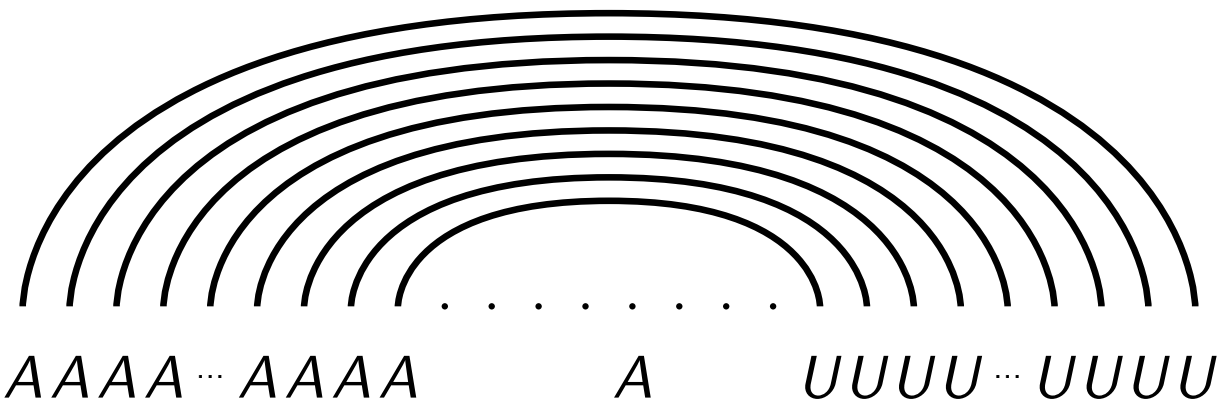


Locality

Seeing S as a tree, the variable gadgets are subtrees of S.

Fact: **a design should be a design for each subtree**

So, a design *has to set each variable to T or F*

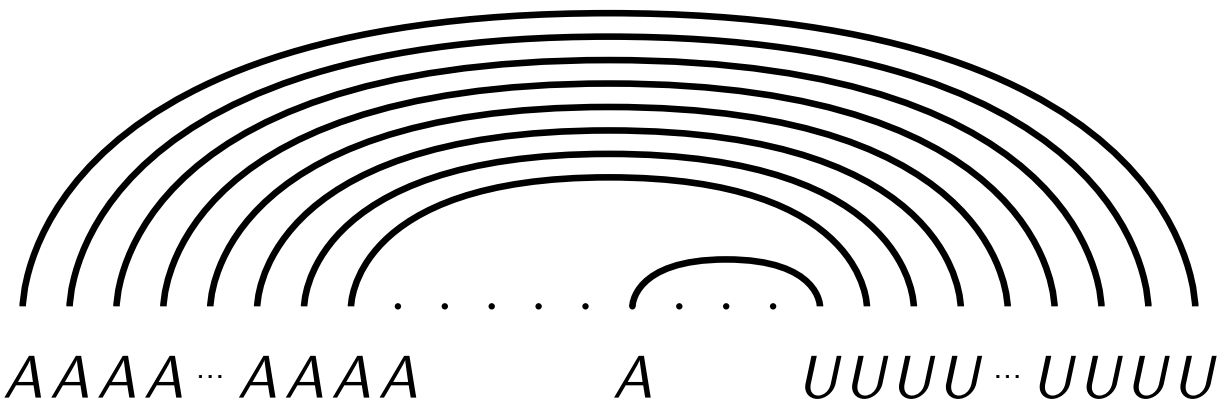


Locality

Seeing S as a tree, the variable gadgets are subtrees of S.

Fact: **a design should be a design for each subtree**

So, a design *has to set each variable to T or F*

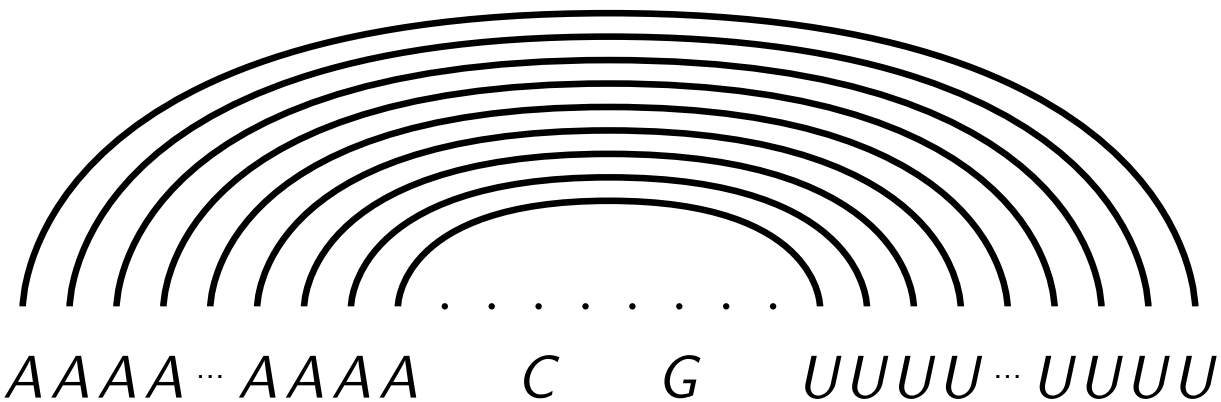


Locality

Seeing S as a tree, the variable gadgets are subtrees of S.

Fact: **a design should be a design for each subtree**

So, a design *has to set each variable to T or F*

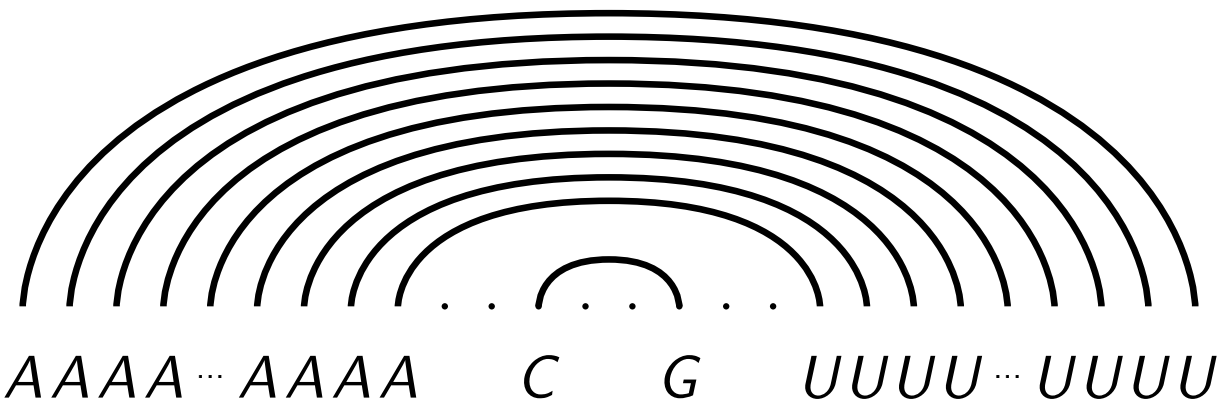


Locality

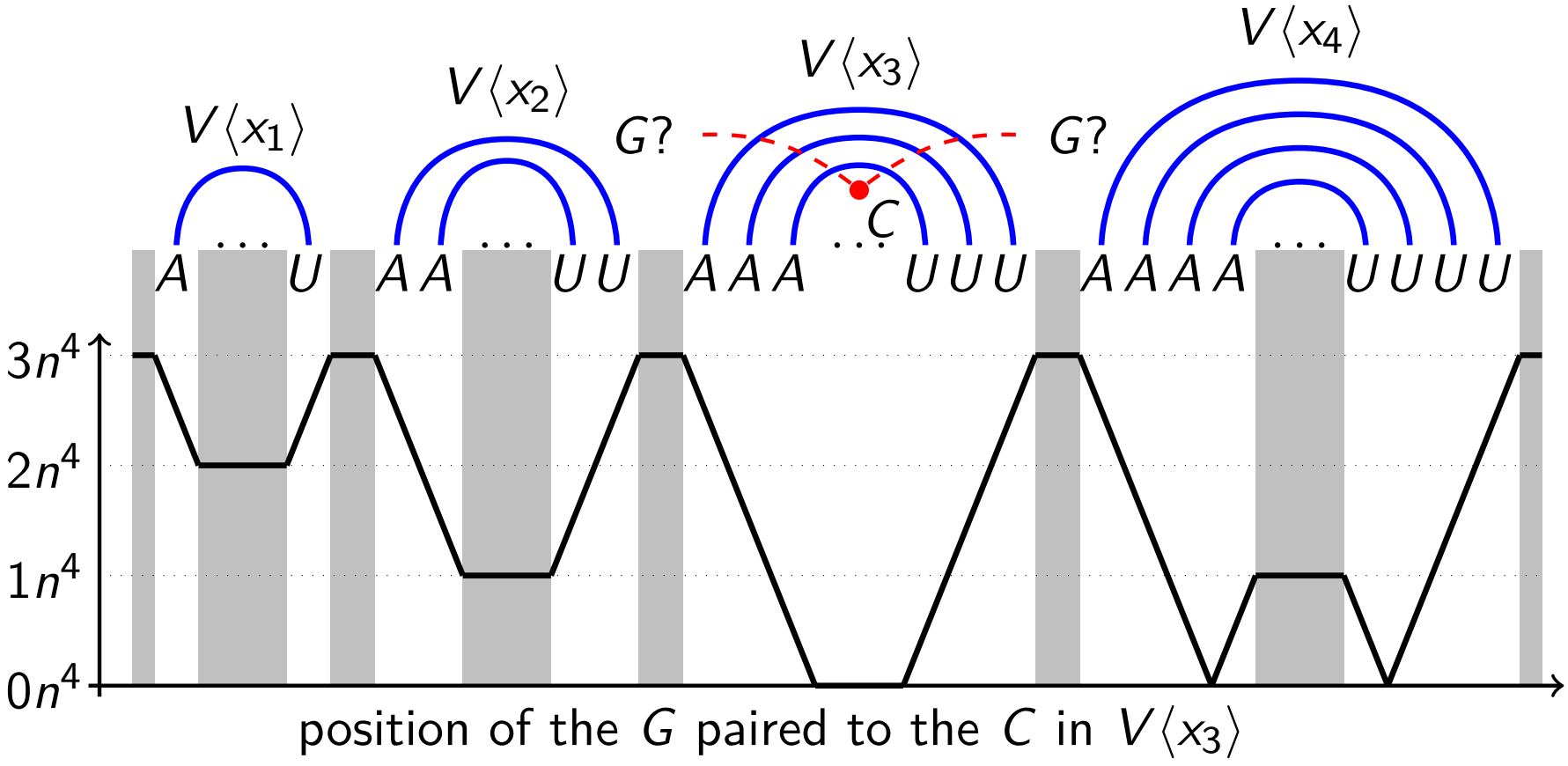
Seeing S as a tree, the variable gadgets are subtrees of S.

Fact: **a design should be a design for each subtree**

So, a design *has to set each variable to T or F*



Why the longer and longer arches?



The y-axis carries the imbalance of A/U:
$|\#_A w - \#_U w|$ where w is overarched by this supposed GC pair

How to make the clause gadget?

Idea: the clause $C_j = \ell_a \vee \ell_b \vee \ell_c$ is encoded by an arch of a bit less than $3n^2$ CG pairs enclosing $L\langle \ell_a \rangle$, $L\langle \ell_b \rangle$, $L\langle \ell_c \rangle$

How to make the clause gadget?

Idea: the clause $C_j = \ell_a \vee \ell_b \vee \ell_c$ is encoded by an arch of a bit less than $3n^2$ CG pairs enclosing $L\langle \ell_a \rangle$, $L\langle \ell_b \rangle$, $L\langle \ell_c \rangle$

Why is this reasonable?

How to make the clause gadget?

Idea: the clause $C_j = \ell_a \vee \ell_b \vee \ell_c$ is encoded by an arch of a bit less than $3n^2$ CG pairs enclosing $L\langle \ell_a \rangle$, $L\langle \ell_b \rangle$, $L\langle \ell_c \rangle$

Why is this reasonable?

Iff none of the 3 literals are "satisfied"

- destroying the CG-arch and,
- rematching $V\langle x_i \rangle$ with $L\langle \ell_i \rangle$

yields more pairs overall

How to make the clause gadget?

Idea: the clause $C_j = \ell_a \vee \ell_b \vee \ell_c$ is encoded by an arch of a bit less than $3n^2$ CG pairs enclosing $L\langle \ell_a \rangle$, $L\langle \ell_b \rangle$, $L\langle \ell_c \rangle$

Why is this reasonable?

Iff none of the 3 literals are "satisfied"

- destroying the CG-arch and,
- rematching $V\langle x_i \rangle$ with $L\langle \ell_i \rangle$

yields more pairs overall

Why is this failing?

How to make the clause gadget?

Idea: the clause $C_j = \ell_a \vee \ell_b \vee \ell_c$ is encoded by an arch of a bit less than $3n^2$ CG pairs enclosing $L\langle \ell_a \rangle$, $L\langle \ell_b \rangle$, $L\langle \ell_c \rangle$

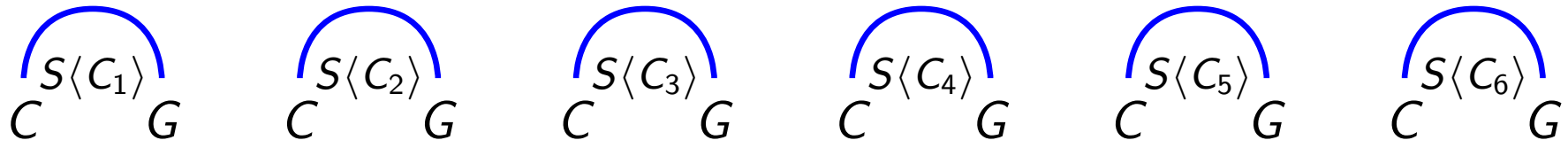
Why is this reasonable?

Iff none of the 3 literals are "satisfied"

- destroying the CG-arch and,
- rematching $V\langle x_i \rangle$ with $L\langle \ell_i \rangle$

yields more pairs overall

Why is this failing?



How to make the clause gadget?

Idea: the clause $C_j = \ell_a \vee \ell_b \vee \ell_c$ is encoded by an arch of a bit less than $3n^2$ CG pairs enclosing $L\langle \ell_a \rangle$, $L\langle \ell_b \rangle$, $L\langle \ell_c \rangle$

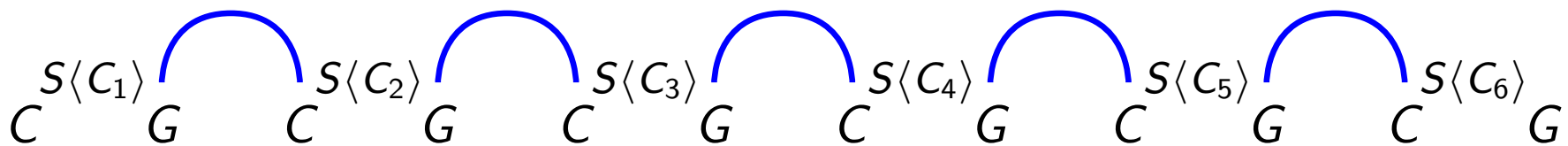
Why is this reasonable?

Iff none of the 3 literals are "satisfied"

- destroying the CG-arch and,
- rematching $V\langle x_i \rangle$ with $L\langle \ell_i \rangle$

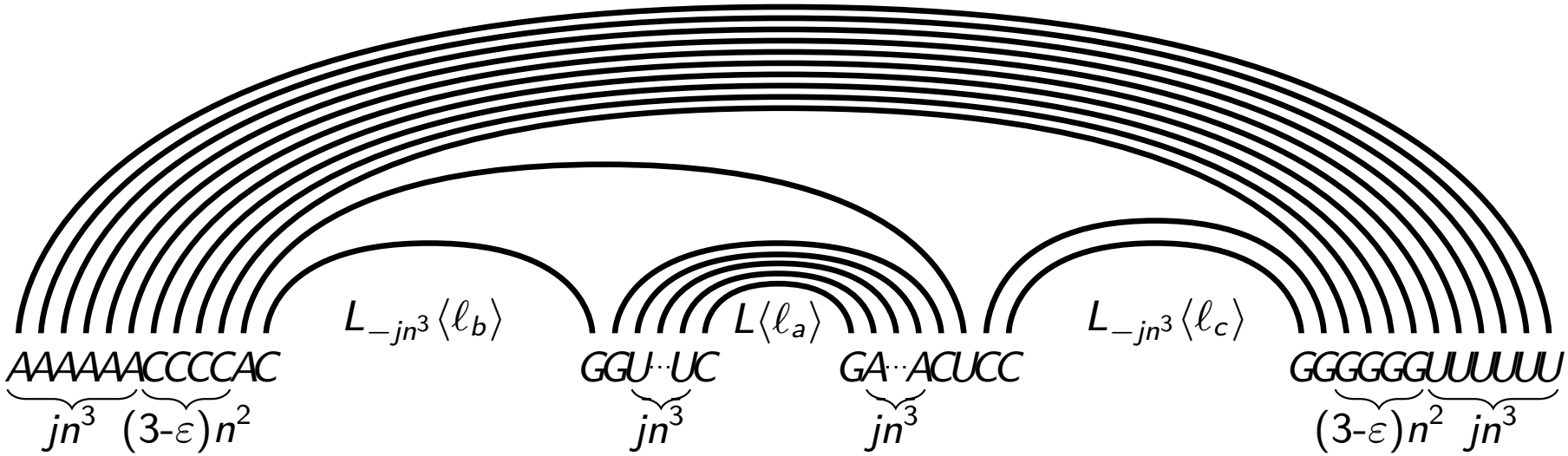
yields more pairs overall

Why is this failing?



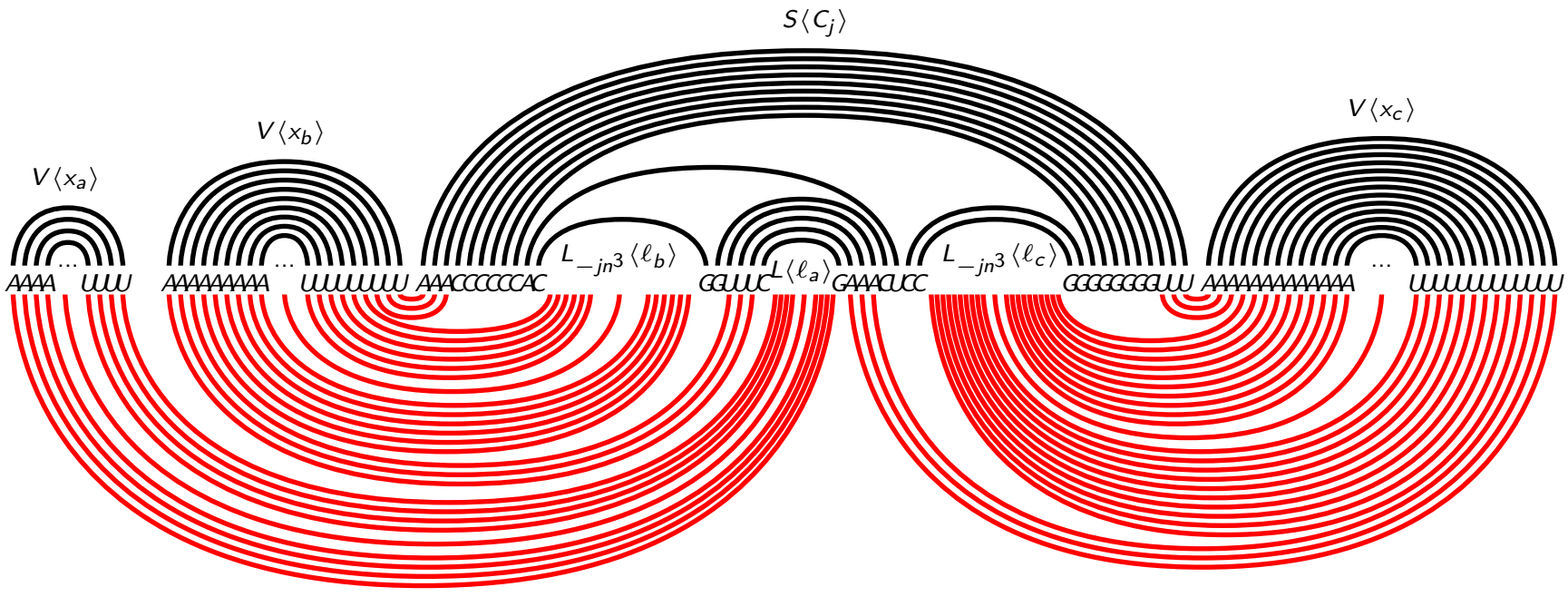
opening all your gifts for the price of one

Fixing the clause gadget $S\langle C_j \rangle$

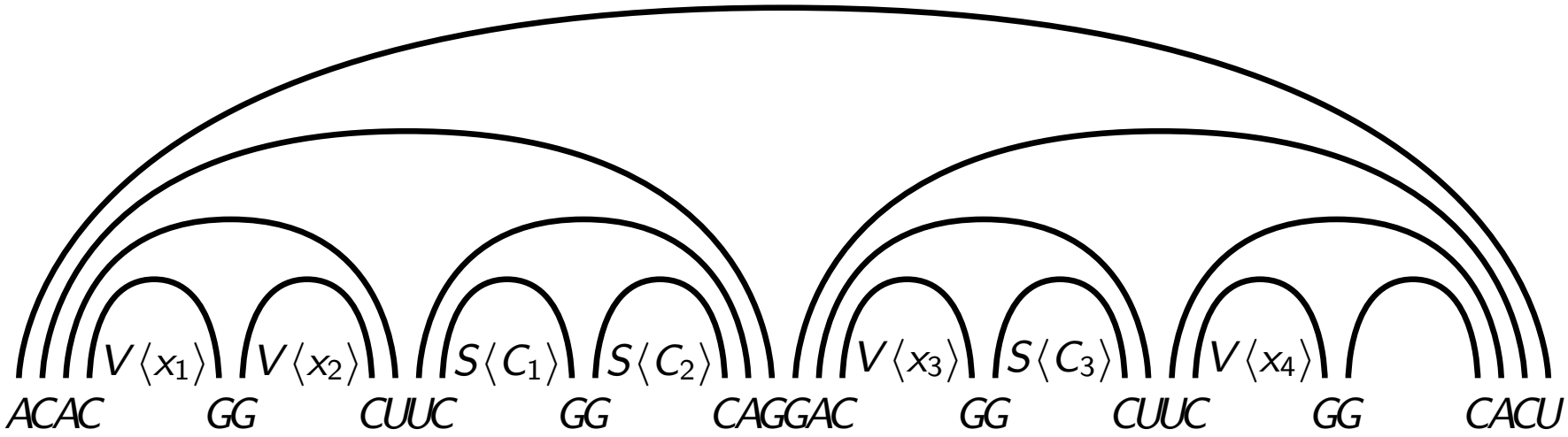


- doubling the CG-arch with a much thicker AU-arch
- putting the literals in the order 2, 1, 3
- placing $S\langle C_j \rangle$ after $V\langle x_a \rangle$, $V\langle x_b \rangle$ and before $V\langle x_c \rangle$

If ϕ is not satisfiable

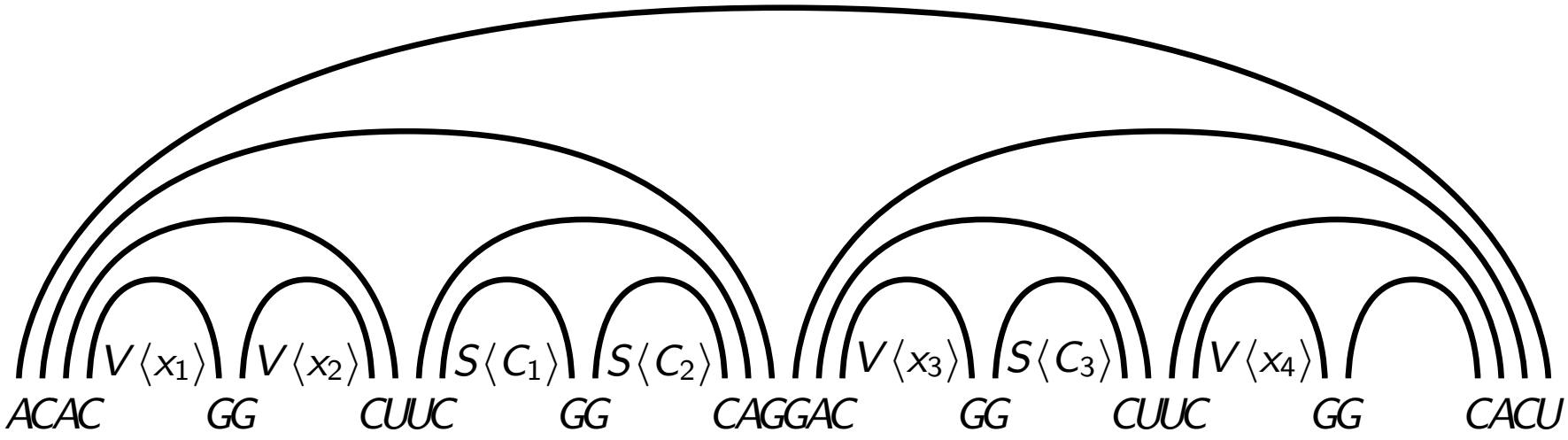


The entire structure S



We order variable/clause gadgets at the leaves of a "binary tree"

The entire structure S



We order variable/clause gadgets at the leaves of a "binary tree"

Hales et al. showed that it is easy to design saturated structures:
any locally good labeling is globally good

Wrapping up

- S has n^3 unlabeled nucleotides, n^2 per variable gadget.
- The only choice in a variable gadget is *all* C (T) or *all* G (F).

Wrapping up

- S has n^3 unlabeled nucleotides, n^2 per variable gadget.
- The only choice in a variable gadget is *all C* (T) or *all G* (F).
- ϕ not satisfiable $\Rightarrow$ one clause gadget admits a rematching.
- ϕ satisfiable $\rightarrow$ we label according to a satisfying assignment.

Wrapping up

- S has n^3 unlabeled nucleotides, n^2 per variable gadget.
- The only choice in a variable gadget is *all* C (T) or *all* G (F).
- ϕ not satisfiable $\Rightarrow$ one clause gadget admits a rematching.
- ϕ satisfiable $\rightarrow$ we label according to a satisfying assignment.

Assume $\exists S'$ compatible structure with more pairs than S

- S' matches at least one dot.
- This has to be between a $V\langle x_i \rangle$ and a $L\langle \ell_i \rangle$ in $S\langle C_j \rangle$
- C_j satisfied $\Rightarrow$ a literal gadget of $S\langle C_j \rangle$ cannot be rematched.
- Contradiction, since the $(3-\varepsilon)n^2$ CG pairs are essentially lost.

Perspectives

- NP-hardness even without the imposed nucleotides?
- Considering a more realistic model might actually help for this.
- Heuristics, parameterized and subexponential algorithms

Perspectives

- NP-hardness even without the imposed nucleotides?
- Considering a more realistic model might actually help for this.
- Heuristics, parameterized and subexponential algorithms

Thank you for your attention!