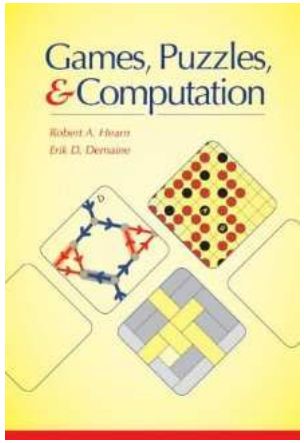
Game Complexity

Rush Hour, . . .

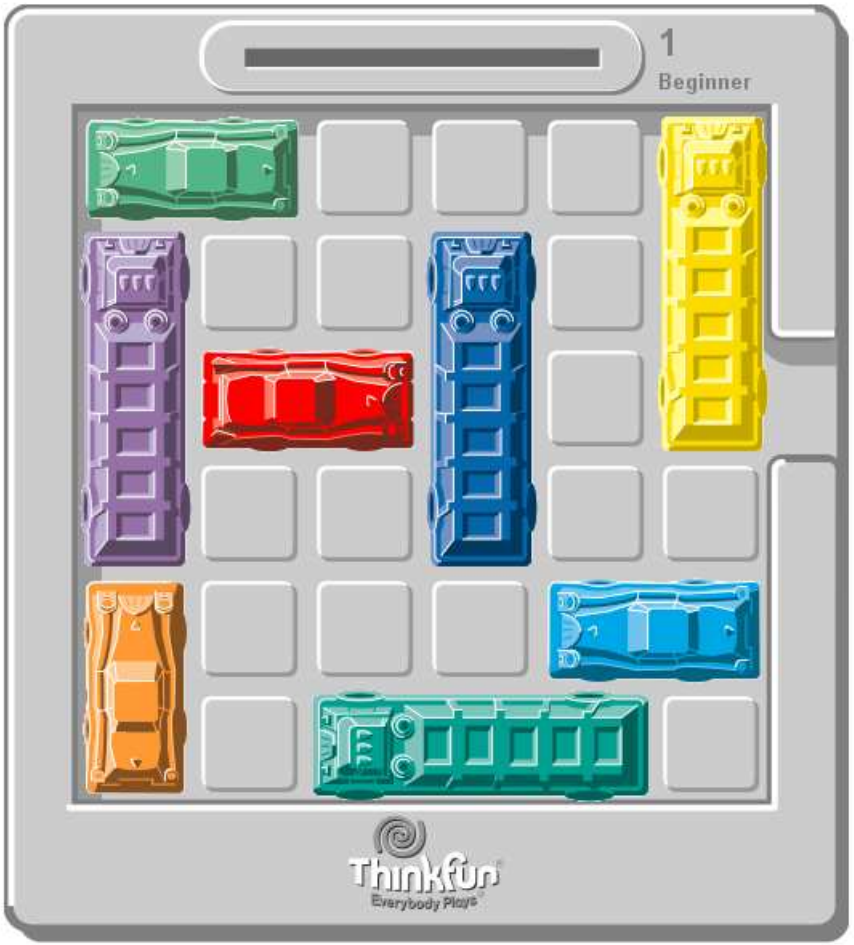


Walter Kosters, Universiteit Leiden

www.liacs.leidenuniv.nl/~kosterswa/

IPA, Eindhoven; Friday, July 8, 2016

Having seen the general picture and some gadgetry, we now examine particular games and puzzles, like Rush Hour®:



www.puzzles.com/products/rushhour.htm

The rules of Rush Hour are easy: cars may move either horizontally or vertically (left/right and up/down), in their natural direction, as long as they do not bump/crash into other cars or the walls.

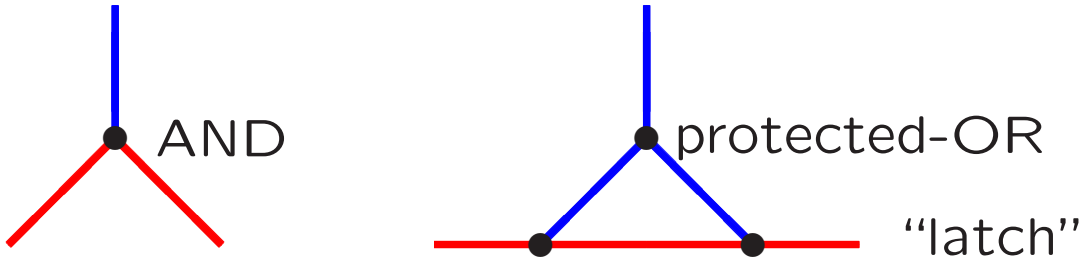
The target is to get the red car out of the garage through the exit.

Theorem Rush Hour is PSPACE-complete.
(Remember Savitch: PSPACE = NPSPACE.)

↑

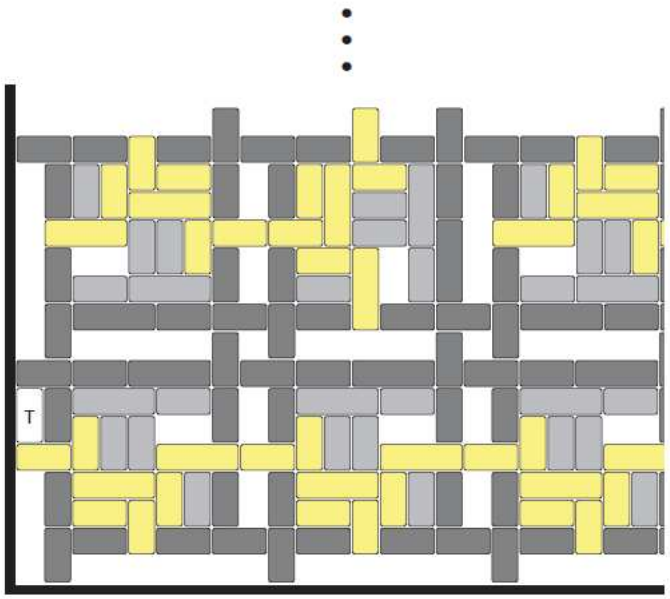
non-deterministic Turing machine with polynomial space

The proof proceeds by reduction from Nondeterministic Constraint Logic (NCL): NCL is PSPACE-complete for planar graphs using only ANDs and protected-ORs.

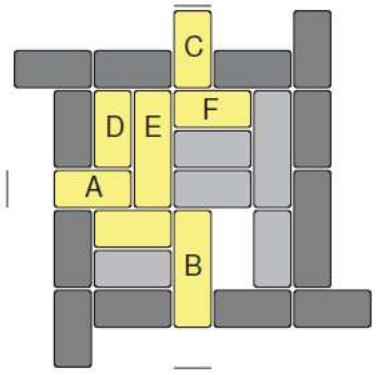


The decision problem is: Given a constraint graph G (including arrows) and a distinguished edge e in G; is there a sequence of edge reversals that eventually reverses e?
Moves may be repeated: it is an *unbounded game*.

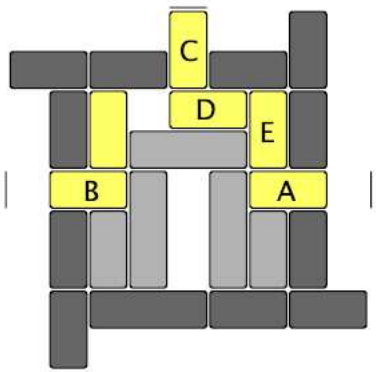
Proof



(a) Layout



(b) AND



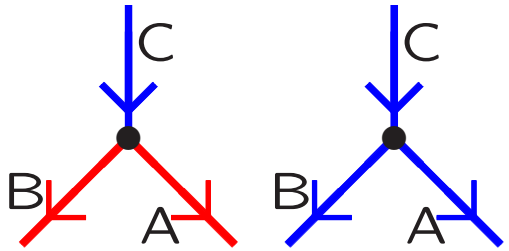
(c) Protected OR

target "car" T must go down

"car" is in

$\Leftrightarrow$

edge points out

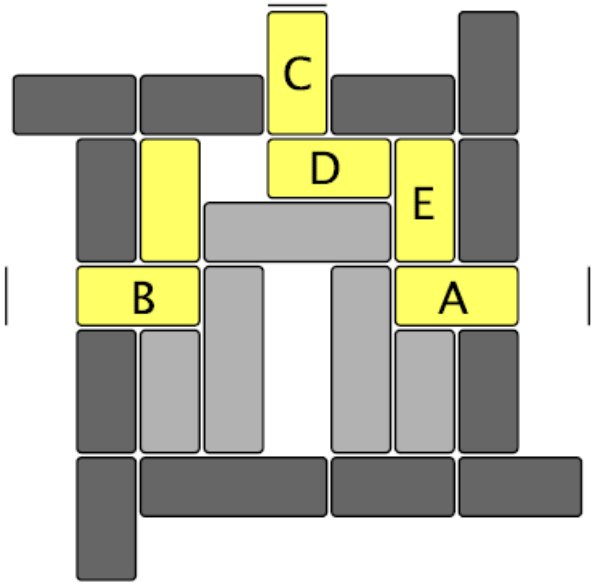


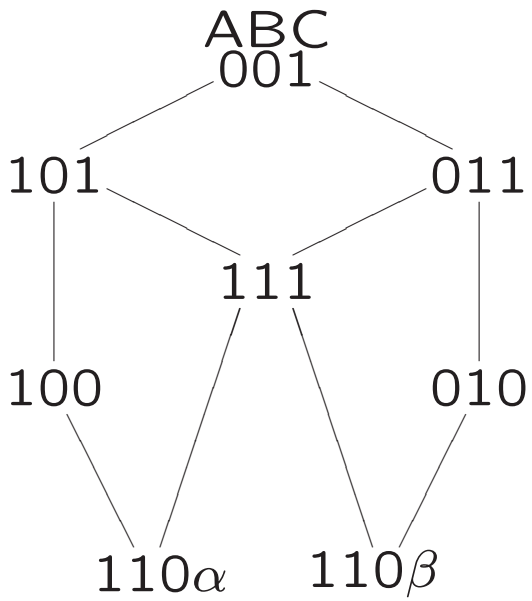
Exercise: Fill in the proof details.

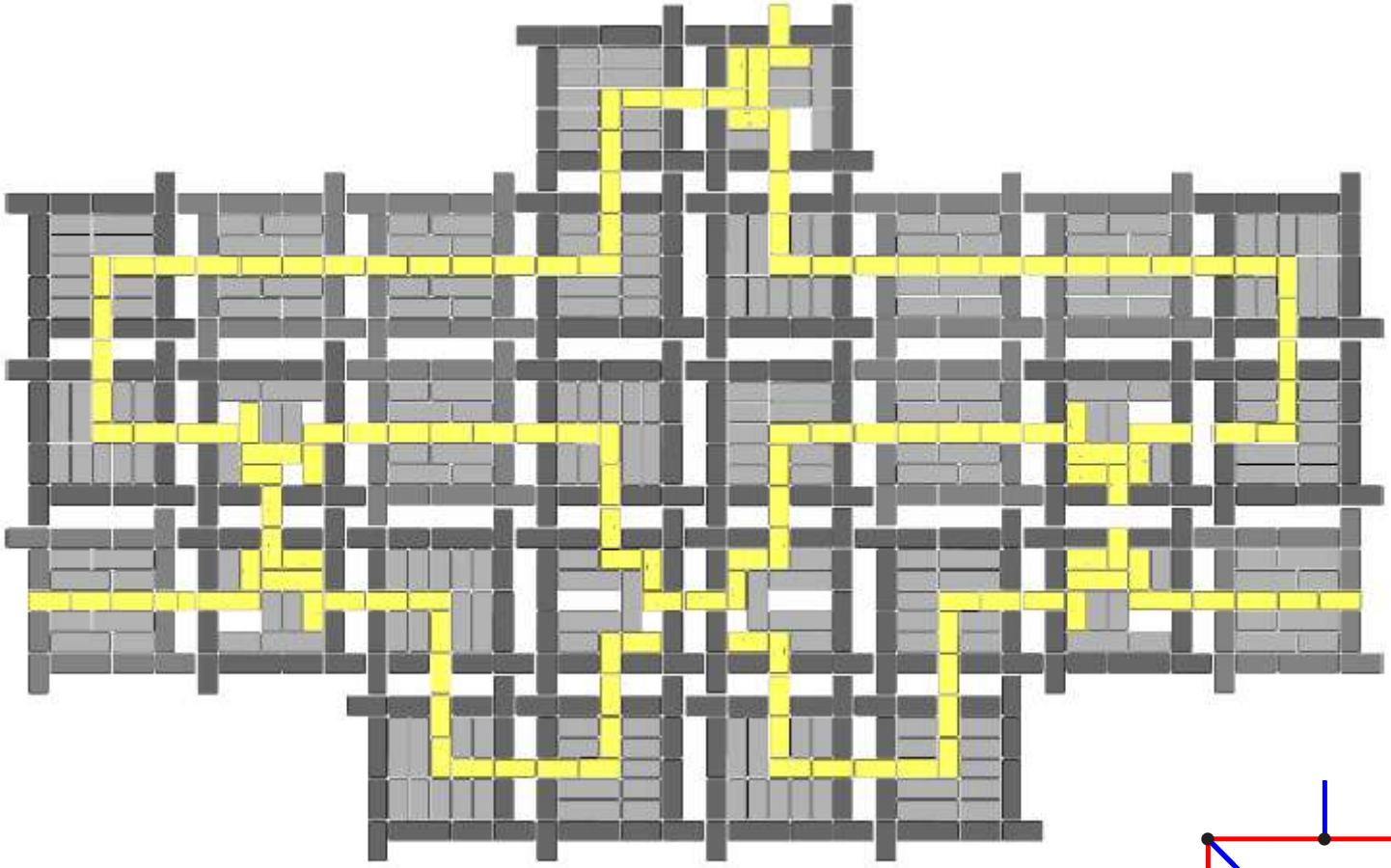
This includes

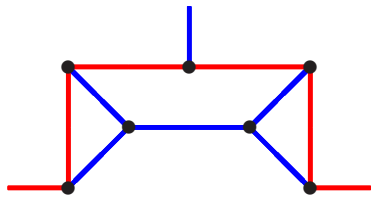
- proper inner working of the gadgets,
- proper communication between gadgets,
- proper glueing together (in polynomial space),
- check that walls do not move,
- ...

The statespace for the Rush-Hour protected-OR gadget is somewhat strange (where 1: car out; 0: car in):









NCL is PSPACE-complete

Clearly, NCL is in NPSPACE (= PSPACE).

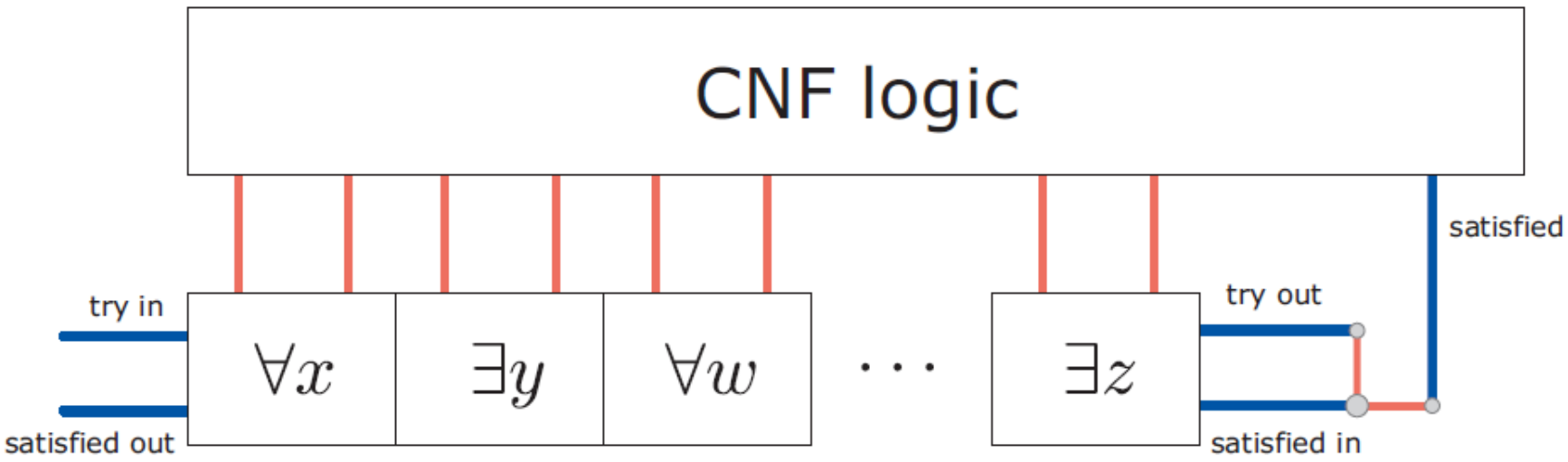
To prove that NCL is PSPACE-hard, we reduce from Quantified Boolean Logic (QBF):

> Given a quantified Boolean formula ϕ (with quantifiers $\exists$ and $\forall$, and of course variables, connectives $\vee$, $\wedge$ and $\neg$, and parentheses), is this ϕ true?

And QBF is known to be PSPACE-complete: it is an element of PSPACE and every problem in PSPACE reduces to QBF — and so, by transitivity, to NCL.

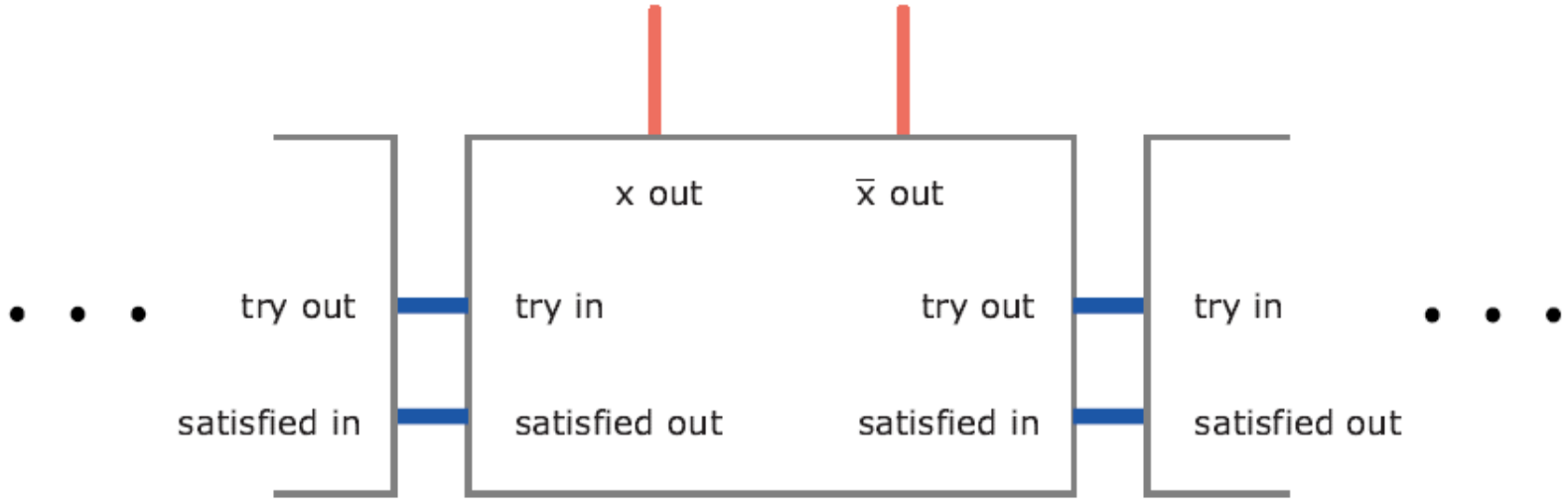
Proof idea — 1

$$\forall x \exists y \forall w \cdots \exists z \, [(x \vee y) \wedge \cdots \wedge (\overline{z} \vee x \vee \overline{w})]$$



CNF = Conjunctive Normal Form

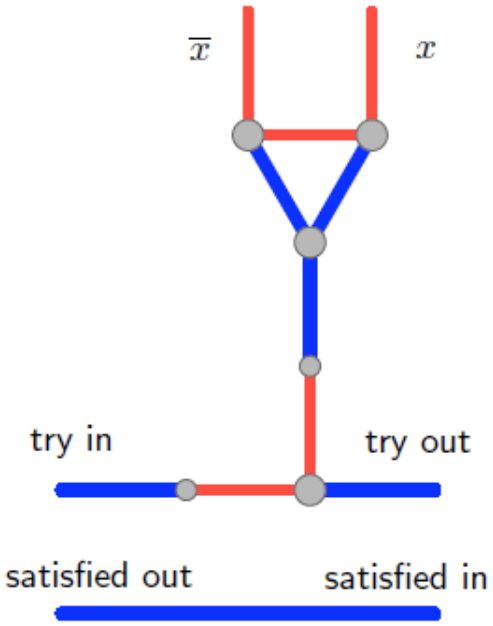
Proof idea — 2



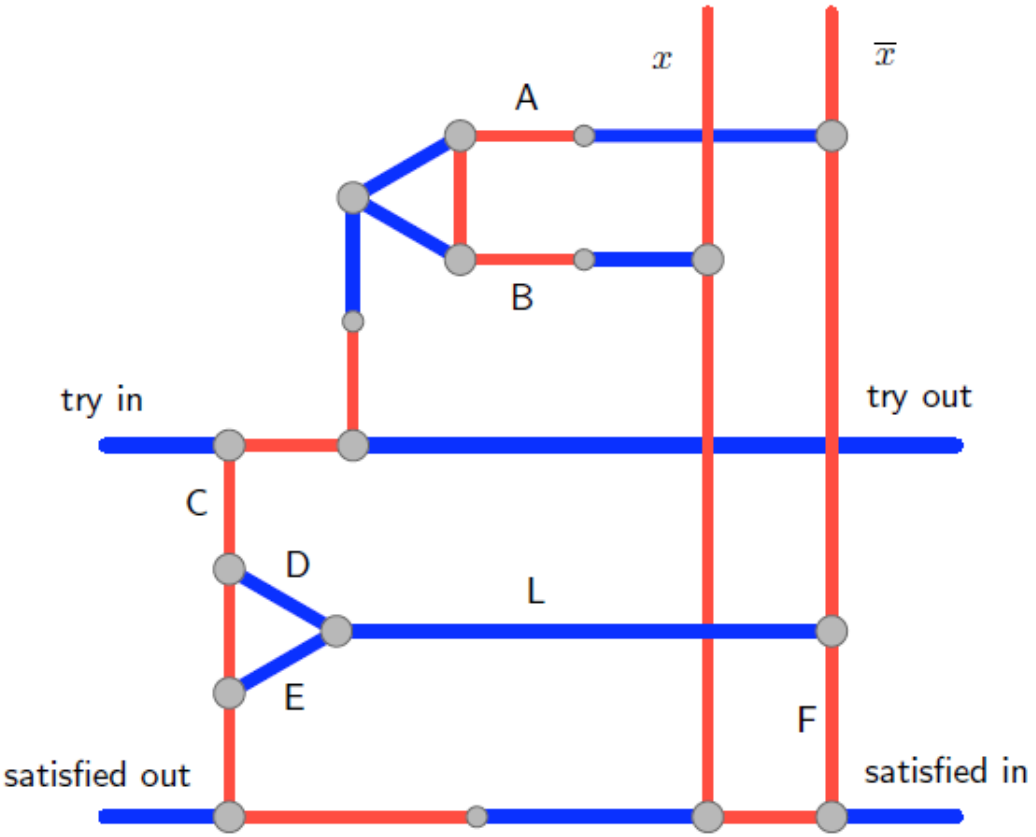
In a $\exists$ gadget, you try (fix) both x and $\overline{x}$ to be `true`, and then try the next quantifier.

In a (more complicated) $\forall$ gadget, you do both, meanwhile "recursively" doing the rest twice.

Proof idea — 3



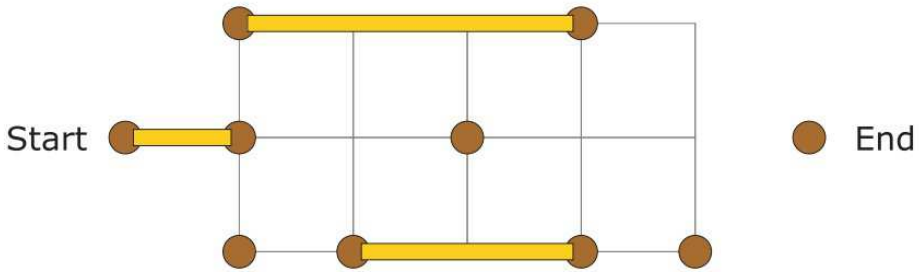
(a) Existential quantifier



(b) Universal quantifier

Plank puzzle — 1

And how about the Plank puzzle = River Crossing^{TM}?

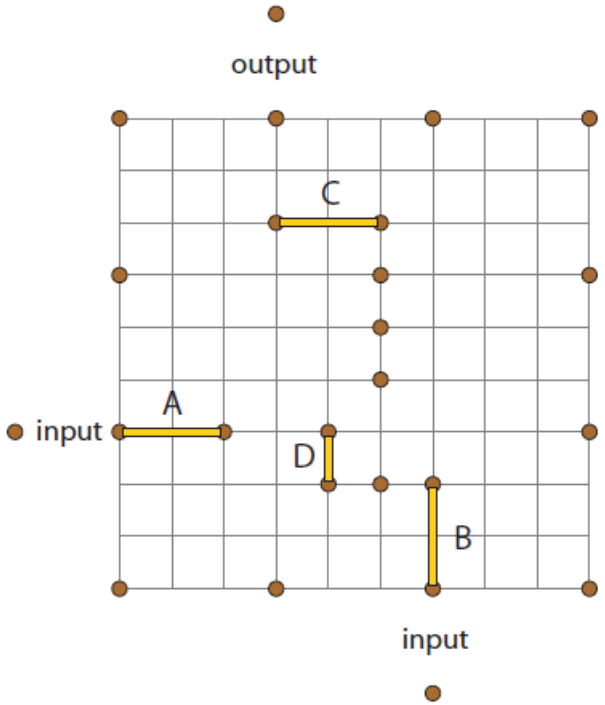


You must travel from Start to End; you can carry and move one plank at a time (if you "have" it), and traverse them in the obvious way.

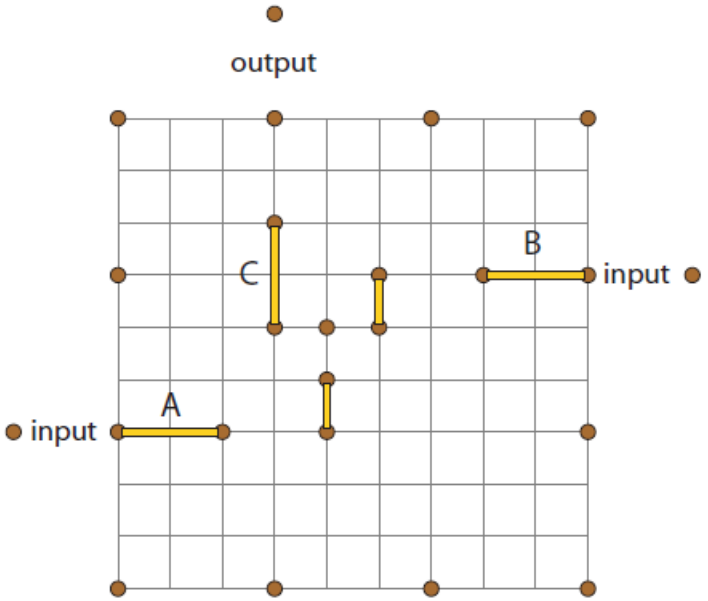
Note added: in the gadgets from the next slide, for the correct behavior it is important that plank A and/or B are inside. You can freely walk around the squares with a length 3 plank.

Plank puzzle — 2

The Plank puzzle is also PSPACE-complete:

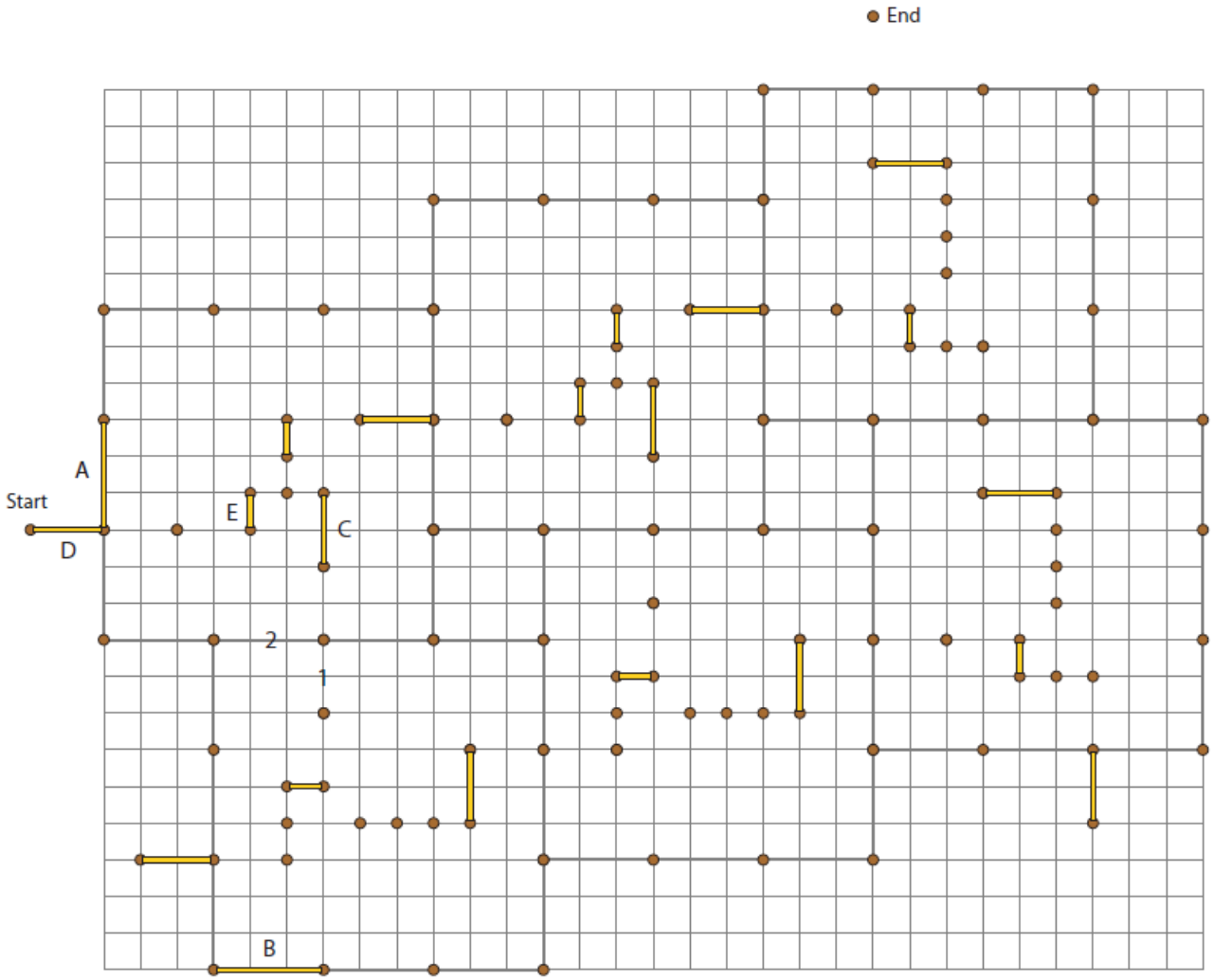


(a) AND



(b) OR

Plank puzzle — 3

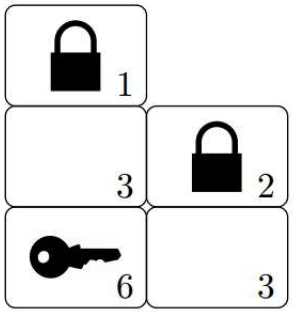


Game rules: two visible stones may be removed if they are the same *and* they are "free" to one or two sides.

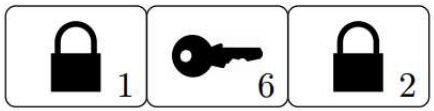
Exercise: Provide AND- and OR-gadgets for Mahjongg.

Hint: keep it simple; find a small set of stones, such that a special one can be “opened” exactly if one (for OR, or both for AND) of two others can be removed.

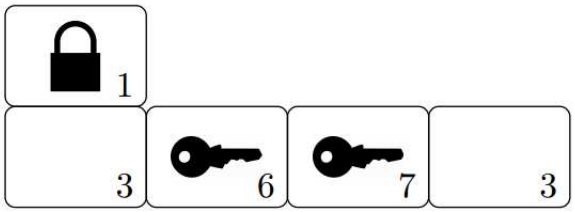
Mahjongg — gadgets continued



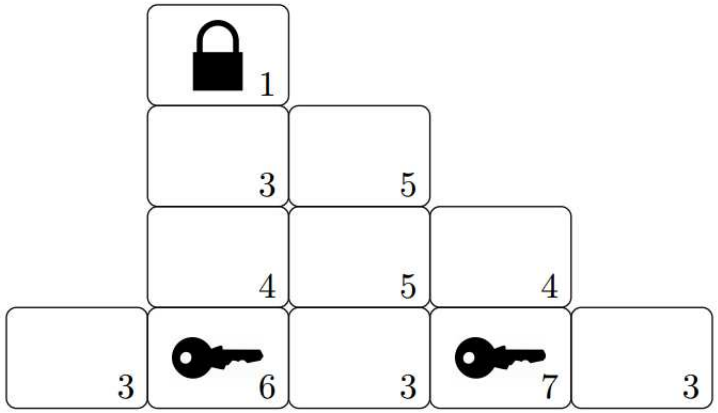
(a) AND gadget



(b) OR gadget

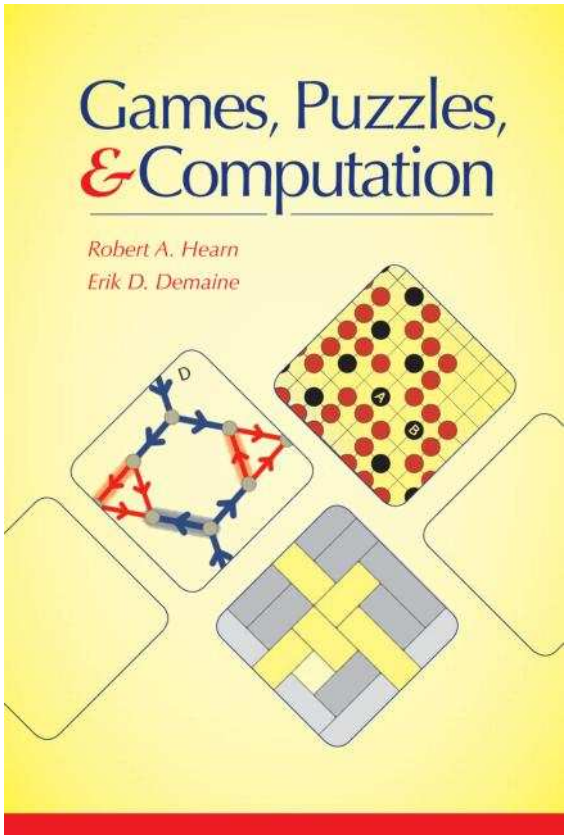


(c) FANOUT gadget

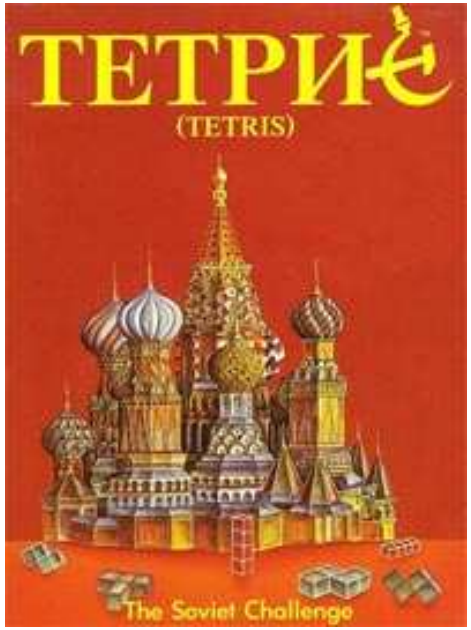


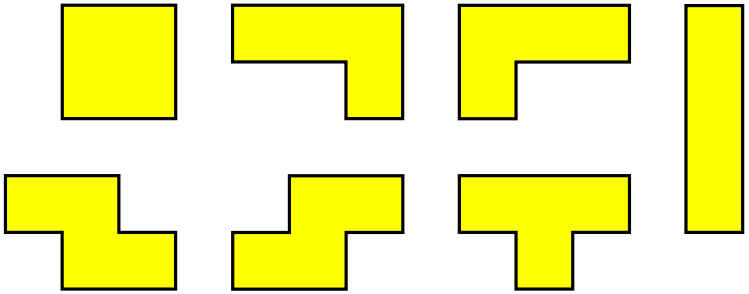
(d) CHOICE gadget

Many games can be proved to be NP-hard, PSPACE-hard, etc., using the Constraint Logic machinery.



Bonus question

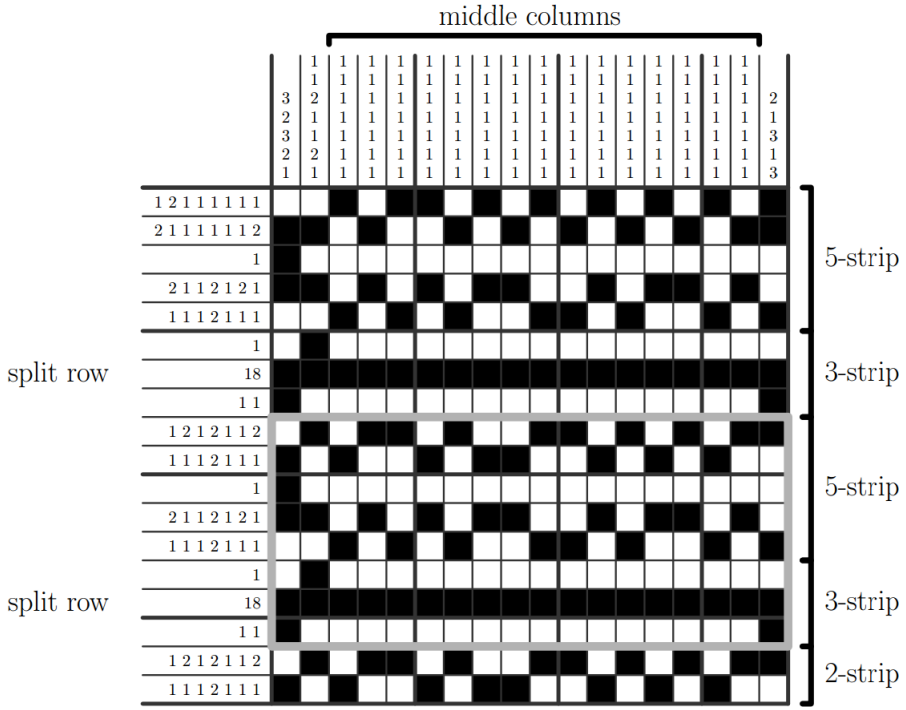




If you can choose the pieces that fall, and their position and orientation, is it then possible to generate each and every Tetris configuration?

www.liacs.leidenuniv.nl/~kosterswa/tetris/

Further research



18 × 18 Nonogram

with difficulty 115

= hori↔verti switches

How to attach a difficulty measure to a puzzle?

www.liacs.leidenuniv.nl/~kosterswa/nono/