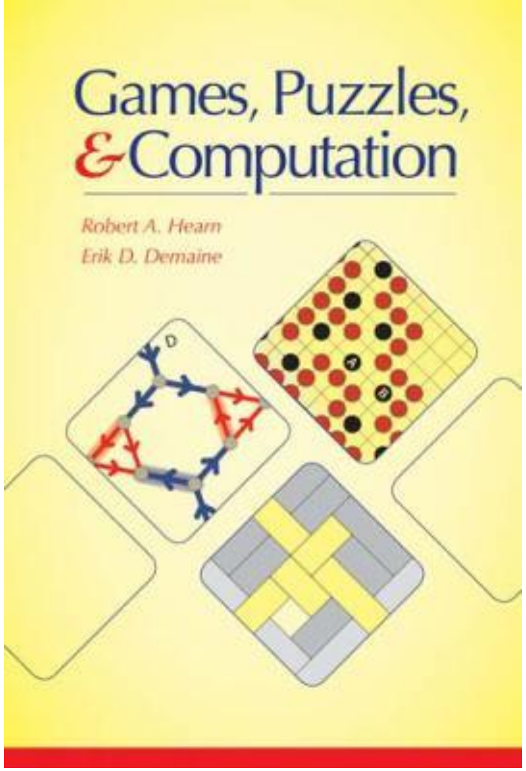
Games, Puzzles & Computation



IPA Advanced Course on Algorithmics and Complexity

Eindhoven, 8 July 2016

Walter Kosters
Hendrik Jan Hoogeboom

LIACS, Universiteit Leiden

schedule

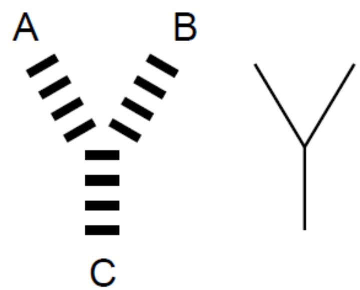
		Framework
10:00-11:00	HJ	Constraint logic, classes
11:15-12:15	W	Gadgets, planarity, exercises
12:30-13:30	*lunch*	
		Concrete Games
14:00-14:45	HJ	Tip-Over is NP-complete
15:00-16:00	W	Rush Hour is PSPACE-complete
		Plank puzzle
16:00-		Wrap-up

games & complexity classes

domino computing

Computing with Planar Toppling Domino Arrangements

William M. Stevens



fork

challenge:
(no) timing & (no) bridges

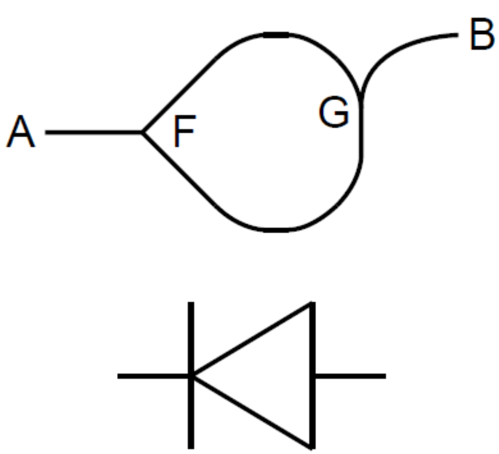


Fig. 3. A one way line

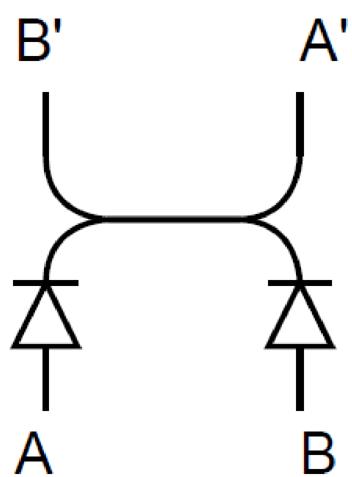


Fig. 4. A single line crossover

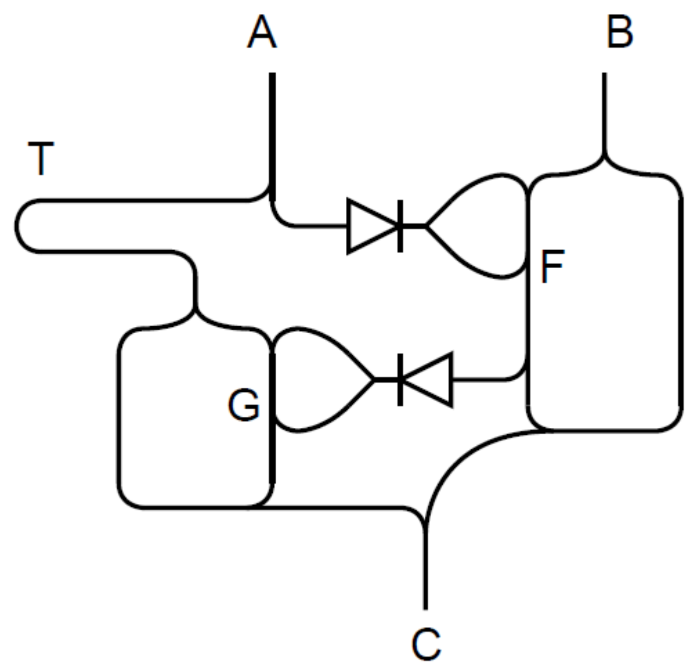


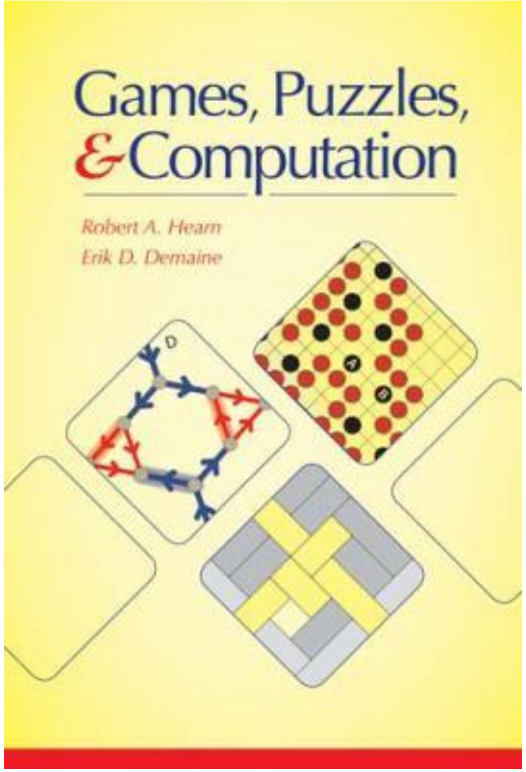
Fig. 5. A both mechanism

Unconventional Computation, 10th International Conference
UC 2011, Turku, Finland, June 6-10, 2011

Games, Puzzles, & Computation

Robert A. Hearn
Erik D. Demaine

(2009, AKPeters)



E. Demaine and R.A. Hearn. Constraint Logic: A Uniform Framework for Modeling Computation as Games. In: Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, June 2008.

R.A. Hearn. Games, Puzzles, and Computation PhD thesis, MIT, 2006.

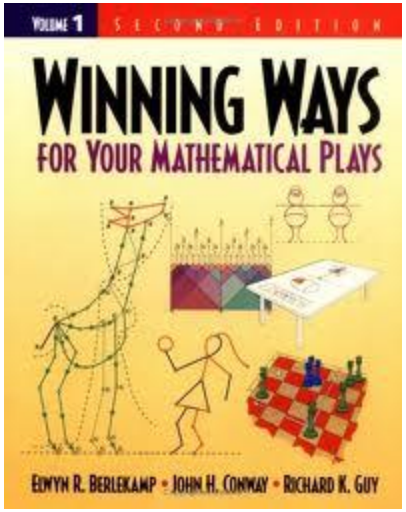
what is a game?

characteristics
- bounded state
- moves, repetition
- players, goal

study the complexity of

- simulation (0p) *game of life*
- puzzles (1p) *rush hour*
- board games (2p) ‘generalized’ *chess*
- teams

fields



combinatorial game theory

algorithms

mathematical theory

economic game theory

von Neumann, Nash

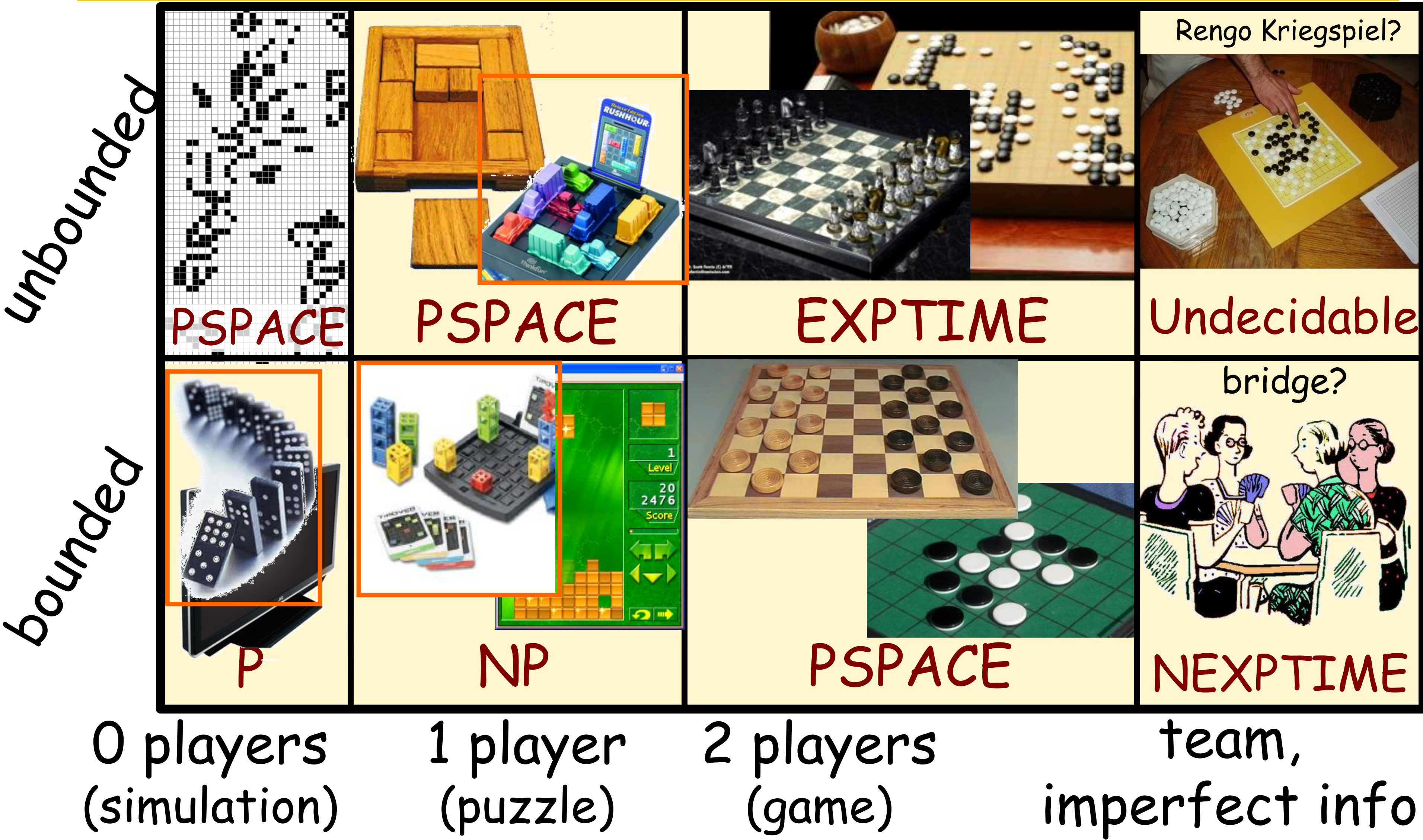
strategy, optimization expected profit

computational complexity

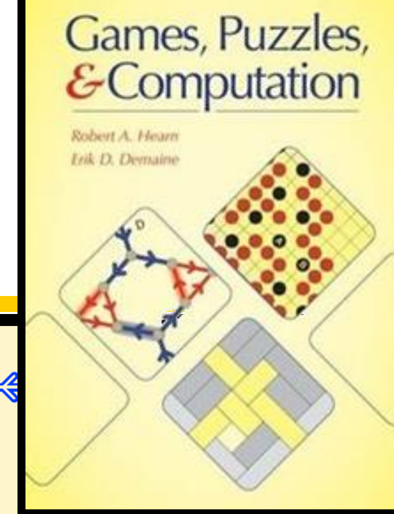
models of computation: *games*

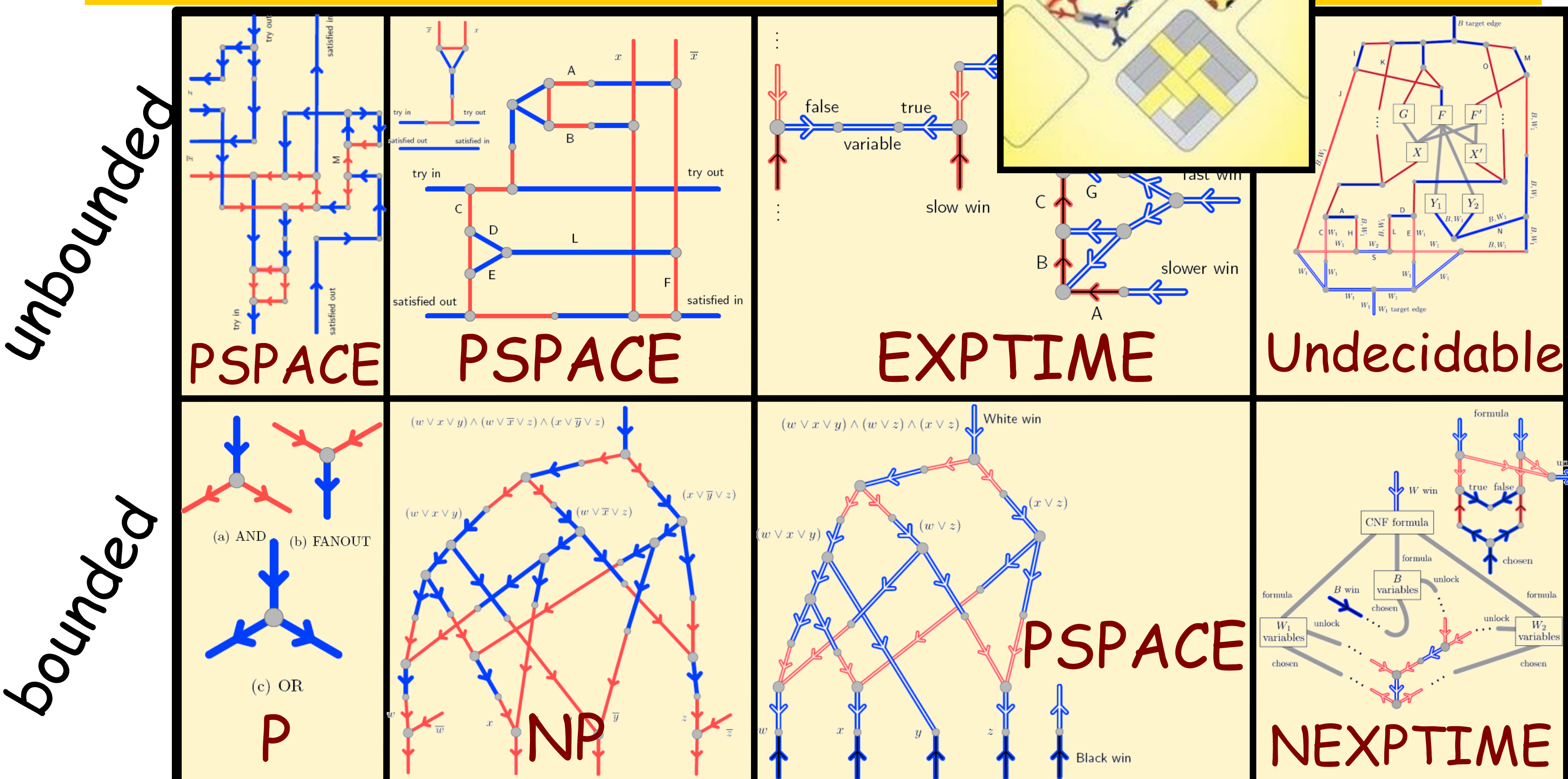
turing machine

Complexity of Games & Puzzles
[Demaine, Hearn & many others]



Constraint Logic
[Hearn & Demaine 2009]





0 players (simulation)

1 player (puzzle)

2 players (game)

team, imperfect info

(details to follow)

constraint logic

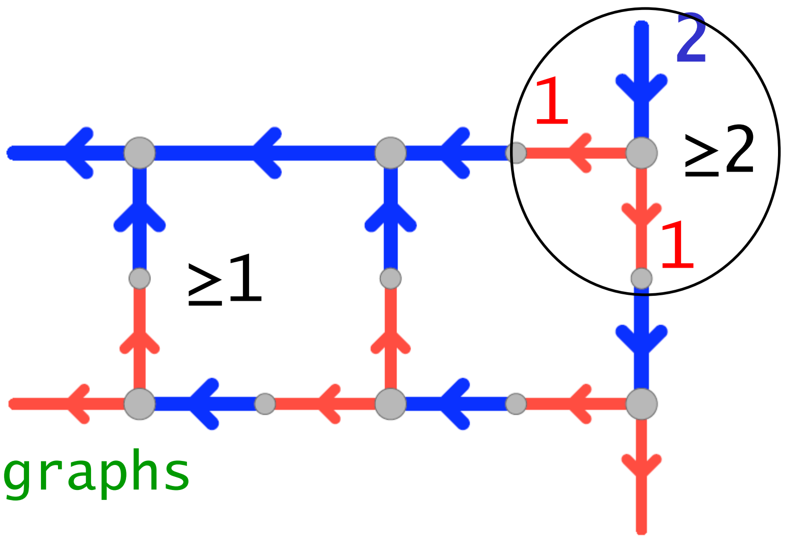
constraint graphs

directed 'oriented'

edge weight 1,2

inflow constraints

legal configuration



game/computation on constraint graphs

move: legal edge reversal

goal: reverse specific edge

NCL - nondet constraint logic

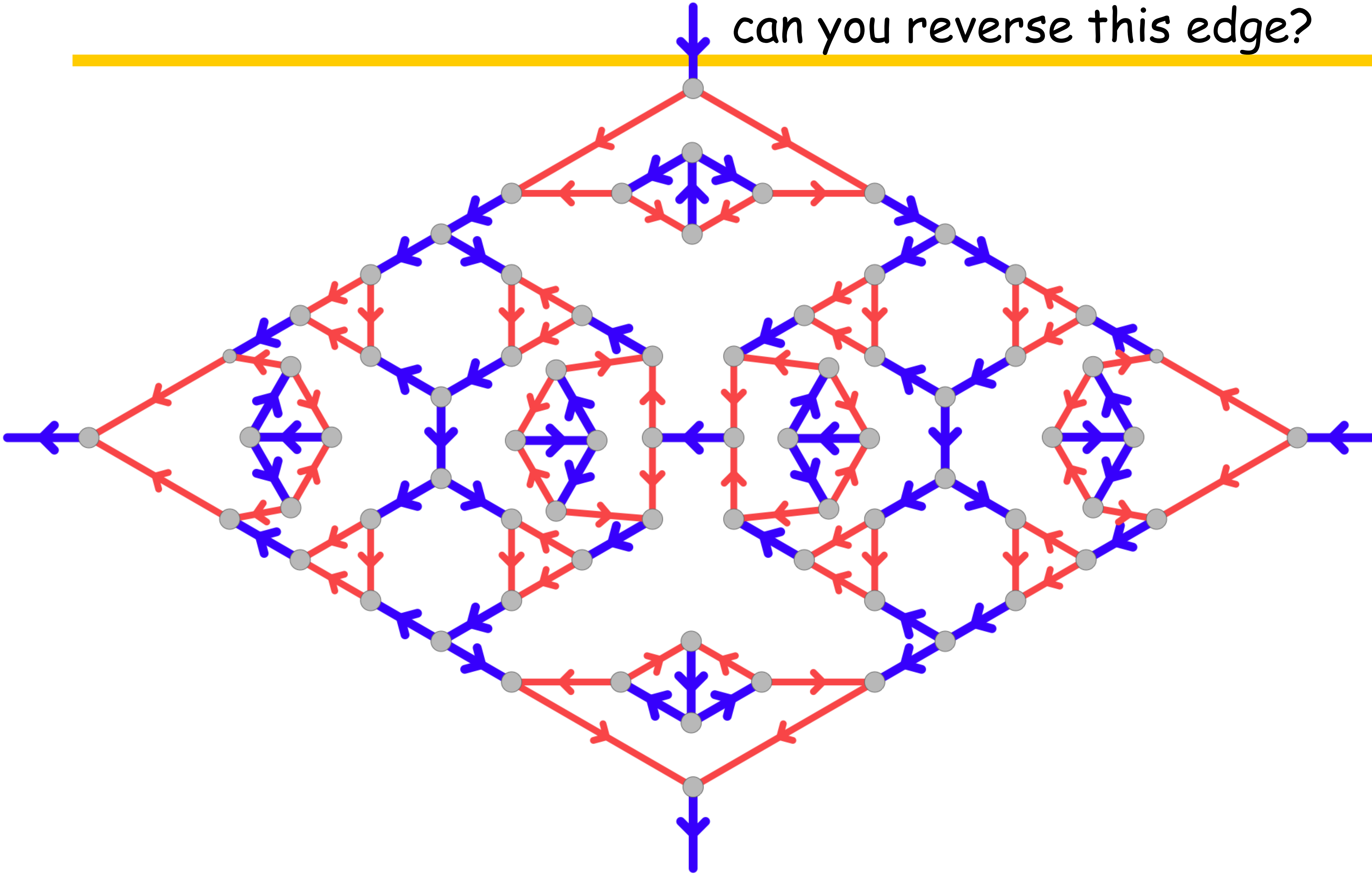
instance: constraint graph G, edge e

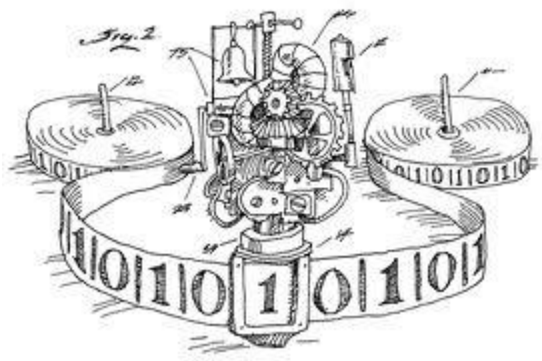
question: sequence which reverses e

BOUNDED NCL

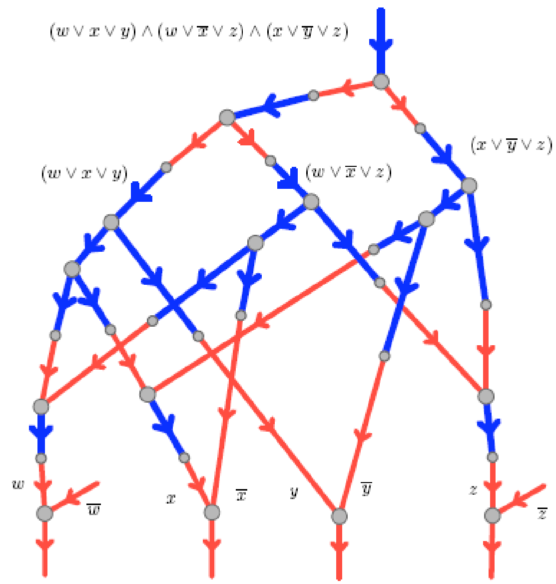
… reverses each edge *at most once*

Decision Problem





$$(w \vee x \vee y) \wedge (w \vee \overline{x} \vee z) \wedge (x \vee \overline{y} \vee z)$$



NP

3SAT

part I

constraint logic

‘graph games’

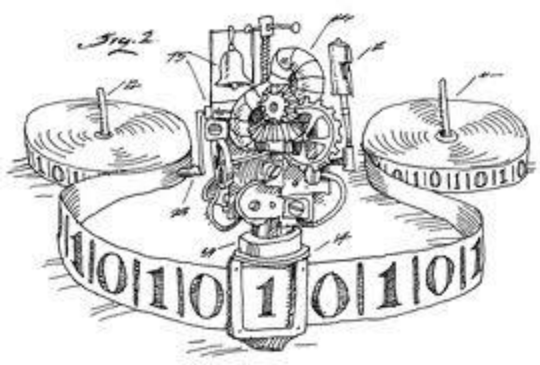
Bounded NCL

part II

games in particular

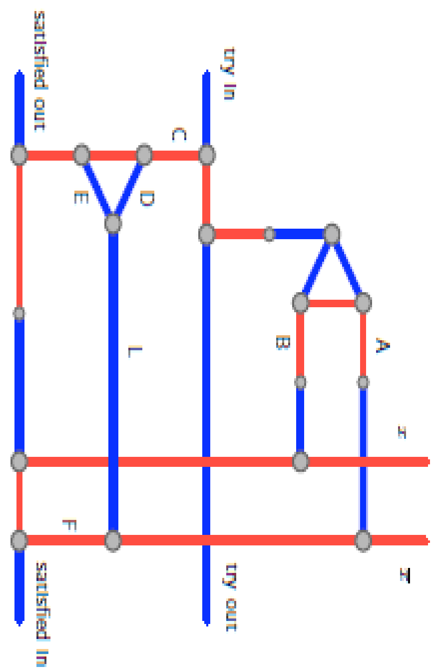
TipOver

PSPACE & Plank Puzzle



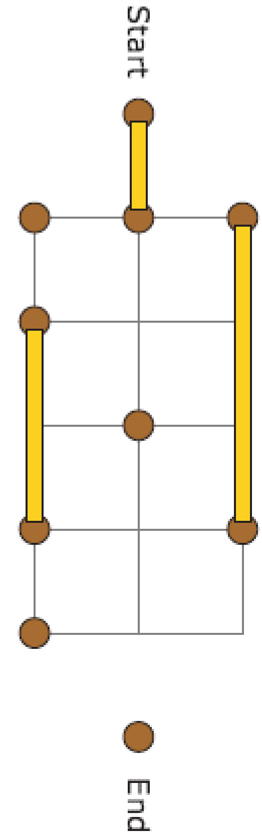
PSPACE

QBF



part I
constraint logic
'graph games'

NCL



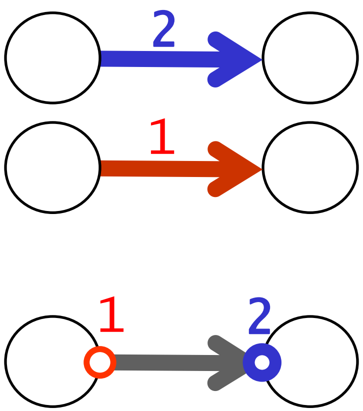
part II
games in particular

plank puzzle
(river crossing)

‘formal’ definition

constraint logic – a graph game

goal: generic ‘graph game’
- several instantiations for specific complexity classes / game types
- reduction to games & puzzles



Hearn & Demaine ‘constraint logic’

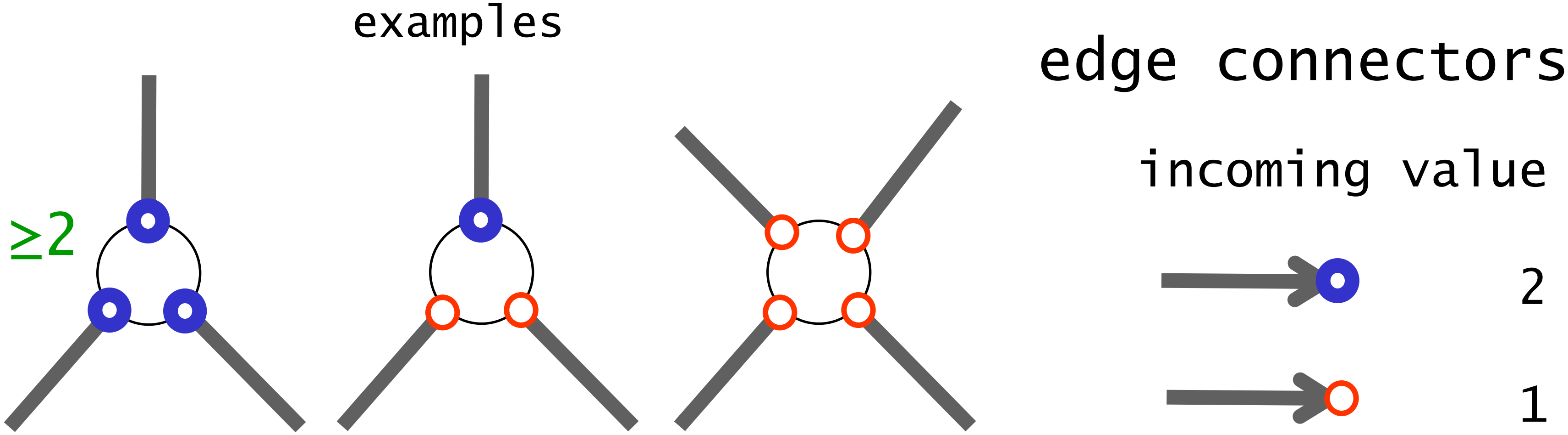
coloured edges / connectors

bounded vs. unbounded

(natural direction of computation)

planarity

basic constraint logic



constraint graph

oriented/directed edges + connectors

vertex constraint

inflow value ≥ 2

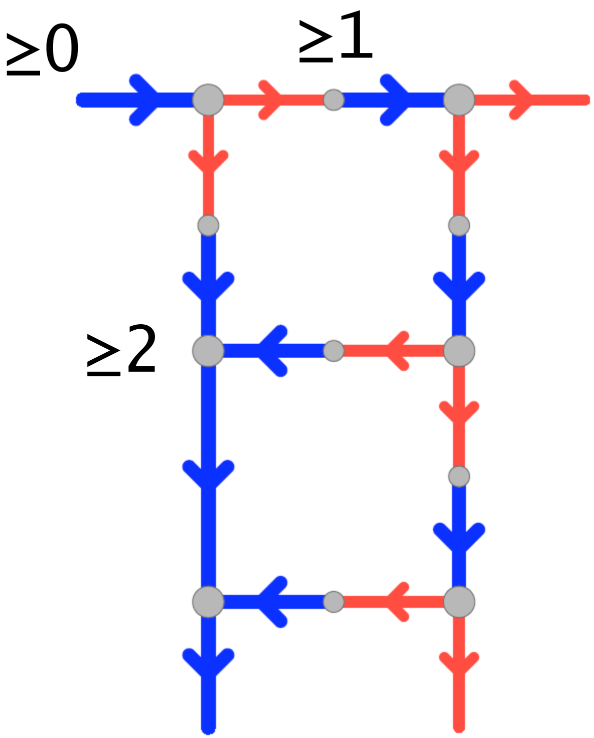
game: legal move

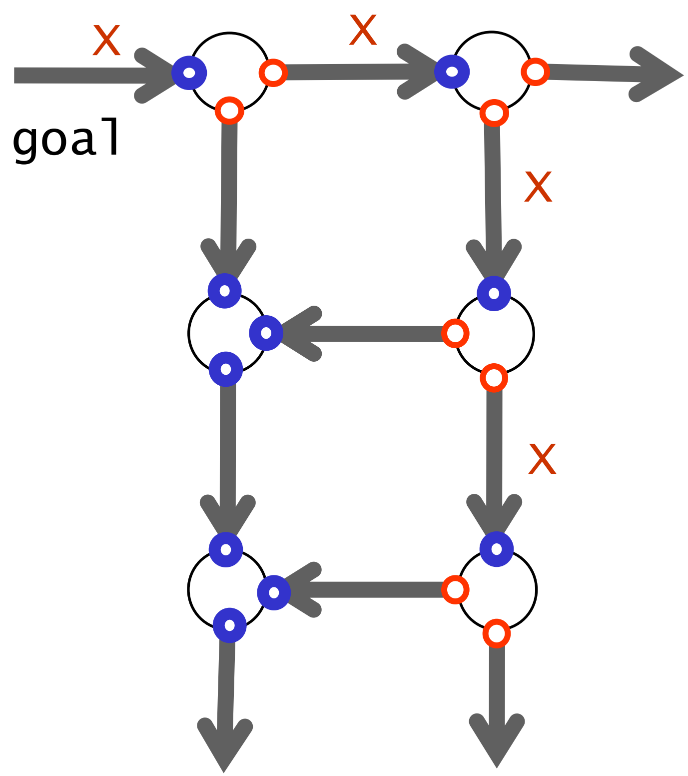
edge reversal satisfying constraint

goal

reversal given edge

‘special’ vertex constraints?



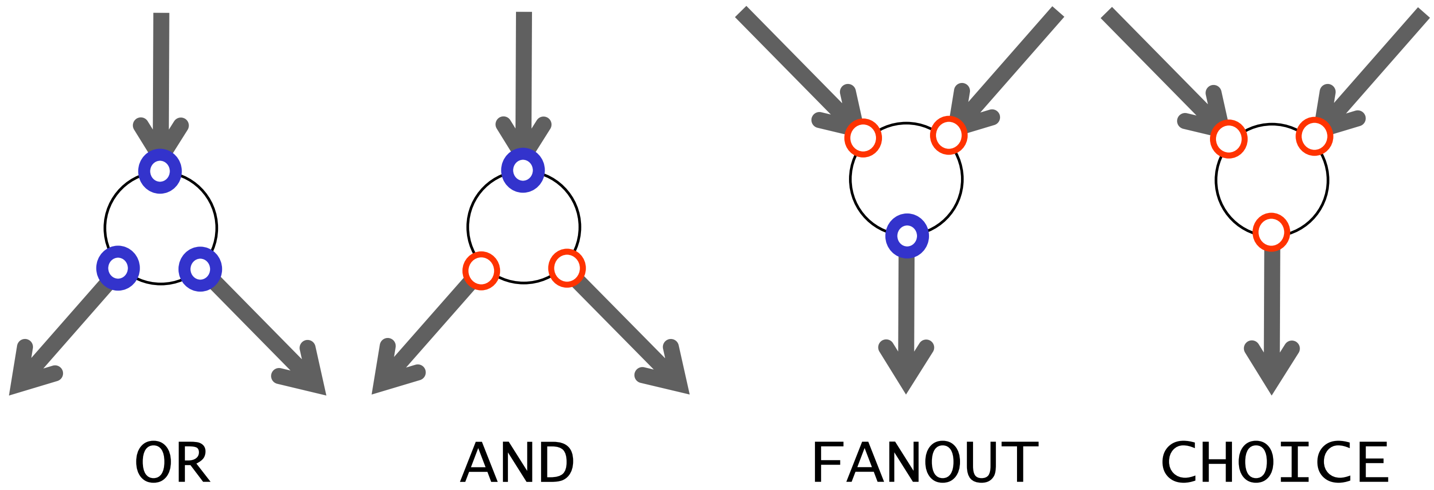


colour conversion

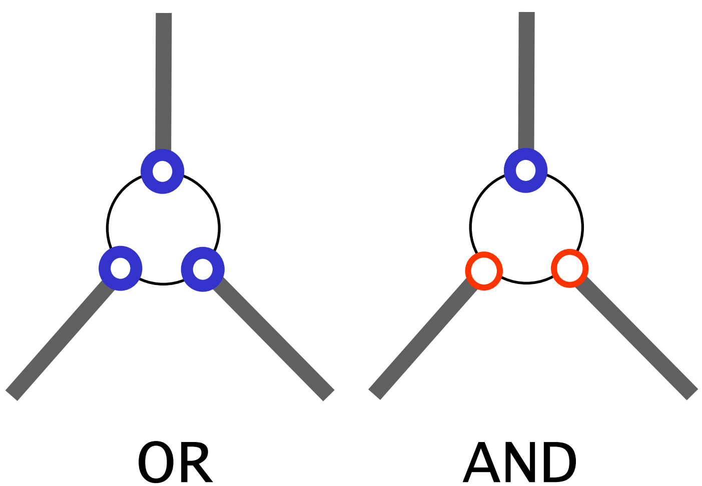
dangling edges *edge terminators*

normal form vertices

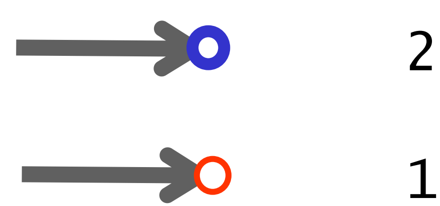
bounded NCL (reverse only once)



NCL



incoming value

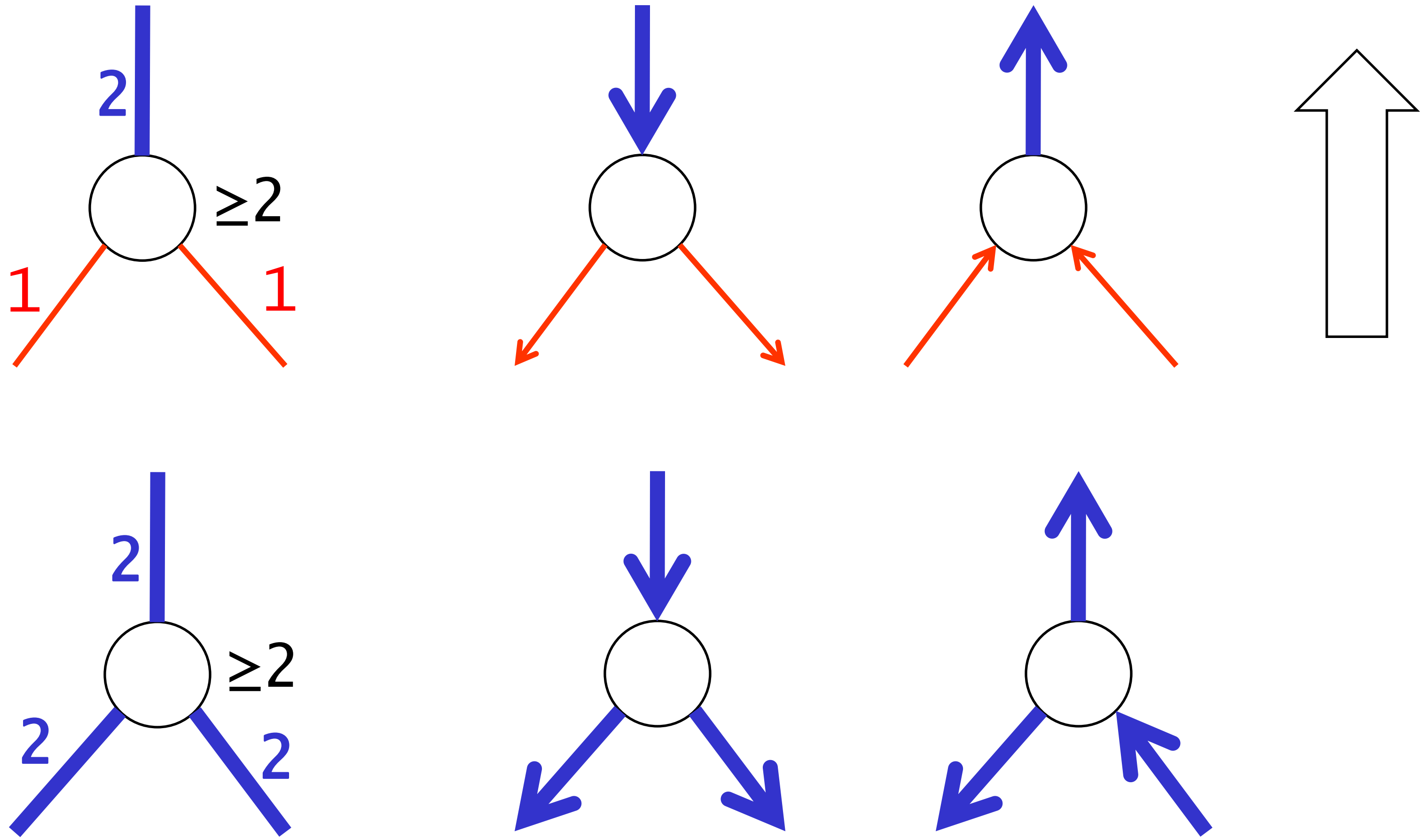


implementing gates

intuitive meaning of vertices

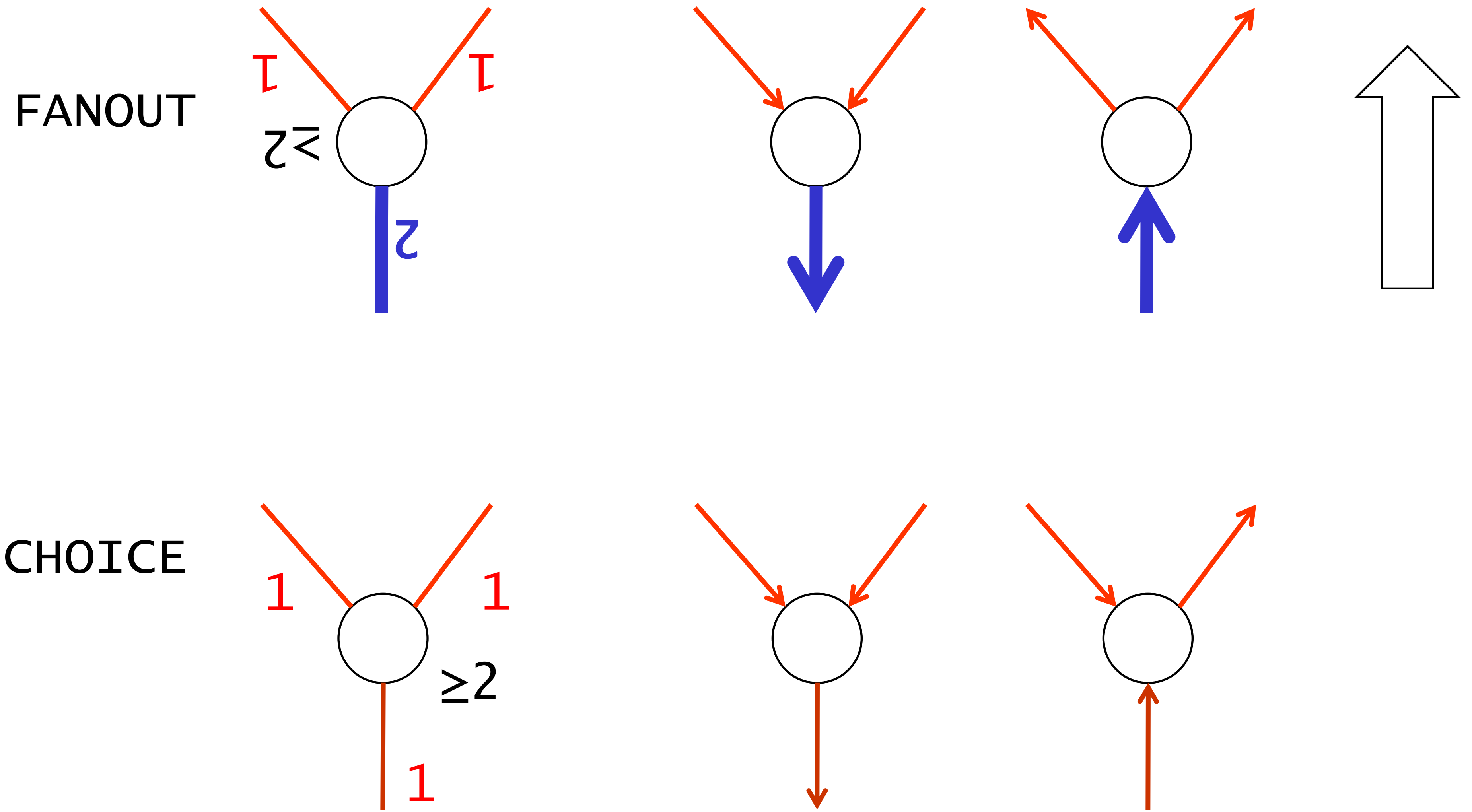
AND

OR



implementing gates

intuitive meaning of vertices



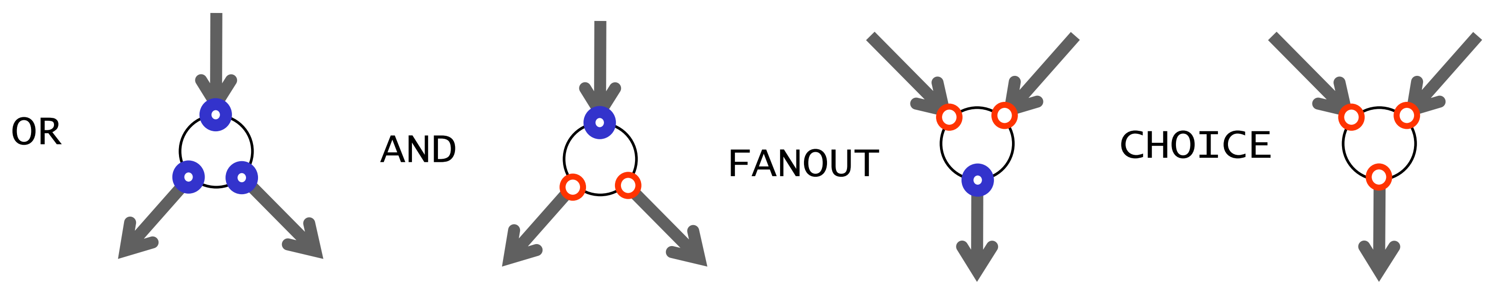
exercise

“emulate” a logical formula as graph game

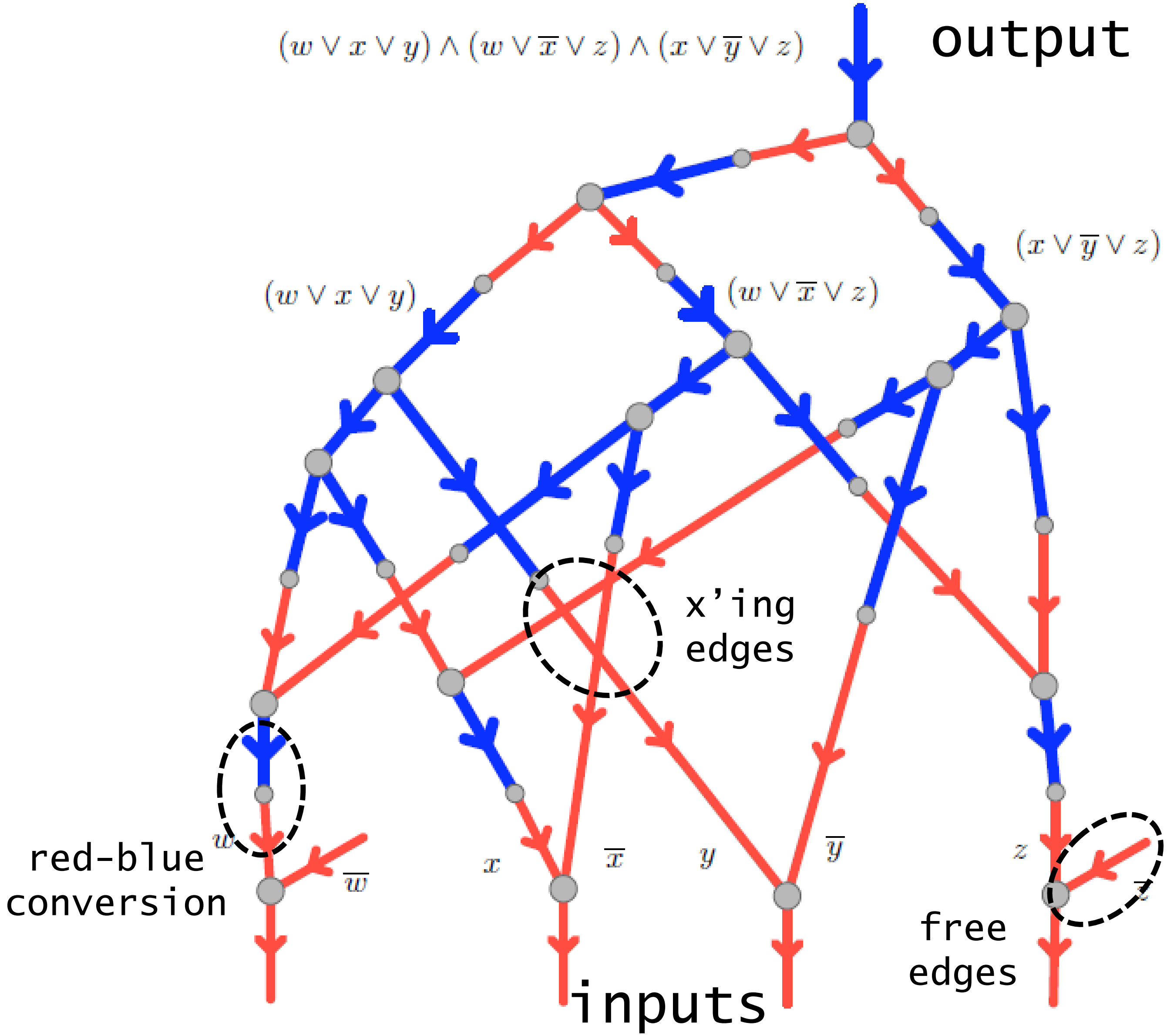
goal:
flip a given edge *iff* formula satisfiable

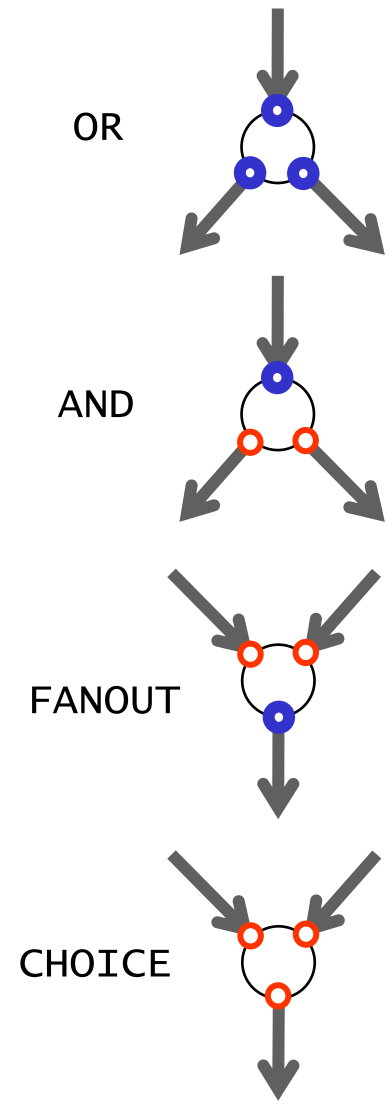
$$(w \vee x \vee y) \wedge (w \vee \neg x \vee z) \wedge (x \vee \neg y \vee z)$$

hint:

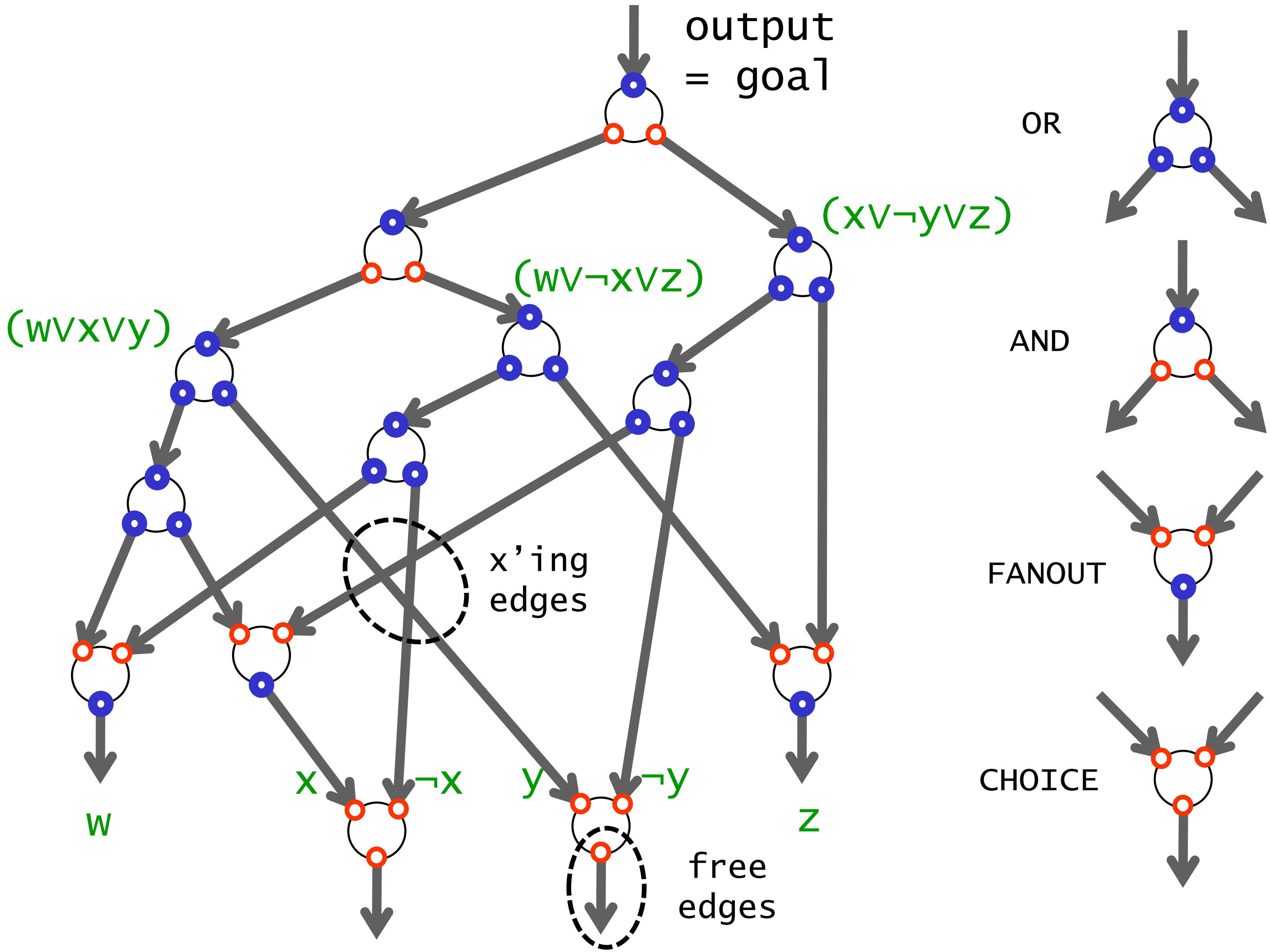


formula constraint graph



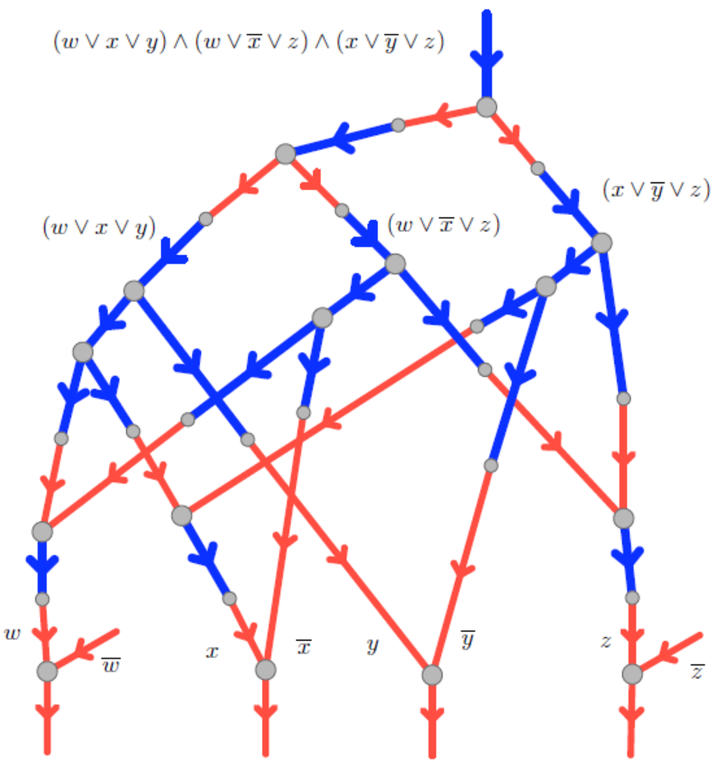


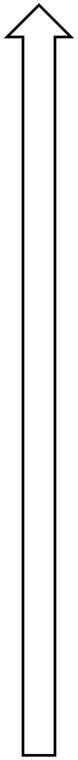
formula constraint graph

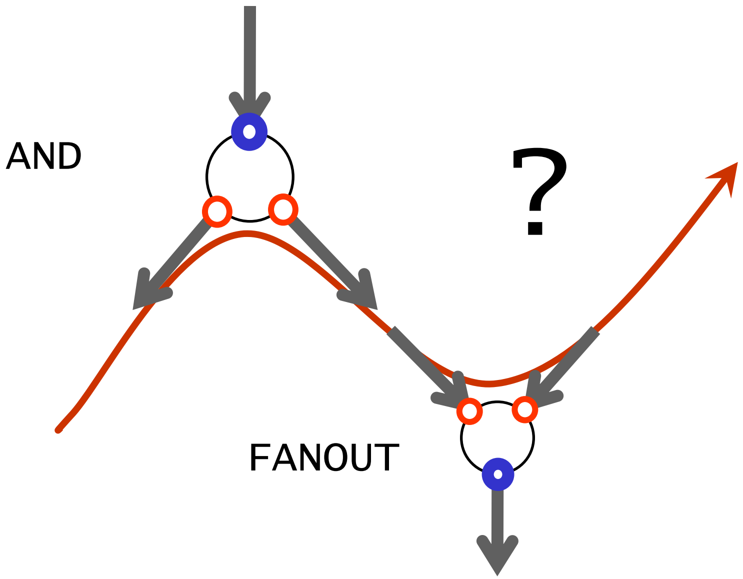


questions

- ‘can’ : not obliged to reverse edges upwards
- can we reverse the ‘wrong way’?
- do we need restriction to reverse edge once?







basic complexity classes

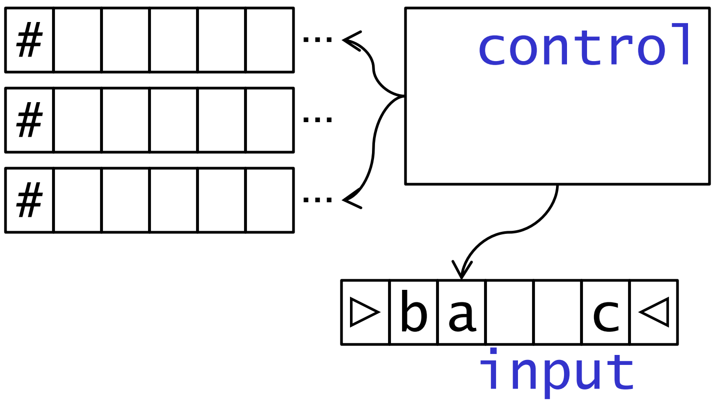
game complexity classes
vs.
TM resources

Cook/Levin NP completeness

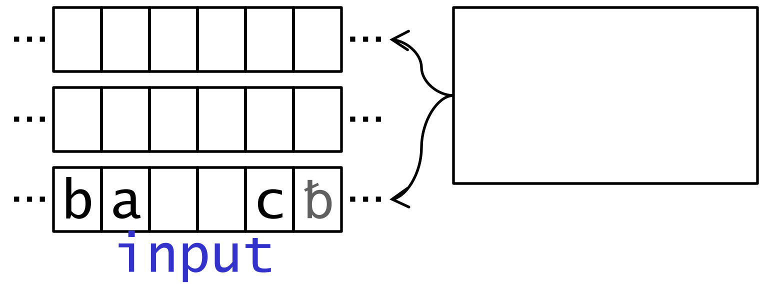
Savitch (N)PSPACE

TM models (H&U)

working tapes



working tapes



<u>space complexity</u>

DSPACE(f) NSPACE(f)

offline
multiple working tapes
single side infinite

<u>time complexity</u>

DTIME(f) NTIME(f)

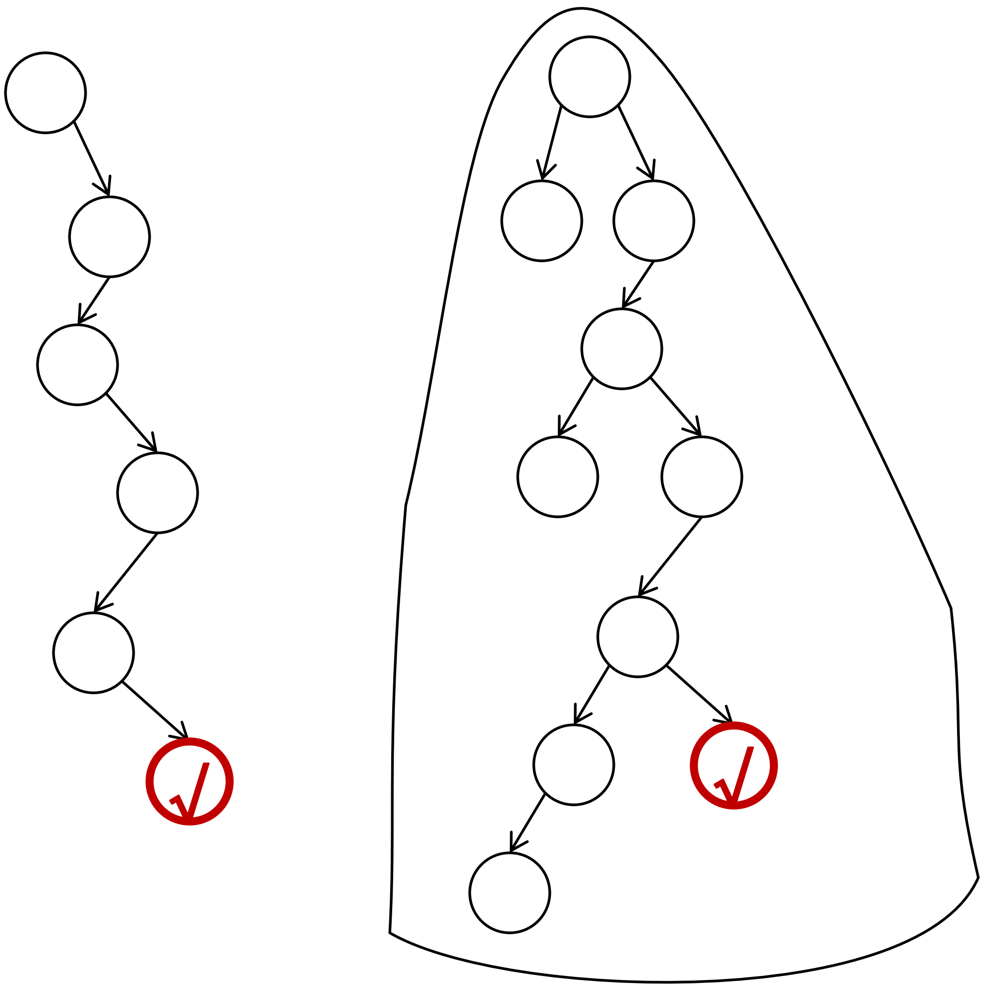
input on tape
multiple working tapes
double sided

for every input word of length n, …

M scans at most f(n) cells on any storage tape …

M makes at most f(n) moves before halting …

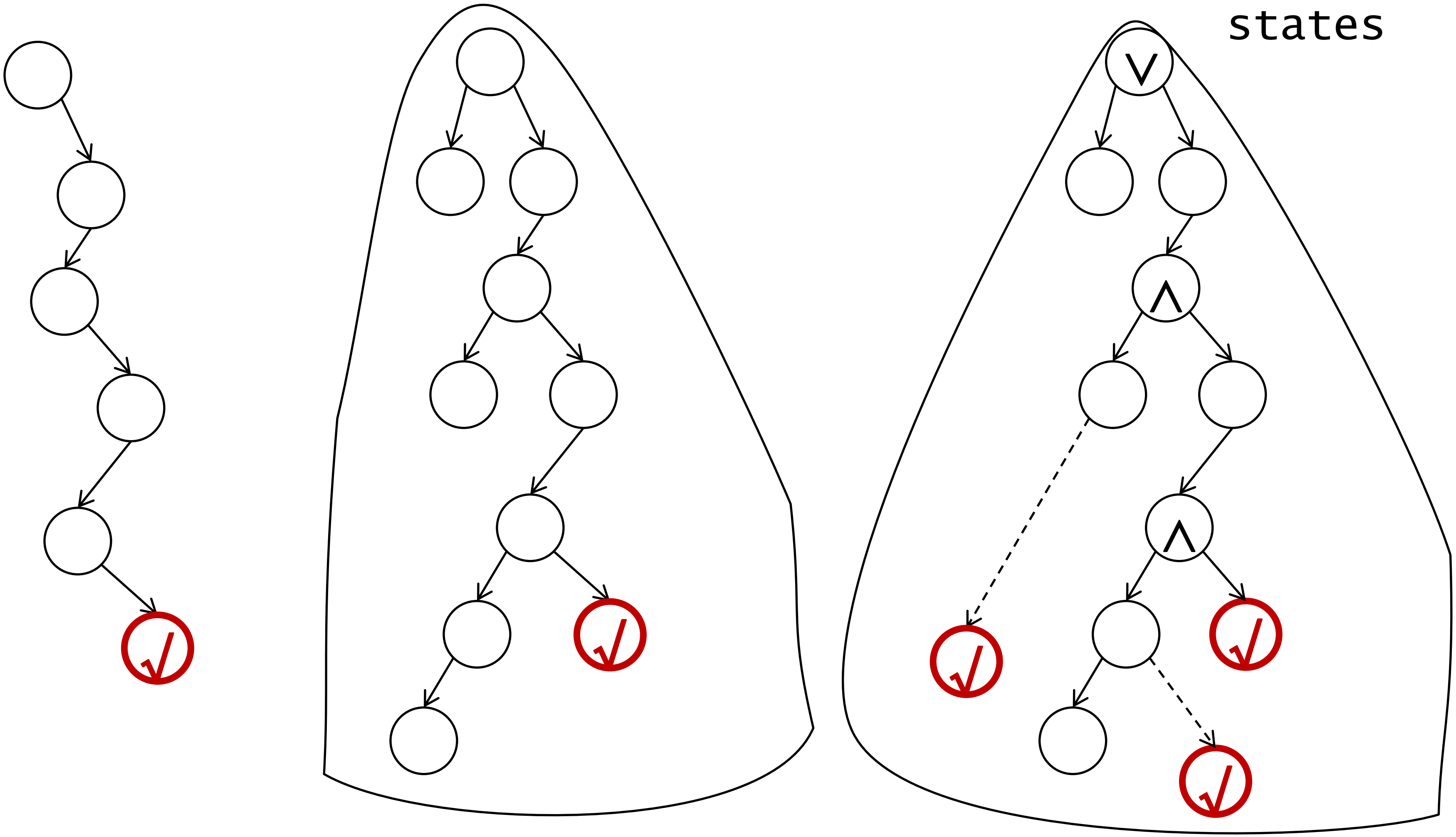
computation tree



determinism nondeterminism

computation tree

existential and *universal* states



determinism nondeterminism alternation

dimensions

existential and *universal* states
computation = tree

	log.	*polynomial*		*exp.*
	space	time	space	time
determinism	L	P	PSPACE	EXPTIME
nondeterminism	NL	NP	NPSPACE	NEXPTIME
alternation	AL	AP	APSPACE	AEXPTIME

AL AP APSPACE AEXPTIME

P ⊆ NP ⊆ PSPACE ⊆ EXPTIME ⊆ NEXPTIME ⊆ EXPSPACE

NPSPACE NEXPSPACE

A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. 'Alternation', Journal of the ACM, Volume 28, Issue 1, pp. 114–133, 1981.

game categories

game categories and their natural complexities

Rush Hour
River Crossing

unbounded	**PSPACE**	**PSPACE**	**EXPTIME**	**undecid**
bounded	**P**	**NP**	**PSPACE**	**NEXPTIME**
#	**zero** *simulation*	**one** *puzzle*	**two** *game*	**team** *imperfect informat.*

TipOver

NL ⊆ P ⊆ NP ⊆ PSPACE ⊆ EXPTIME ⊆ NEXPTIME

game categories

game categories and their natural complexities

(polynomial)
TM resources

Rush Hour
River Crossing

unbounded **SPACE**	**PSPACE**	**PSPACE** NPSPACE	**EXPTIME** APSPACE	undecid
bounded **TIME**	**P**	**NP**	**PSPACE** AP	NEXPTIME
#	**zero** *simulation* ***determ.***	**one** *puzzle* ***nondeterm.***	**two** *game* ***alternat.***	team *imperfect informat.*

TipOver

NL ⊆ P ⊆ NP ⊆ PSPACE ⊆ EXPTIME ⊆ NEXPTIME

NPSPACE

Savitch

$NSPACE(s(n)) \subseteq SPACE(s^2(n))$

can we reach a halting configuration?
at most exponentially many steps $s(n)|\Sigma|^{s(n)}$

solve recursively "re-use space"

reach(*ini*,*fin*,1) = step(*ini*,*fin*)
reach(*ini*,*fin*,2k)
    foreach configuration mid
        test reach(*ini*,*mid*,k) ∧ reach(*mid*,*fin*,k)

stack depth s(n) of configs, each size s(n)

NPSPACE = PSPACE

$NSPACE(s(n)) \subseteq ATIME(s^2(n))$ "parallel in time"

3SAT

literal
(negated) variable

clause

$(w \vee x \vee y) \wedge (w \vee \neg x \vee z) \wedge (x \vee \neg y \vee z) \wedge \ldots$

3 conjunctive normalform

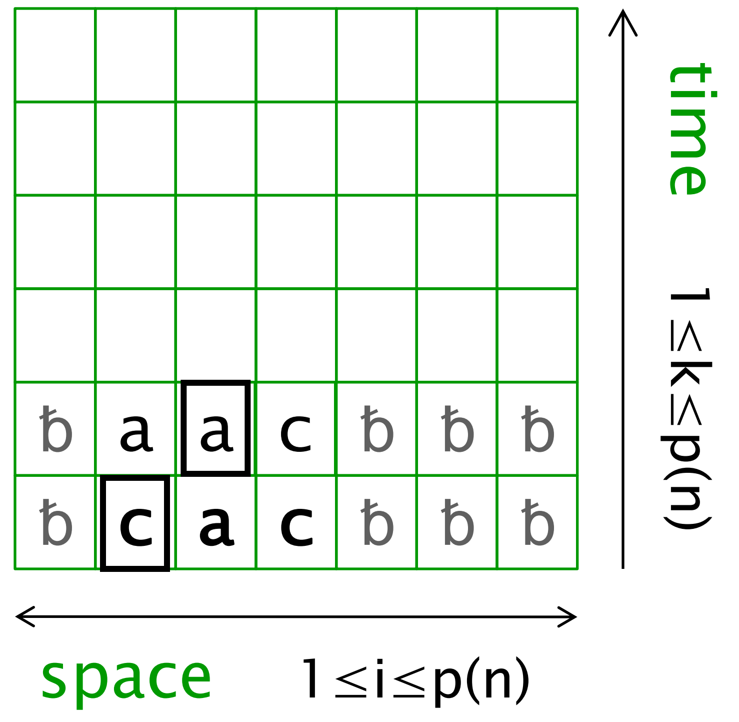
3SAT
given: given formula ϕ in 3CNF
question: is ϕ satisfiable?
(can we find a variable assignment making formula true)

Cook/Levin 3SAT is NP-complete

specify computation

at step k …

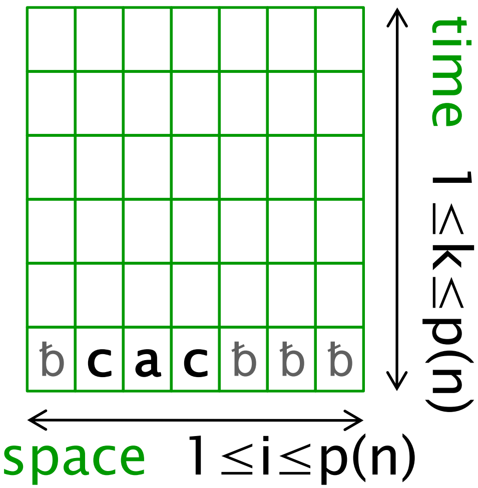
T_{iak}	cell i contains a
H_{ik}	head at position i
Q_{qk}	state q



Cook/Levin

specify computation
at step k …

T_{iak} cell i contains a
H_{ik} head at position i
Q_{qk} state q



(p,a,p,a,0) for each p,a

conjunction of

$T_{ix[i]0}$ initial tape $x[i]=x_i$ or $x[i]=ƀ$

Q_{q00} initial state

H_{00} initial position

$T_{iak} \rightarrow \neg T_{ibk}$ single symbol $a \neq b$

$Q_{pk} \rightarrow \neg Q_{qk}$ single state $p \neq q$

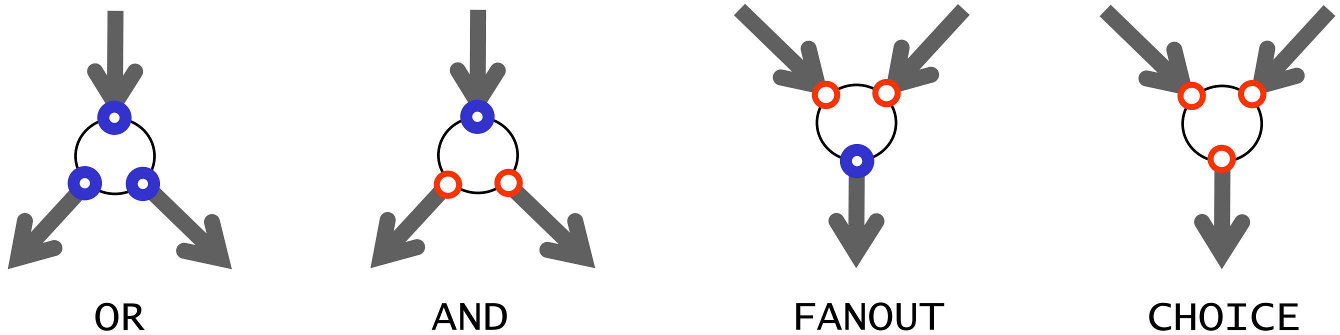
$H_{ik} \rightarrow \neg H_{jk}$ single head $i \neq j$

$T_{iak} \wedge T_{ib.k+1} \rightarrow H_{ik}$ changed only if written $a \neq b$

$H_{ik} \wedge Q_{pk} \wedge T_{iak} \rightarrow \bigvee_{(p,a,q,b,d)} H_{i+d.k+1} \wedge Q_{q.k+1} \wedge T_{ib.k+1}$

$Q_{h.p(n)}$ accept

conclusion



BOUNDED NCL - nondet constraint logic

instance: constraint graph G, edge e

question: sequence which reverses *each edge at most once*, ending with e

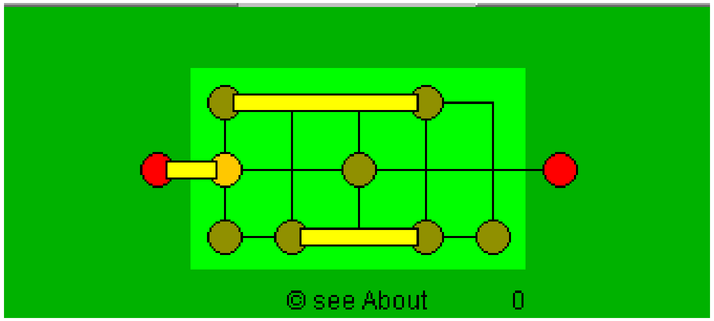
- reduction from 3SAT into Bounded NCL
- Bounded NCL is in NP

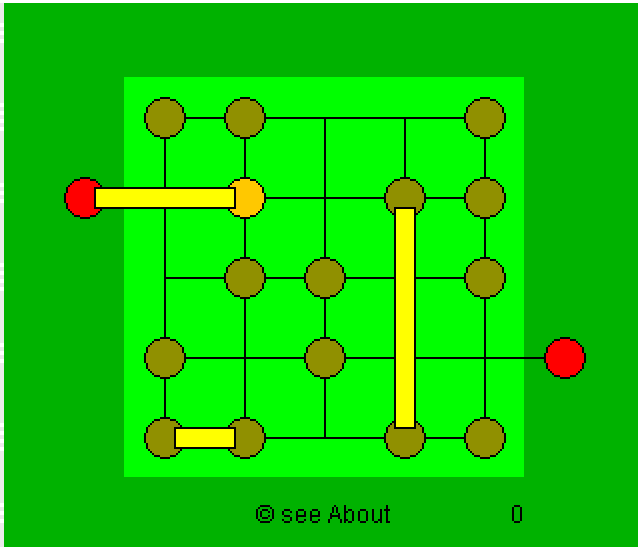
Bounded NCL is NP-complete

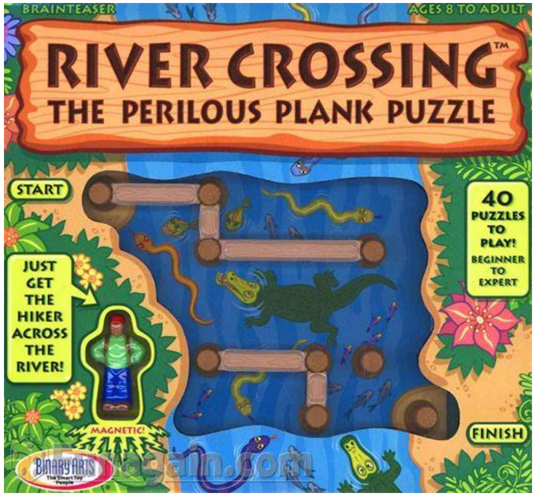
however: topling domino's cannot cross

plank puzzles

http://www.clickmazes.com/planks/ixplanks.htm



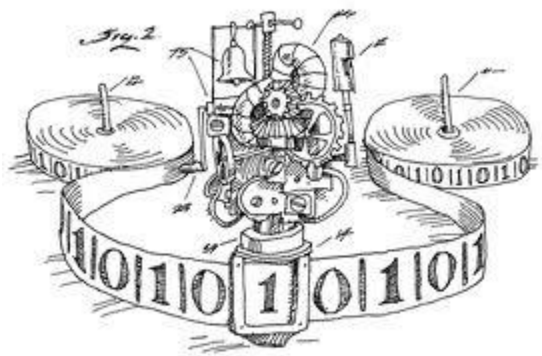




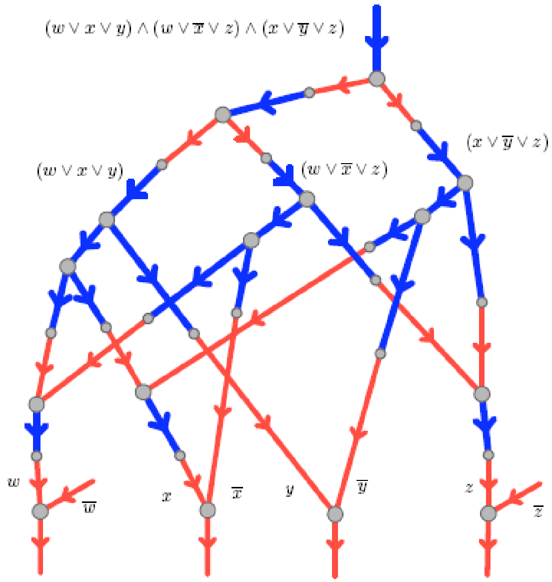
Your challenge is to find a route across a crocodile infested swamp using just a handful of rather short planks. Fortunately the planks are light enough to move around, and the swamp is full of old tree-stumps which will support the planks to form temporary bridges. So by careful planning, and re-use of planks, you might just find a route. Needless to say you can only move planks you can physically reach, so try not to leave any too far behind.

TipOver is NP-Complete

NP & TipOver



$(w \vee x \vee y) \wedge (w \vee \overline{x} \vee z) \wedge (x \vee \overline{y} \vee z)$



NP

3SAT

part I

constraint logic

'graph games'

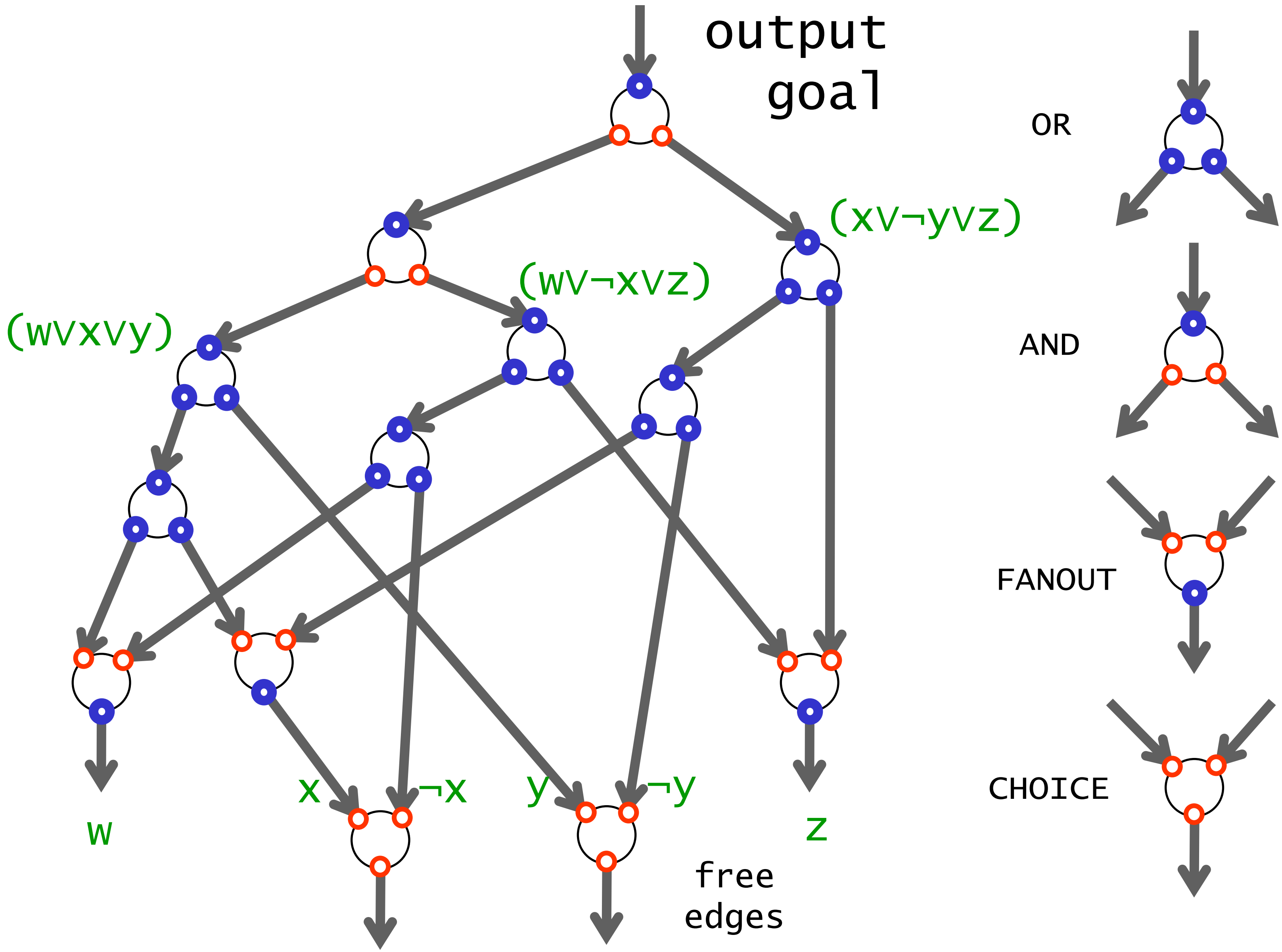
Bounded NCL

part II

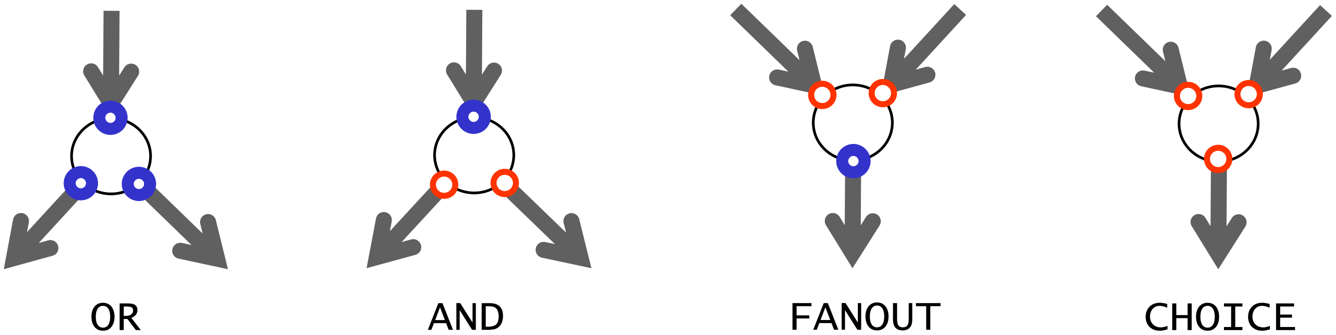
games in particular

TipOver

formula constraint graph



conclusion



BOUNDED NCL - nondet constraint logic

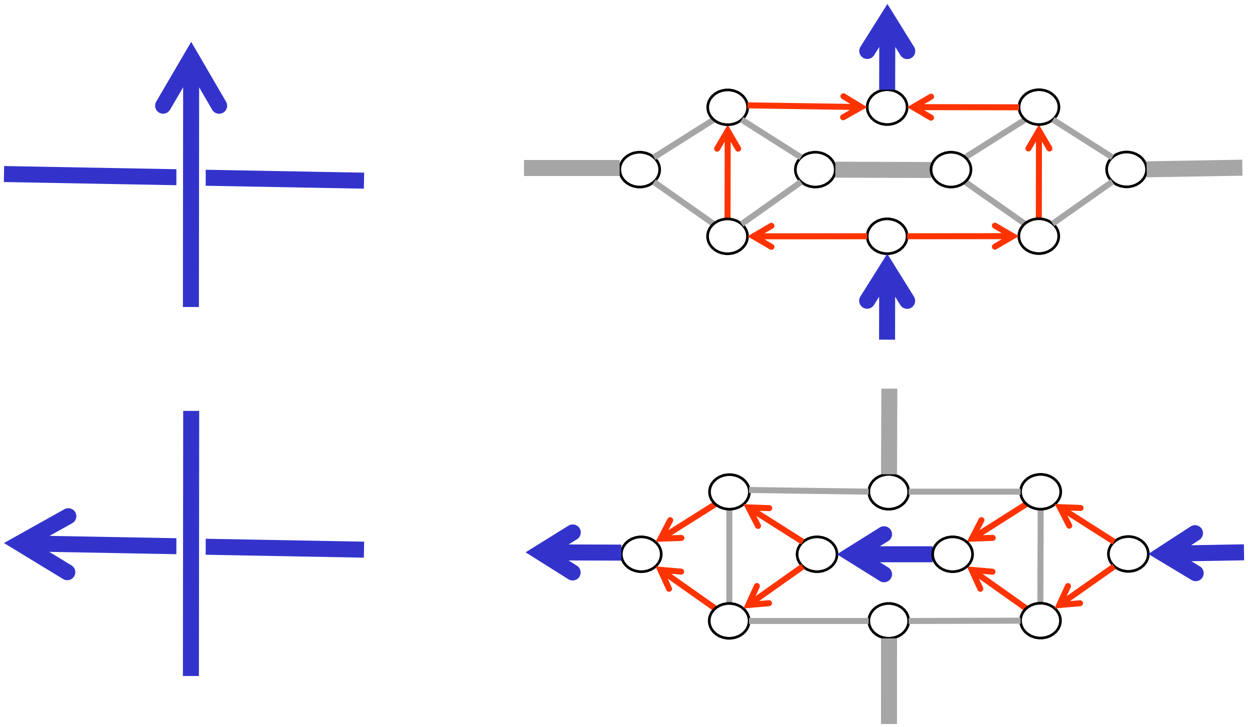
instance: constraint graph G, edge e

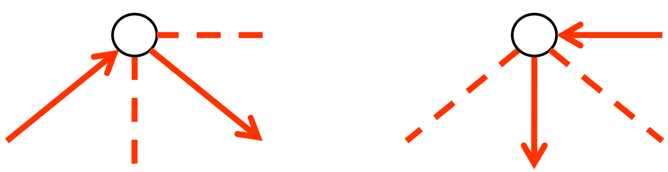
question: sequence which reverses *each edge at most once*, ending with e

Bounded NCL is NP-complete

however: topling domino's cannot cross

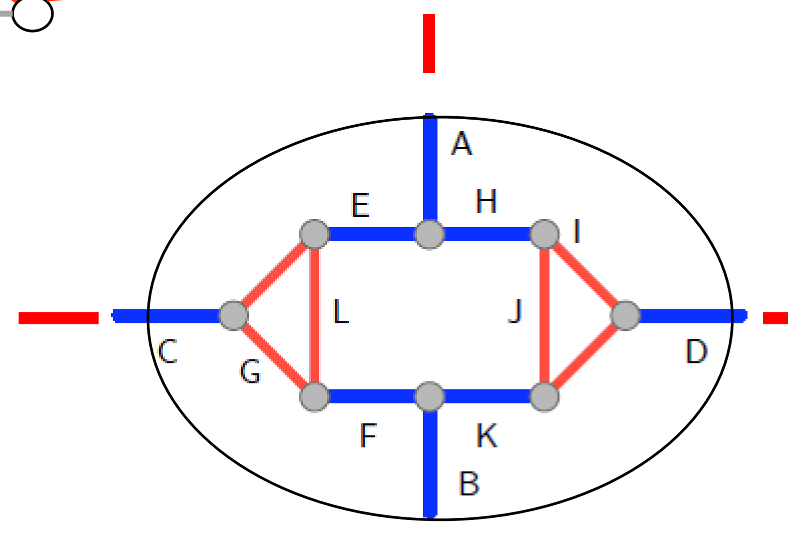
planar crossover gadget





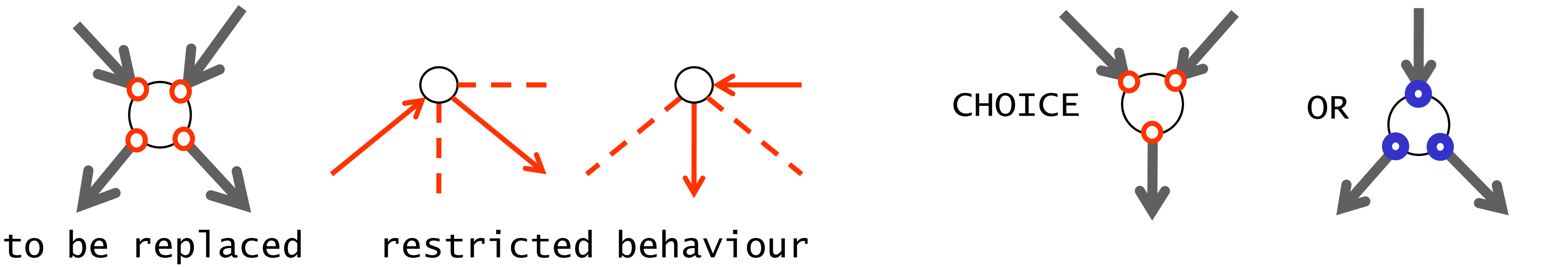
necessary behaviour

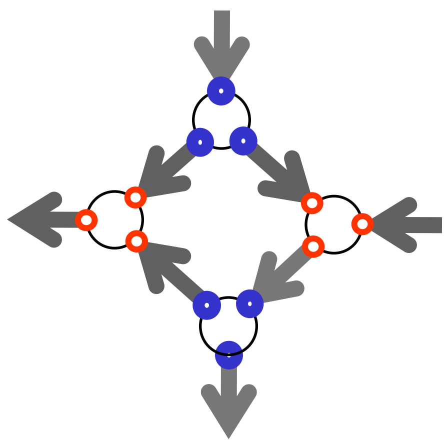
either of them or both!

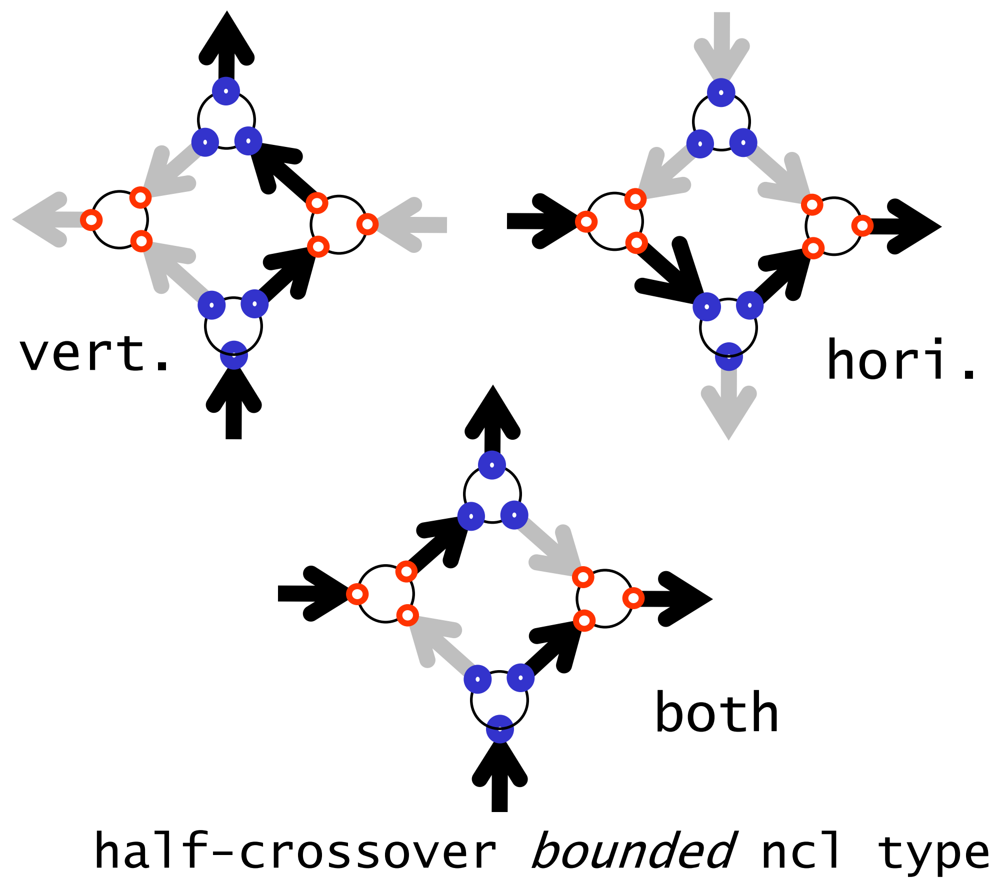


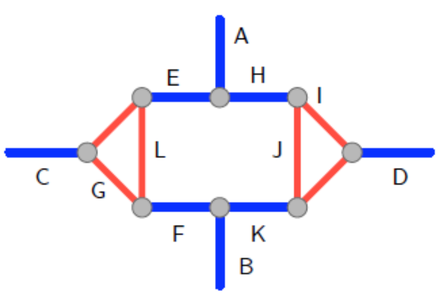
half-crossover ncl type

bounded NCL half-crossover



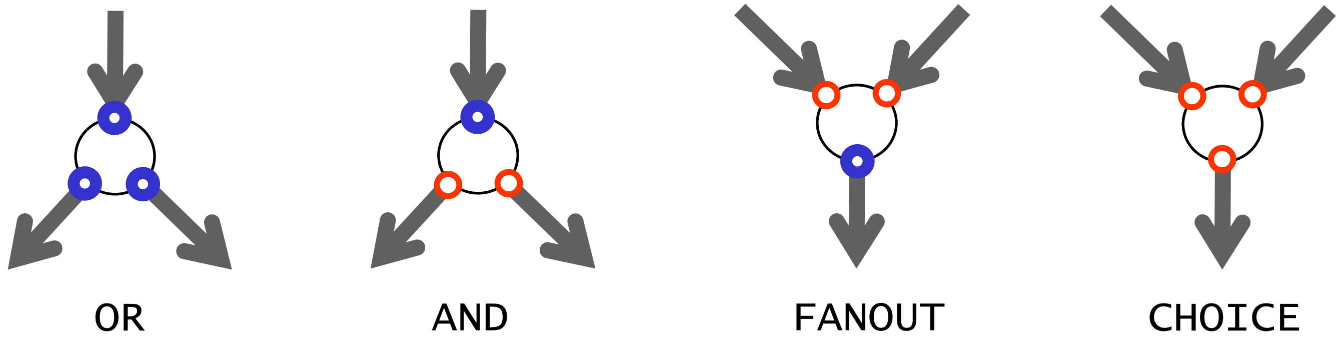






(b) Half-crossover

conclusion(2)



BOUNDED NCL - nondet constraint logic

instance: constraint graph G, edge e

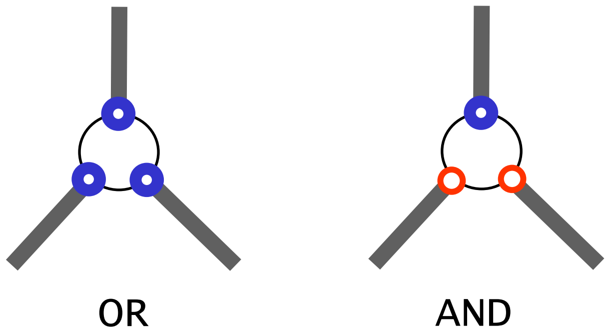
question: sequence which reverses *each edge at most once*, ending with e

Bounded NCL is NP-complete,

even for planar graphs,

with restricted vertices

conclusion (next hour)



NCL - nondet constraint logic

instance: constraint graph G, edge e

question: sequence which reverses e

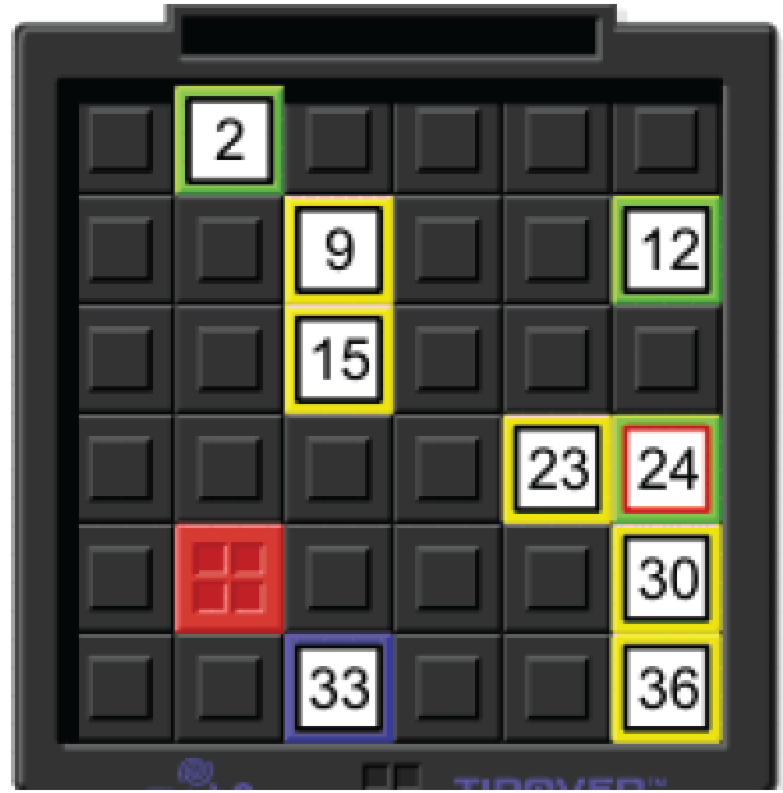
NCL is PSPACE-complete,

even for planar graphs,

with restricted vertices

application: TipOver

http://www.puzzles.com/products/tipover/PlayOnLine.htm



- 2
- 3
- 4
- goal

24 initial position

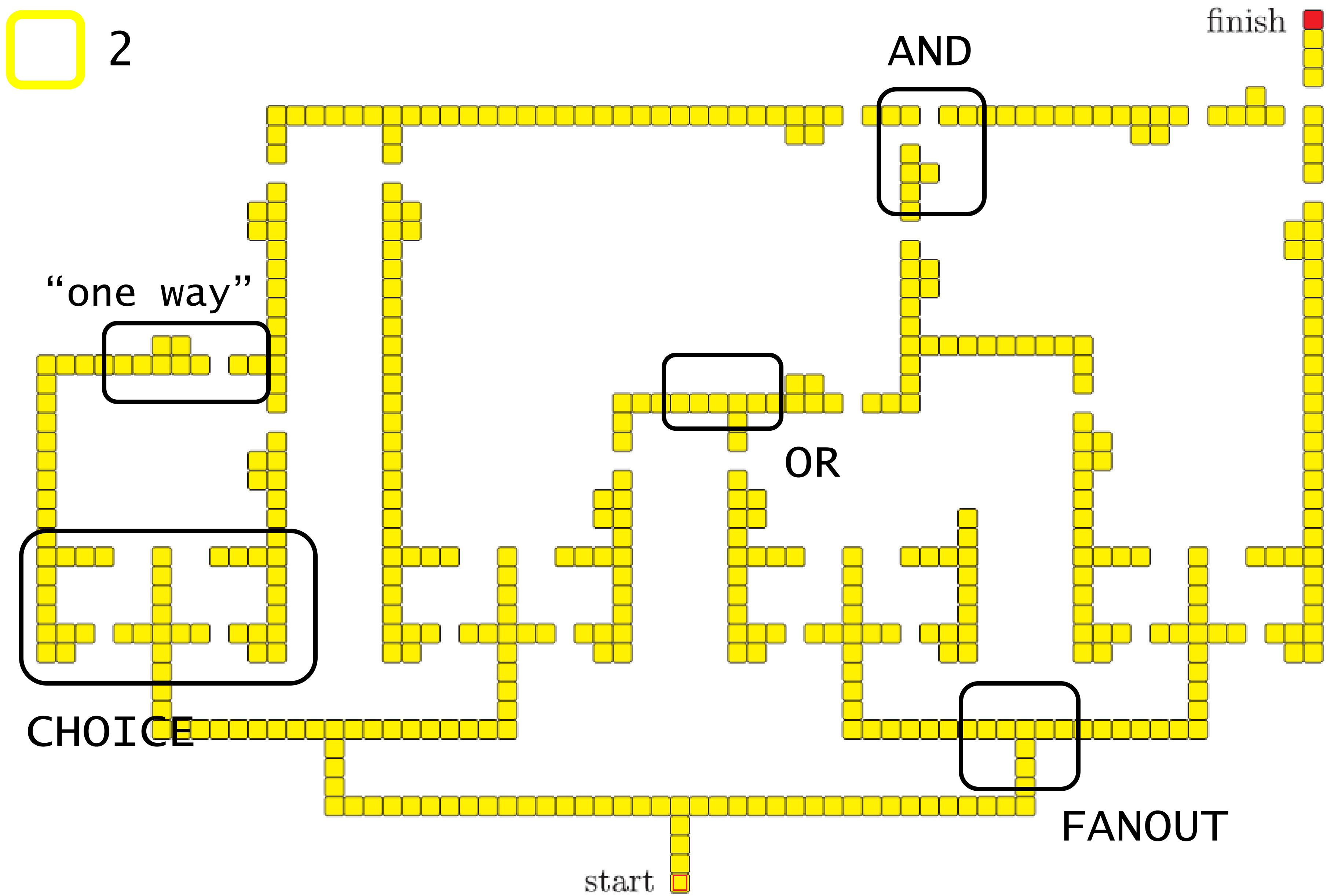


Figure 9-7: TipOver puzzle for a simple constraint graph.

gadgets: "one way", OR

invariant:

- can be reached ⇔ can be inverted
- all visited positions remain connected

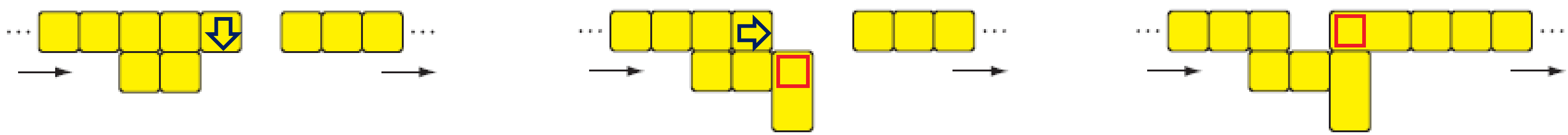
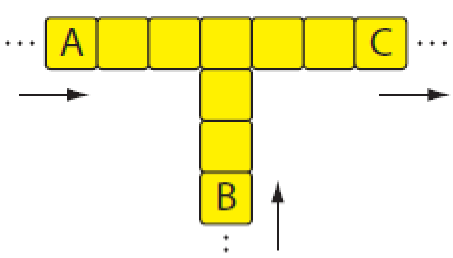
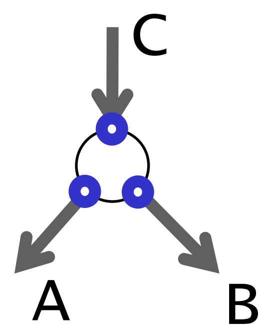


Figure 9-3: A wire that must be initially traversed from left to right. All crates are height two.

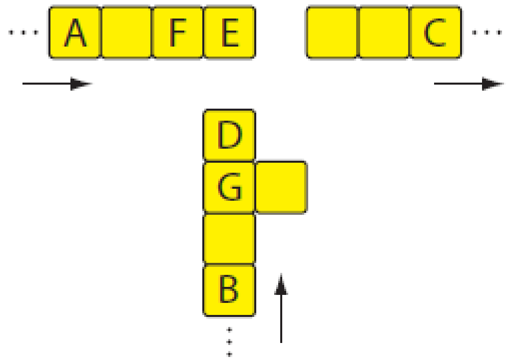


(a) OR gadget. If the tipper can reach either A or B, then it can reach C.

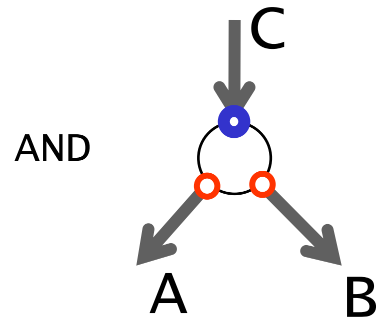
OR



gadgets: AND



(b) AND gadget. If the tipper can reach both A and B, then it can reach C.



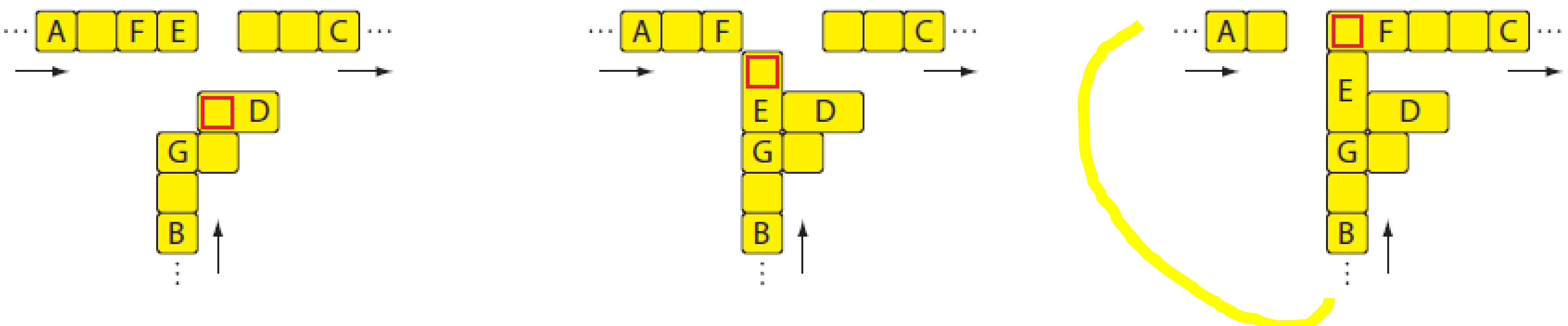
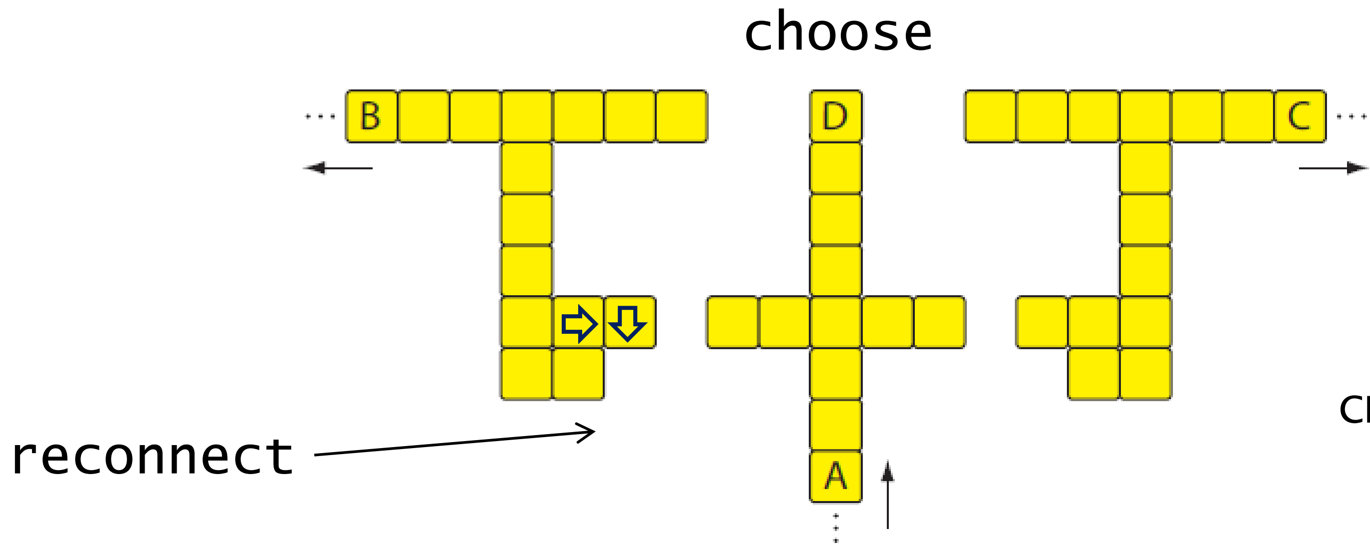


Figure 9-5: How to use the AND gadget.

remains connected

gadgets: CHOICE, FANOUT



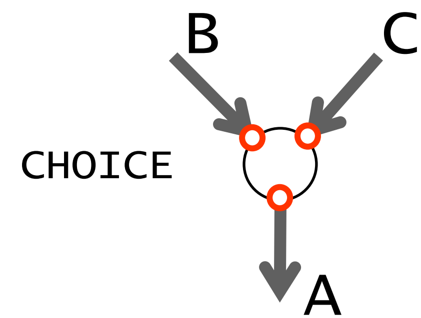
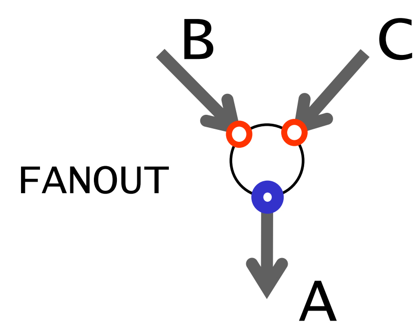
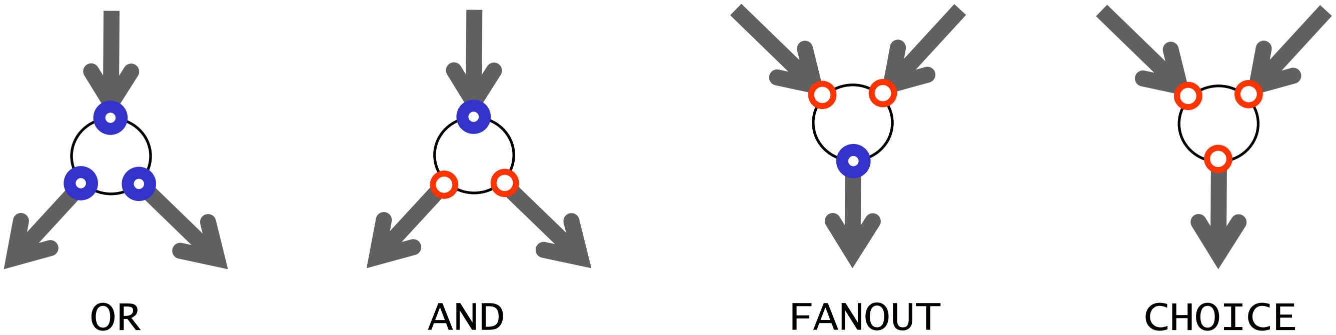


Figure 9-6: TipOver CHOICE gadget. If the tipper can reach A, then it can reach B or C, but not both.



use one-way gadgets at B and C
(control information flow)

conclusion

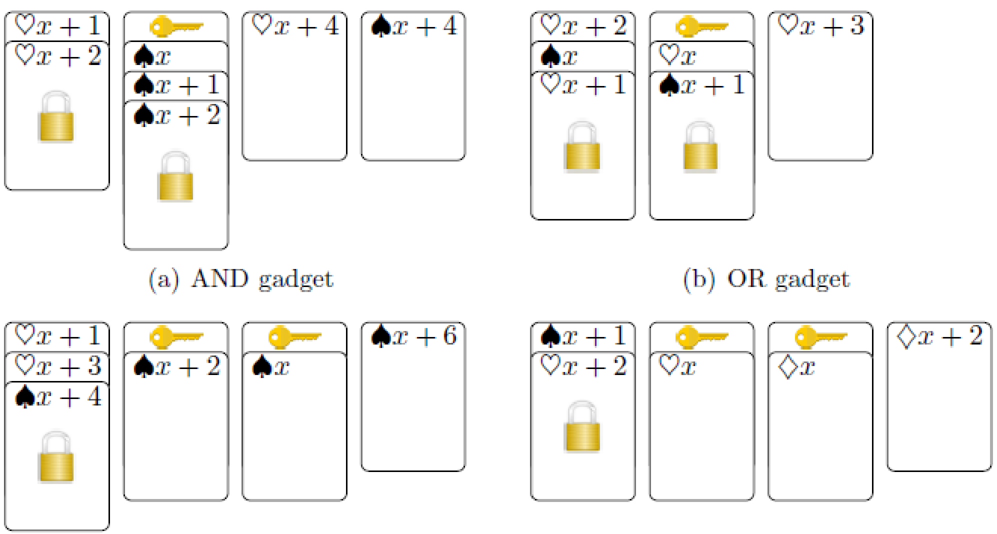


Bounded NCL is NP-complete,
even for planar graphs,
with restricted vertices

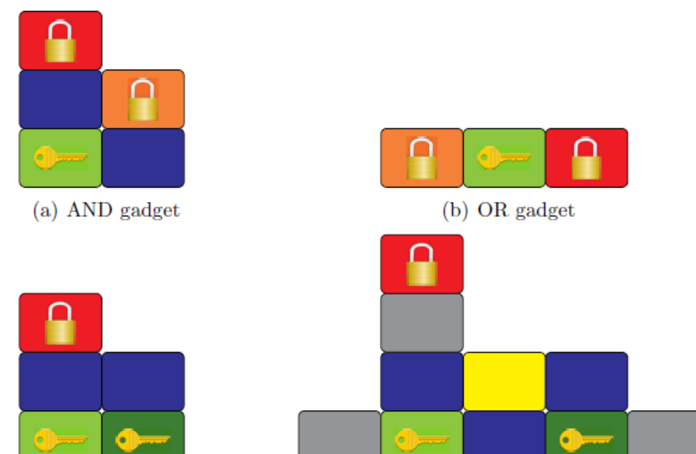
thm. TipOver is NP-complete

NP complete bounded games

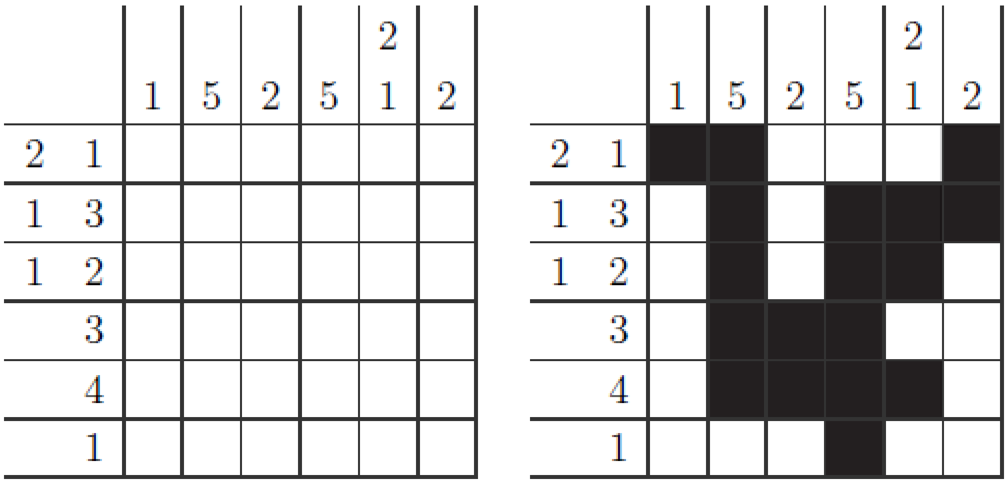
Jan van Rijn: Playing Games:
The complexity of Klondike, Mahjong, Nonograms
and Animal Chess
(Master Thesis, 2013, Leiden)



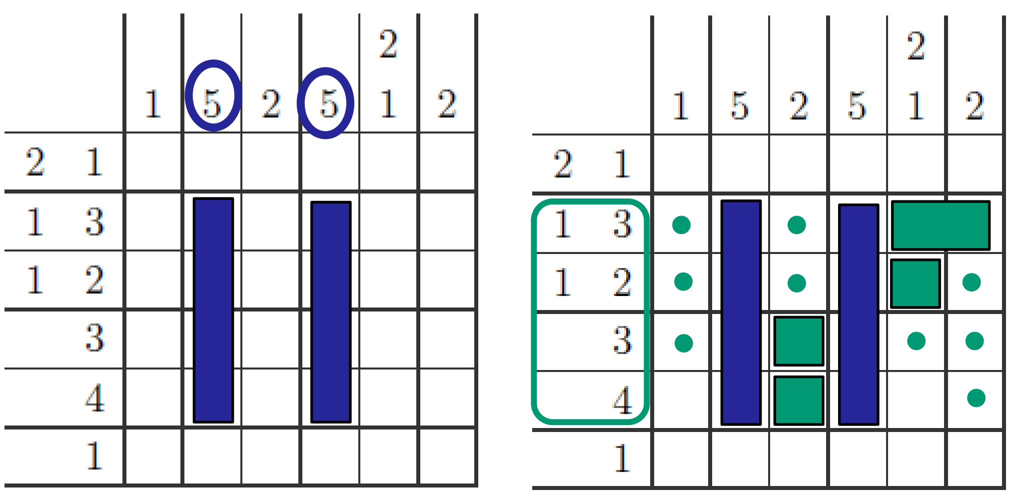
(a) AND gadget (b) OR gadget
(c) FANOUT gadget (d) CHOICE gadget

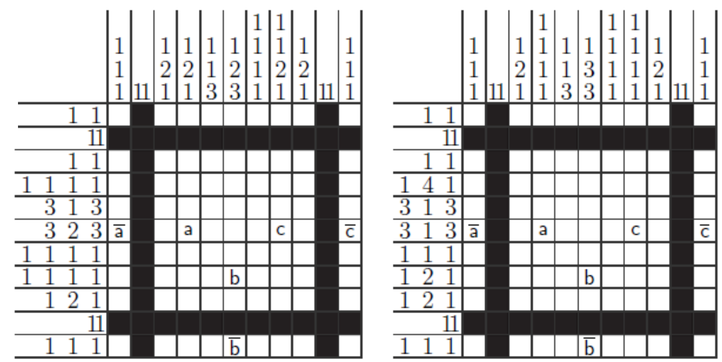

(a) AND gadget (b) OR gadget
(c) FANOUT gadget (d) CHOICE gadget

(ctd.) nonograms

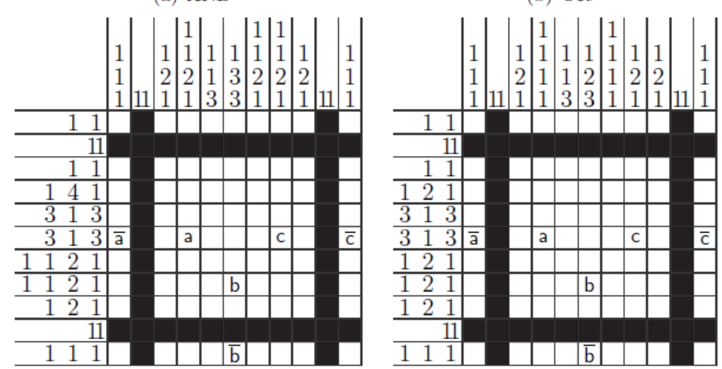


(a) 6 × 6 Nonogram (b) Solved Nonogram





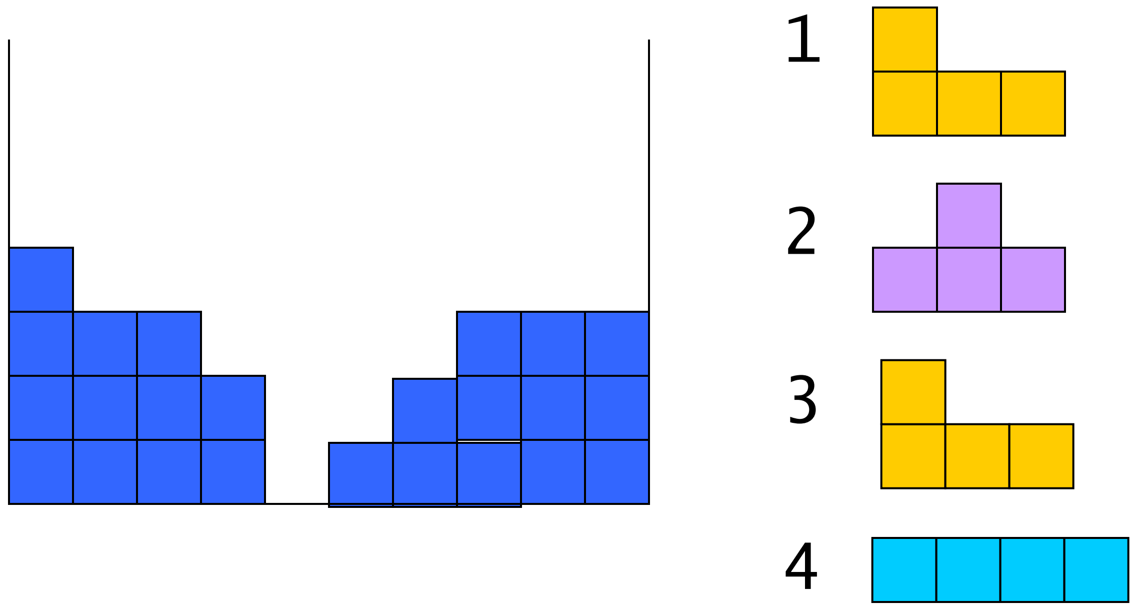
(a) AND (b) OR



(c) FANOUT (d) CHOICE

Tetris is NP complete

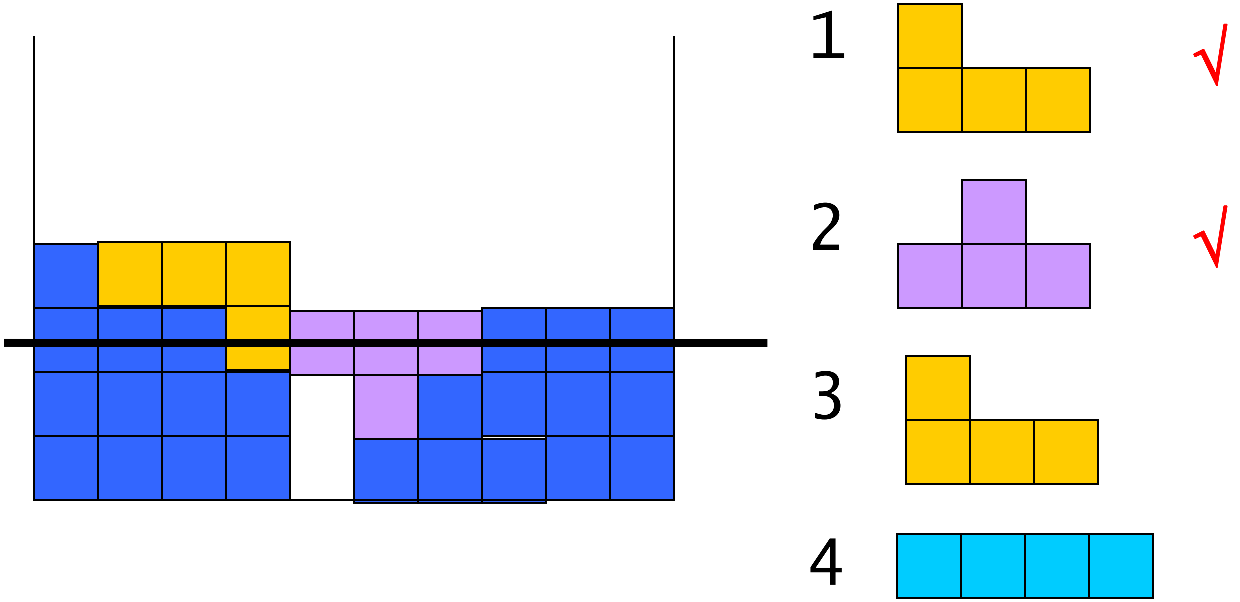
"Given an initial game board and a sequence of pieces, can the board be cleared?"



Breukelaar, Demaine, Hohenberger, Hoogeboom, Kosters, Liben-Nowell. Tetris is Hard, Even to Approximate. Selected Papers from the Ninth Int. Computing and Combinatorics Conf. (COCOON 2003). *Int. J. of Computational Geometry and Applications* 14 (2004) 41-68.

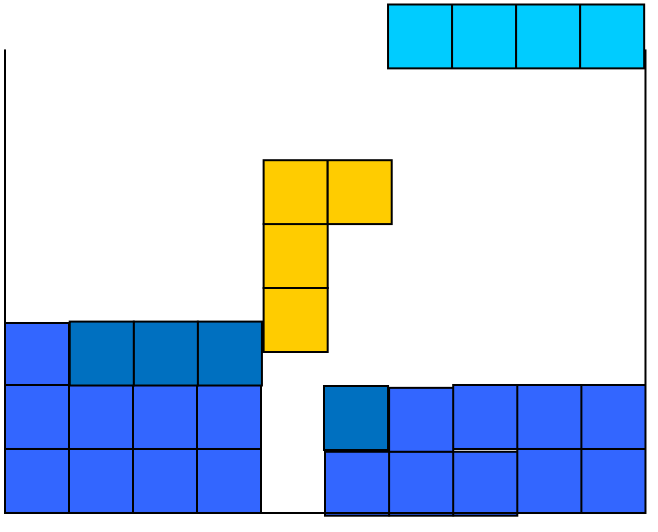
Tetris is NP complete

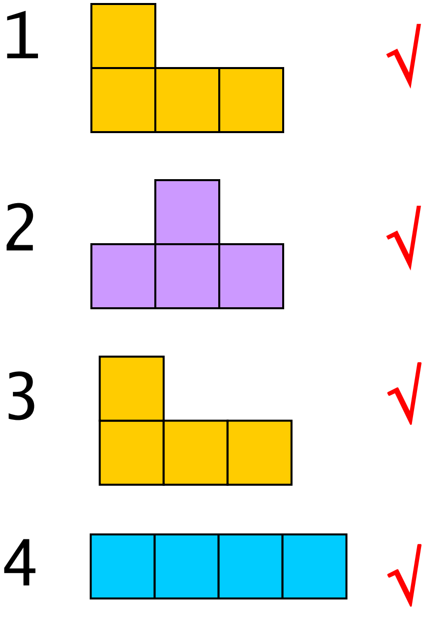
"Given an initial game board and a sequence of pieces, can the board be cleared?"



Tetris is NP complete

"Given an initial game board and a sequence of pieces, can the board be cleared?"



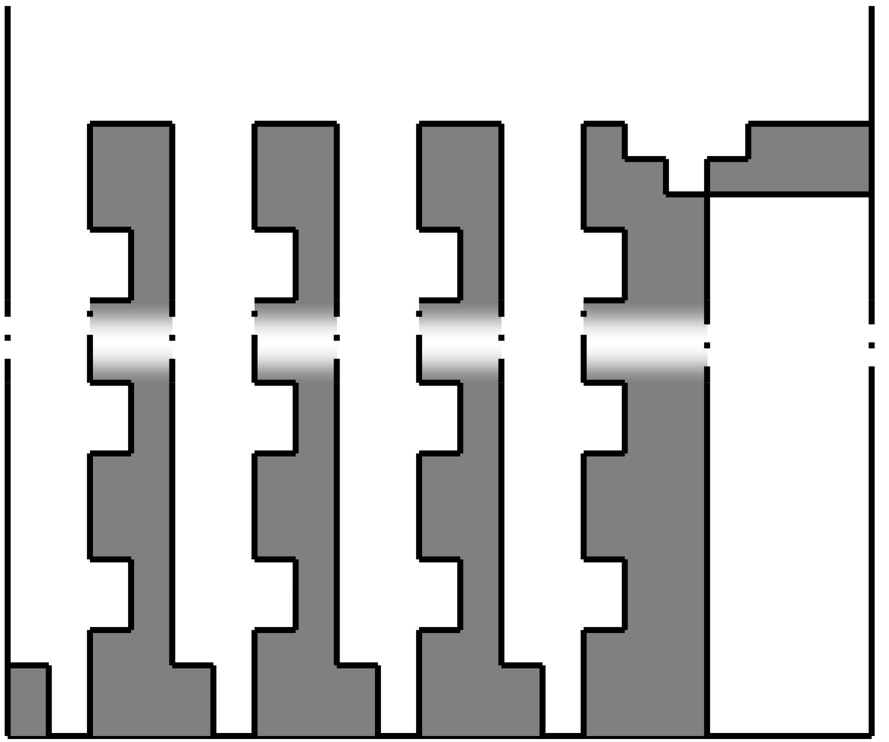


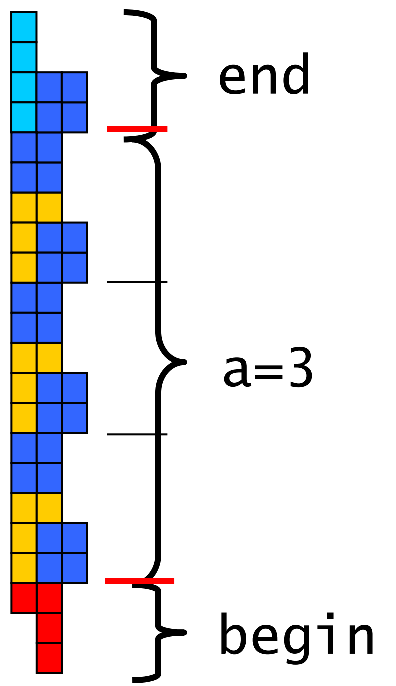
yes!

Tetris is NP complete

"Given an initial game board and a sequence of pieces, can the board be cleared?"

reduction from 3-partitioning problem
(can we divide set of numbers into triples?)



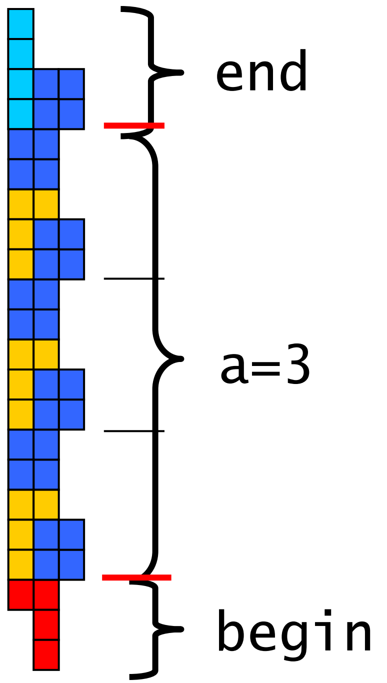


Tetris is NP complete

reduction from 3-partitioning problem
(can we divide set of numbers into triples?)

OPEN: directly with Bounded NCL ?

find OR, AND, FANOUT, CHOICE



Games, Puzzles & Computation

IPA Advanced Course on Algorithmics and Complexity

Eindhoven, 8 July 2016

Walter Kosters
Hendrik Jan Hoogeboom

LIACS, Universiteit Leiden

Wrap Up

conclusion: nice uniform family of graph games, suitable for the various game classes

not in this presentation:

deterministic classes are hard to prove complete: timing constraints
bounded det. ncl has no known planar normal form

2pers. games need two types of edges (apart from colours), for each of the players

for teams one needs hidden info, otherwise equivalent to 2p games

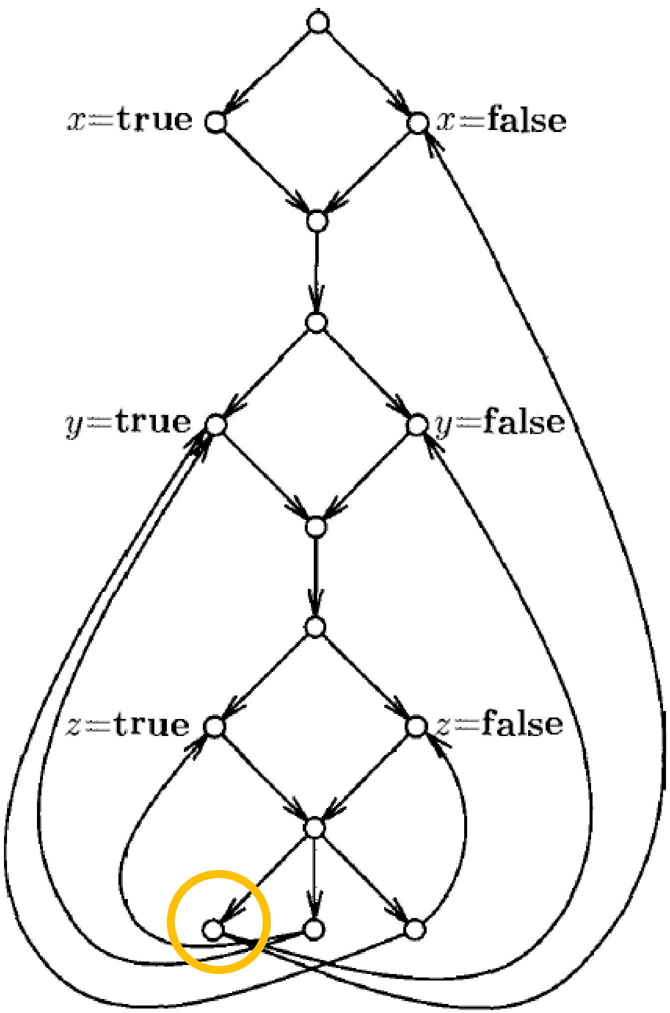
conclusion

roots can be found in the literature
(see Geography)

take care: game of life (what is the ‘goal’?)
is PSPACE, it also is undecidable ☺
(on infinite grid)

example of P complete:
the domino topling simulation

geography



[Schäfer, J.CSS, 1978]

THM. Geography is PSPACE-complete

two players on directed graph alternately pick next vertex, without repetition

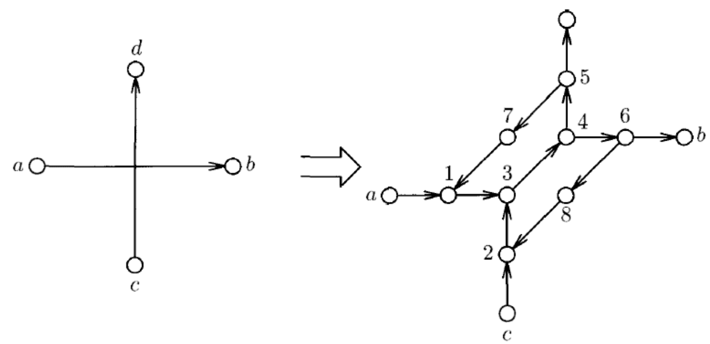
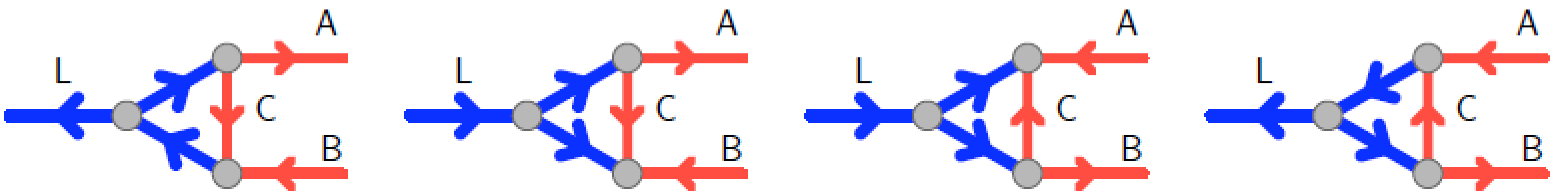


Figure 19-4. Crossing edges in GEOGRAPHY.

application to GO
[Lichtenstein&Sipser, J.ACM, 1980]

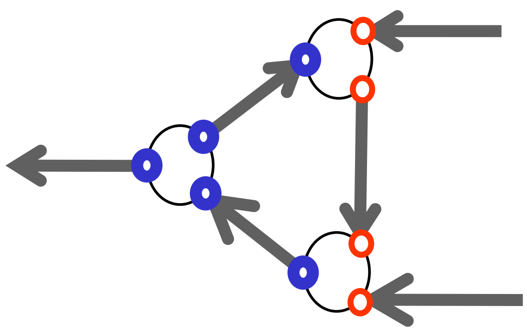
$$\exists x \forall y \exists z [(\neg x \vee \neg y) \wedge (y \vee z) \wedge (y \vee \neg z)].$$

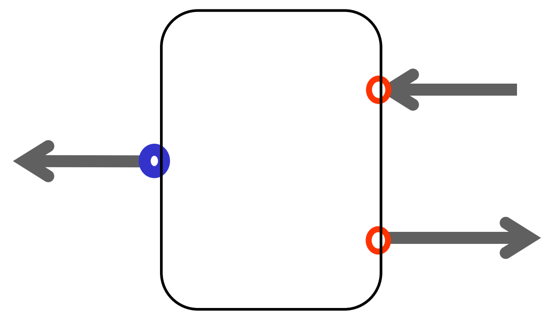
latch / protected OR



(a) Locked, A active (b) Unlocked, A active (c) Unlocked, B active (d) Locked, B active

Figure 5-6: Latch gadget, transitioning from state A to state B.

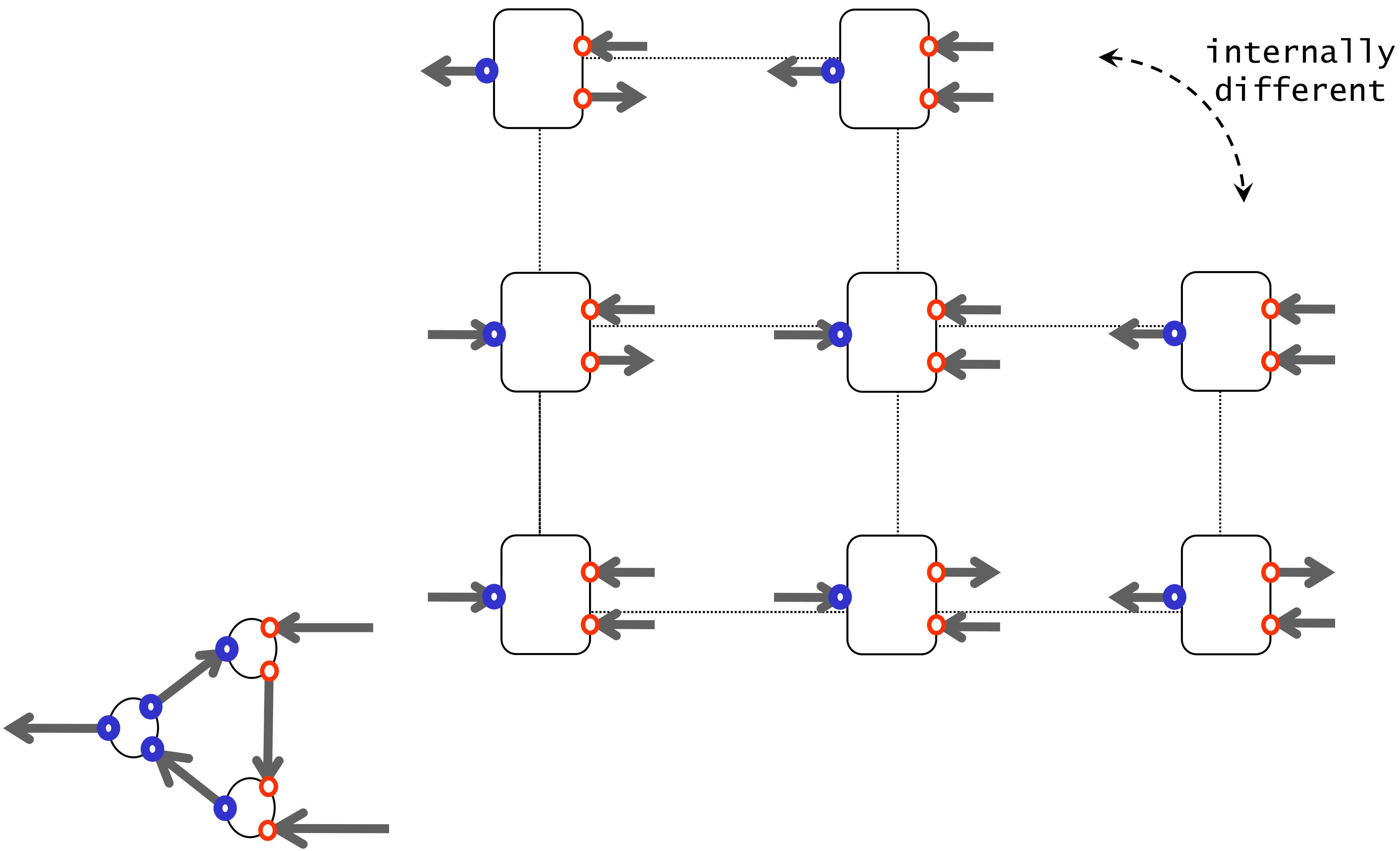




we are not (really) interested in the internal states
only in the external connections:
input-output behaviour

synchronization, silent actions & simulation

latch behaviour



formula games – complete problems

NL **2SAT**

$(x_1 \vee x_3) \wedge (\neg x_5 \vee \neg x_3) \wedge (x_5 \vee x_1)$

P **HORN-SAT**

$(\neg x_3 \vee \neg x_2 \vee \neg x_5 \vee x_1)$ i.e. $(x_3 \wedge x_2 \wedge x_5 \rightarrow x_1)$

NP **SAT** satisfiablity

$\exists x_1\ \exists x_3\ \exists x_5\ (x_1 \vee x_3 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_5 \vee x_1)$

PSPACE **QBF** aka QSAT

$\exists x_1\ \forall x_3\ \exists x_5\ (x_1 \vee x_3 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_5 \vee x_1)$

formula games – complete problems

NL **2SAT**

$(x_1 \lor x_3) \land (\neg x_5 \lor \neg x_3) \land (x_5 \lor x_1)$

P **HORN-SAT**

$(\neg x_3 \lor \neg x_2 \lor \neg x_5 \lor x_1)$ i.e. $(x_3 \land x_2 \land x_5 \rightarrow x_1)$

NP **SAT** satisfiablity

$\exists x_1\ \exists x_3\ \exists x_5\ (x_1 \lor x_3 \lor \neg x_5) \land (\neg x_1 \lor \neg x_3) \land (x_5 \lor x_1)$

PSPACE **QBF** aka QSAT

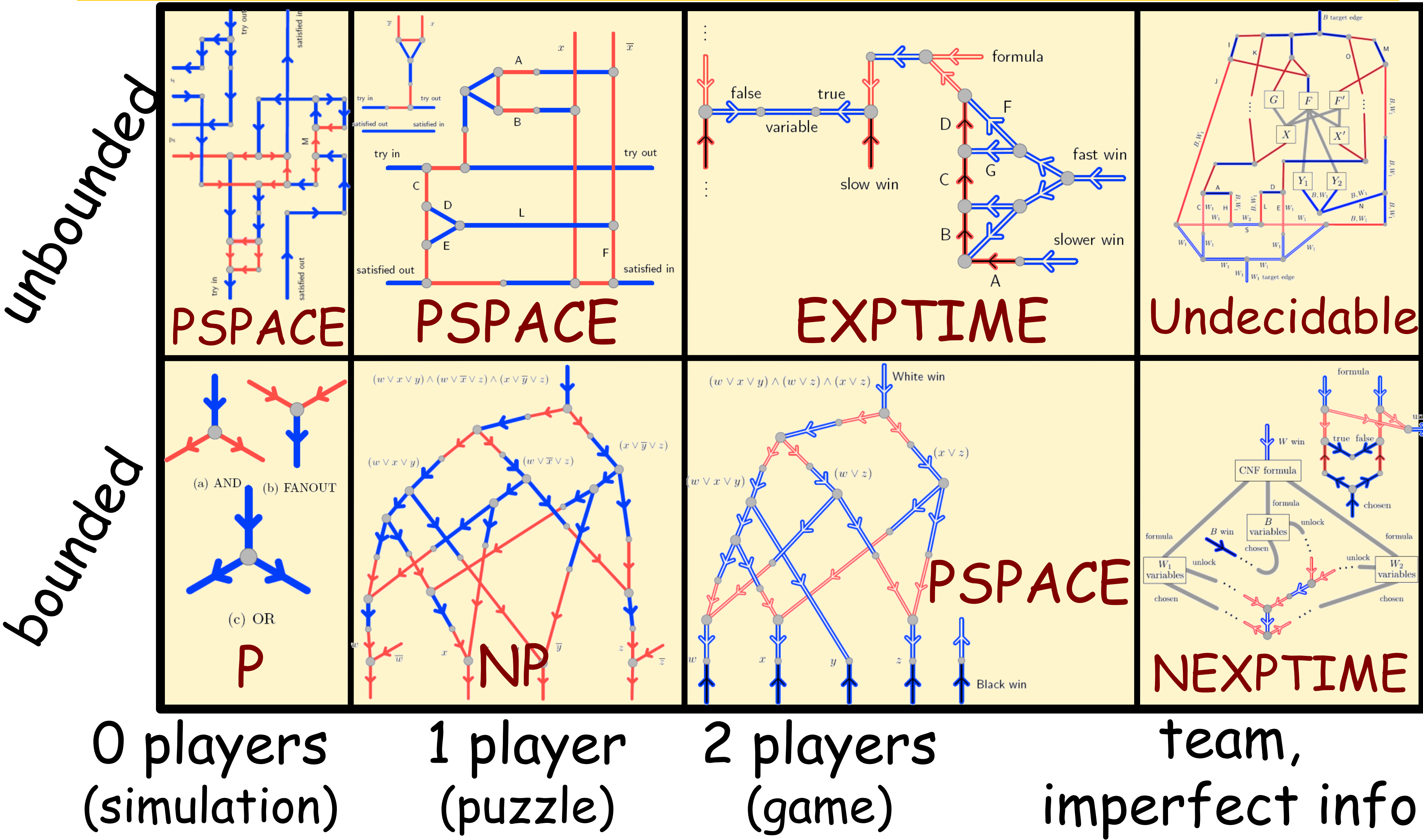
$\exists x_1\ \forall x_3\ \exists x_5\ (x_1 \lor x_3 \lor \neg x_5) \land (\neg x_1 \lor \neg x_3) \land (x_5 \lor x_1)$

EXPTIME **G_6** [Stockmeyer & Chandra]

given: CNF formula F variables X Y
initial assignment
player I (and II) change single variable in X (Y)
taking turns, passing allowed
player I wins if F becomes true

question: does I have a forced win?

Constraint Logic
[Hearn & Demaine 2009]



Games, Puzzles & Computation

IPA Advanced Course on Algorithmics and Complexity

Eindhoven, 8 July 2016

Walter Kosters
Hendrik Jan Hoogeboom

LIACS, Universiteit Leiden