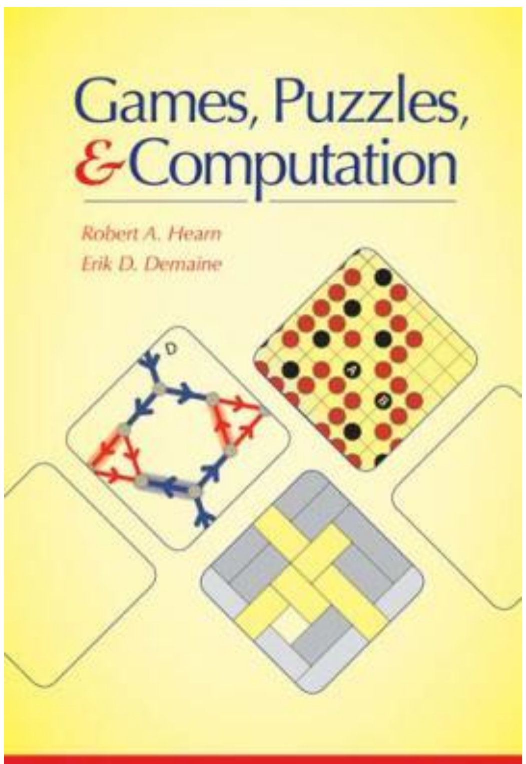
Games, Puzzles & Computation



IPA Advanced Course on Algorithmics and Complexity

Eindhoven, 8 July 2016

Walter Kosters
Hendrik Jan Hoogeboom

LIACS, Universiteit Leiden

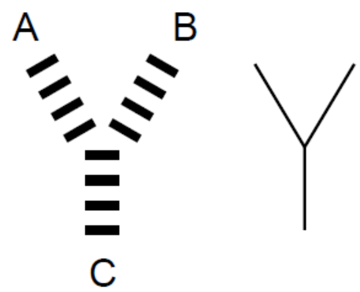
schedule

		Framework
10:00-11:00	HJ	Constraint logic, classes
11:15-12:15	W	Gadgets, planarity, exercises
12:30-13:30	*lunch*	
		Concrete Games
13:45-14:45	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

Computing with Planar Toppling Domino Arrangements

William M. Stevens



fork

challenge:
(no) timing & (no) bridges

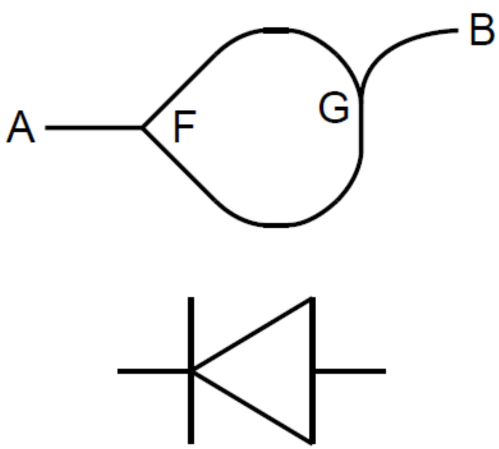


Fig. 3. A one way line

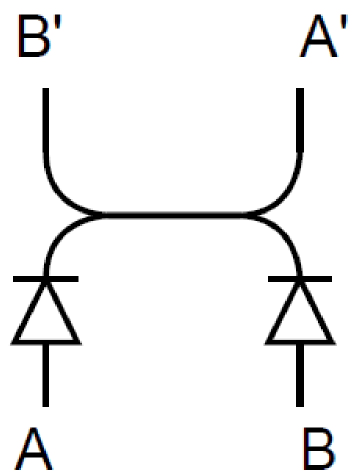


Fig. 4. A single line crossover

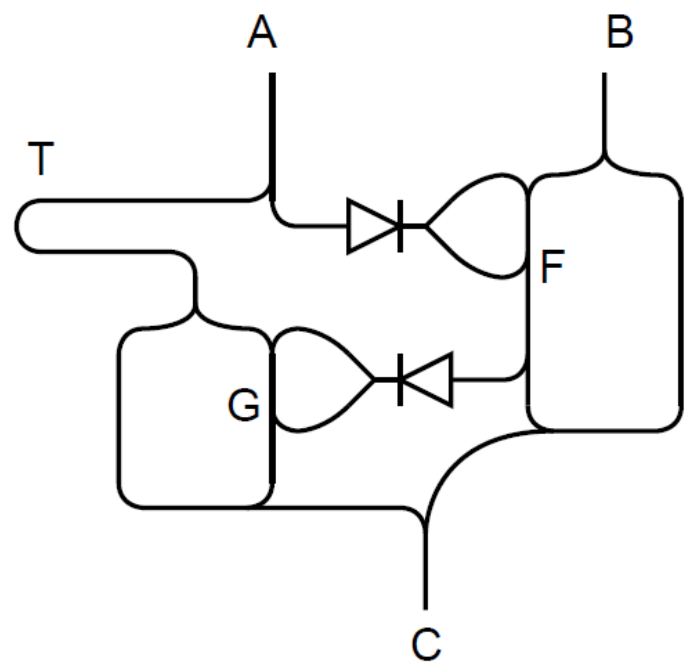


Fig. 5. A both mechanism

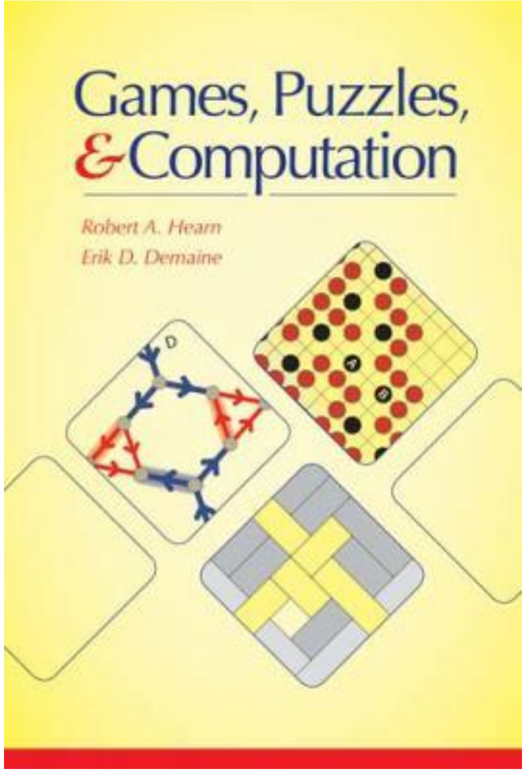
Unconventional Computation, 10th International Conference
UC 2011, Turku, Finland, June 6-10, 2011

reference

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?

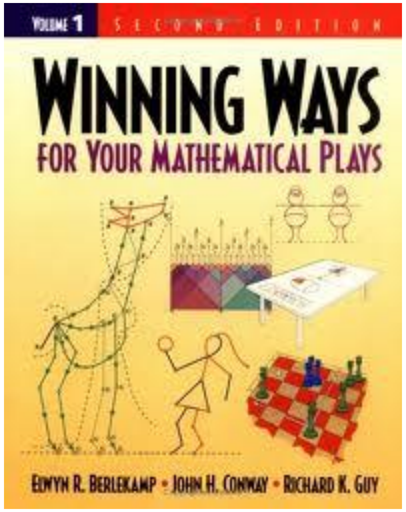
characteristics

- 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

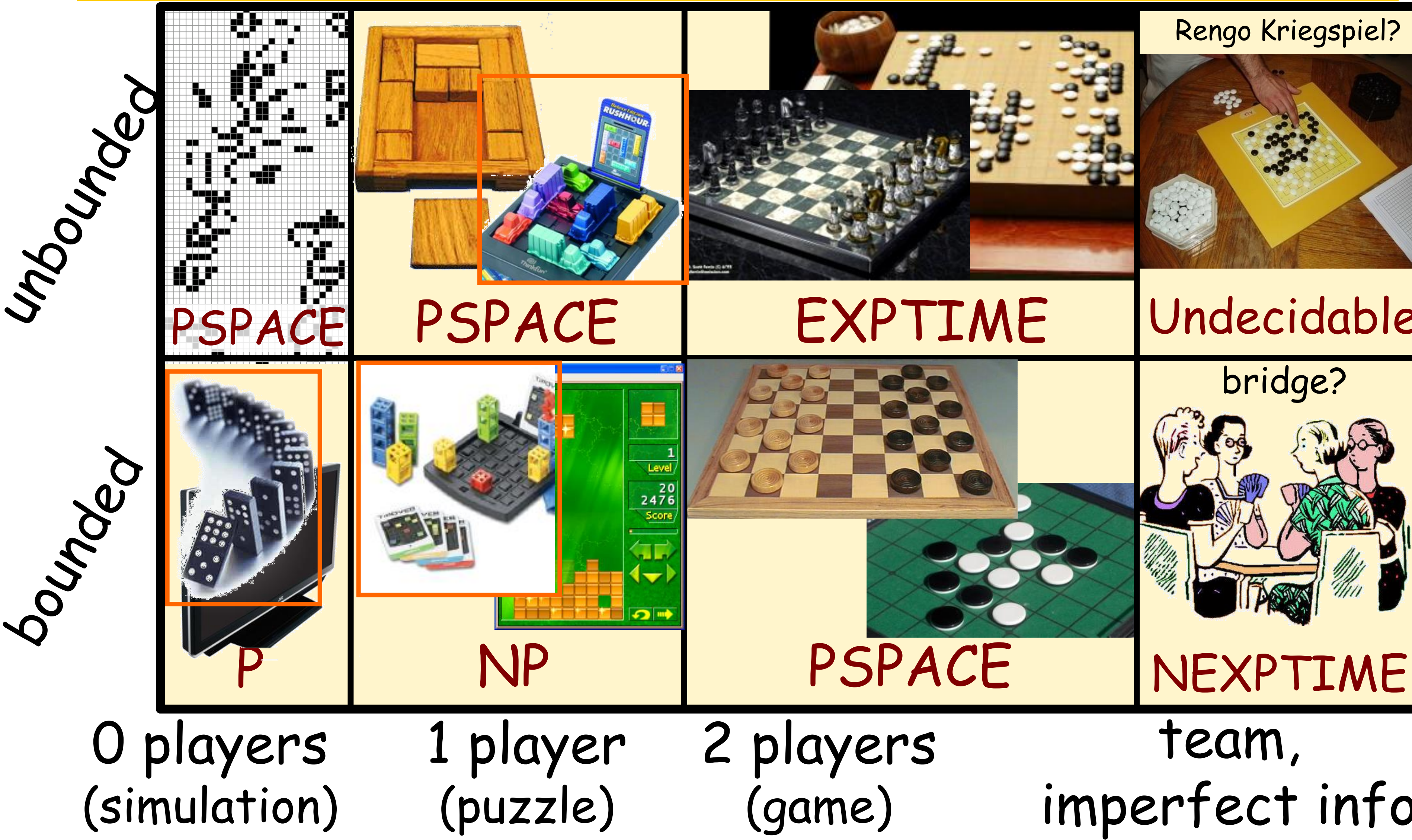
computational complexity

models of computation: *games*

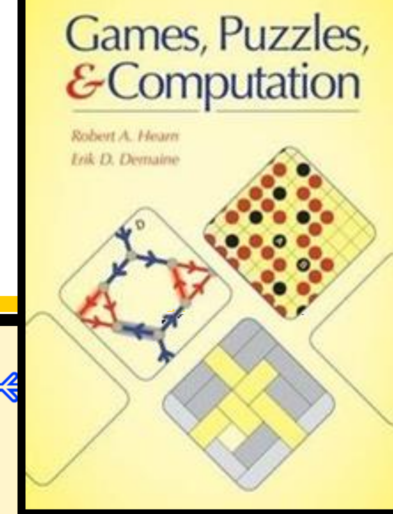
turing machine

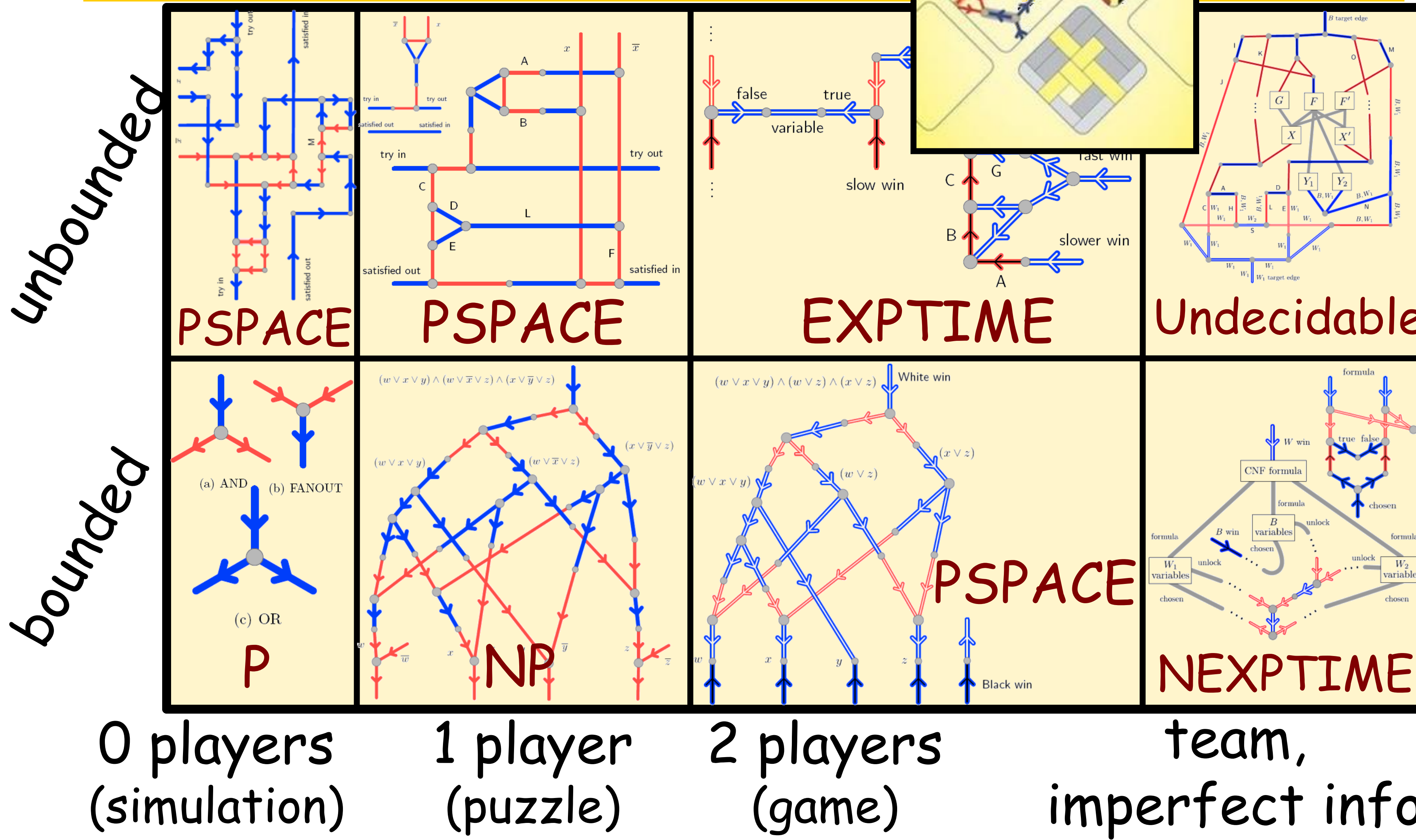
Complexity of Games & Puzzles

[Demaine, Hearn & many others]



Constraint Logic
[Hearn & Demaine 2009]





(details to follow)

constraint logic

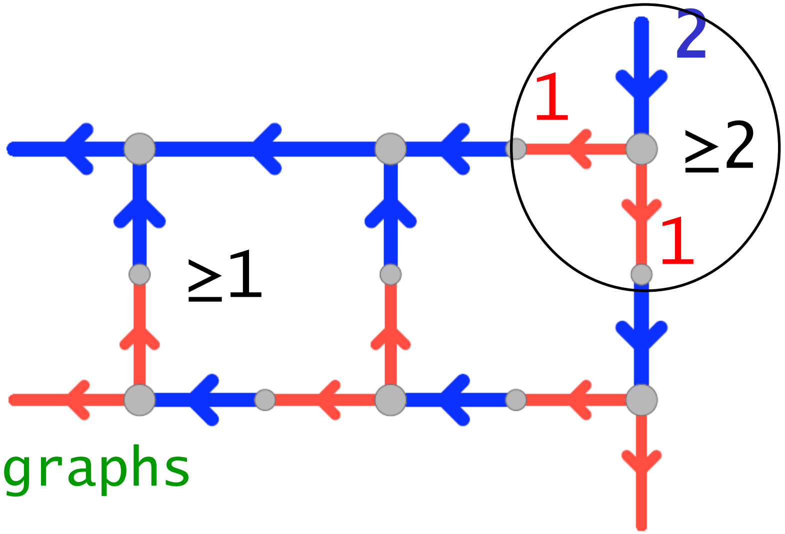
constraint graphs

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

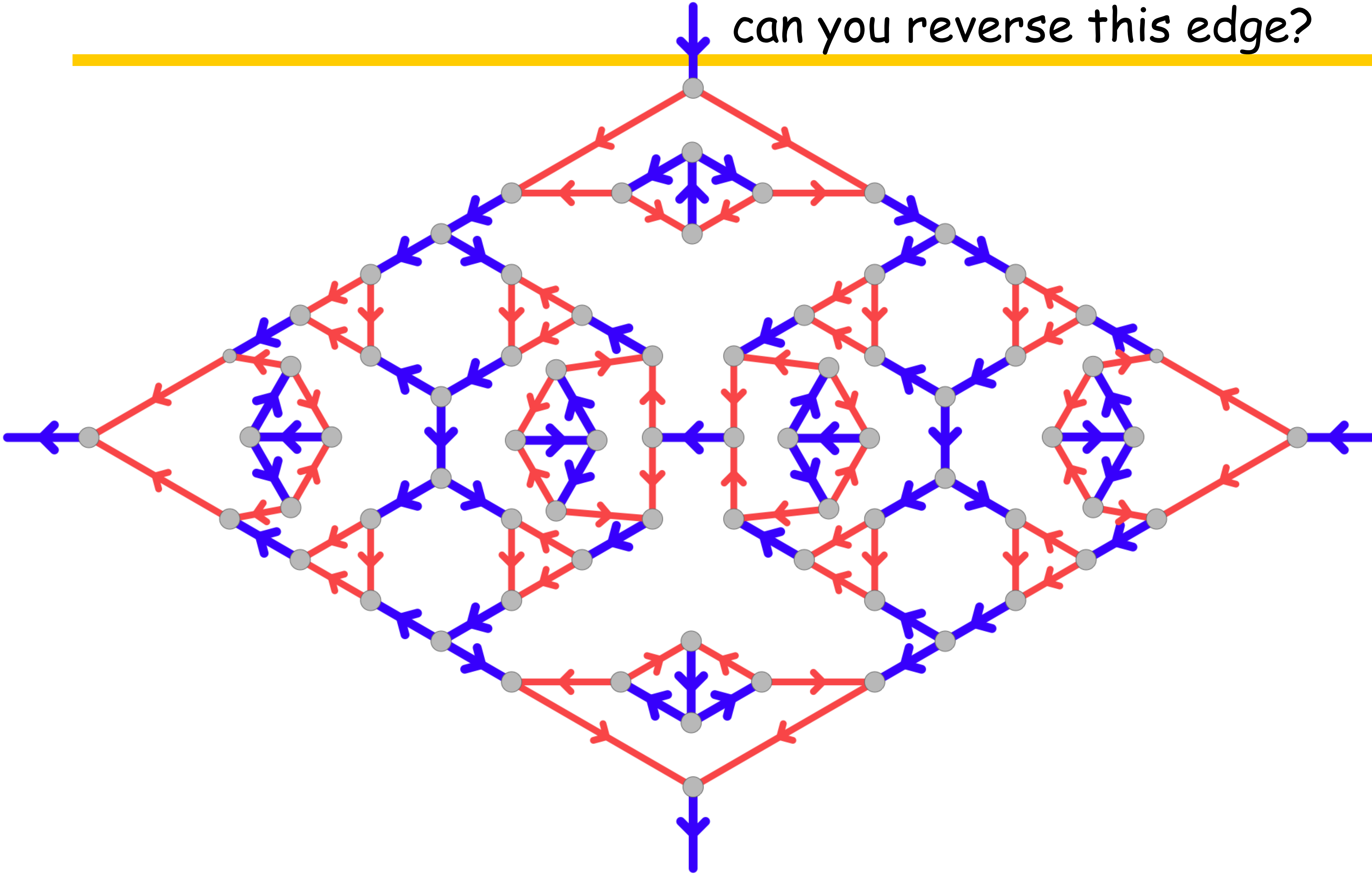
instance: constraint graph G, edge e

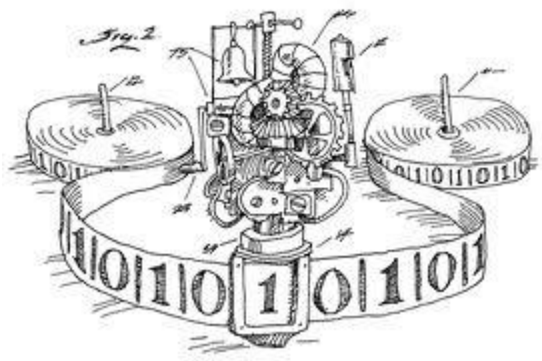
question: sequence which reverses e

BOUNDED NCL

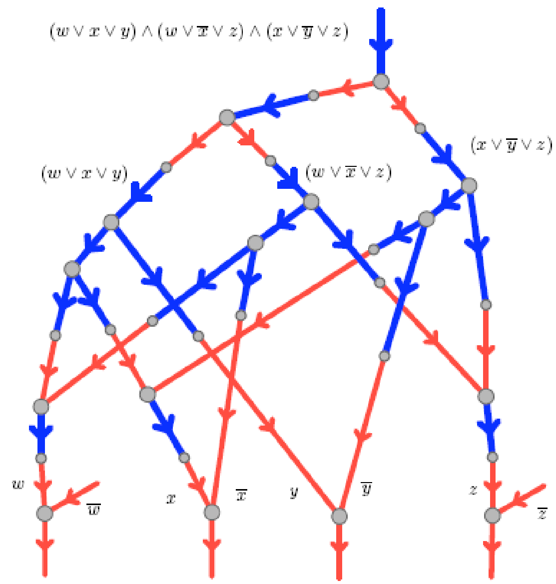
… reverses each edge *at most once*

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'

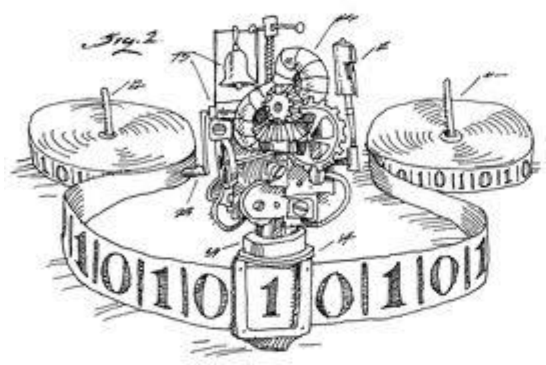
Bounded NCL

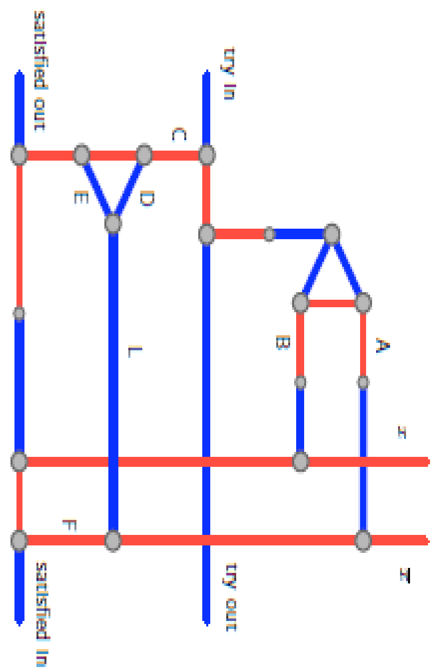
part II

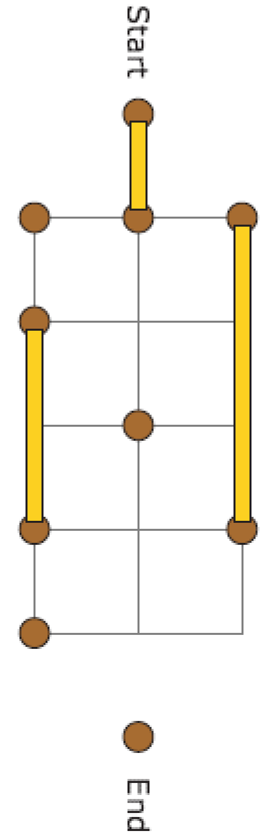
games in particular

TipOver

PSPACE & Plank Puzzle







PSPACE

QBF

part I
constraint logic
‘graph games’

NCL

part II
games in particular

plank puzzle
(river crossing)

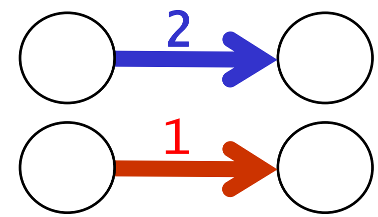
‘formal’ definition

constraint logic – a graph game

issues

goal: generic ‘graph game’

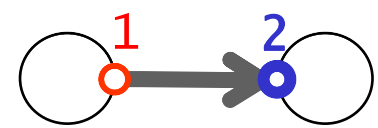
- several instantiations for specific complexity classes / game types
- reduction to games & puzzles



Hearn & Demaine ‘constraint logic’

coloured edges

problem: *colour conversions*



coloured ‘connectors’

problem: *internal behaviour*

state transitions (Tromp & Cilibrasi)

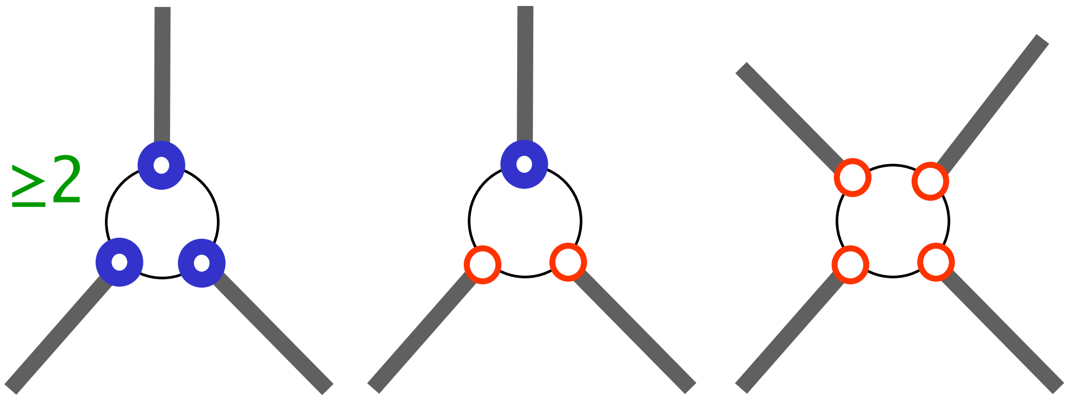
bounded vs. unbounded

(natural direction of computation)

planarity

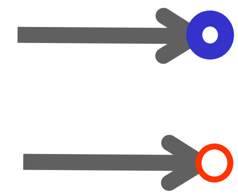
basic constraint logic

examples



edge connectors

incoming value



2

1

constraint graph

oriented/directed edges + connectors

vertex constraint

inflow value ≥ 2

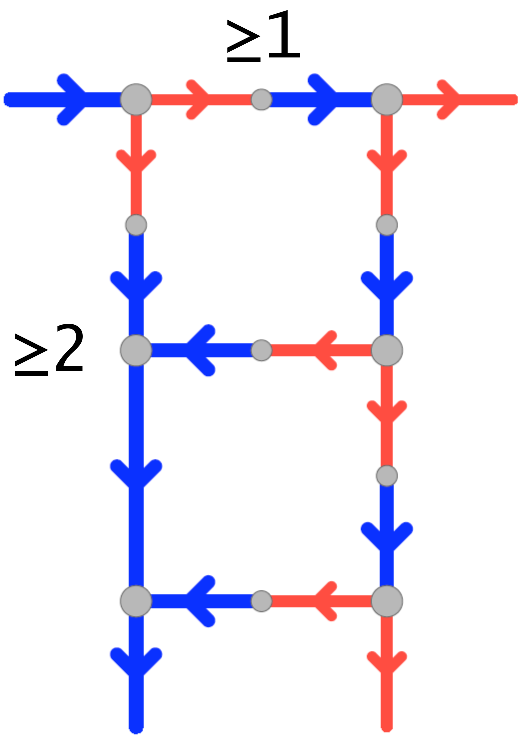
game: legal move

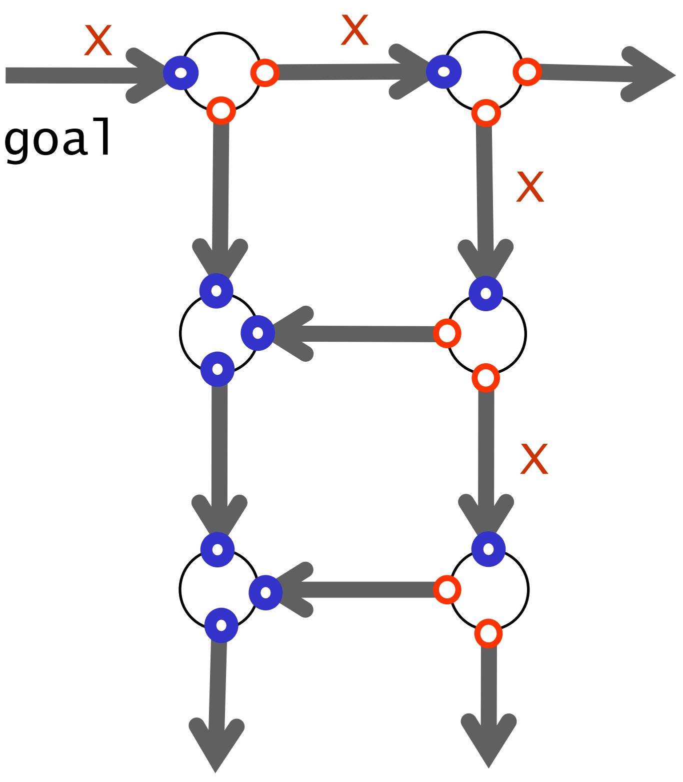
edge reversal satisfying constraint

goal

reversal given edge

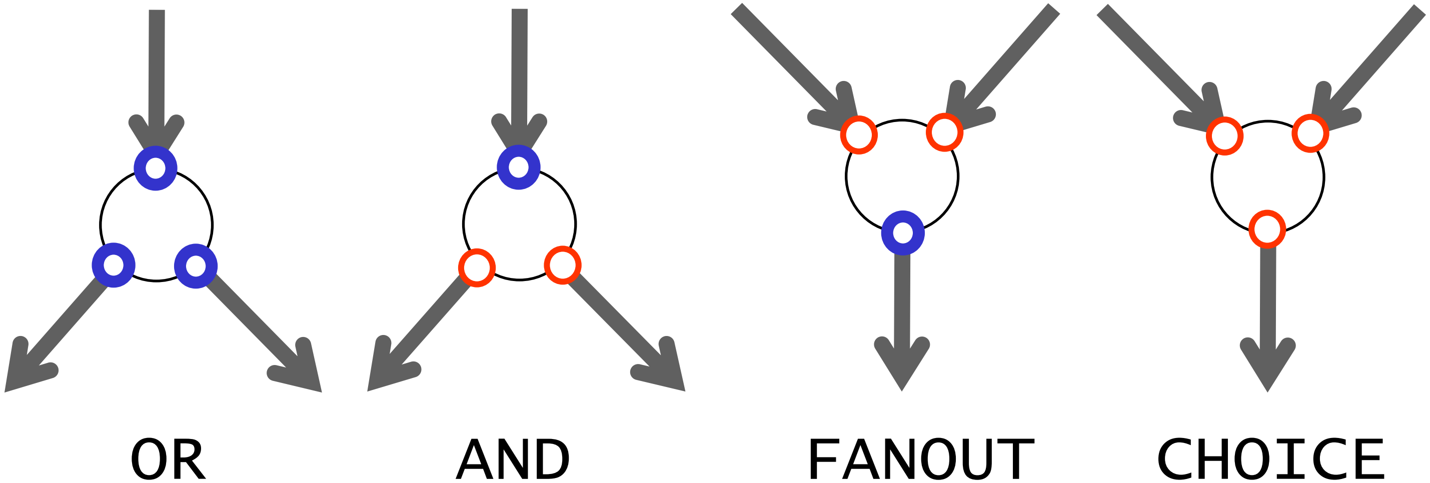
colour conversion



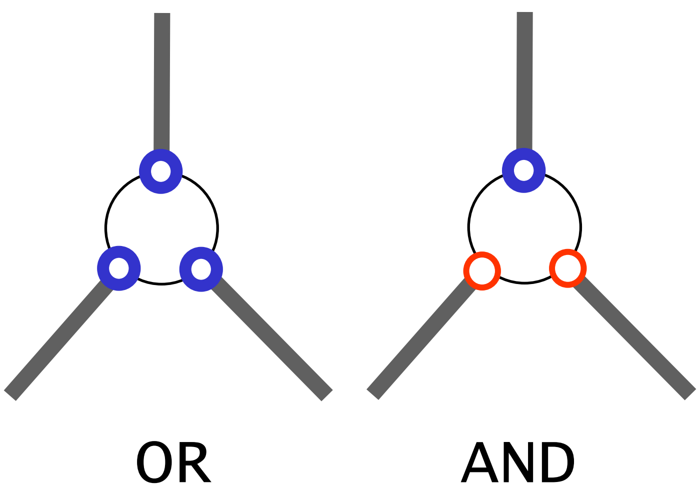


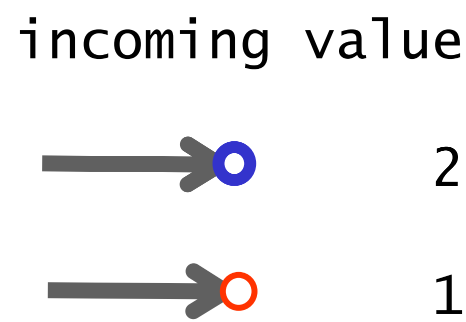
normal form vertices

bounded NCL



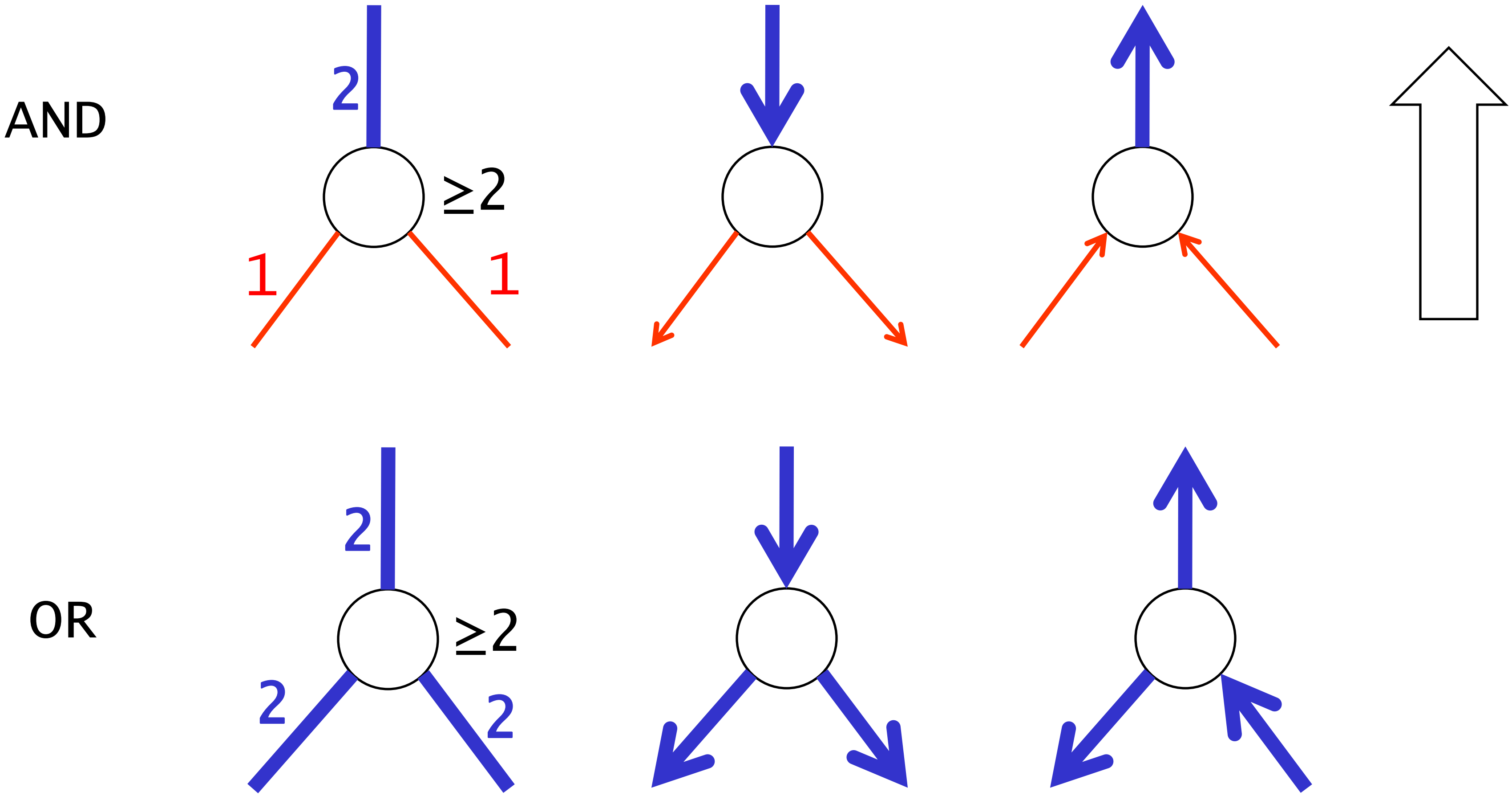
NCL





implementing gates

intuitive meaning of vertices

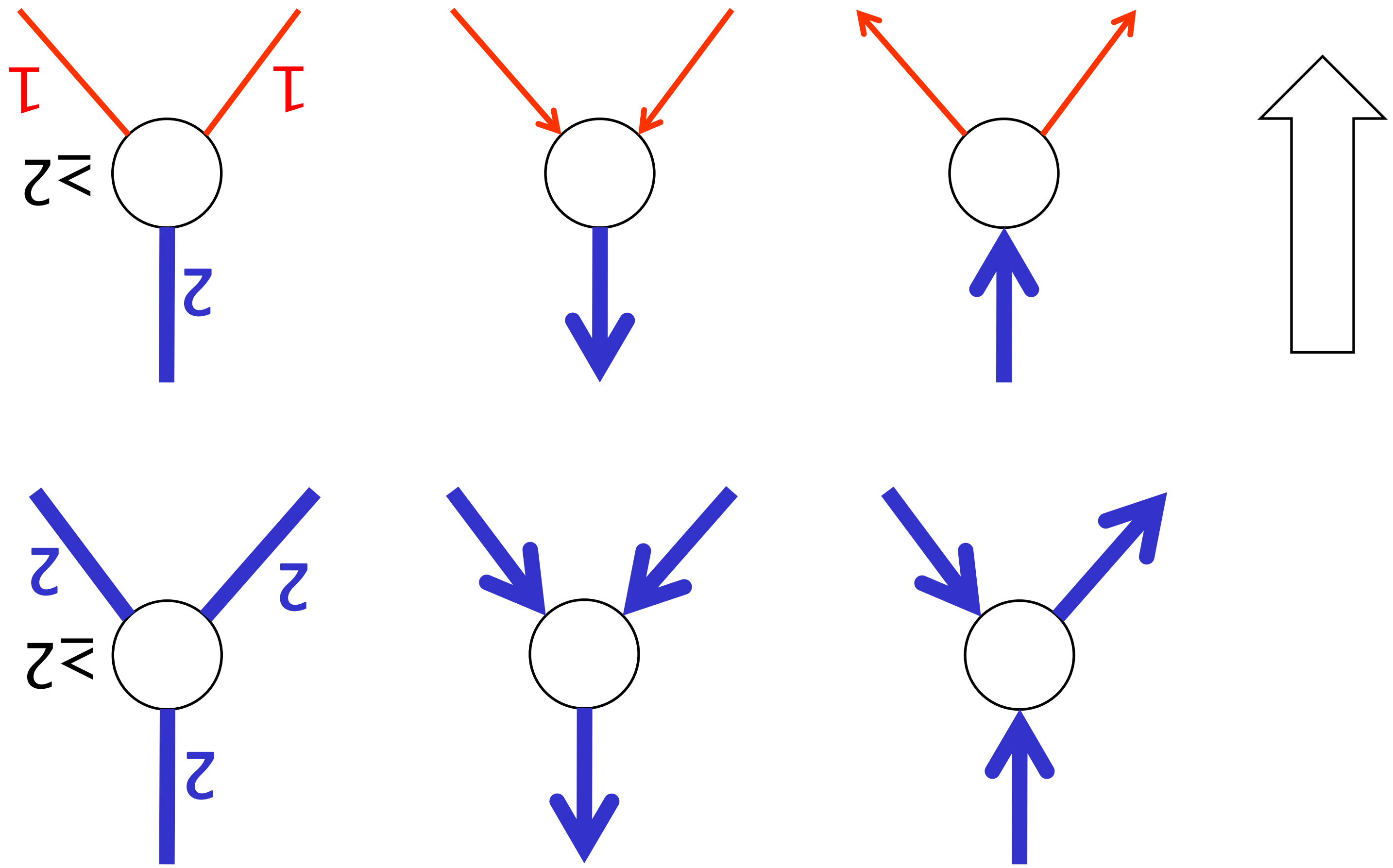


implementing gates

intuitive meaning of vertices

FANOUT

CHOICE



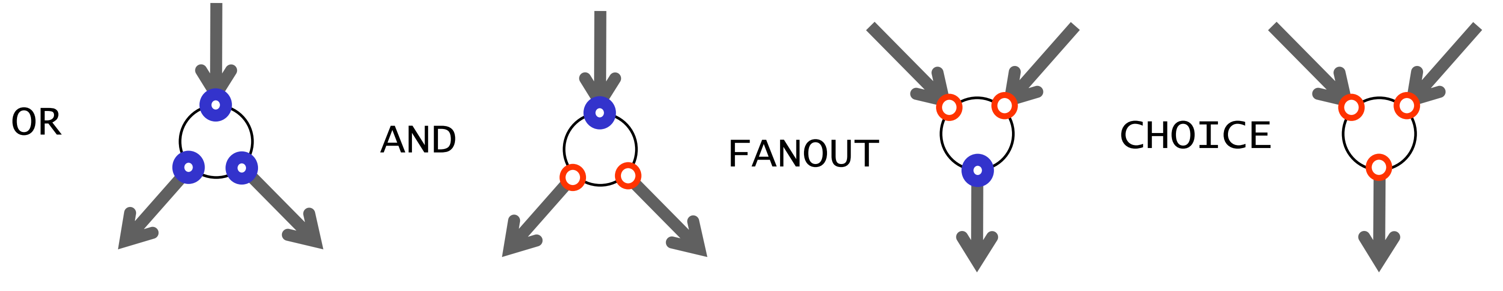
exercise

“emulate” a logical formula as graph game

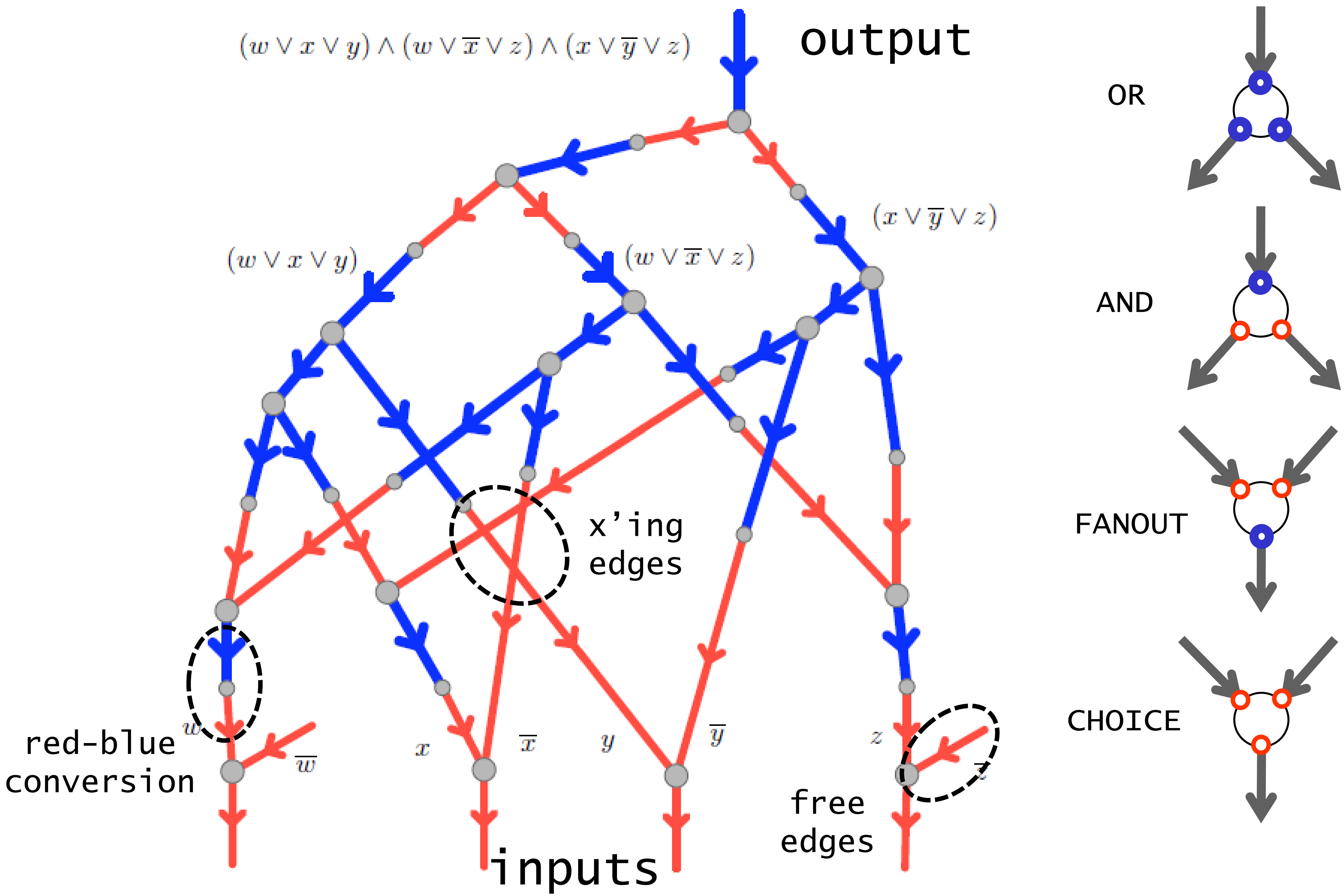
goal:
flip a given edge *iff* formula satisfiable

(w∨x∨y) ∧ (w∨¬x∨z) ∧ (x∨¬y∨z)

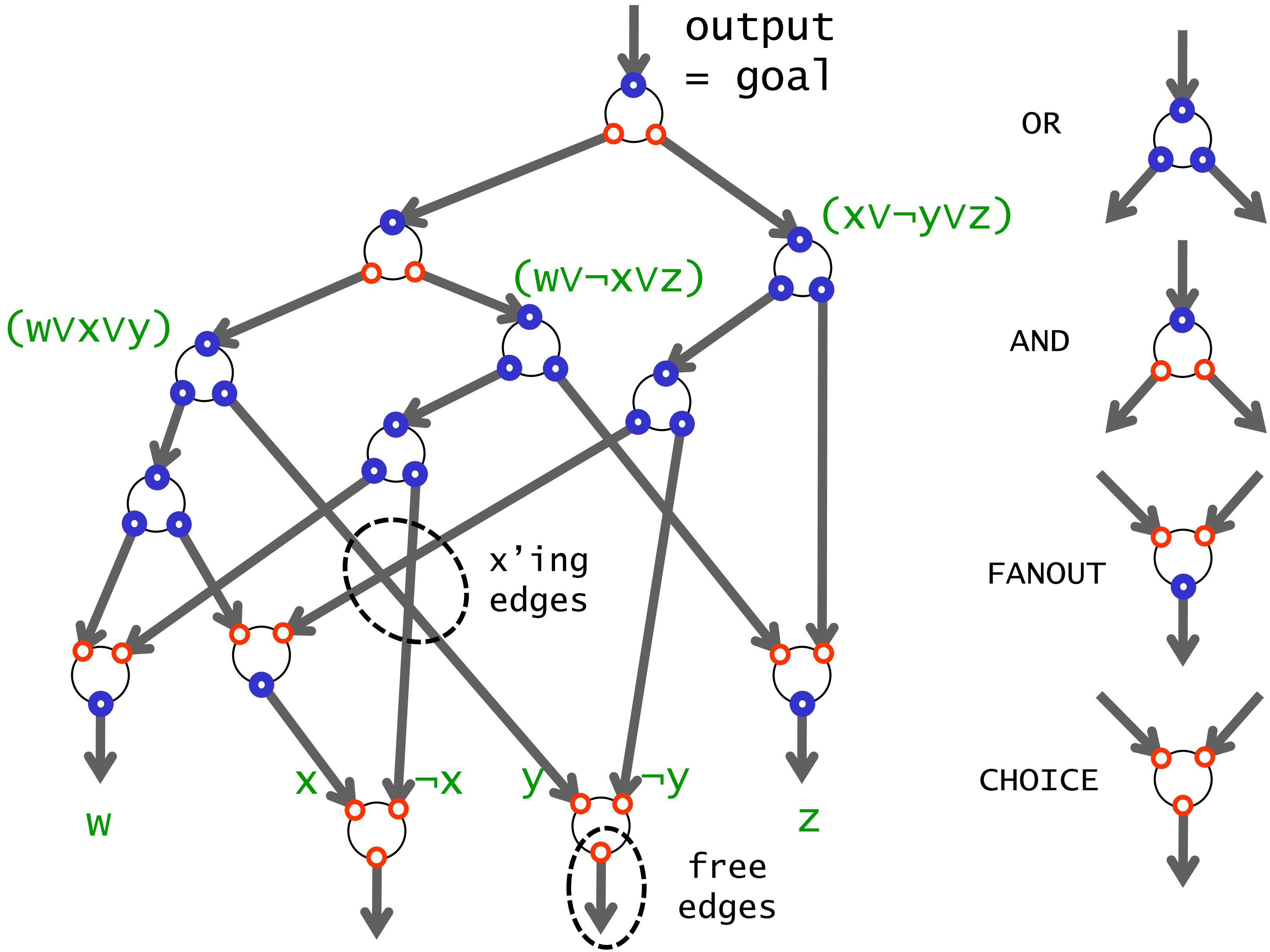
hint:



formula constraint graph

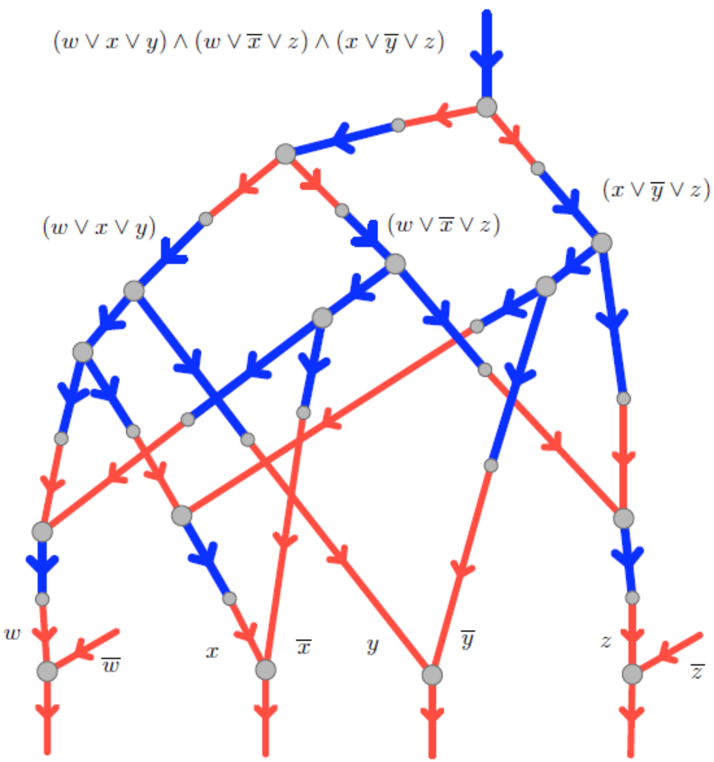


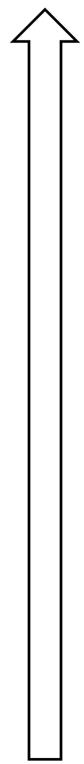
formula constraint graph

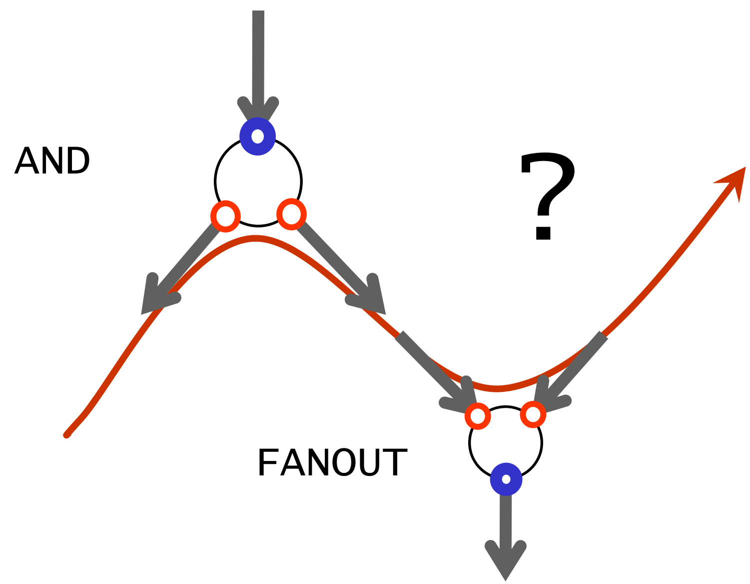


questions

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







basic complexity classes

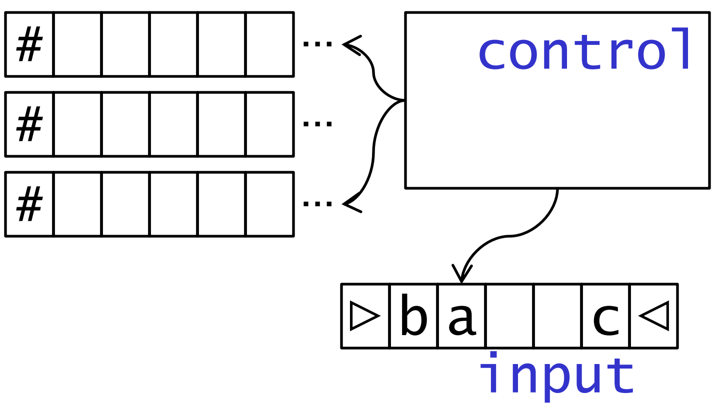
game complexity classes
vs.
TM resources

Cook/Levin NP completeness

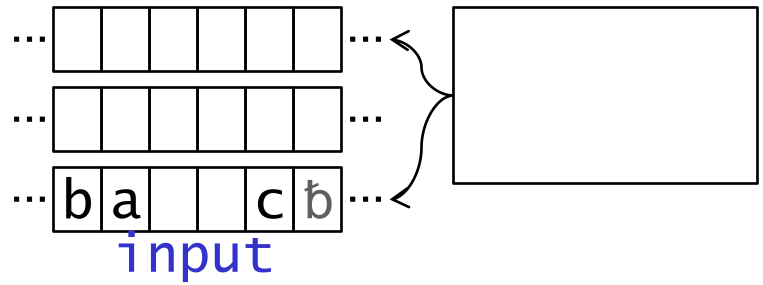
Savitch (N)PSPACE

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)

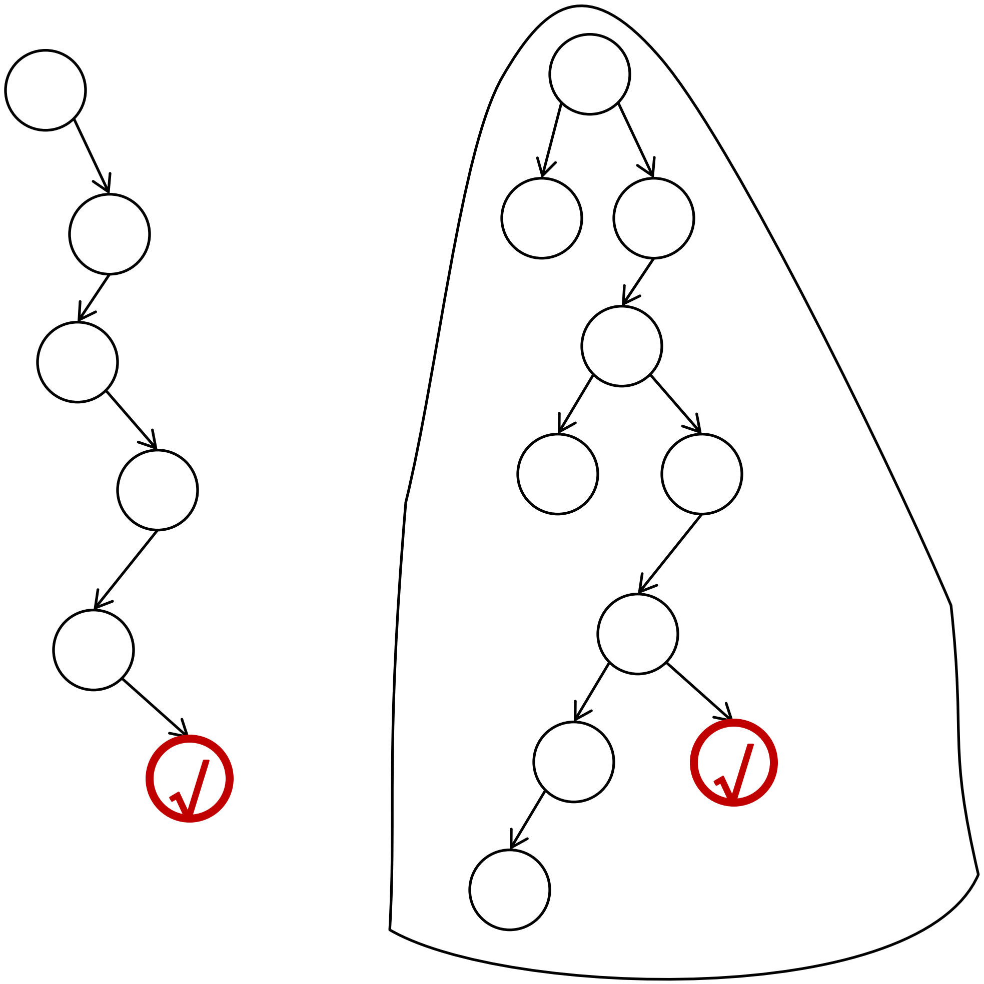
input on tape
multiple working tapes
double sided

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 …

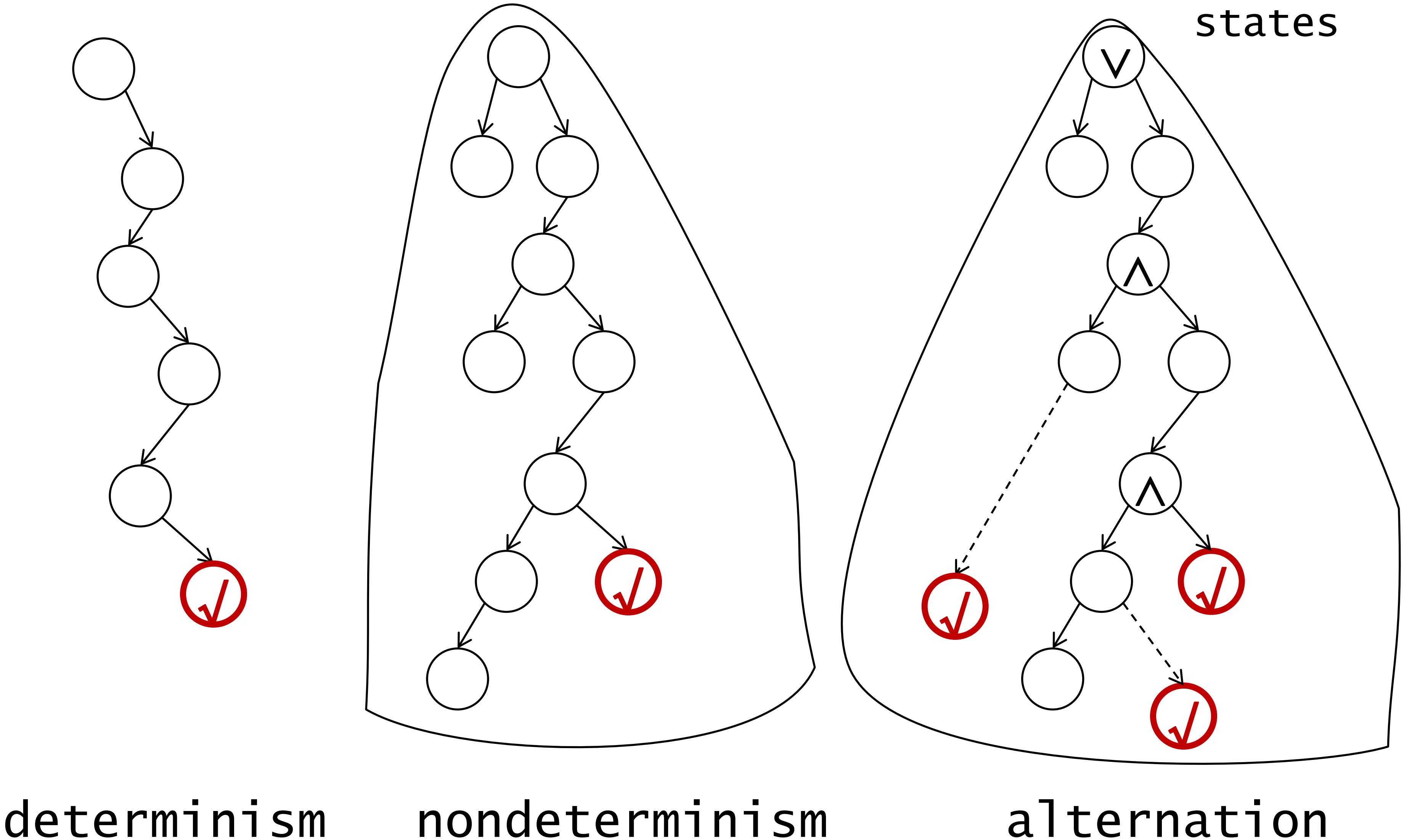
computation tree



determinism nondeterminism

computation tree

existential and *universal* states



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

$NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \subseteq NEXPTIME$

NPSPACE

Savitch

NSPACE(s(n)) ⊆ 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)) ⊆ ATIME(s^2(n)) "parallel in time"

3SAT

literal
(negated) variable

clause

$(w \lor x \lor y) \land (w \lor \neg x \lor z) \land (x \lor \neg y \lor z) \land \ldots$

3 conjunctive normalform

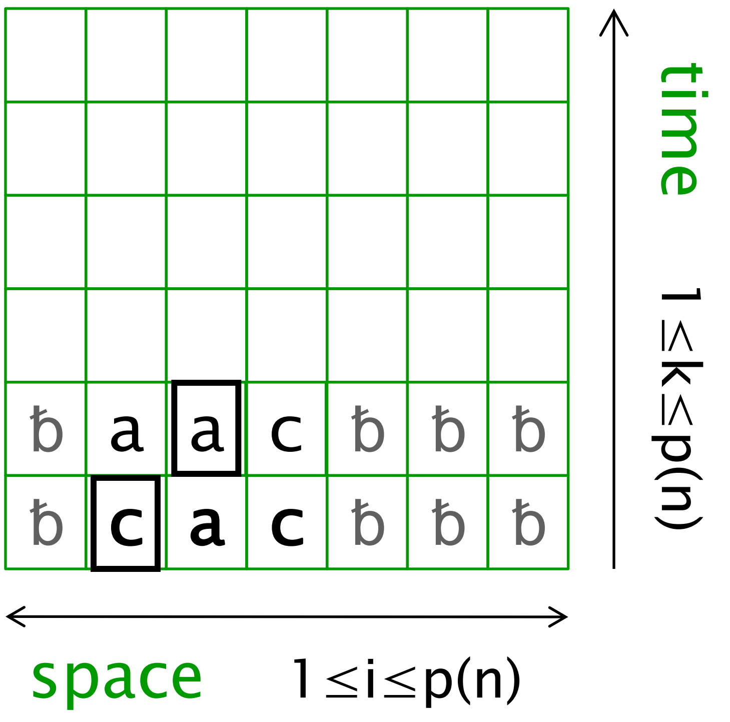
3SAT
given: given formula ϕ in 3CNF
question: is ϕ satisfiable?
(can we find a variable assignment making formula true)

Cook/Levin   3SAT is NP-complete

specify computation

at step k …

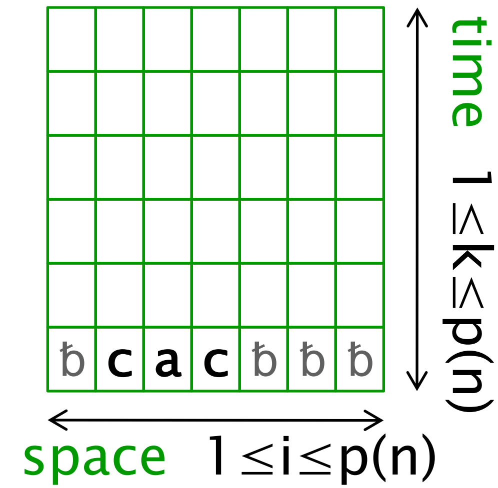
T_{iak}	cell i contains a
H_{ik}	head at position i
Q_{qk}	state q



Cook/Levin

specify computation
at step k …

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

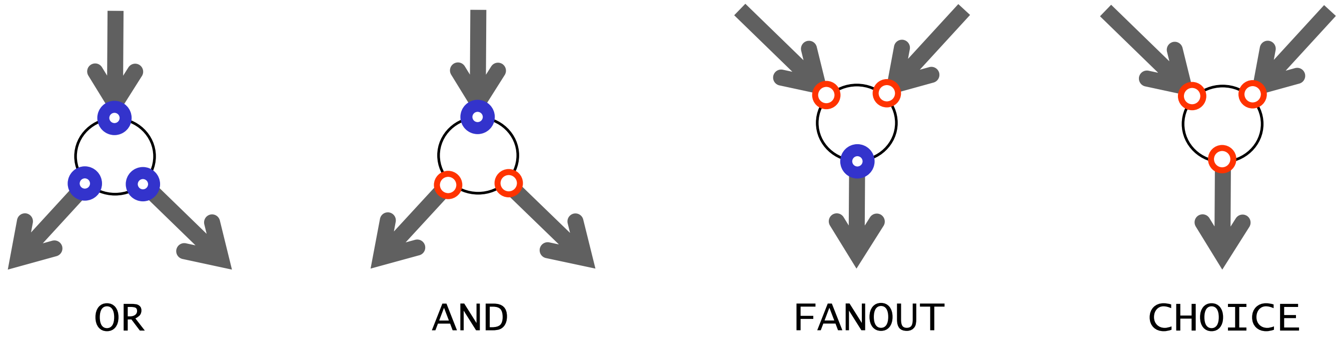


(p,a,p,a,0) for each p,a

conjunction of

$T_{ix[i]0}$ initial tape $x[i]=x_i$ or $x[i]=ƀ$
Q_{q00} initial state
H_{00} initial position
$T_{iak} \to \neg T_{ibk}$ single symbol $a \neq b$
$Q_{pk} \to \neg Q_{qk}$ single state $p \neq q$
$H_{ik} \to \neg H_{jk}$ single head $i \neq j$
$T_{iak} \wedge T_{ib.k+1} \to H_{ik}$ changed only if written $a \neq b$
$H_{ik} \wedge Q_{pk} \wedge T_{iak} \to \bigvee_{(p,a,q,b,d)} H_{i+d.k+1} \wedge Q_{q.k+1} \wedge T_{ib.k+1}$
$Q_{h.p(n)}$ 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

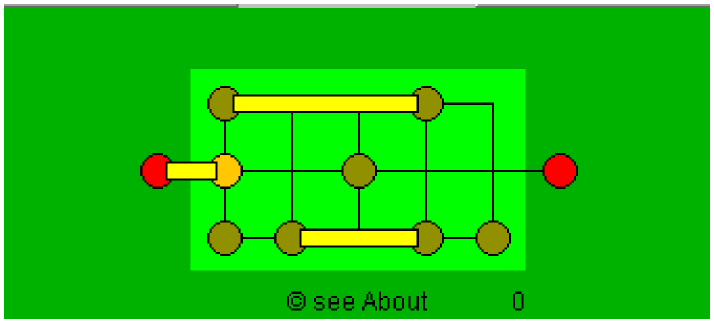
- reduction from 3SAT into Bounded NCL
- Bounded NCL is in NP

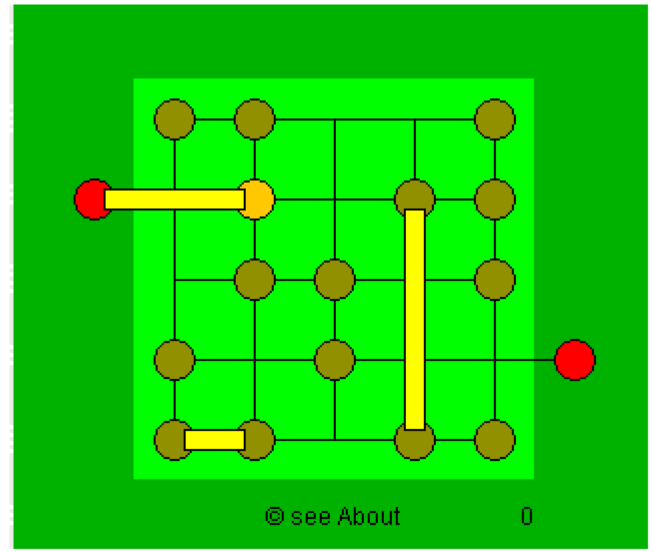
Bounded NCL is NP-complete

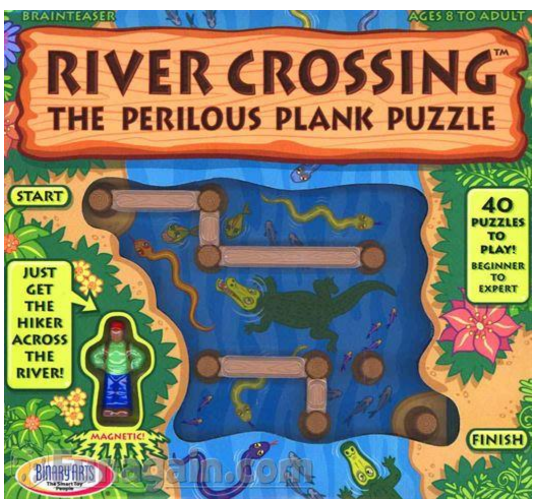
however: topling domino's cannot cross

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.