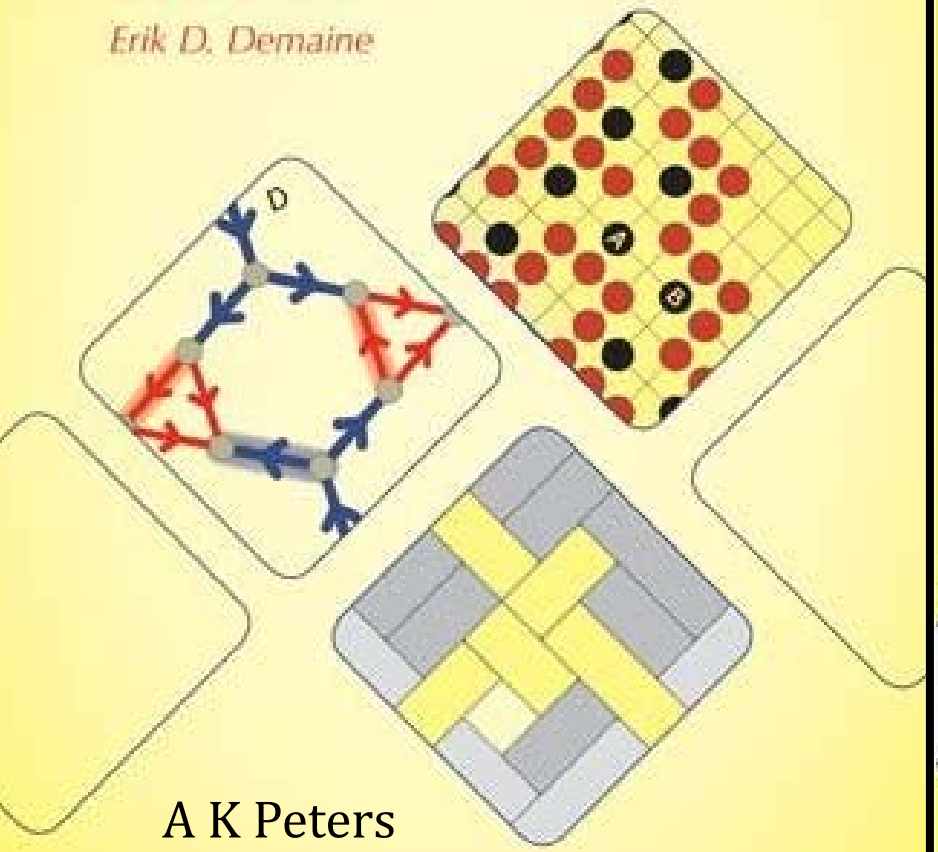
Games, Puzzles, & Computation

Robert A. Hearn
Erik D. Demaine



A K Peters
July 2009

ゲームとパズルの計算量

Games, Puzzles, & Computation

Robert A. Hearn
Erik D. Demaine

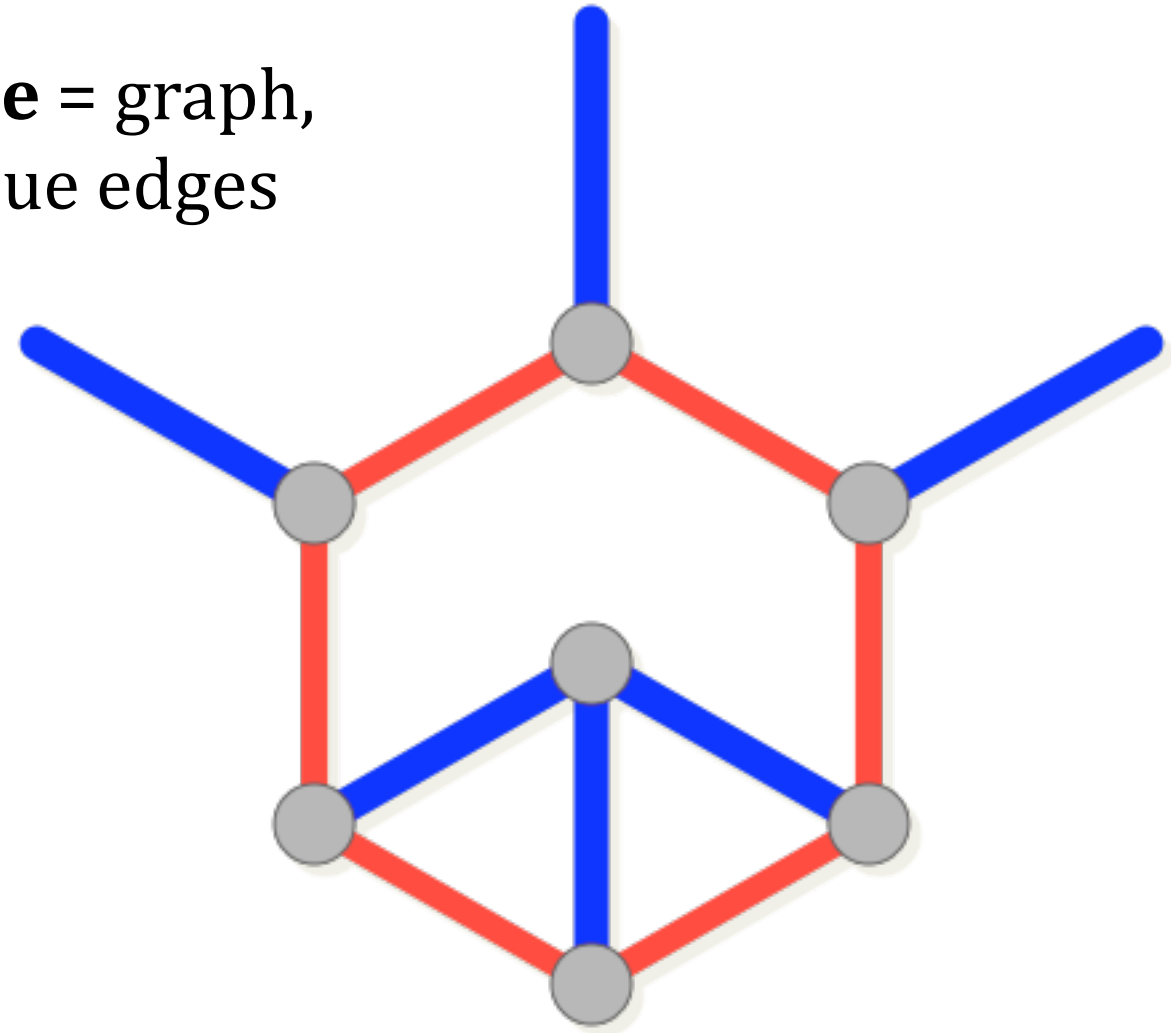
ロバート・A・ハーン ©著
エリック・D・ドメイン ©著
上原隆平 ©訳

translated by
Ryuhei Uehara

近代科学社

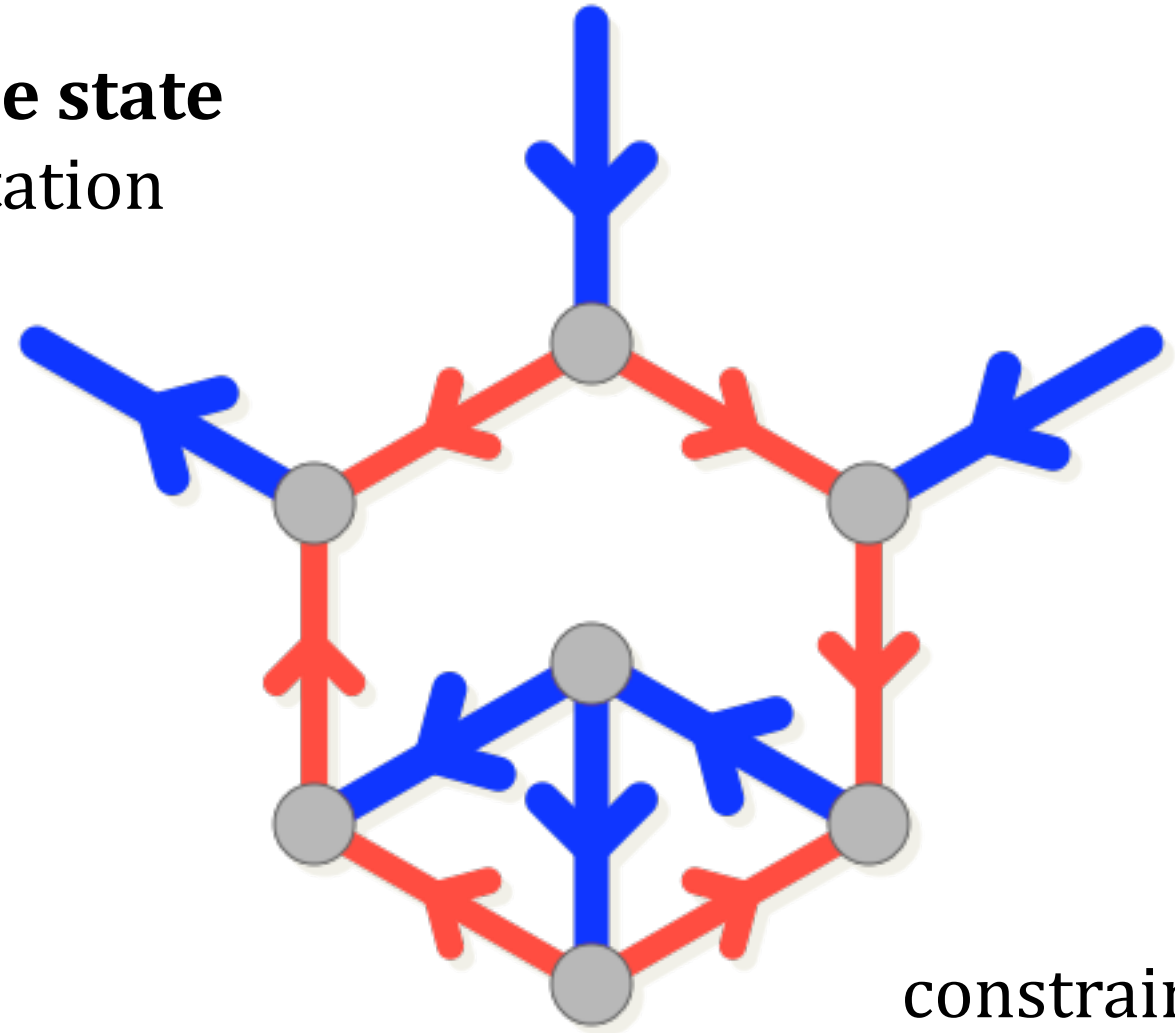
Constraint Graphs

Machine = graph,
red & blue edges



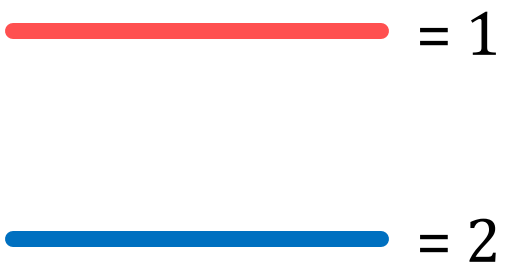
Constraint Graphs

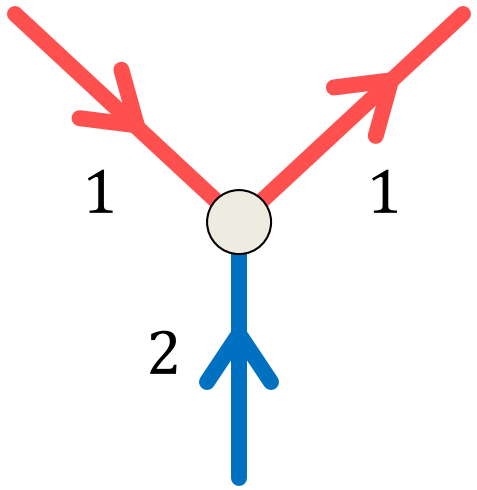
Machine state
= orientation



constraint graph

Constraint Logic

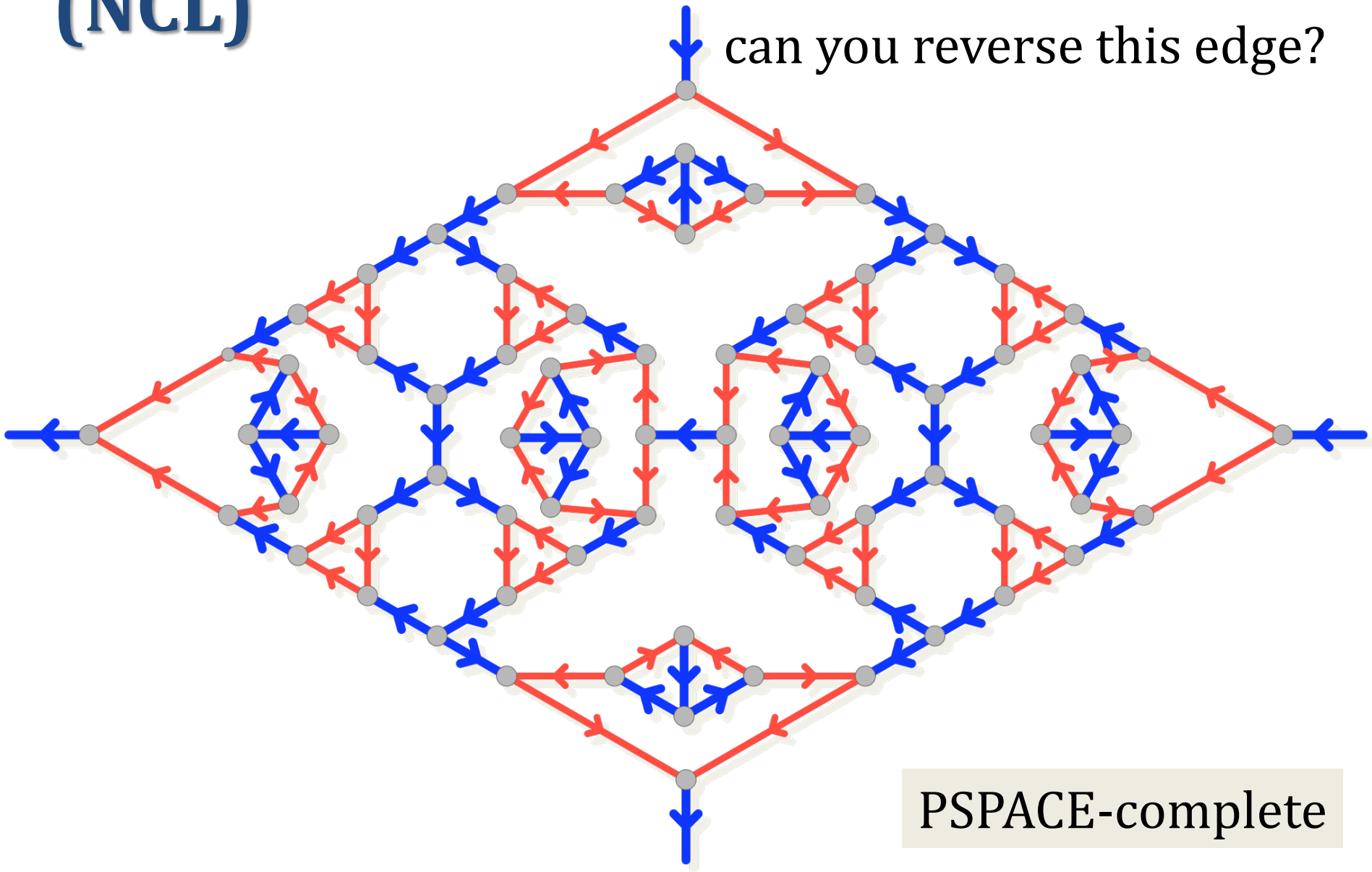




Rule: at least 2 units incoming at a vertex

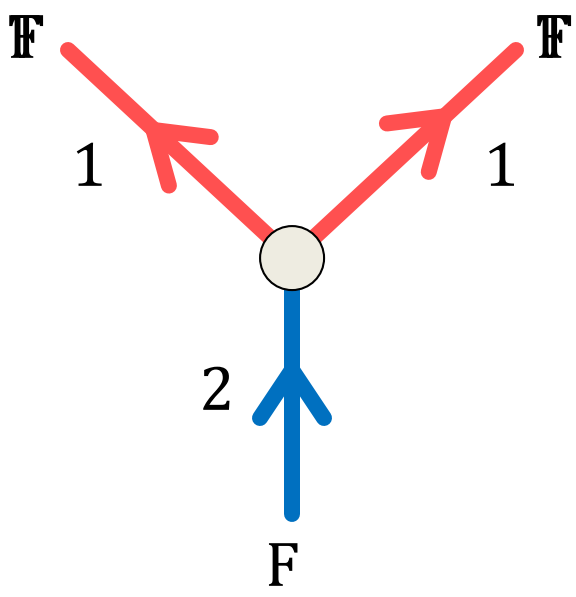
Move: reverse an edge, preserving Rule

Nondeterministic Constraint Logic (NCL)

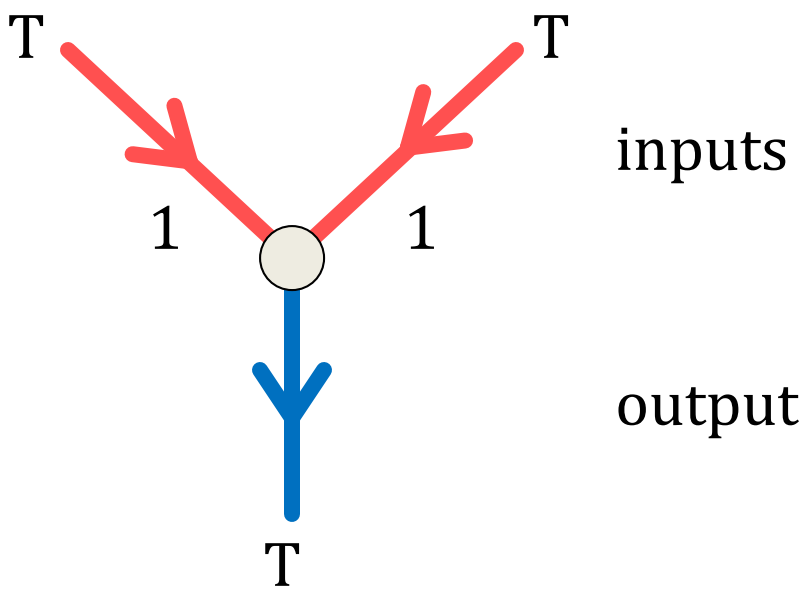


PSPACE-complete

AND vertex

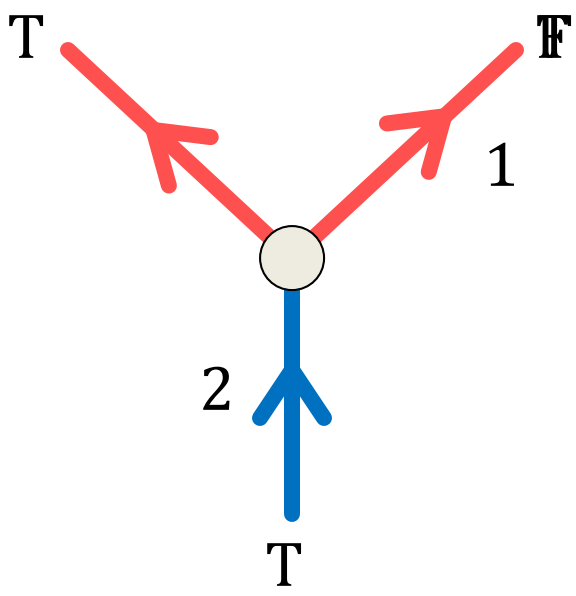


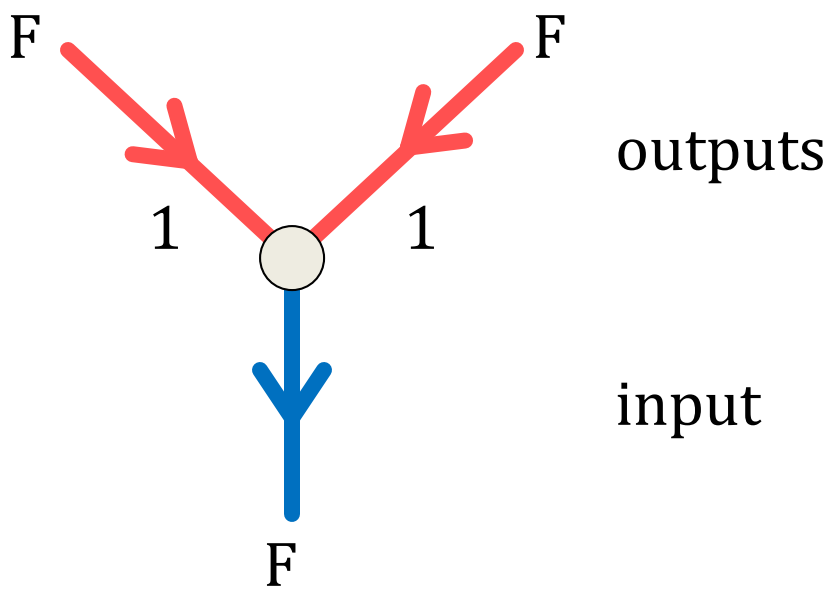
not your usual AND gate!



Rule: at least 2 units incoming at a vertex

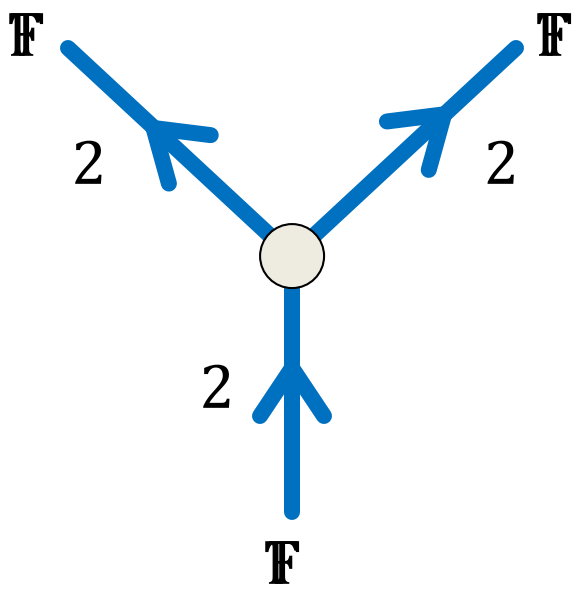
SPLIT vertex



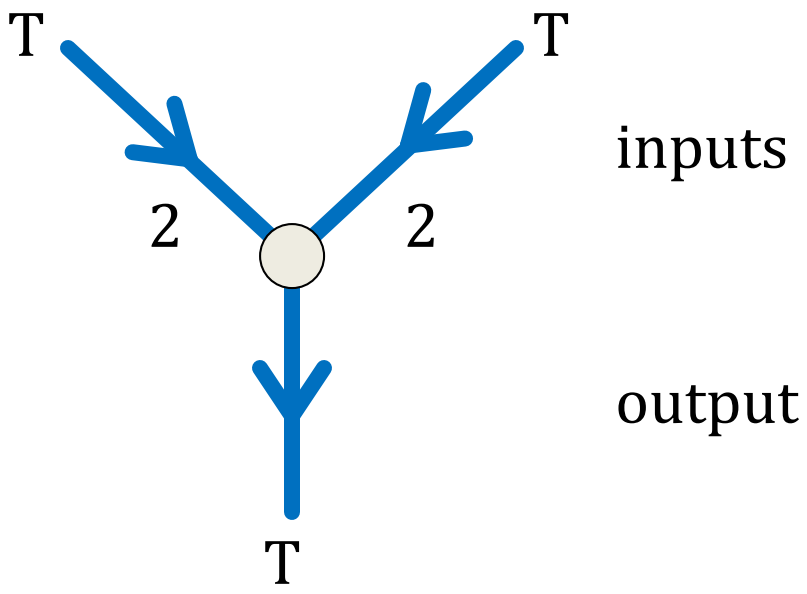


Rule: at least 2 units incoming at a vertex

OR vertex

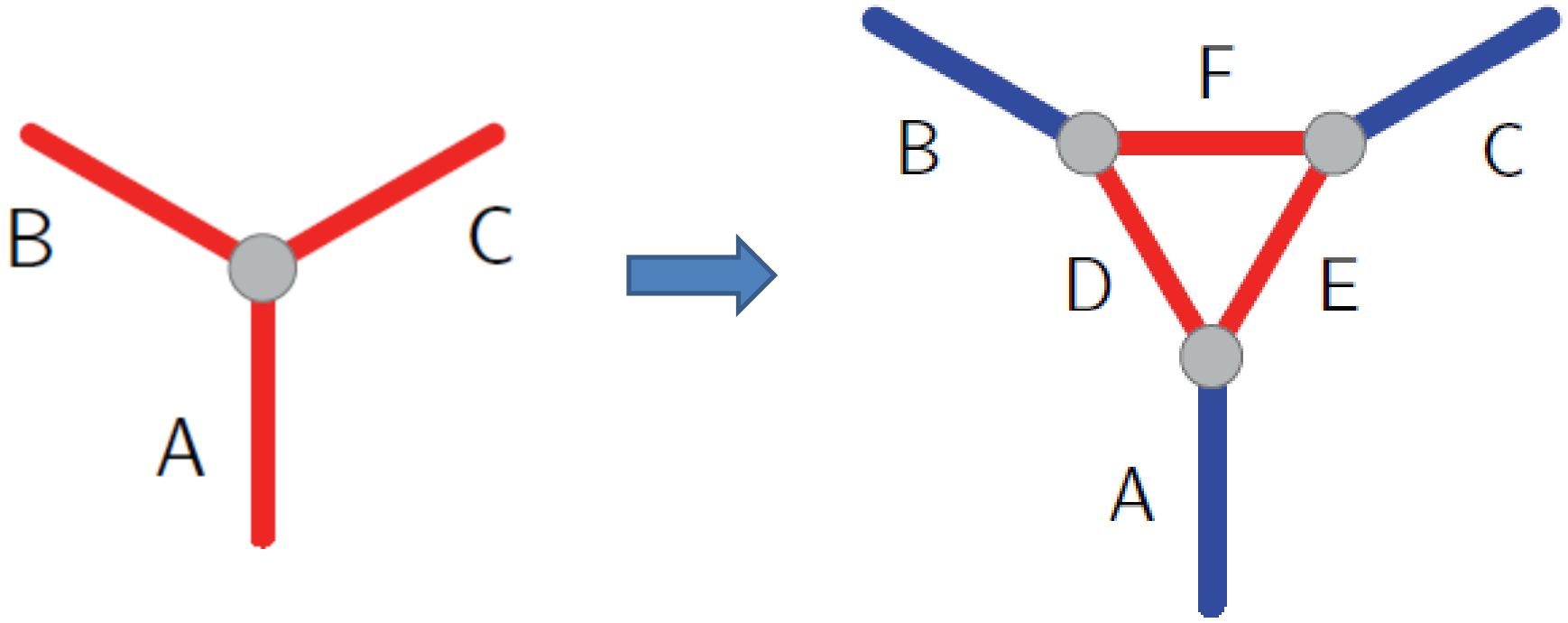


not your usual OR gate!

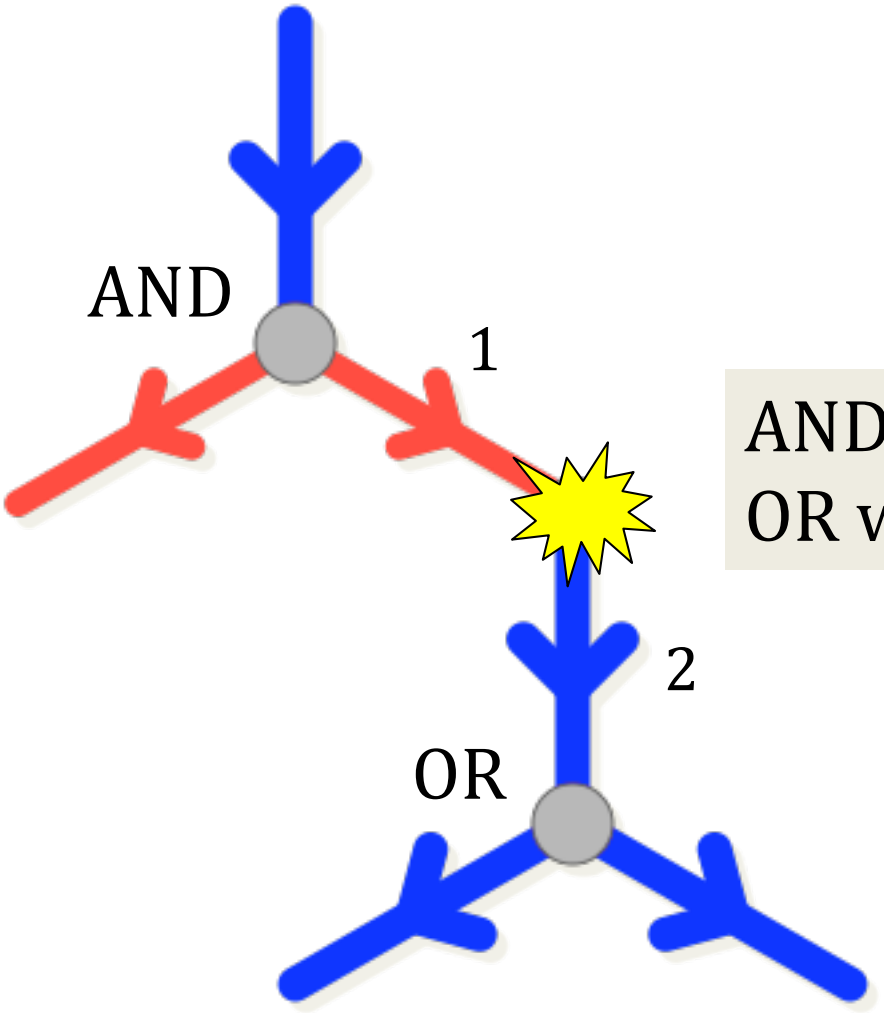


Rule: at least 2 units incoming at a vertex

CHOICE vertex



Wiring Vertices Together

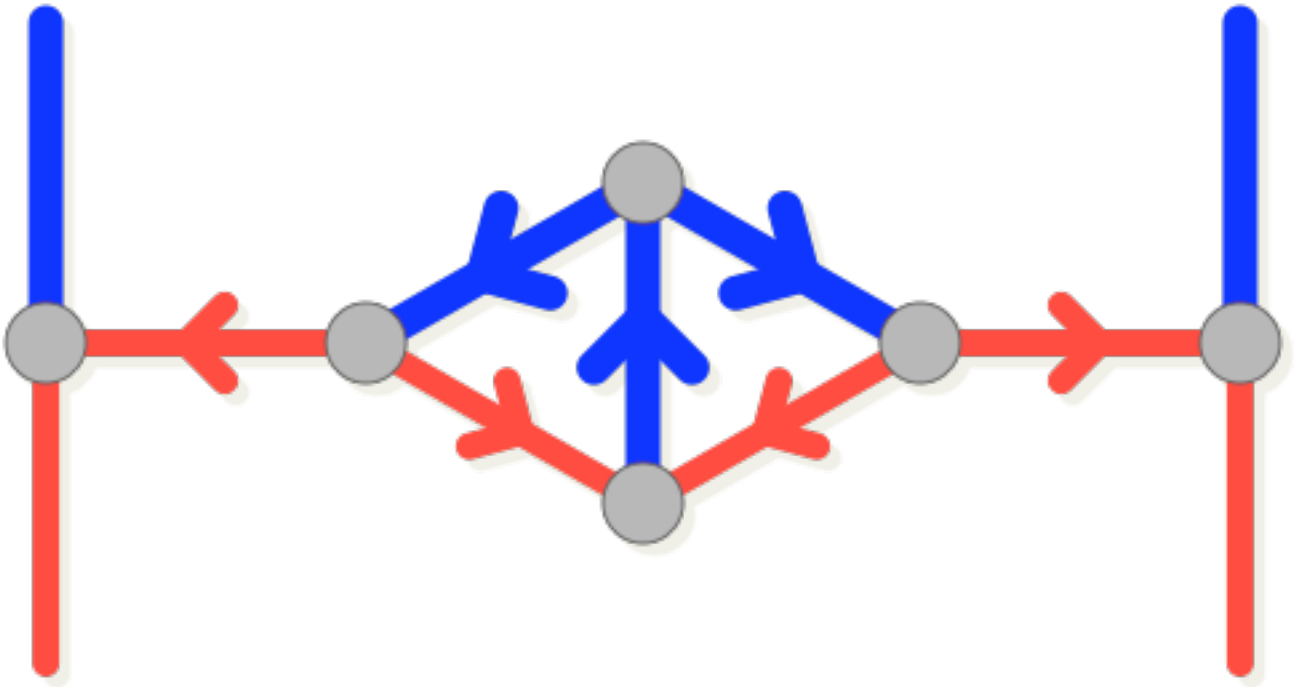


AND wants red
OR wants blue

Red-Blue Conversion

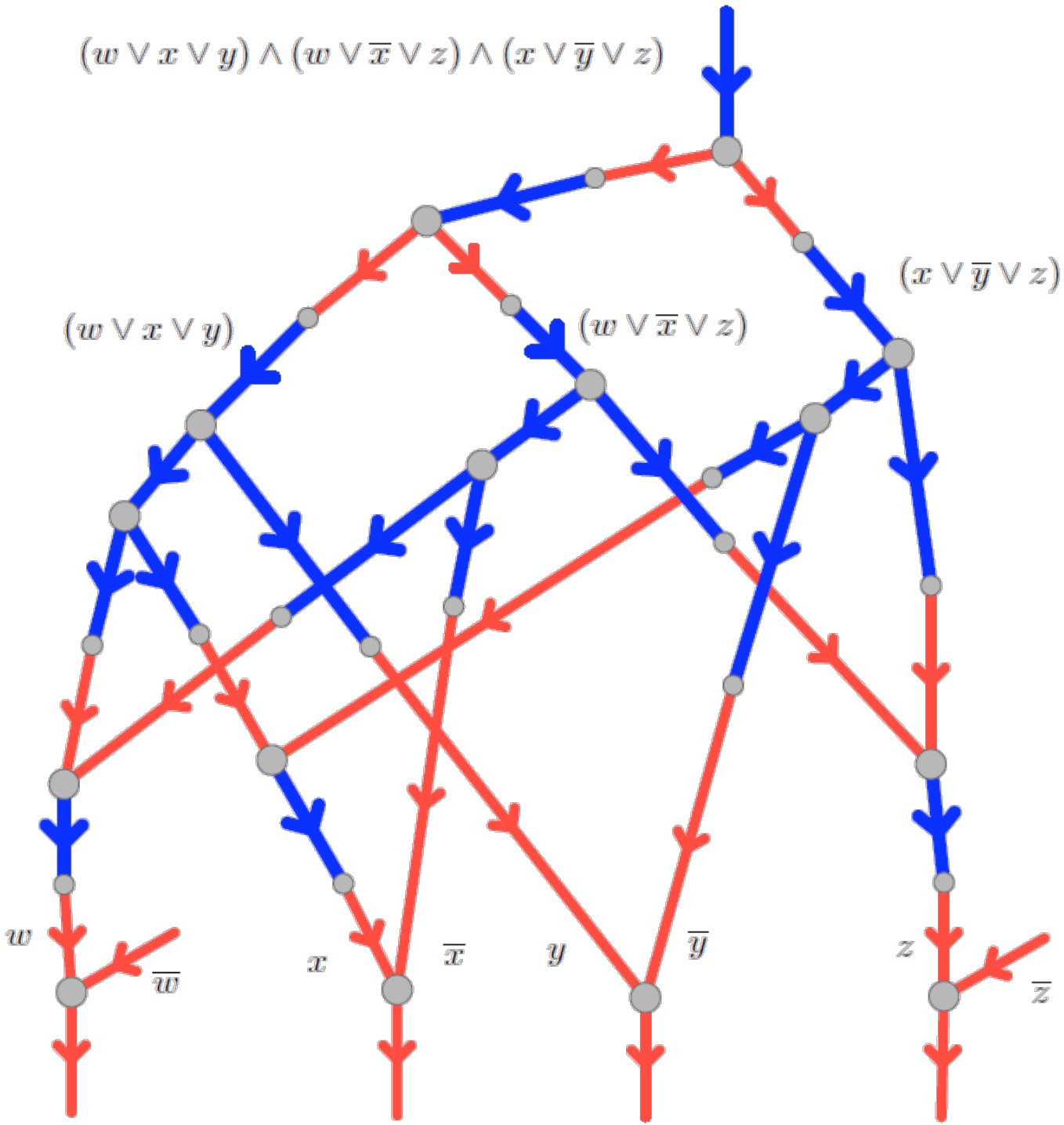
assume an even number of conversions

Red-Blue Conversion

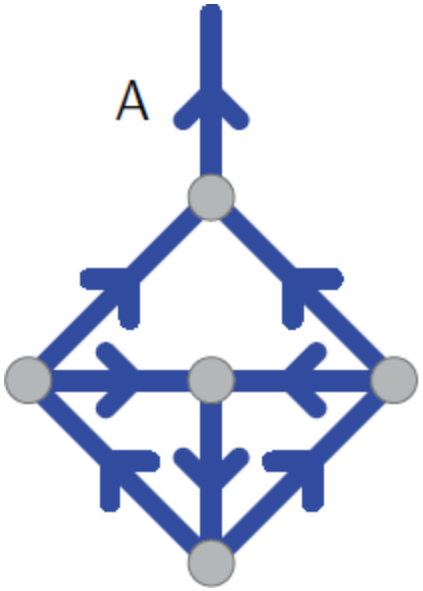


assume an even number of conversions

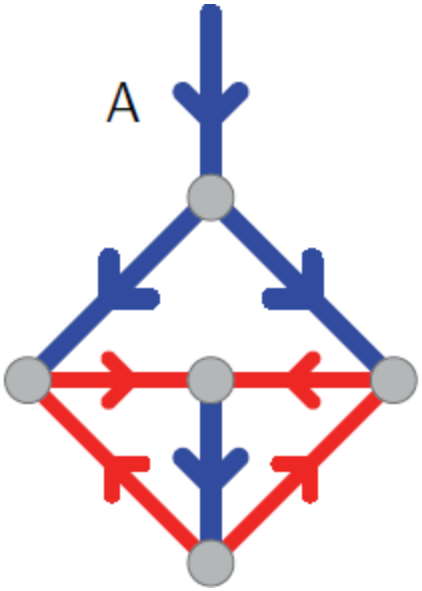
CNF Formula


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

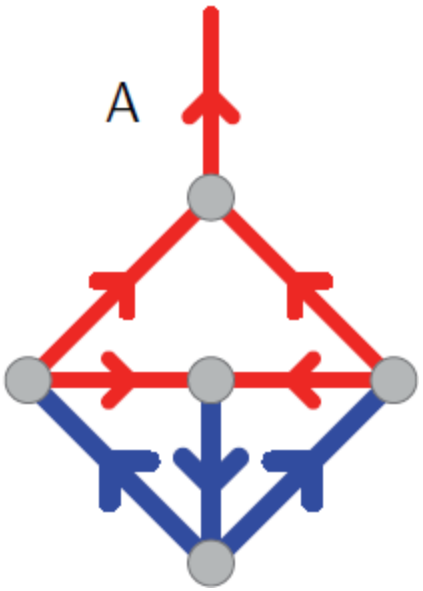
Wire Terminators



unconstrained
blue terminator

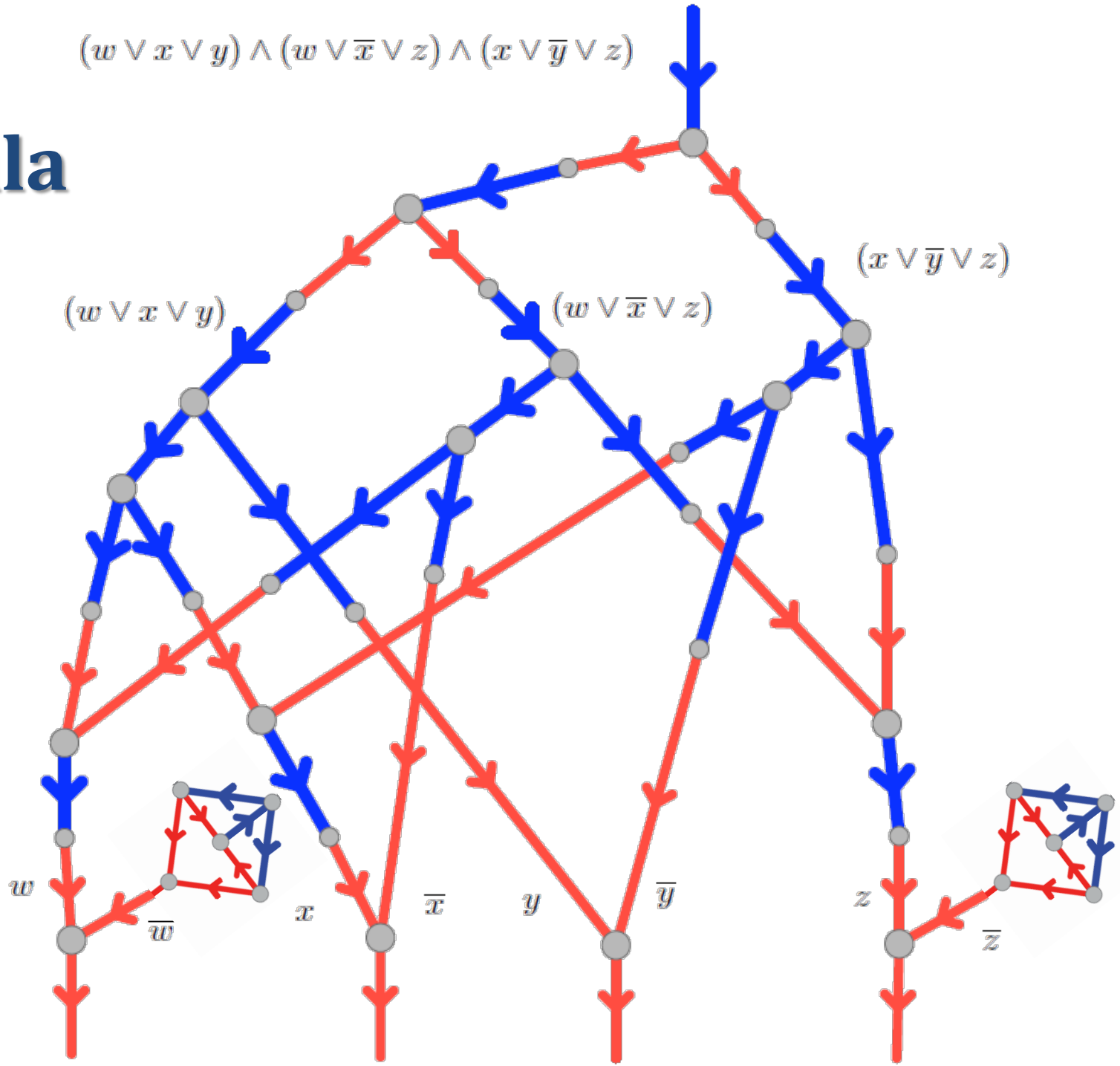


forced-inward
blue terminator

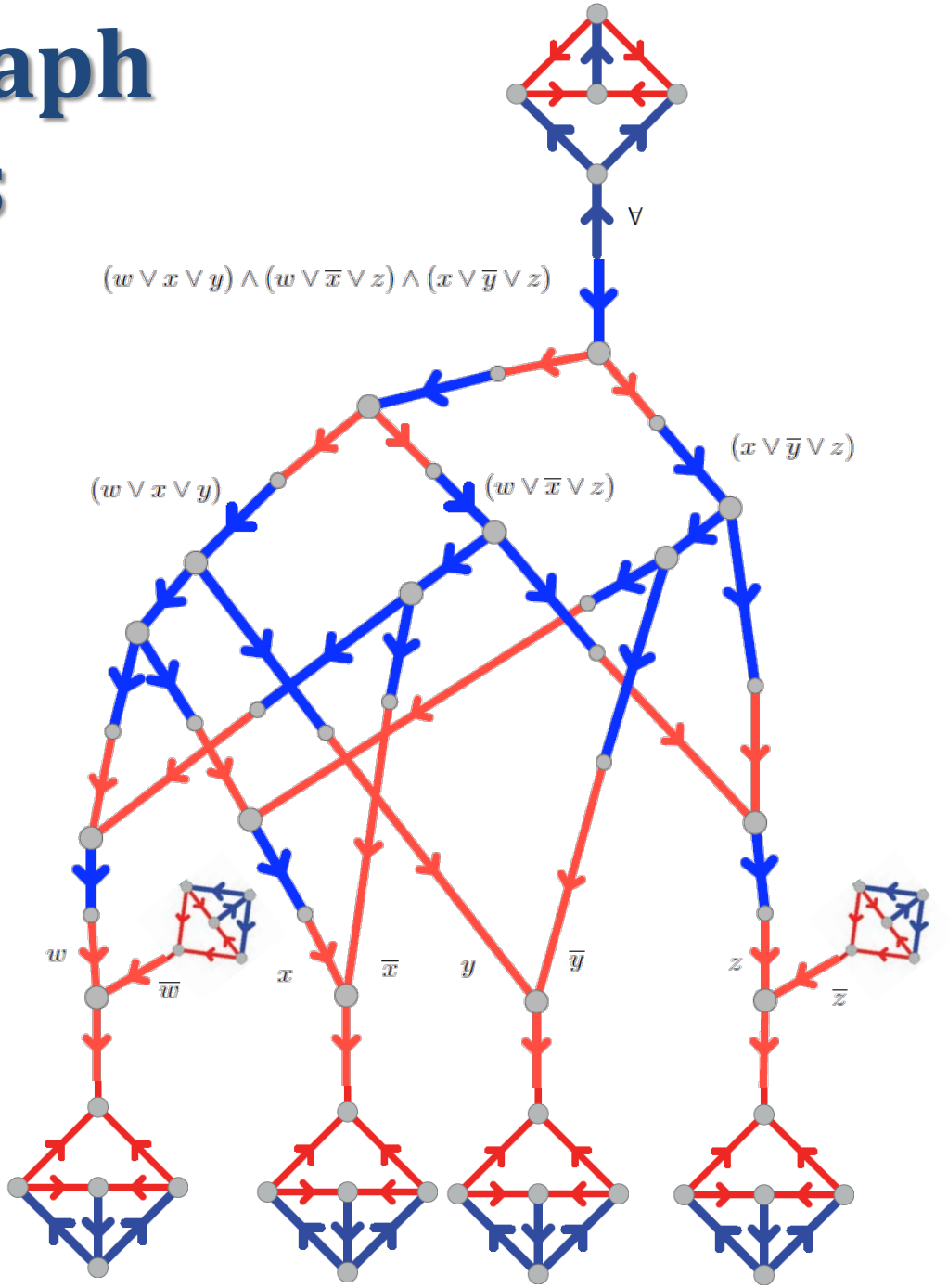


unconstrained
red terminator

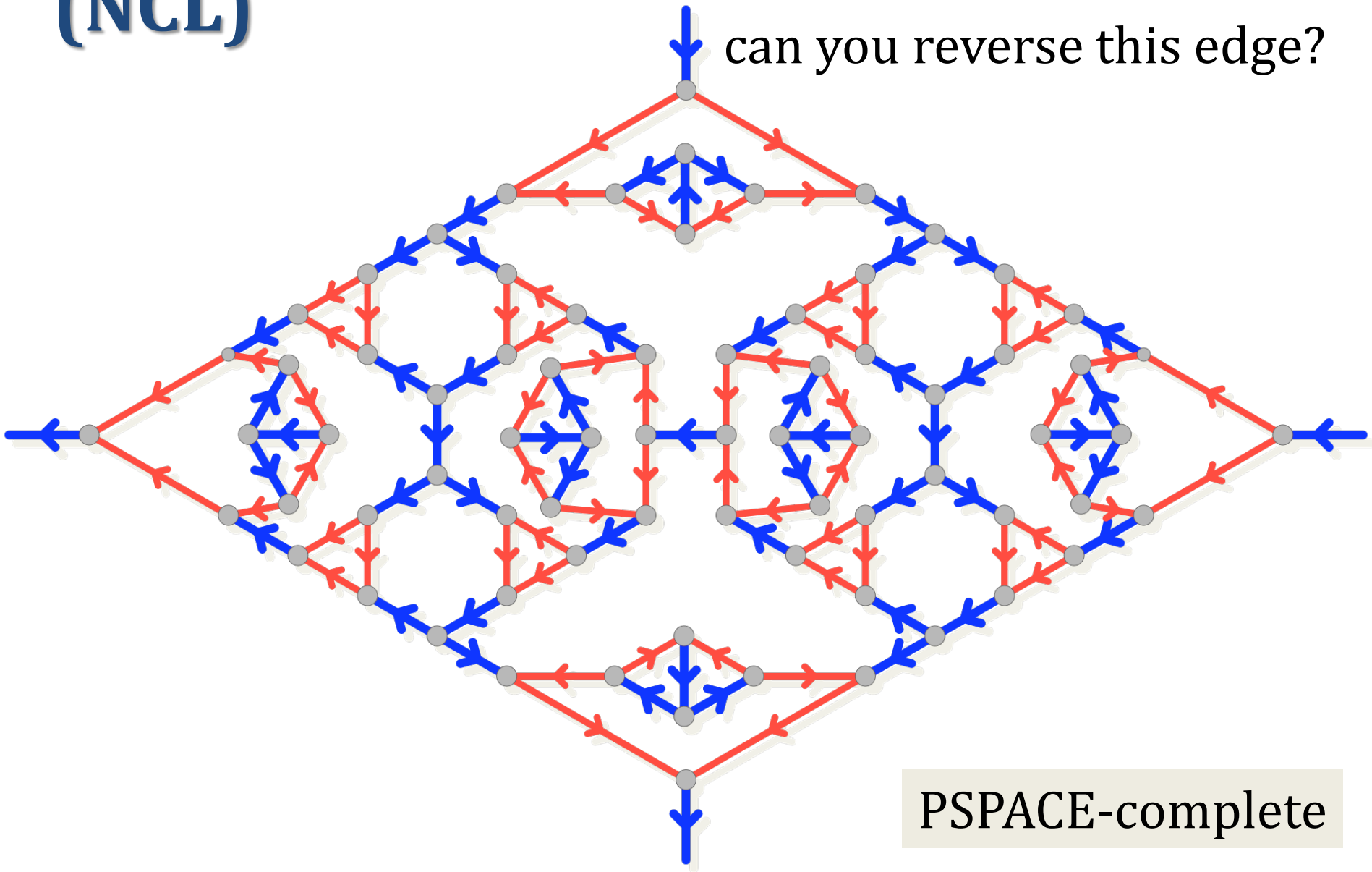
CNF Formula


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

Constraint Graph Satisfaction is NP-complete



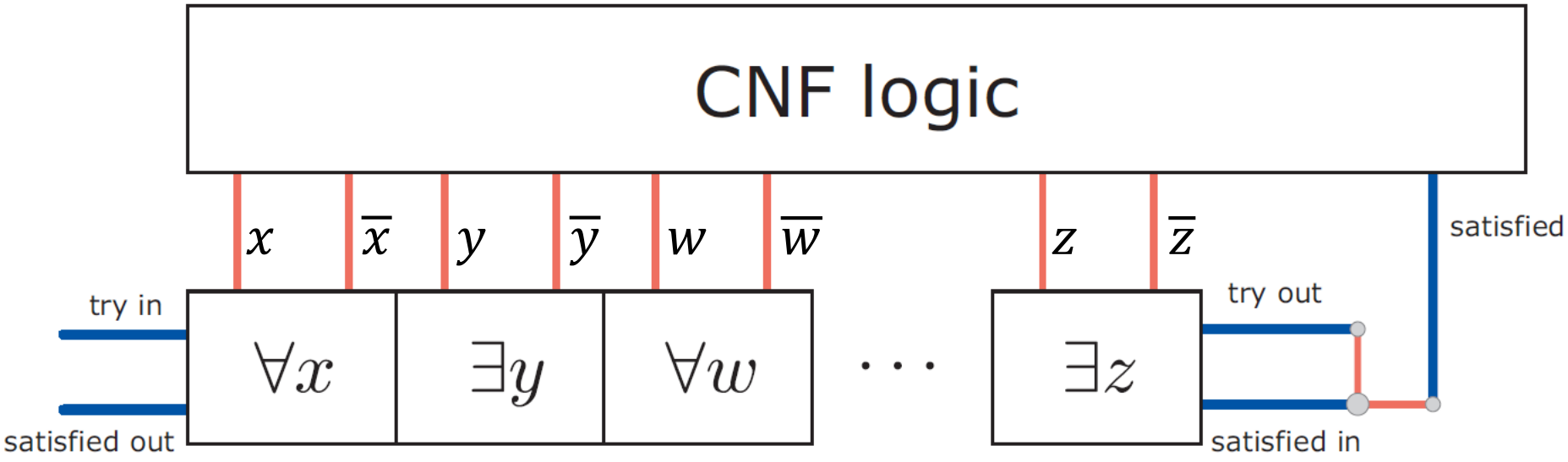
Nondeterministic Constraint Logic (NCL)



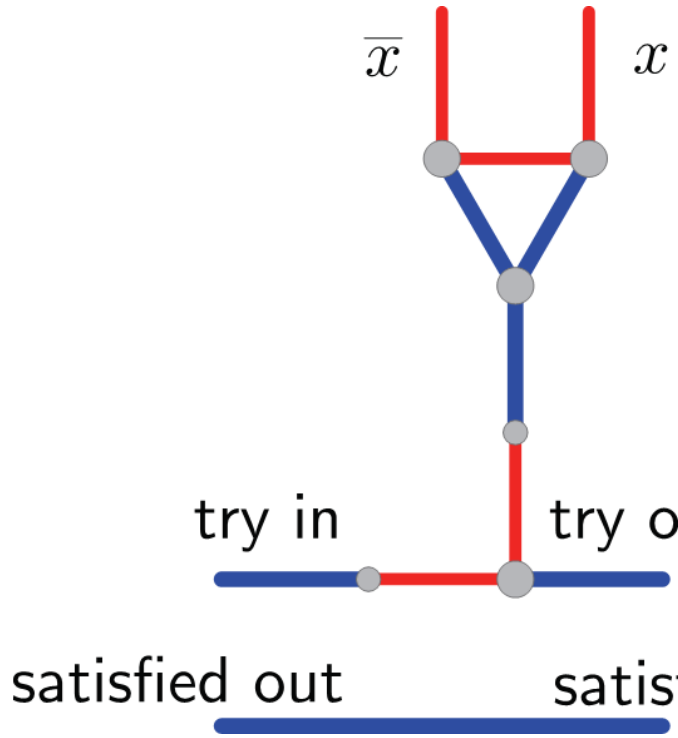
PSPACE-complete

Quantified Boolean Formulas (QBF/QSAT)

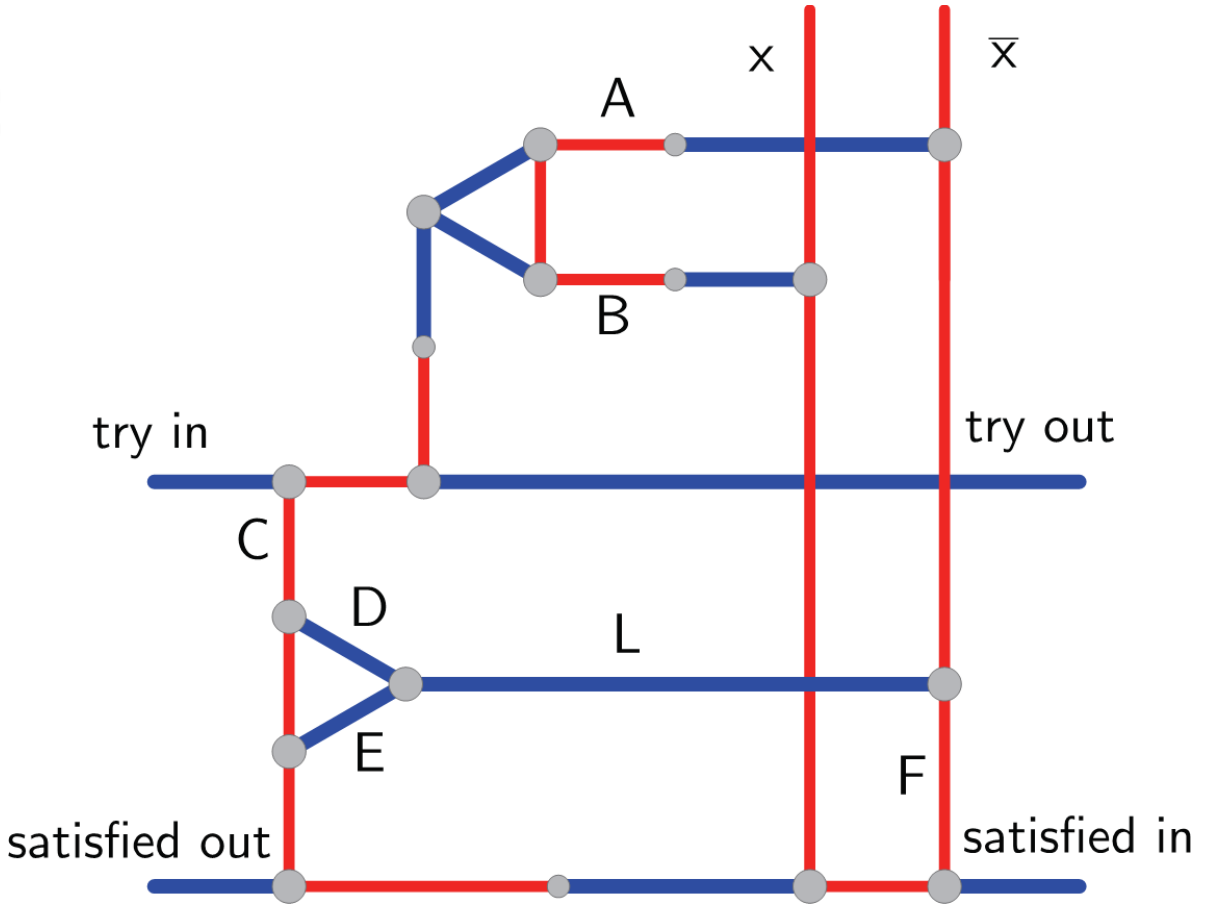
$$\forall x \exists y \forall w \cdots \exists z \left[(x \vee y) \wedge \cdots \wedge (\overline{z} \vee x \vee \overline{w}) \right]$$



Quantifiers

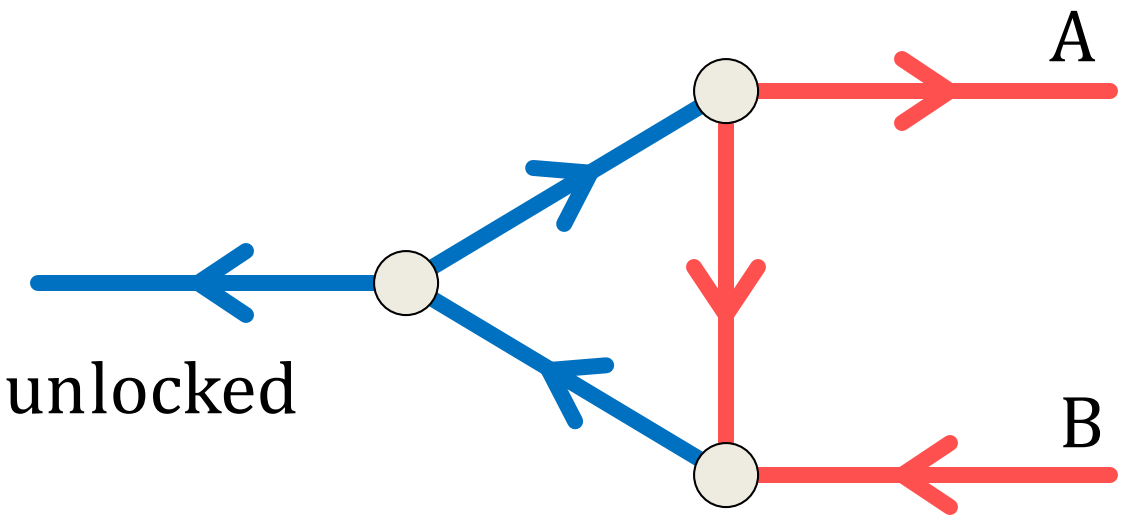


existential

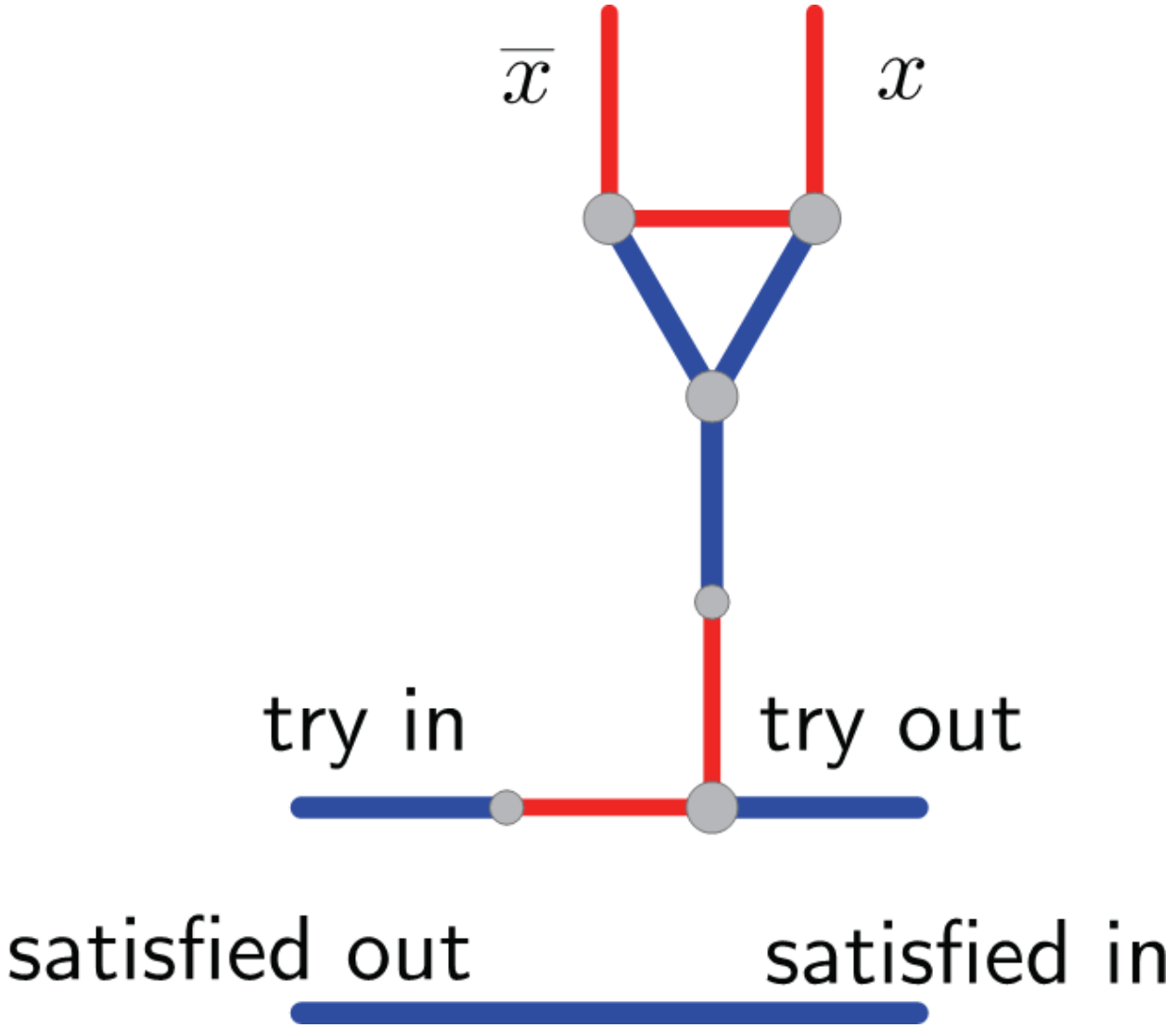


universal

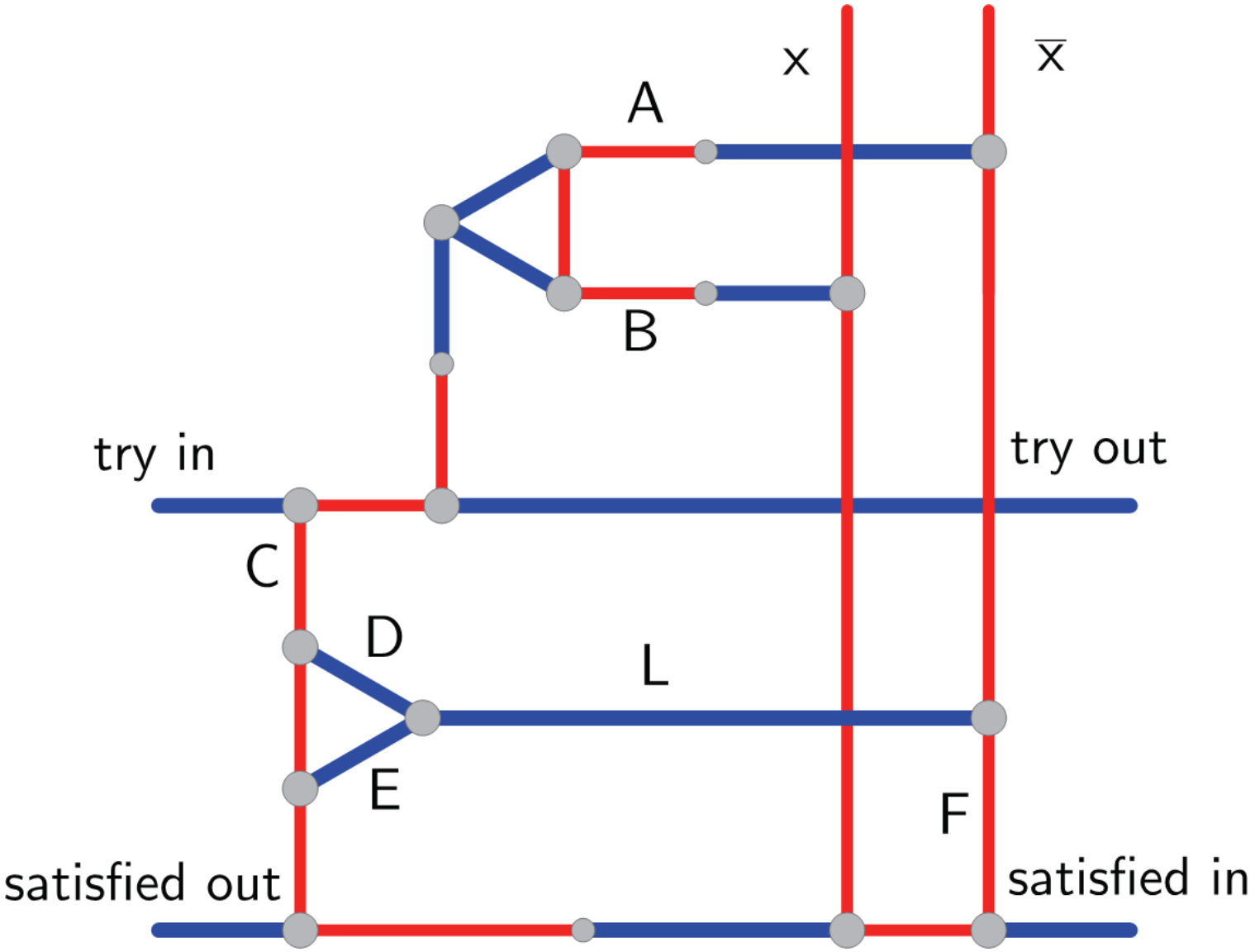
Latch



Existential Quantifier

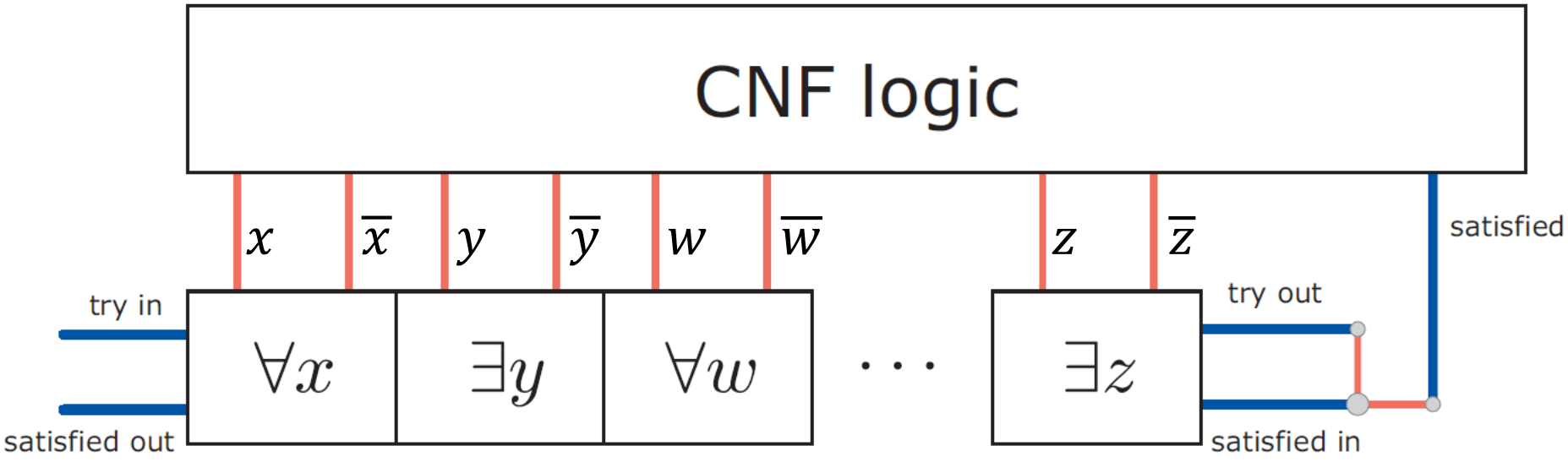


Universal Quantifier

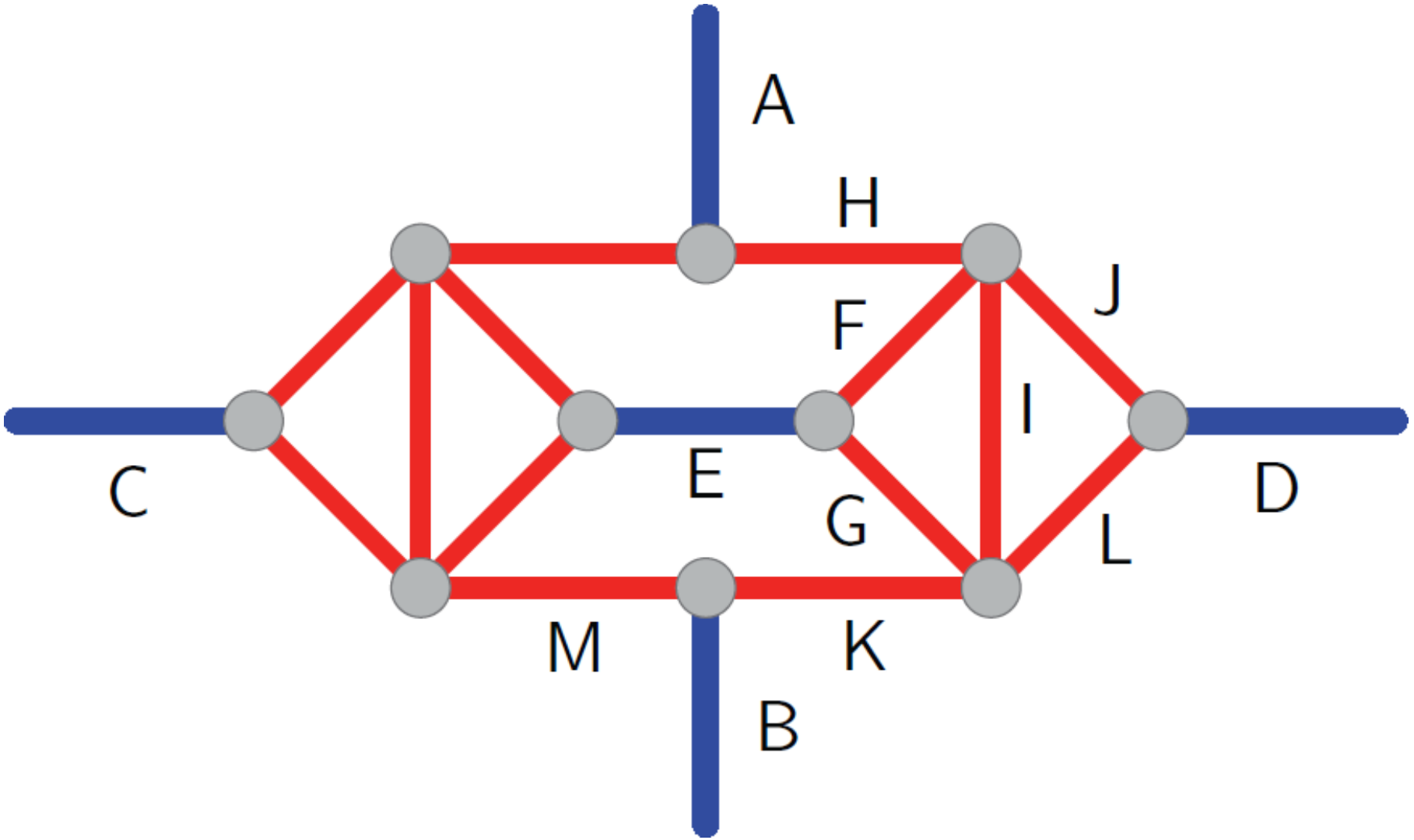


Quantified Boolean Formulas (QBF/QSAT)

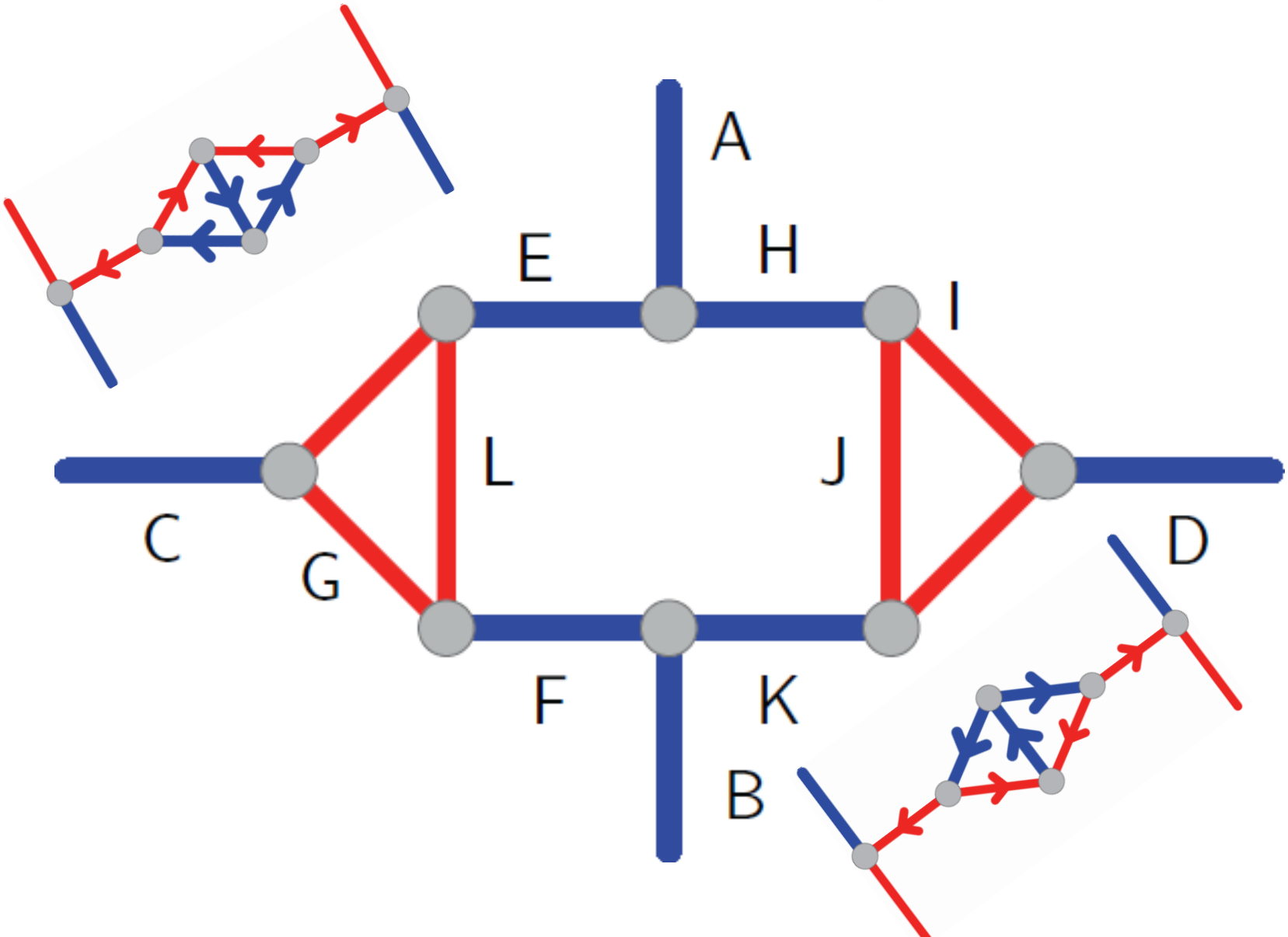
$$\forall x \exists y \forall w \cdots \exists z \left[(x \vee y) \wedge \cdots \wedge (\overline{z} \vee x \vee \overline{w}) \right]$$



Crossover Gadget

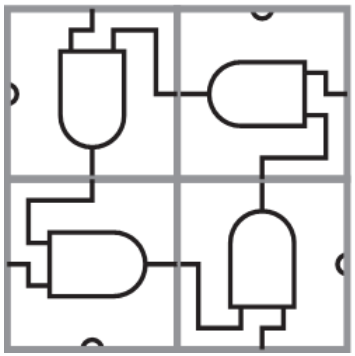


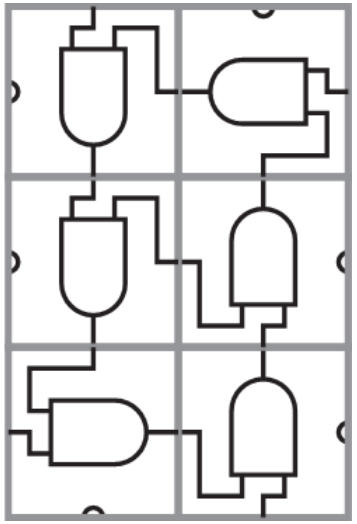
Vertex with 4 Red Edges



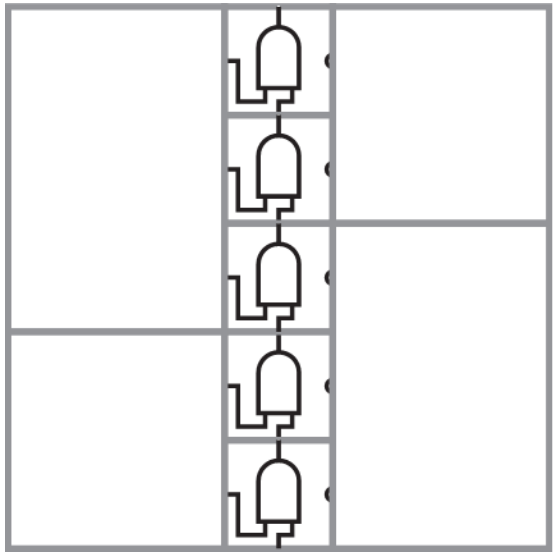
Grid Constraint Graphs: Straights & Turns

filler

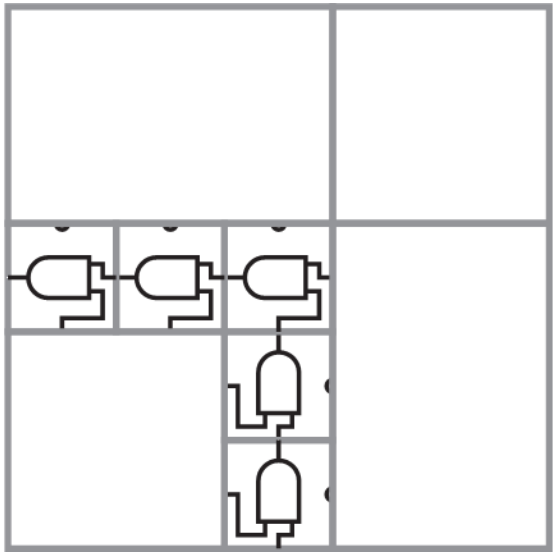




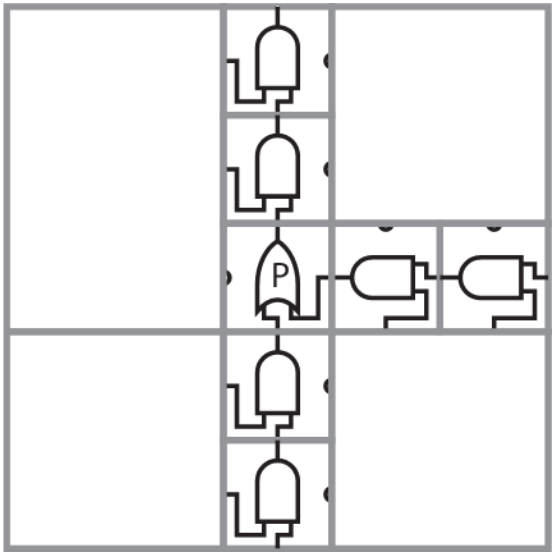
straight



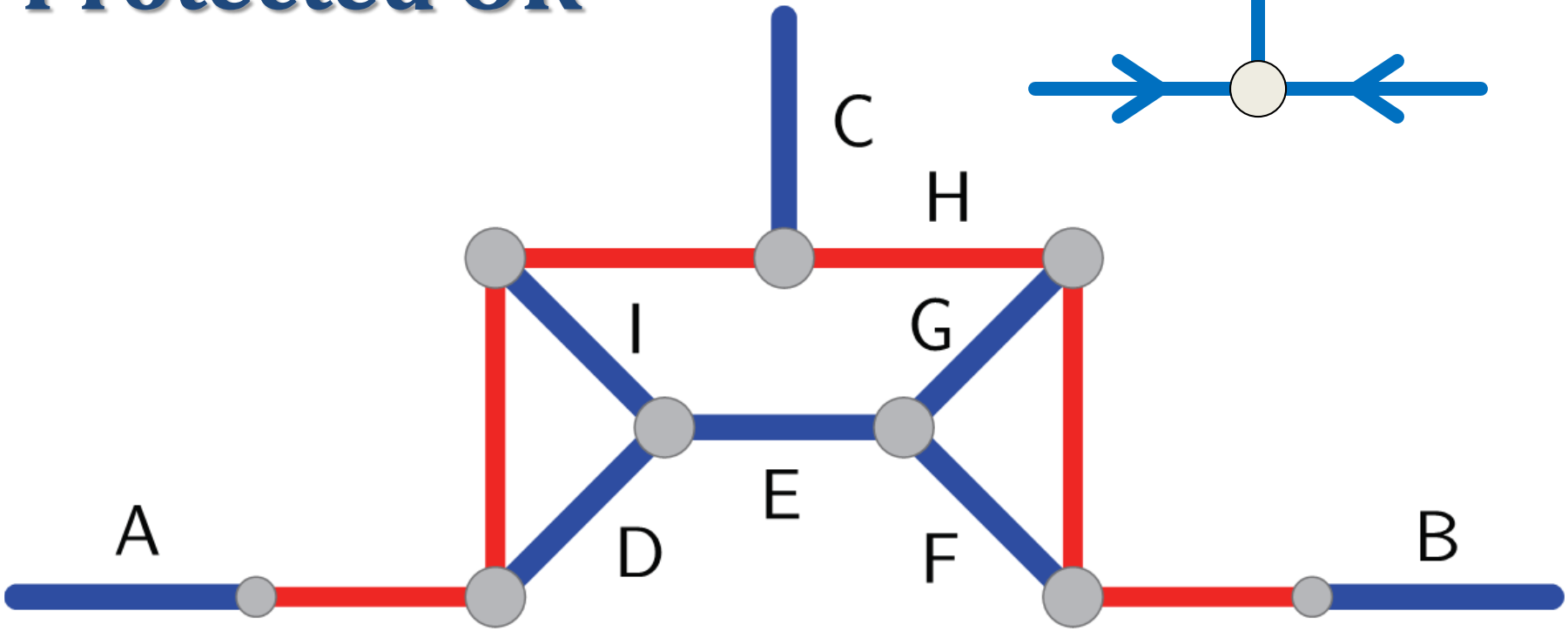
turn

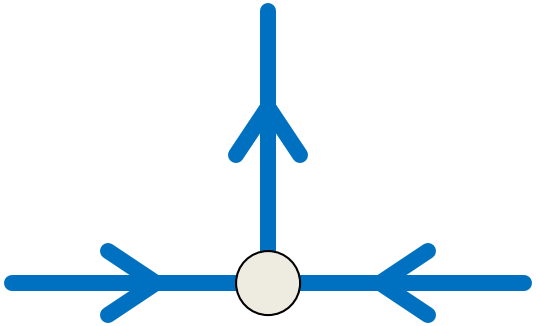


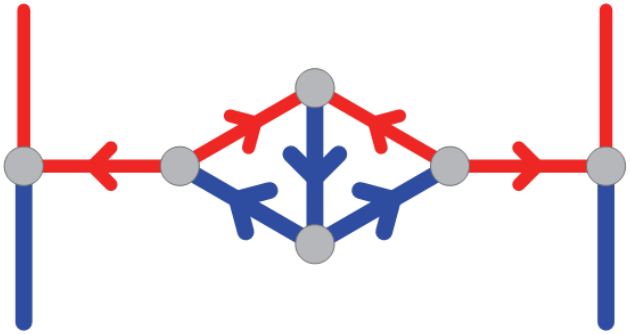
OR

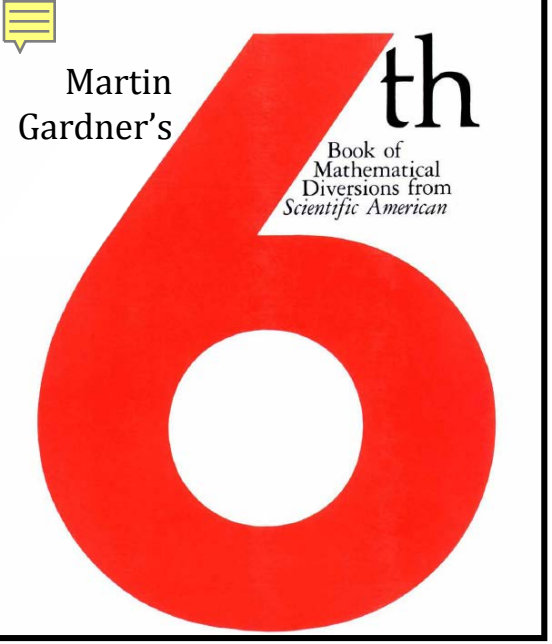


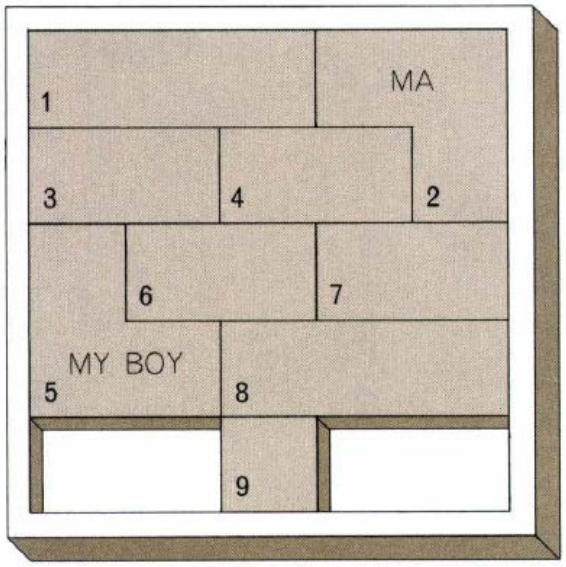
OR from Protected OR



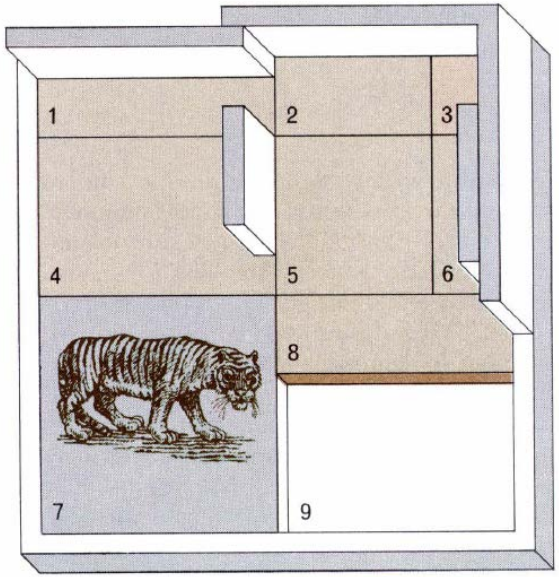






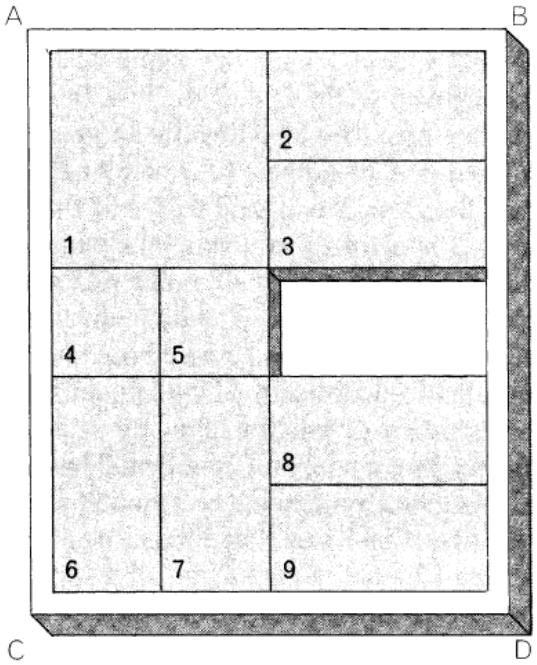


Ma's Puzzle

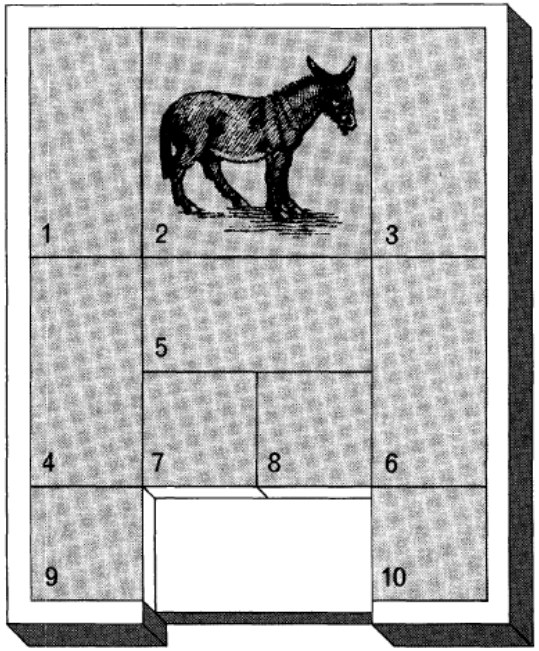


L'Ane Rouge

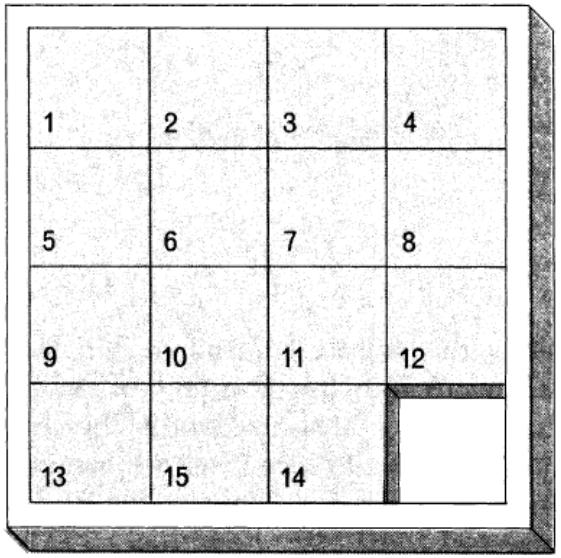
7. Sliding-Block Puzzles



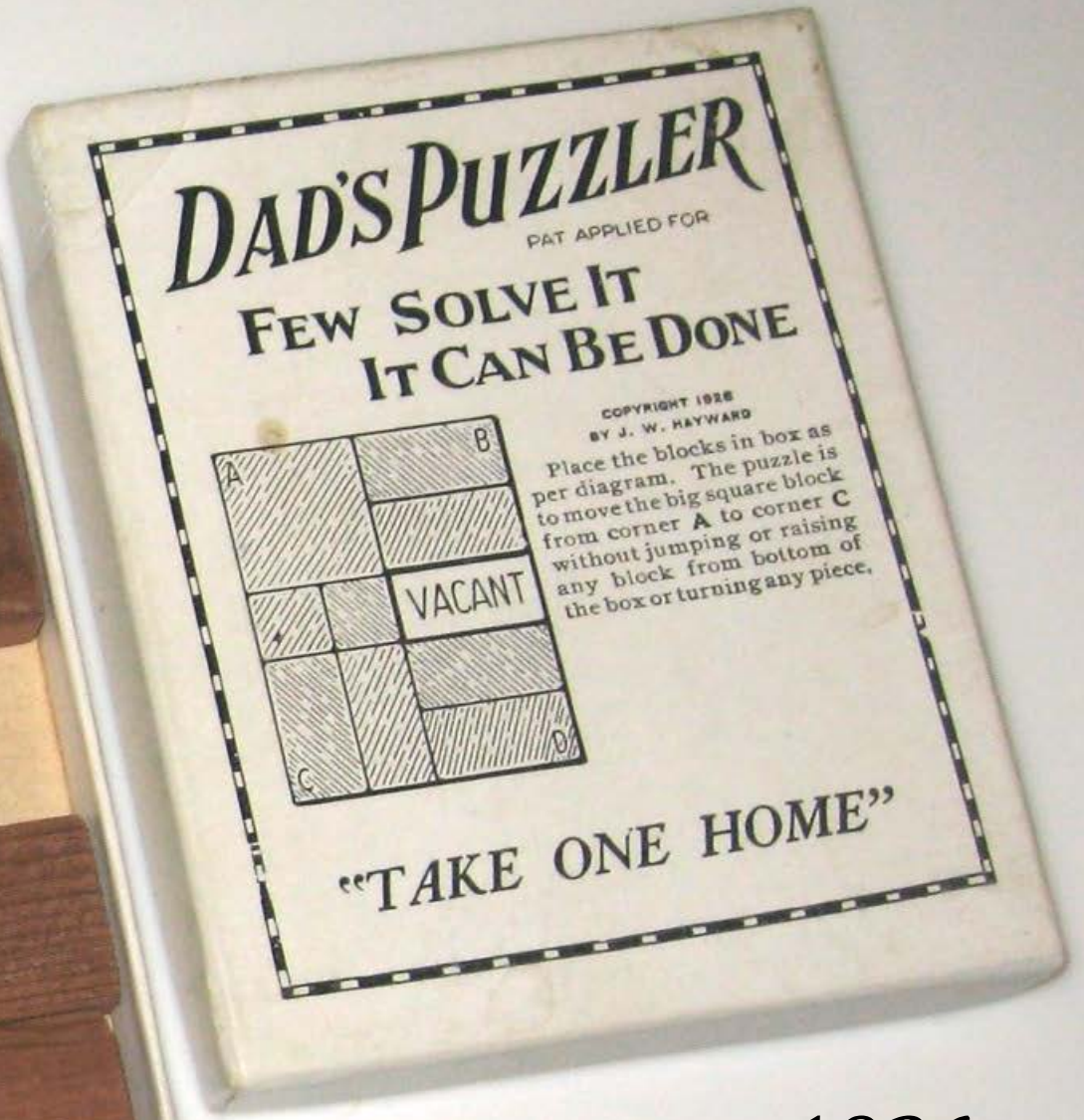
Dad's Puzzle (1926)



Sam Loyd's 15 Puzzle



Sliding-Block Puzzles



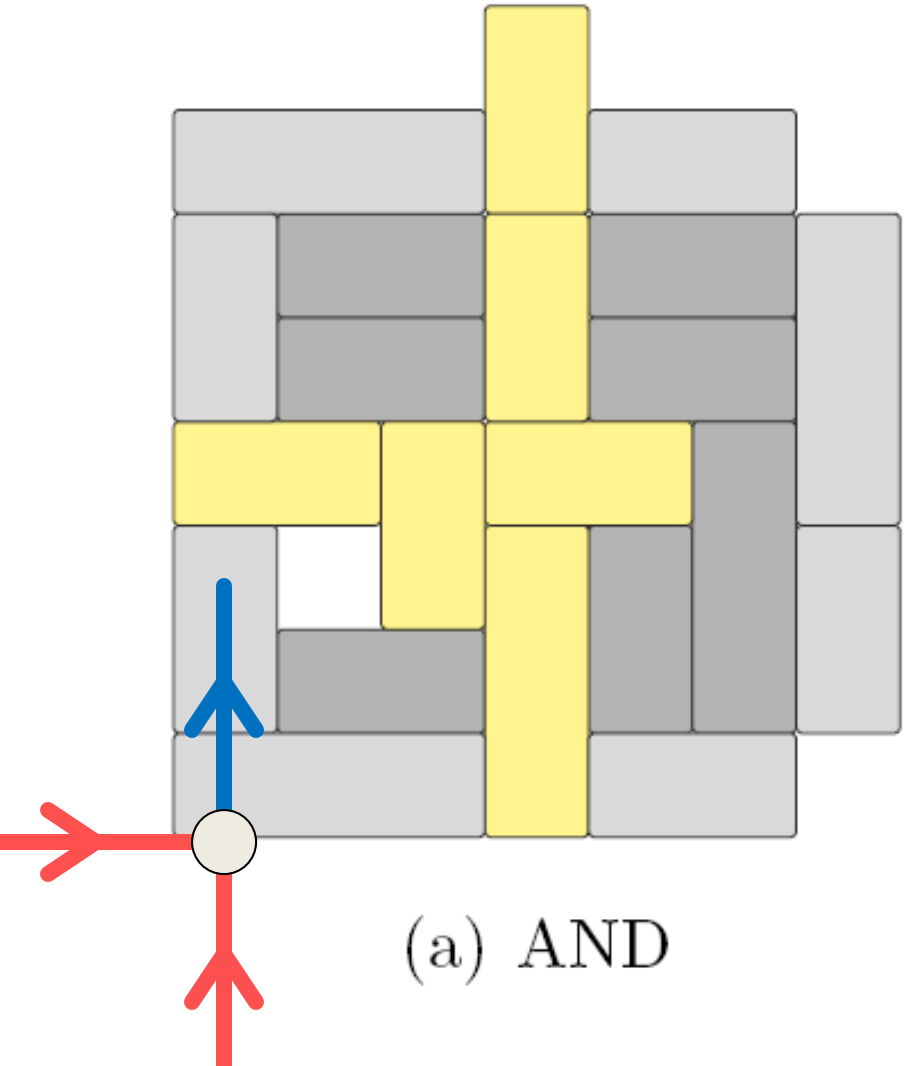
1926

83 moves

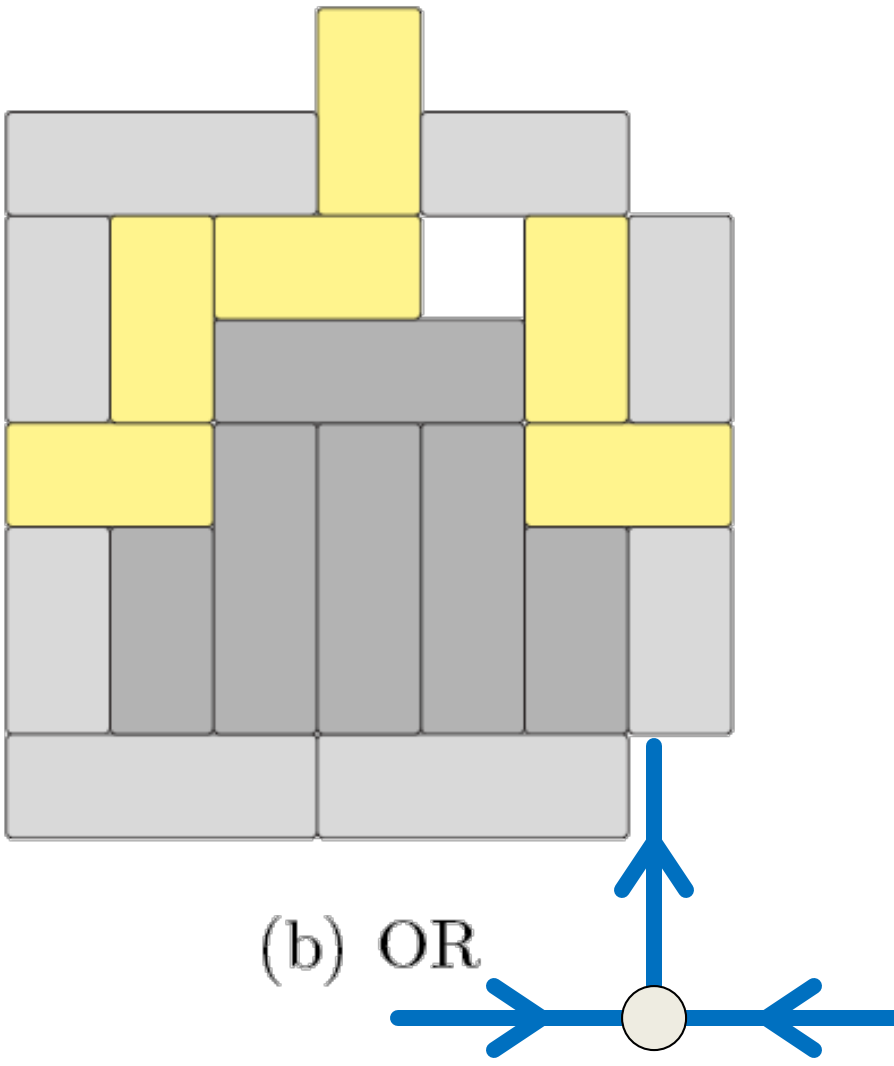
Sliding-Block Puzzles

[Hearn & Demaine 2002]

PSPACE-complete



(a) AND

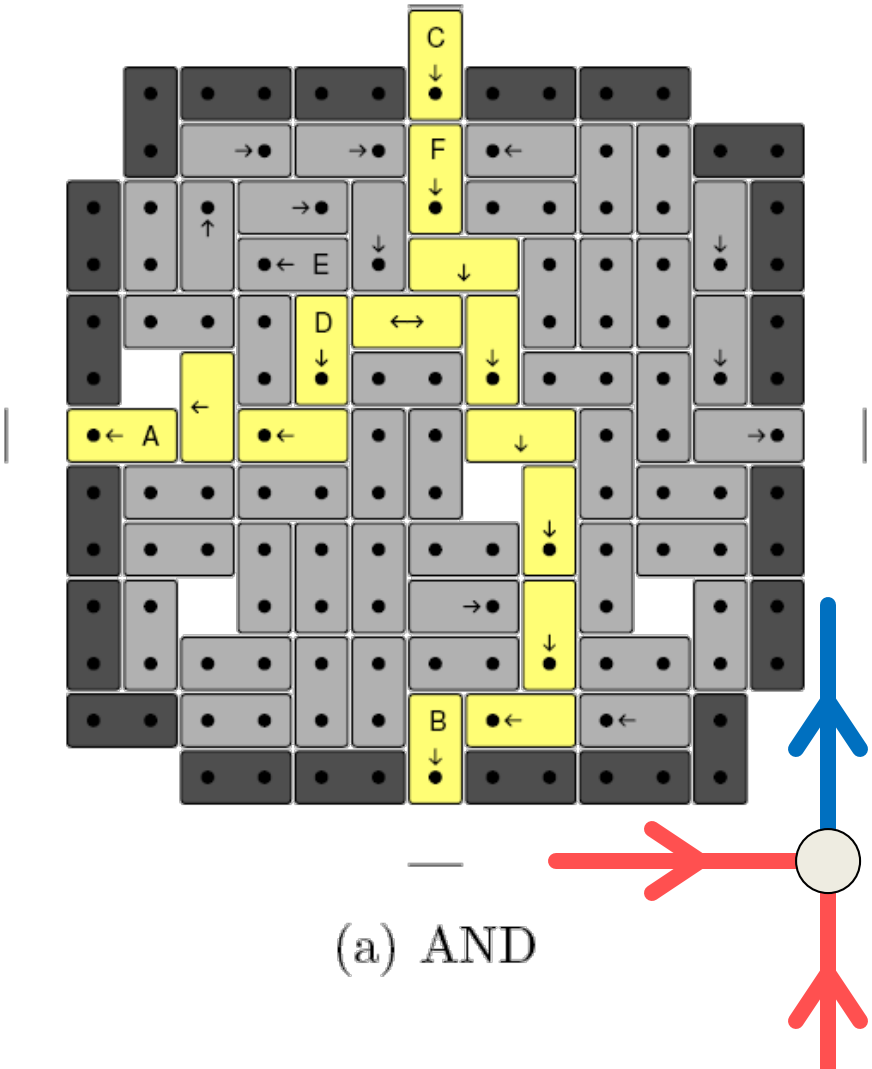


(b) OR

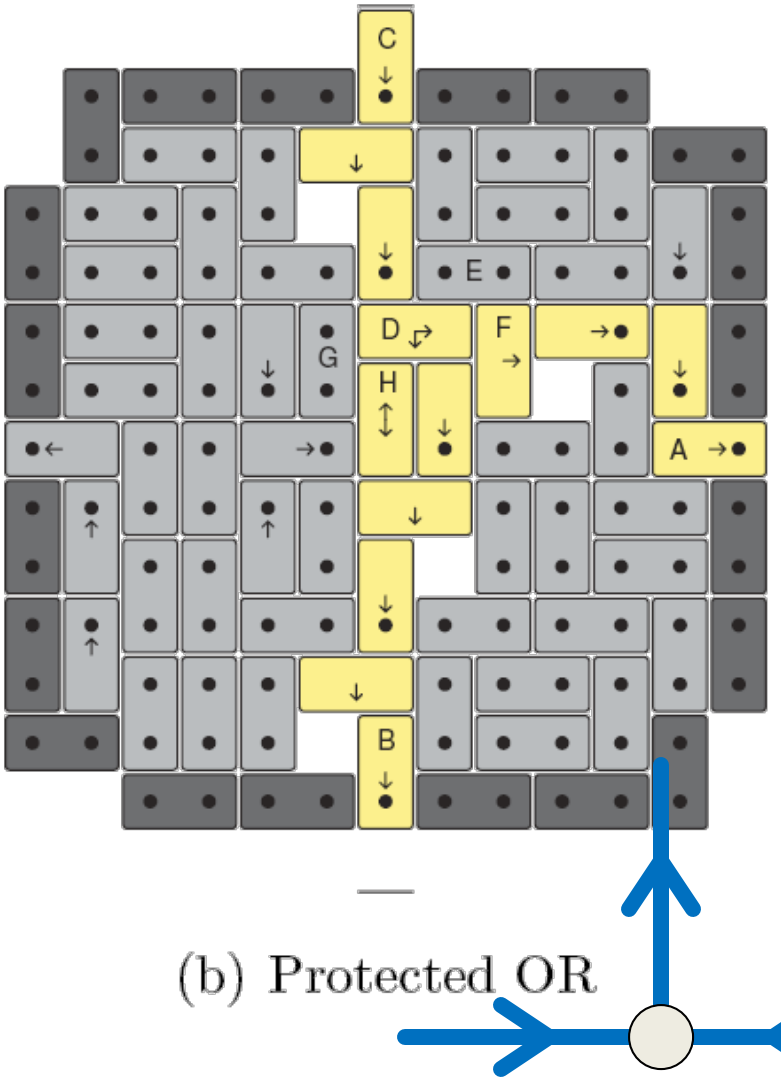
Sliding-Block Puzzles

[Hearn & Demaine 2002]

PSPACE-complete

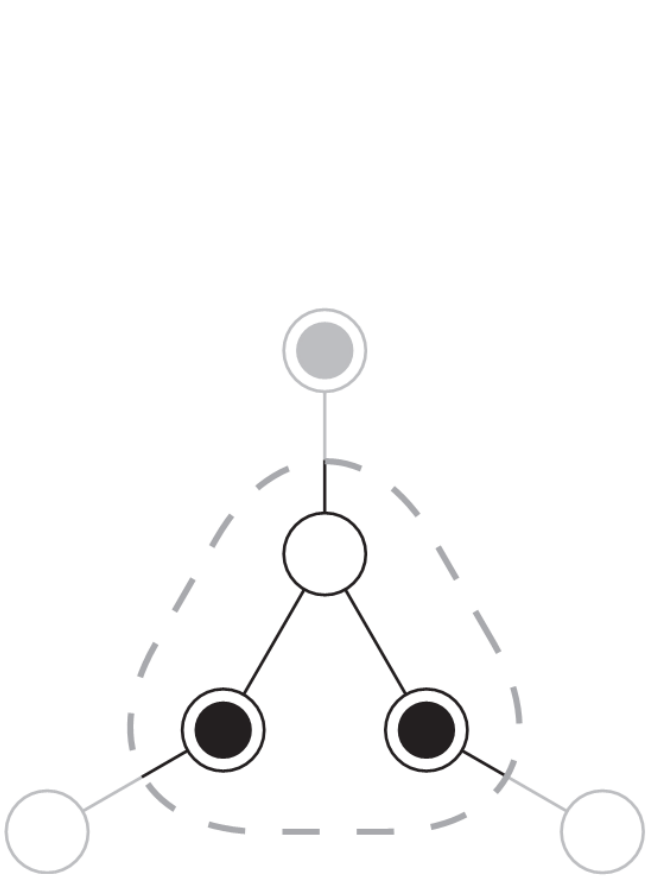


(a) AND

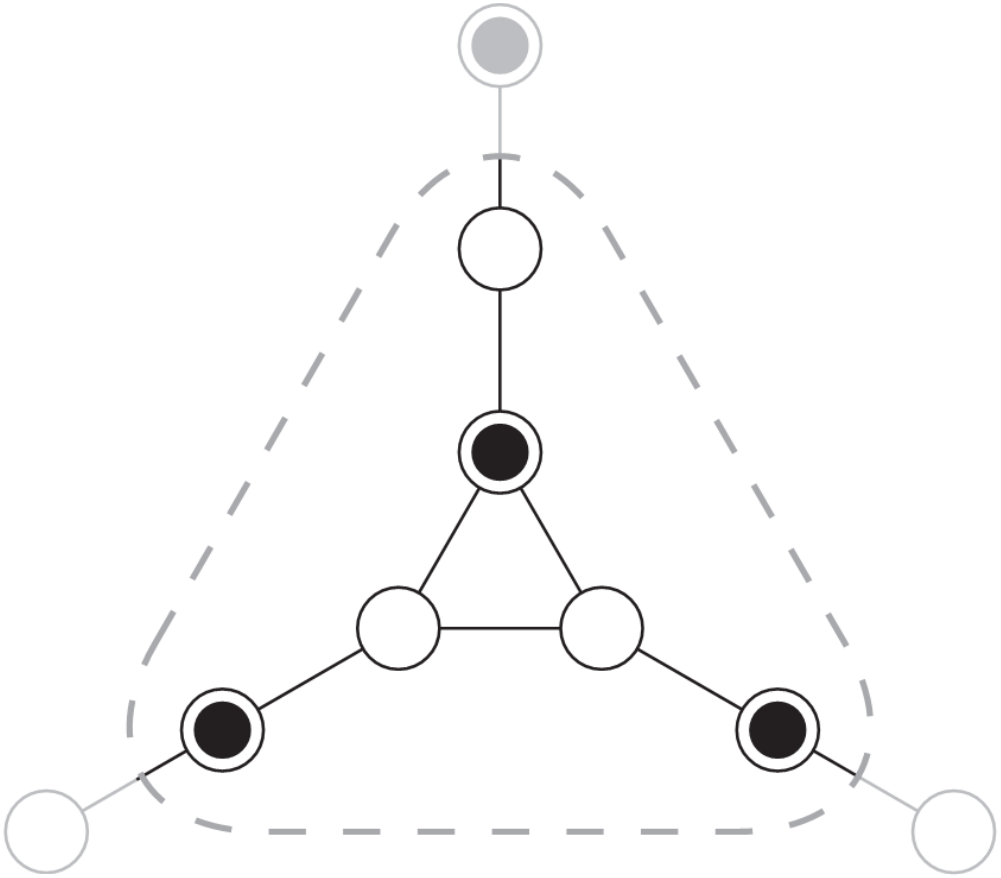


(b) Protected OR

Sliding Tokens (Reconfiguration Independent Set)



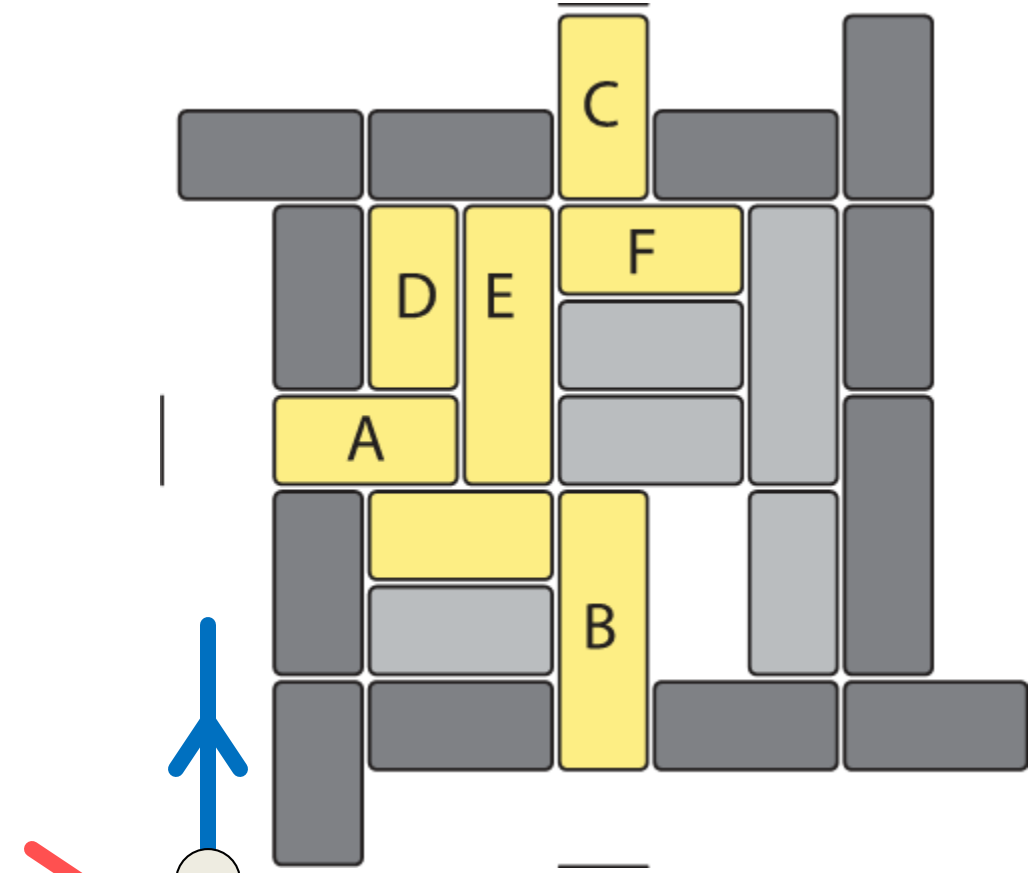
(a) AND



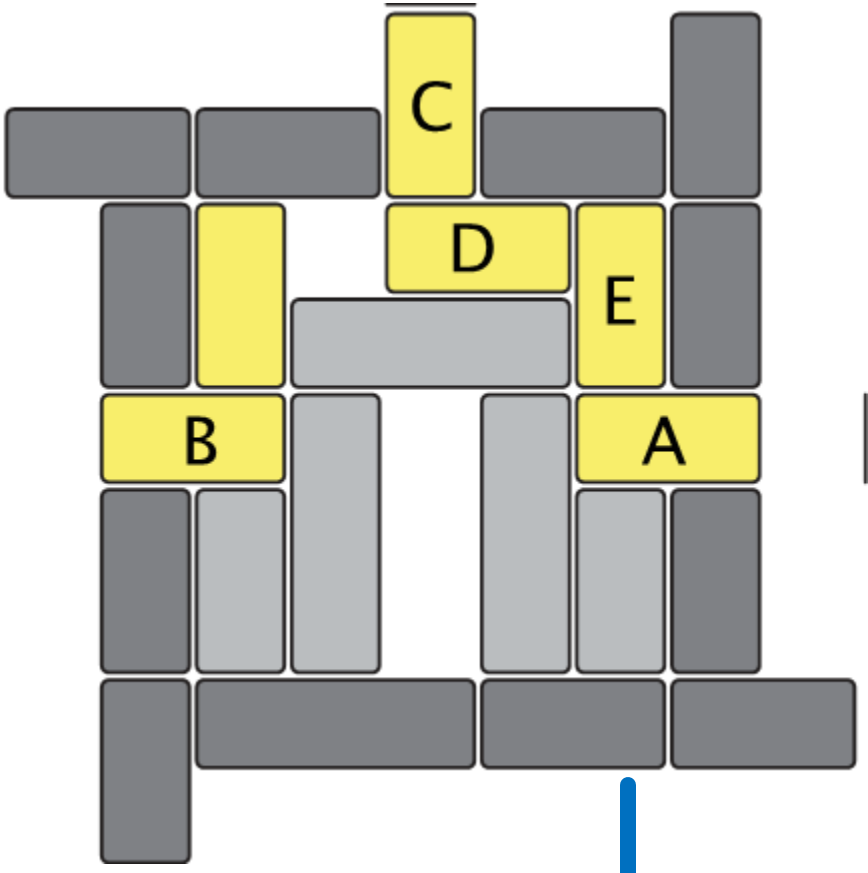
(b) OR

Rush Hour is PSPACE-complete

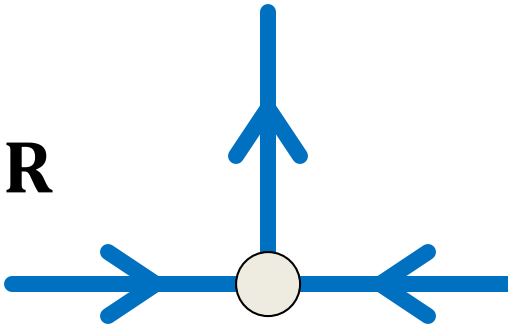
[Flake & Baum 2002; Hearn & Demaine 2002]



AND

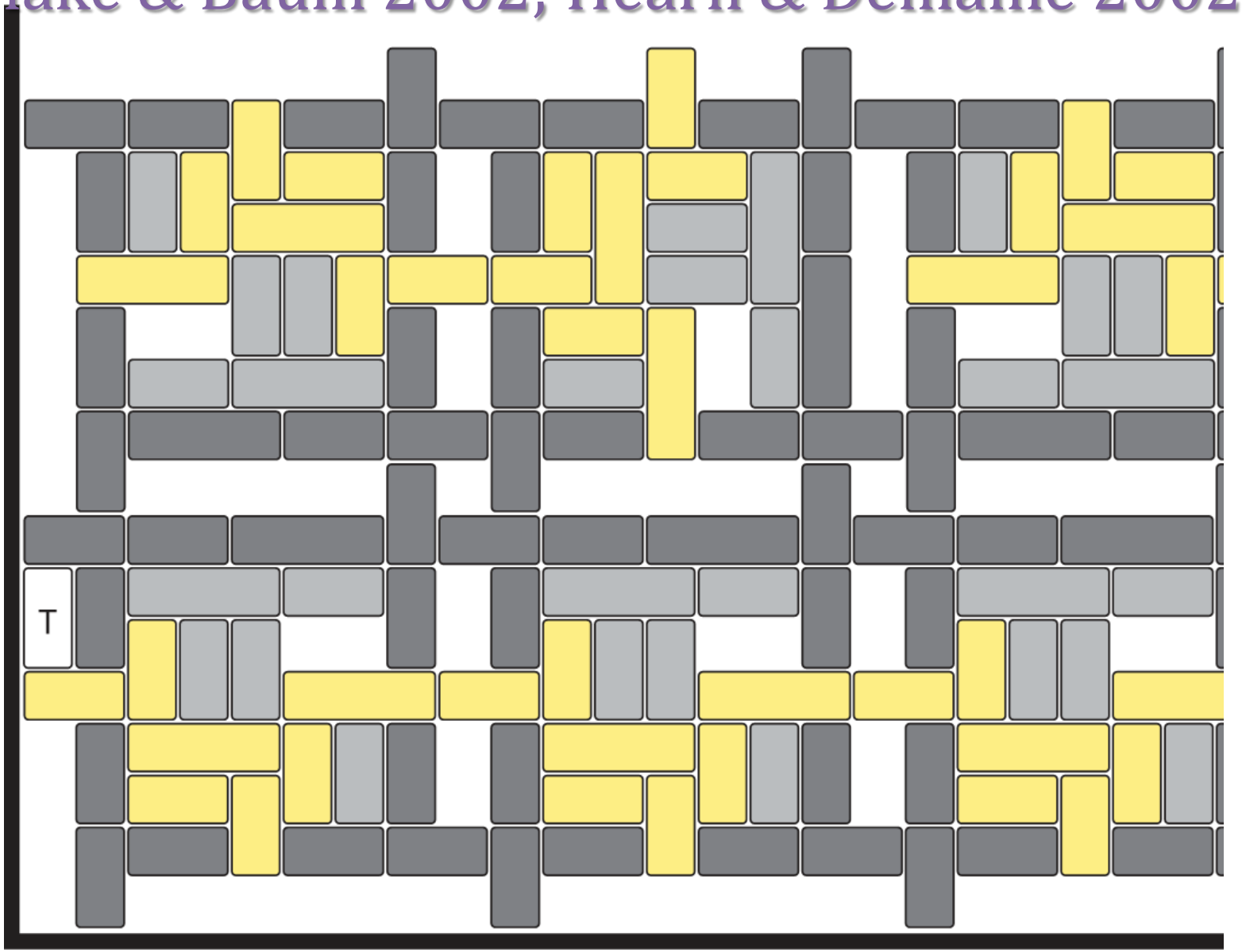


Protected **OR**



Rush Hour is PSPACE-complete

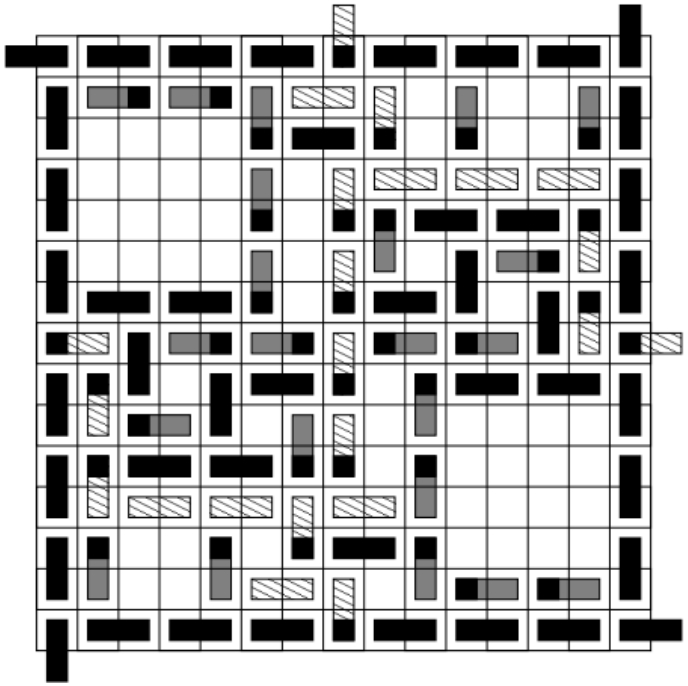
[Flake & Baum 2002; Hearn & Demaine 2002]

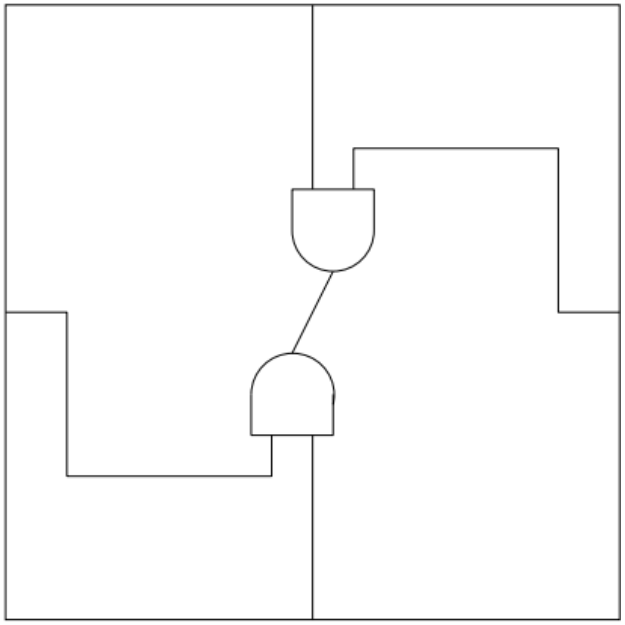


Rush Hour

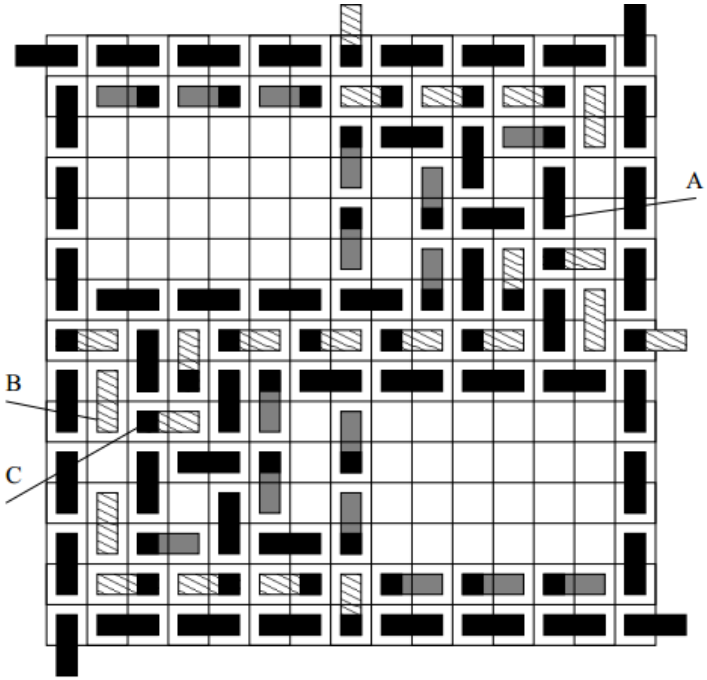
[Tromp & Cilibrasi 2008]

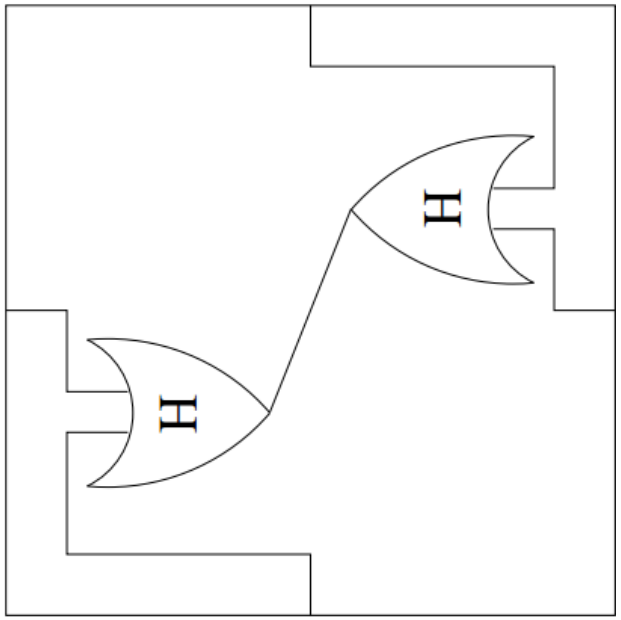
2 ANDs





2 protected ORs

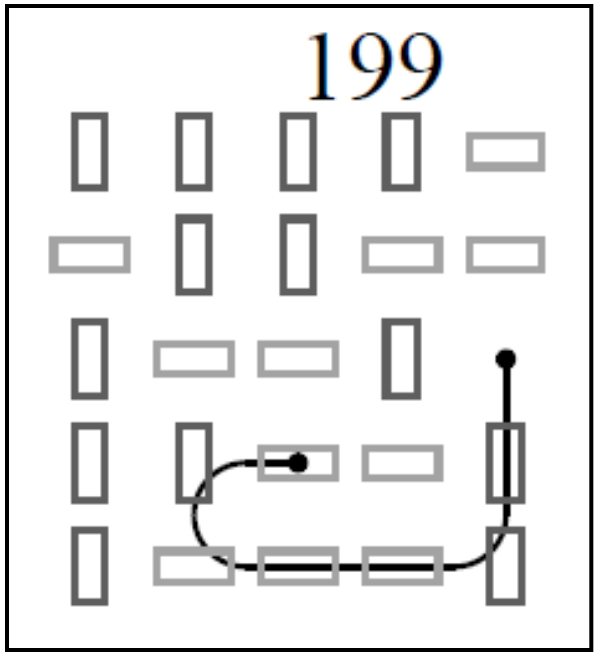


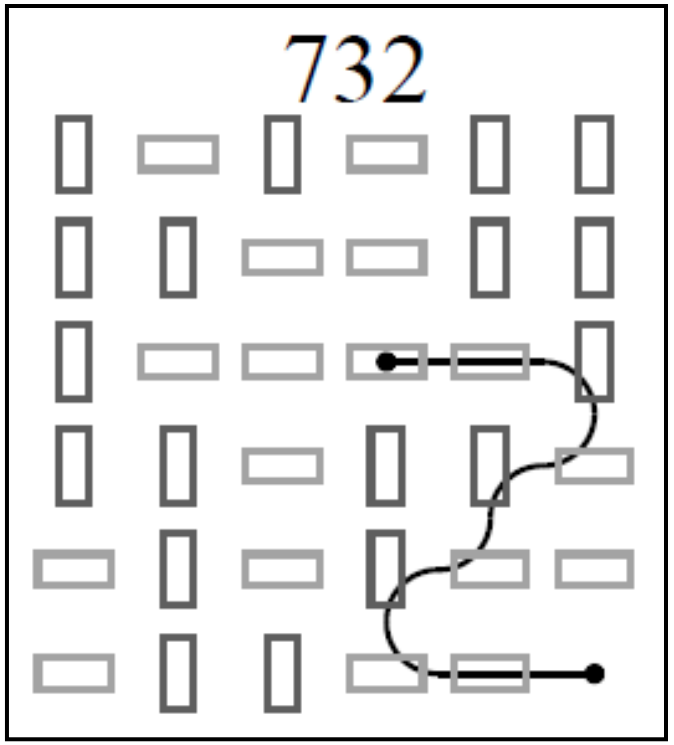


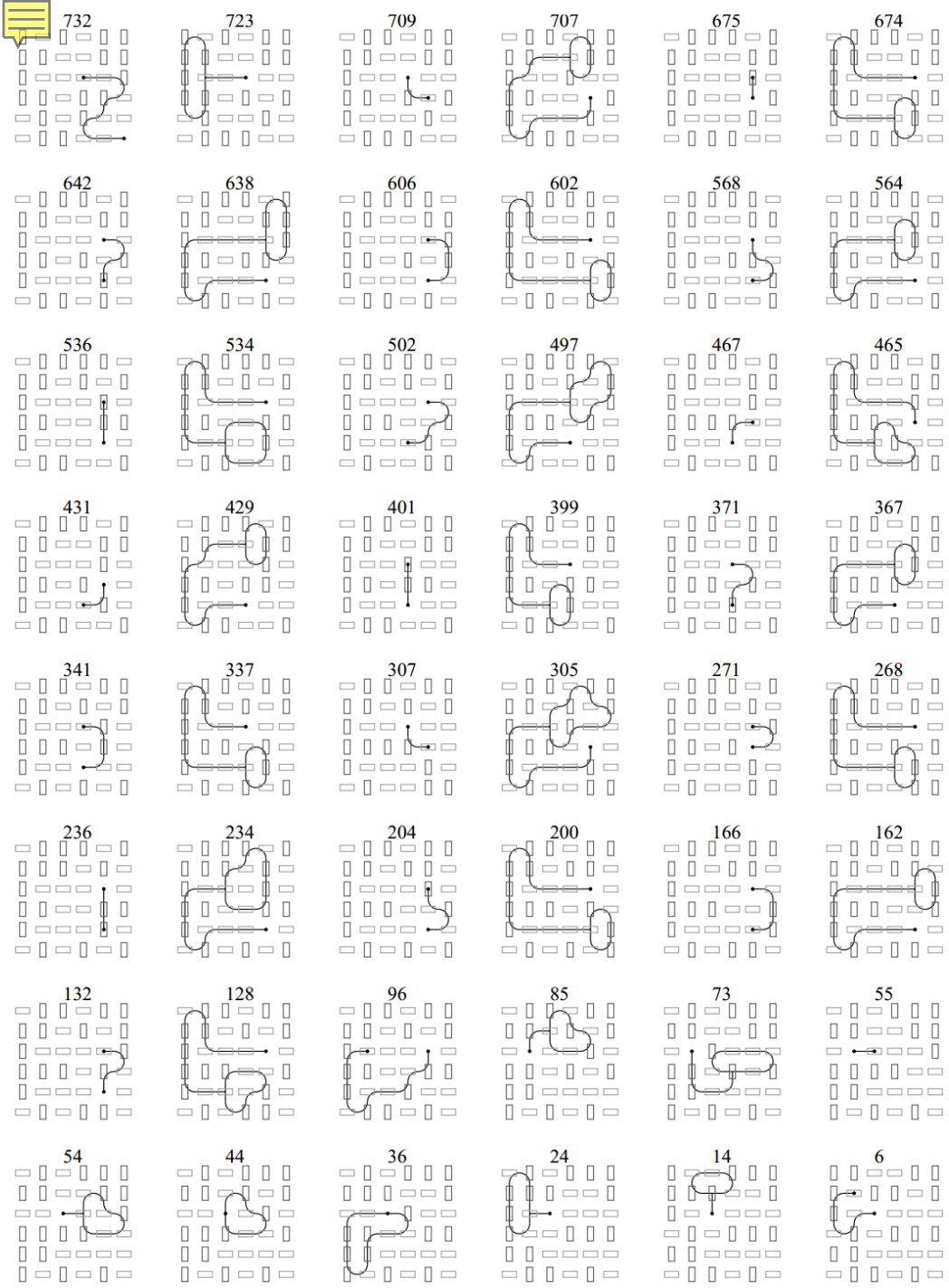
Open: 1 × 1 Rush Hour

[Tromp & Cilibrasi 2008]

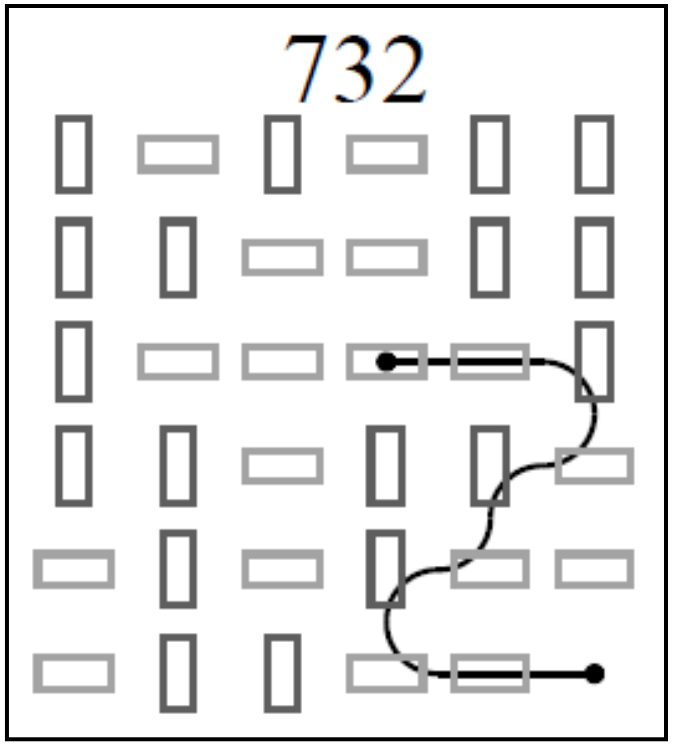
- P or PSPACE-complete or …?





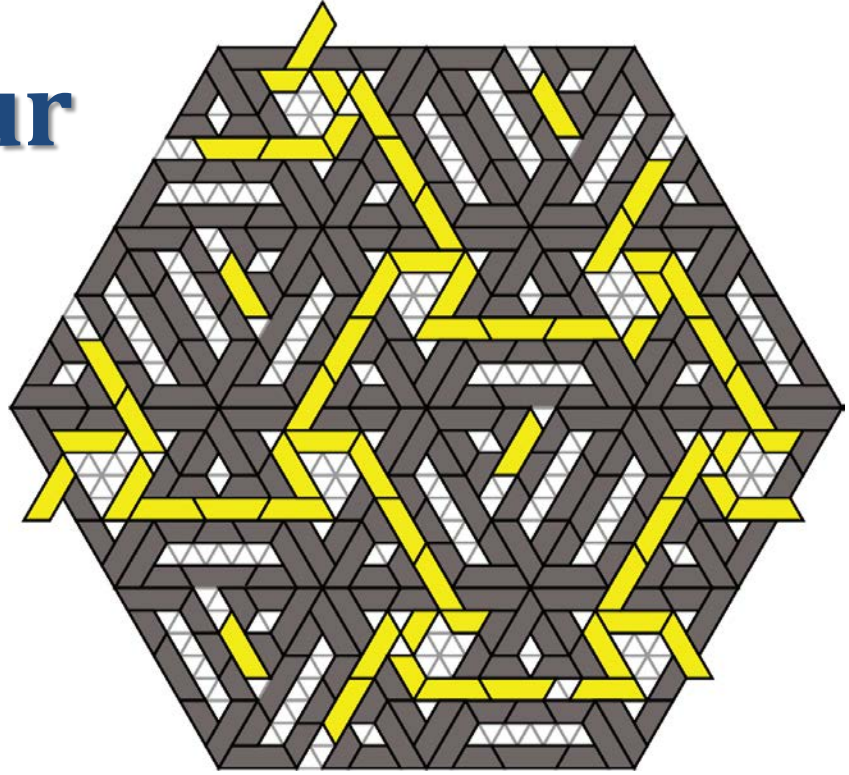


[Tromp & Cilibrasi 2008]

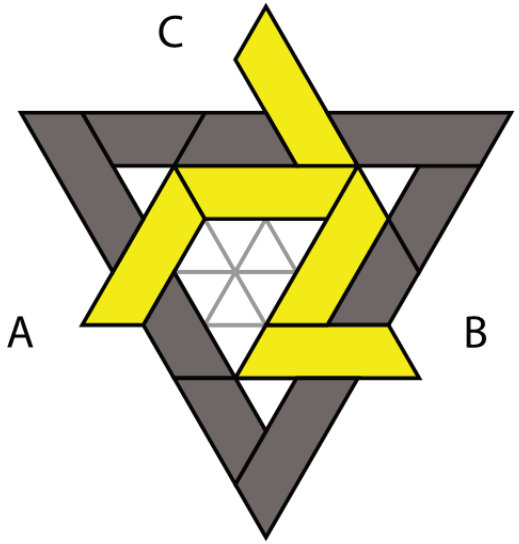


Triangular Rush Hour

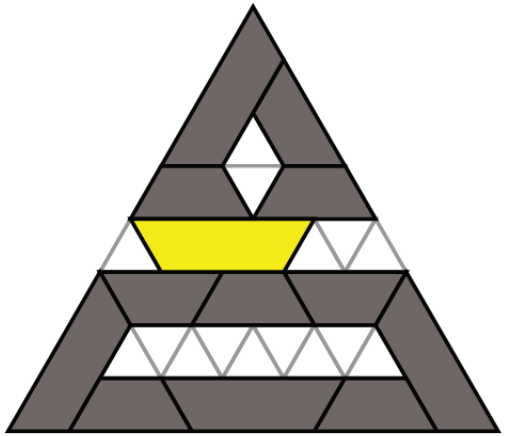
[Hearn & Demaine 2009]



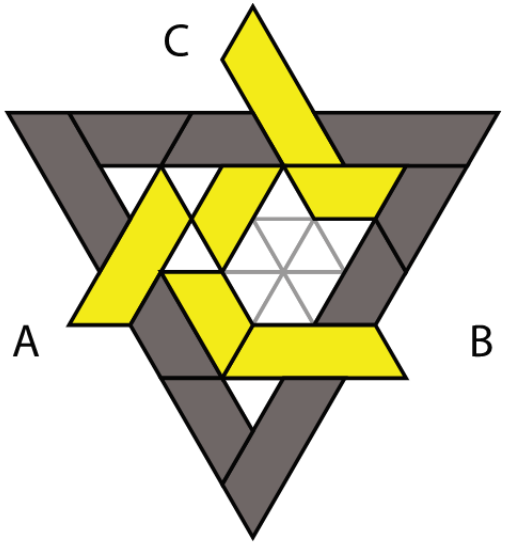
PSPACE-complete



(a) AND vertex

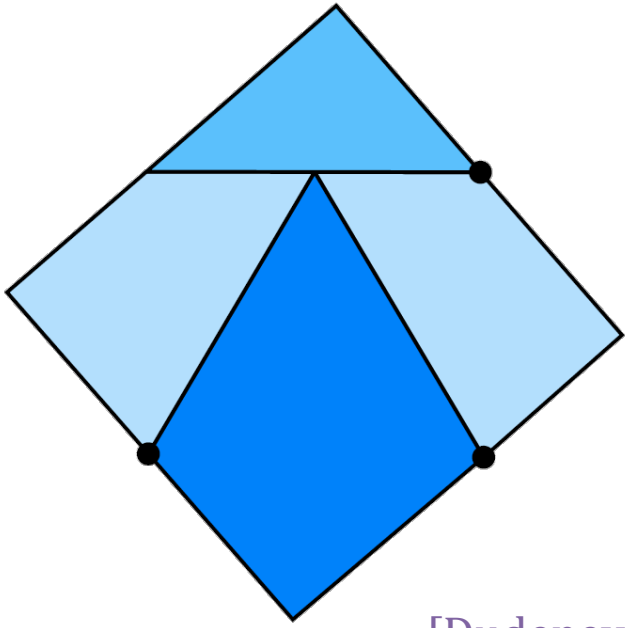


(b) Connector

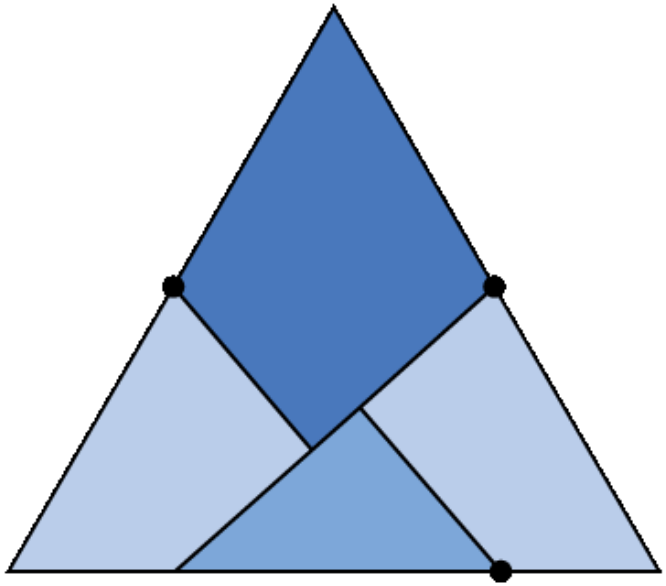


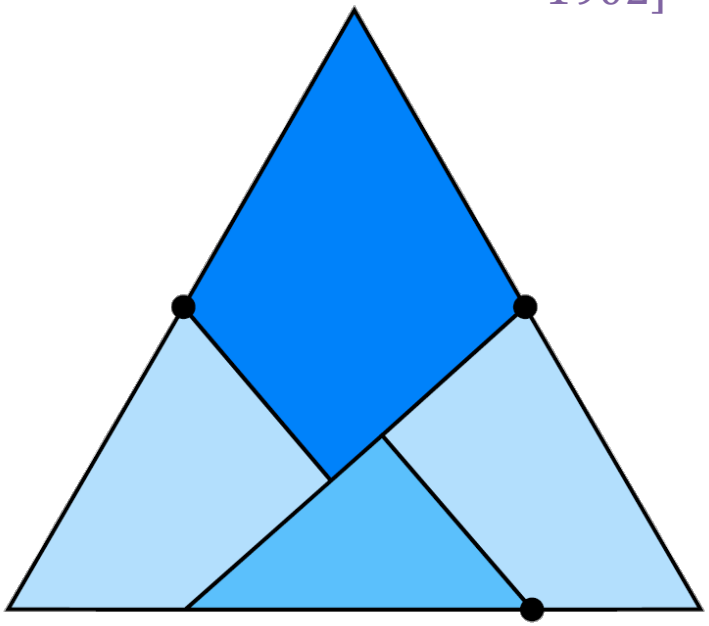
(c) OR vertex

Hinged Dissection



[Dudeney 1902]

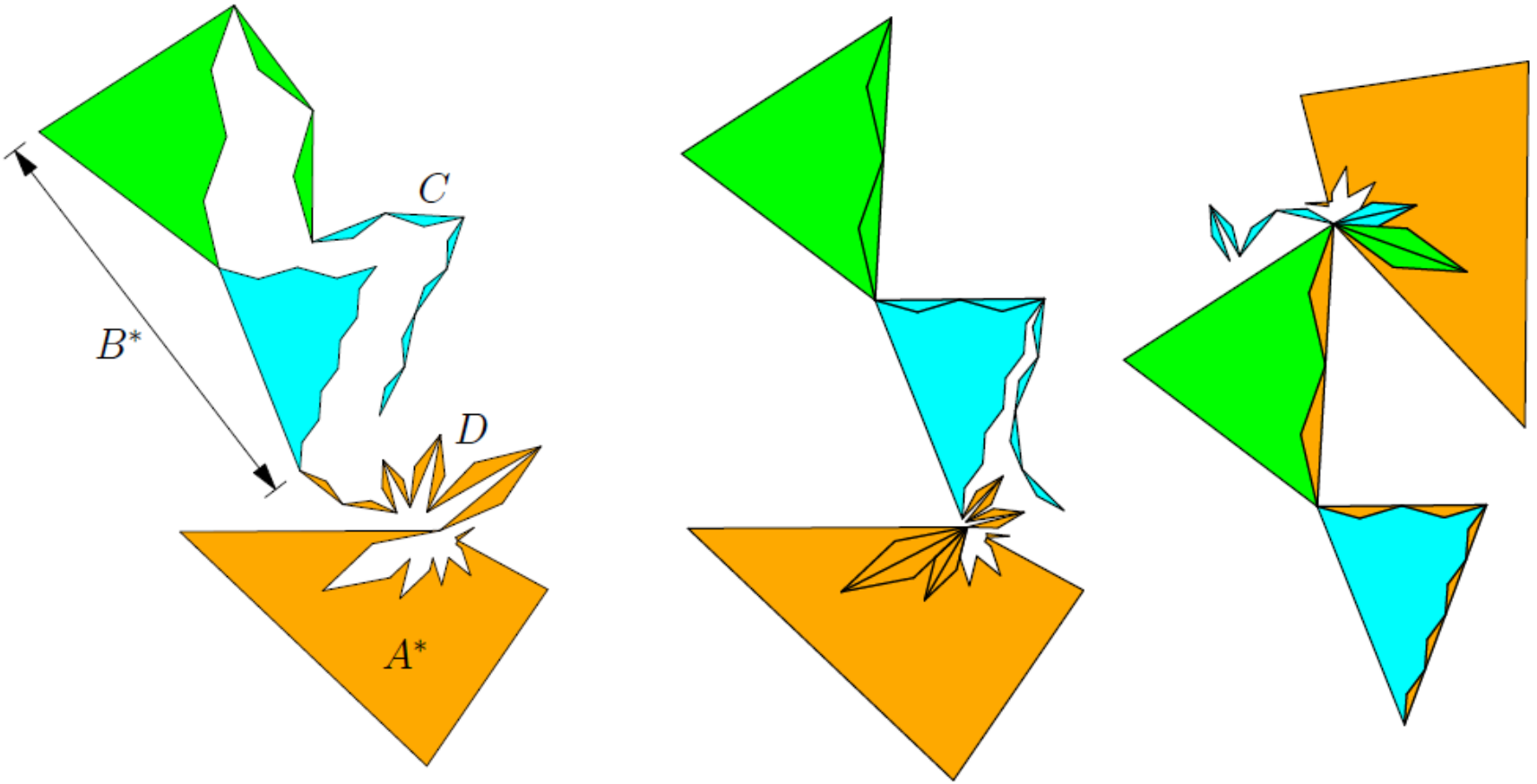




Hinged Dissection Universality

[Abbott, Abel, Charlton, Demaine, Demaine, Kominers 2008]

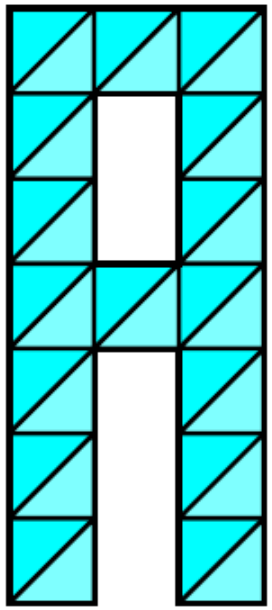
Polygons of equal area have a hinged dissection that folds *continuously without self-intersection*

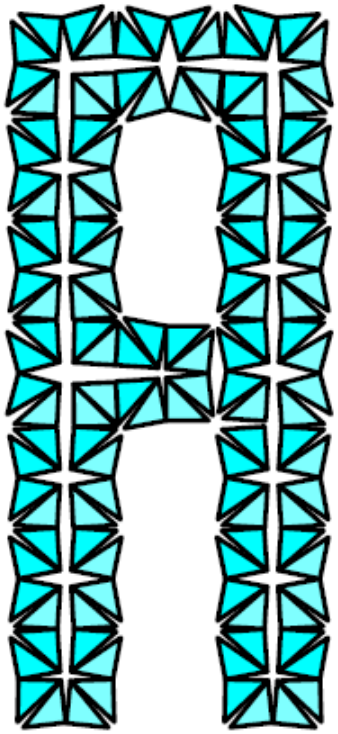


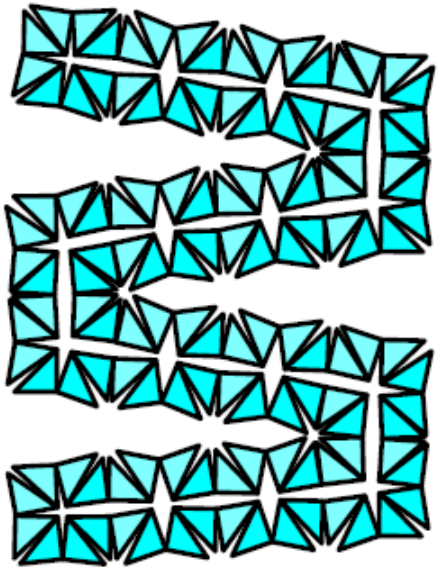
“Hinged alphabet”

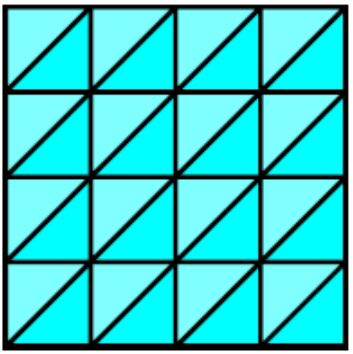
Erik & Martin Demaine

2003



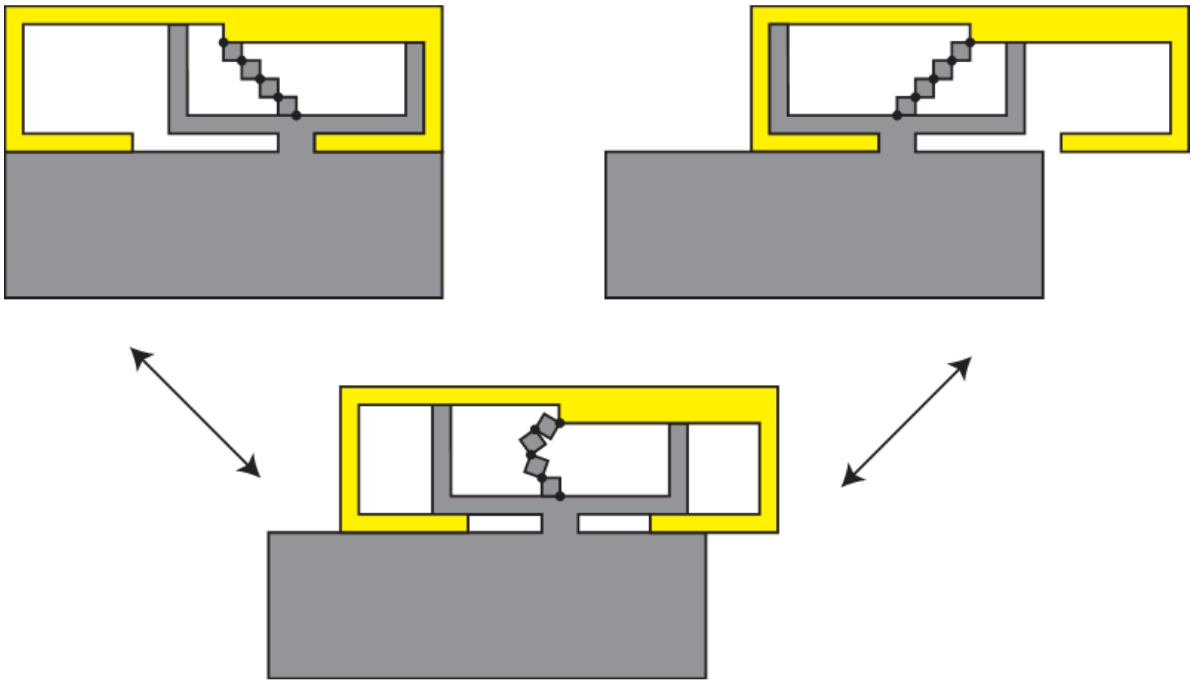




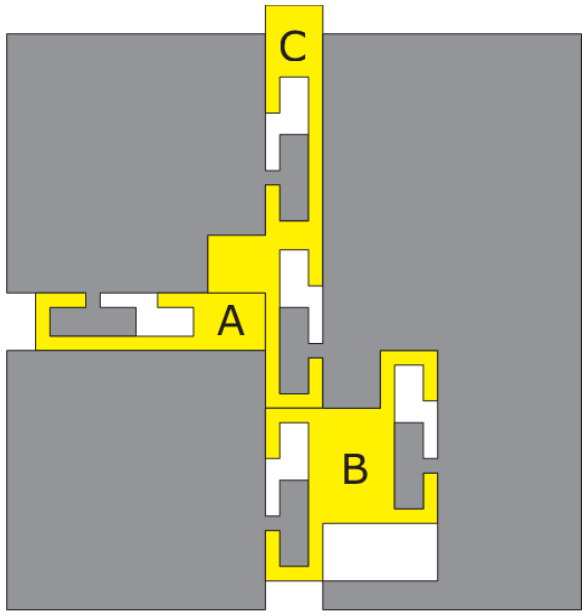


Hinged Dissection Motion

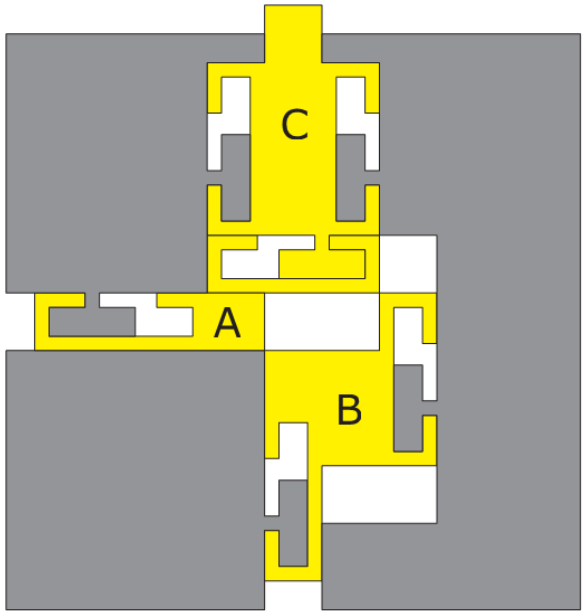
[Hearn & Demaine 2009]



(c) Hinged slider



(a) AND



(b) OR

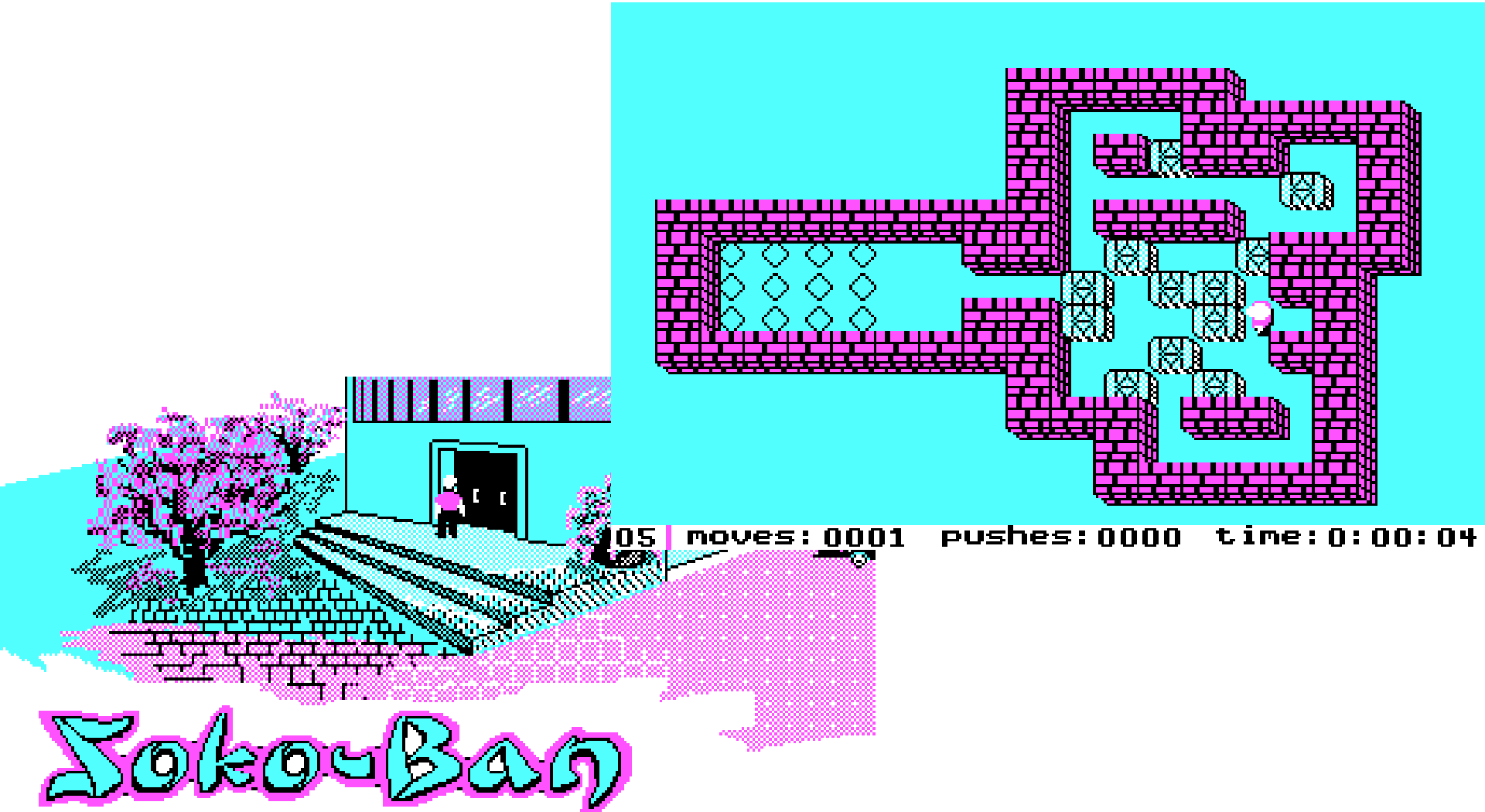
PSPACE-complete

Pushing 1×1 Blocks Complexity

Name	Push	Fixed	Slide	Goal	Complexity	Reference
Push-k	$k \geq 1$	no	min	path	NP-hard	D, D, O'Rourke 2000
Push-$*$	∞	no	min	path	NP-hard	Hoffmann 2000
PushPush-k	$k \geq 1$	no	max	path	PSPACE-complete	D, Hoffmann, Holzer 2004
PushPush-$*$	∞	no	max	path	NP-hard	Hoffmann 2000
Push-1F	1	yes	min	path	NP-hard	DDO 2000
Push-kF	$k \geq 2$	yes	min	path	PSPACE-complete	D, Hearn, Hoffmann 2002
Push-$*$F	∞	yes	min	path	PSPACE-complete	Bremner, O'Rourke, Shermer 1994
Push-kX	$k \geq 1$	no	min	simple path	NP-complete	D, Hoffmann 2001
Push-$*$X	∞	no	min	simple path	NP-complete	Hoffmann 2000
Sokoban	1	yes	min	storage	PSPACE-complete	Culberson 1998

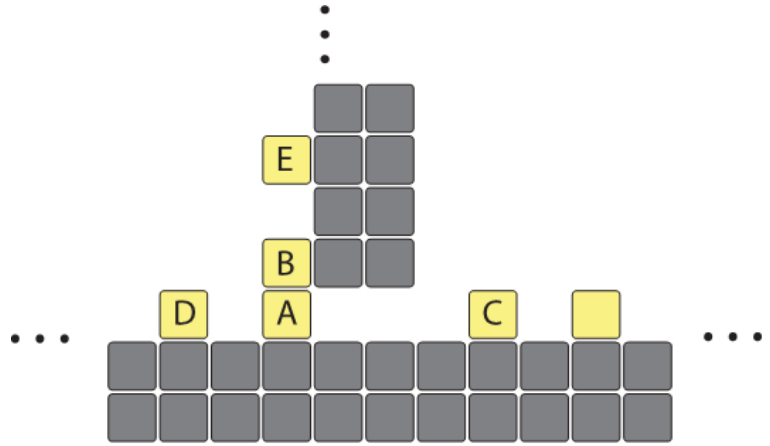
Sokoban

[Thinking Rabit, Hiroyuki Imabayashi, 1982]

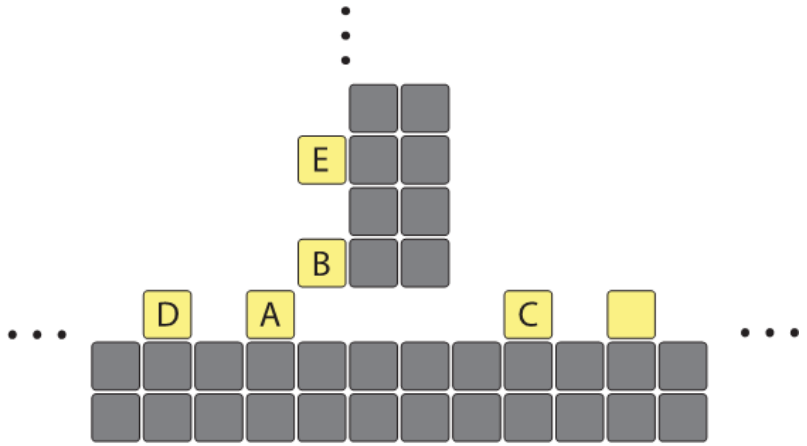


Sokoban is PSPACE-complete

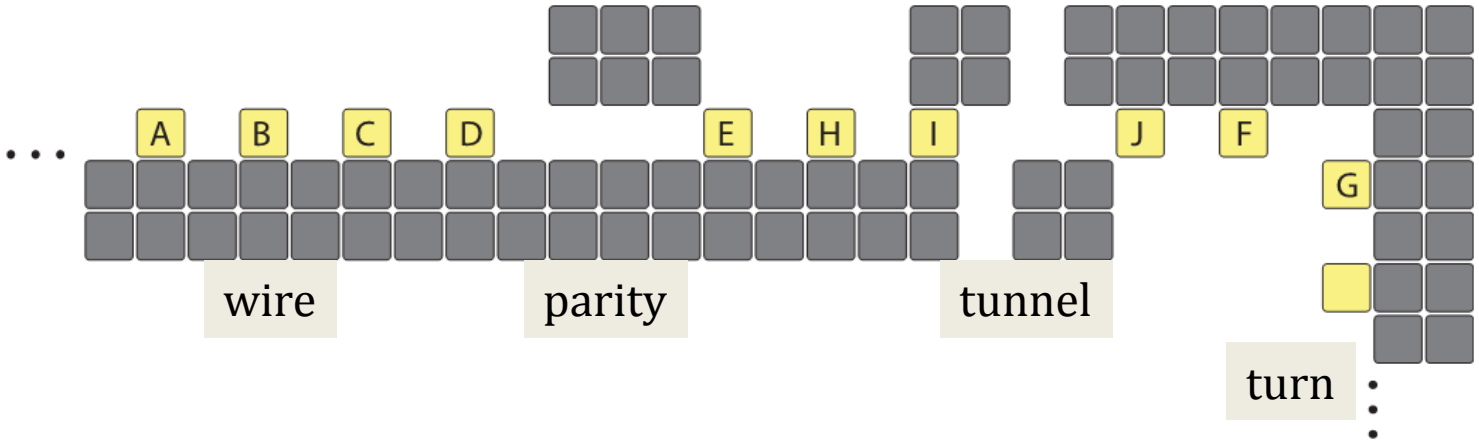
[Culberson 1998; Hearn & Demaine 2002]



(a) AND



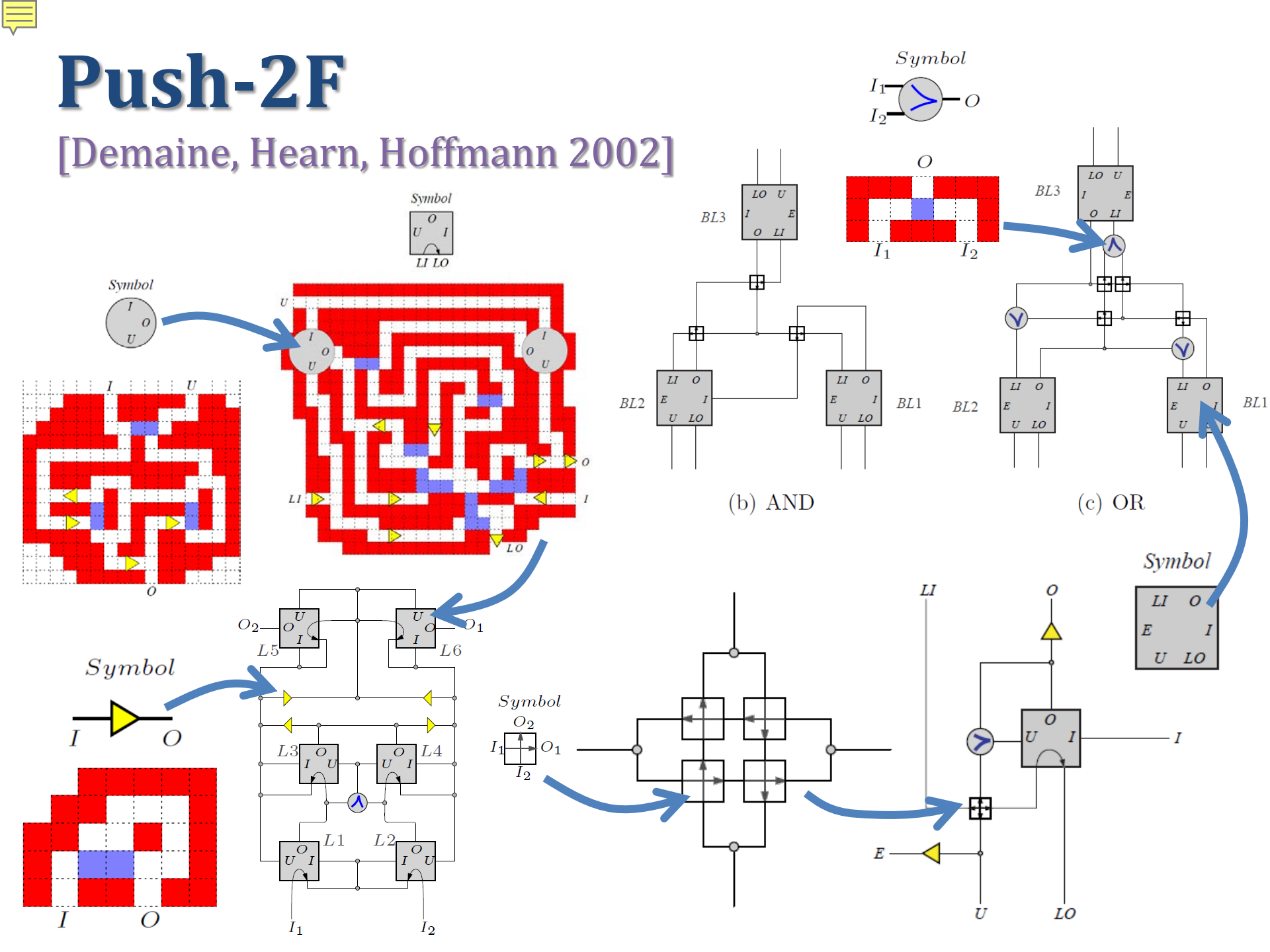
(b) OR



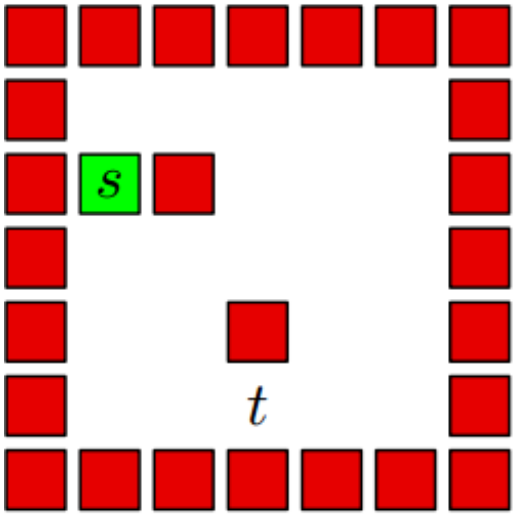
(c) Utility gadgets

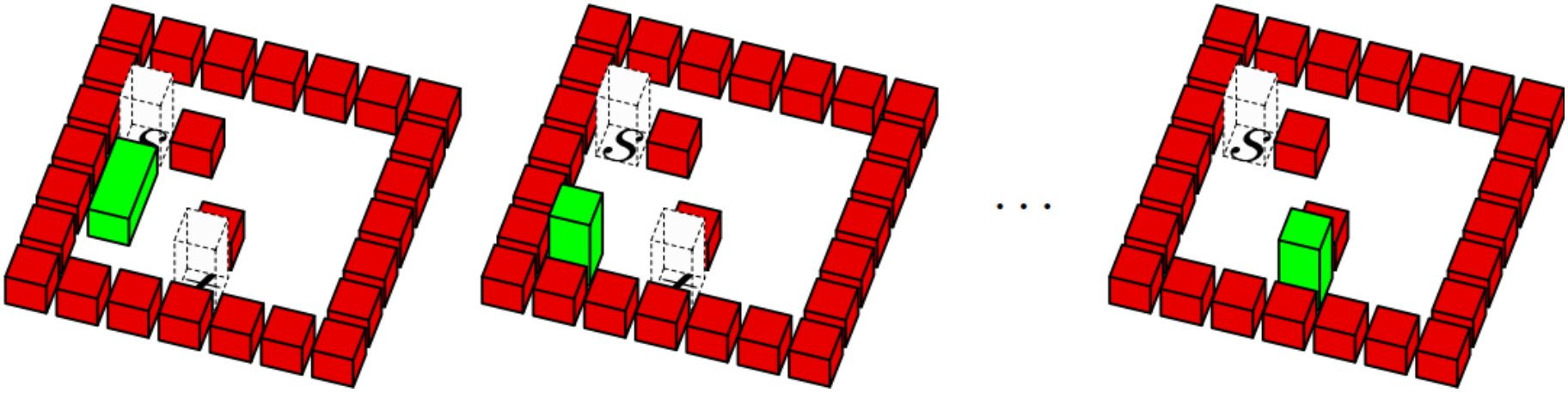
Push-2F

[Demaine, Hearn, Hoffmann 2002]



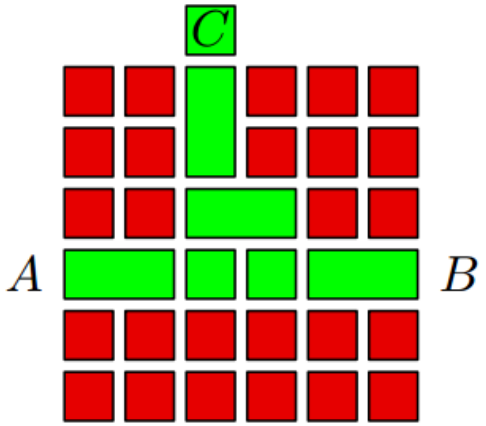
Rolling Block Mazes

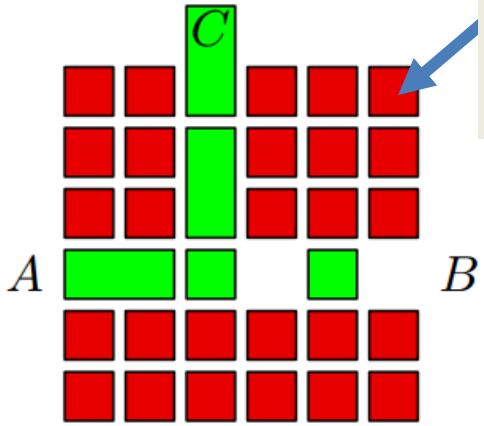




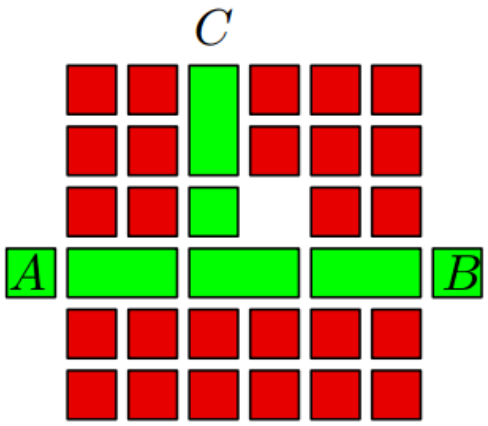
Rolling Block Mazes PSPACE-complete

[Holzer & Jakobi 2012]

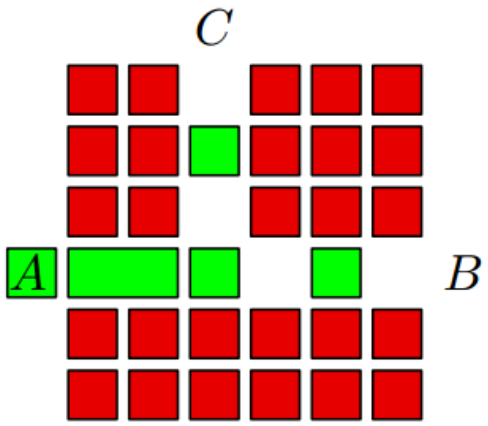


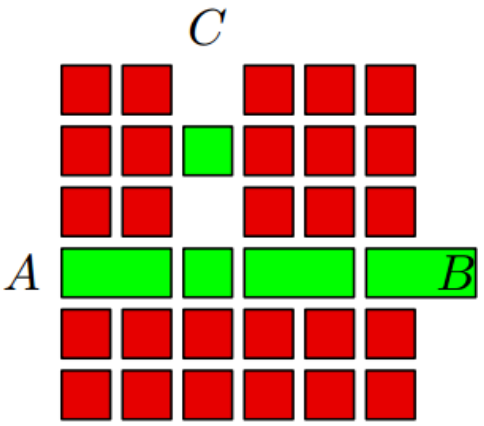


effectively immovable $1 \times 1 \times 2$ blocks



AND

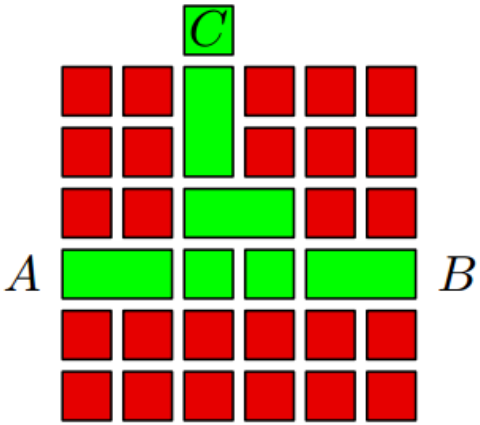




OR

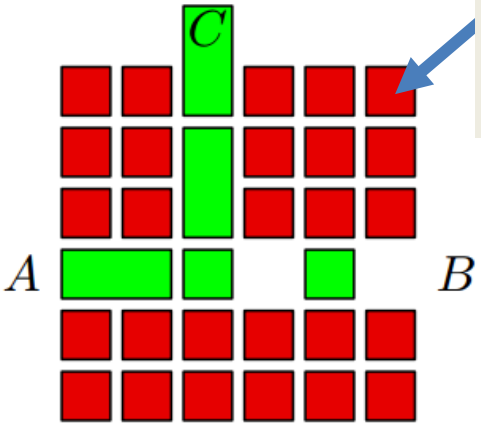
Rolling Block Mazes PSPACE-complete

[Holzer & Jakobi 2012]

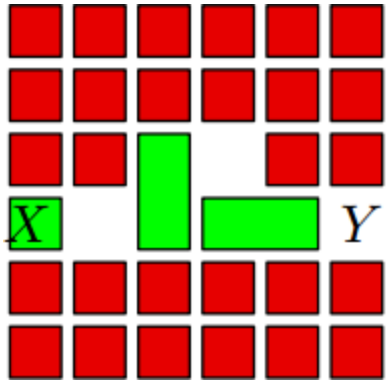


AND

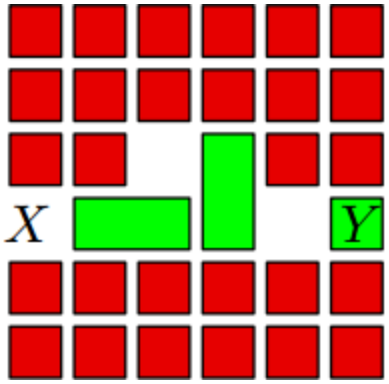
OR



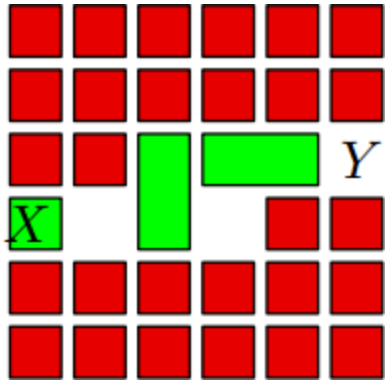
effectively immovable $1 \times 1 \times 2$ blocks



rightward edge



leftward edge



shift

Plank Puzzles

[Andrea Gilbert; ThinkFun]

HOW TO PLAY:

Hiker STARTS TO CROSS.

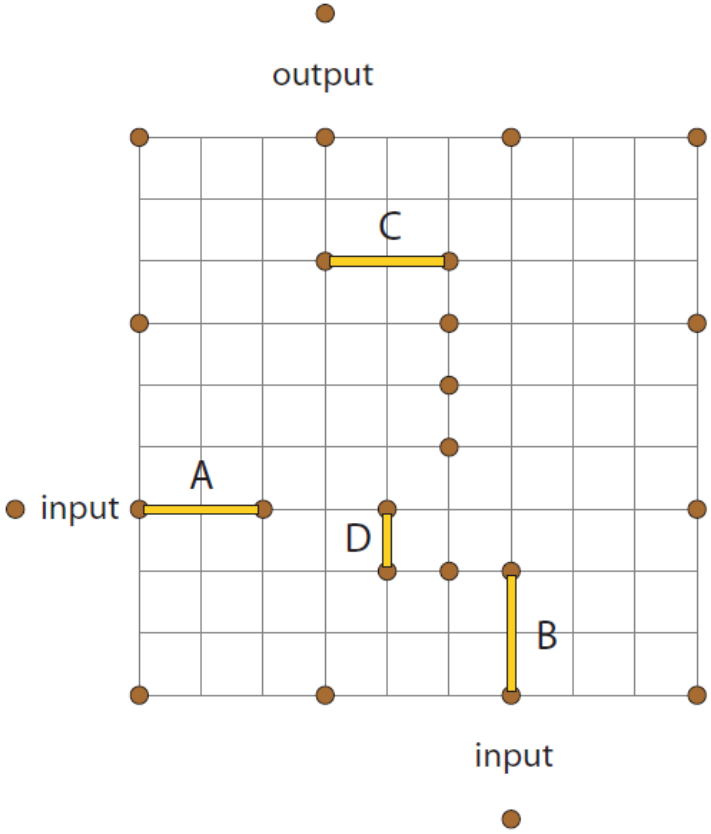
Hiker MOVES the medium plank TO A NEW POSITION.

Hiker MOVES the small plank TO A NEW POSITION.

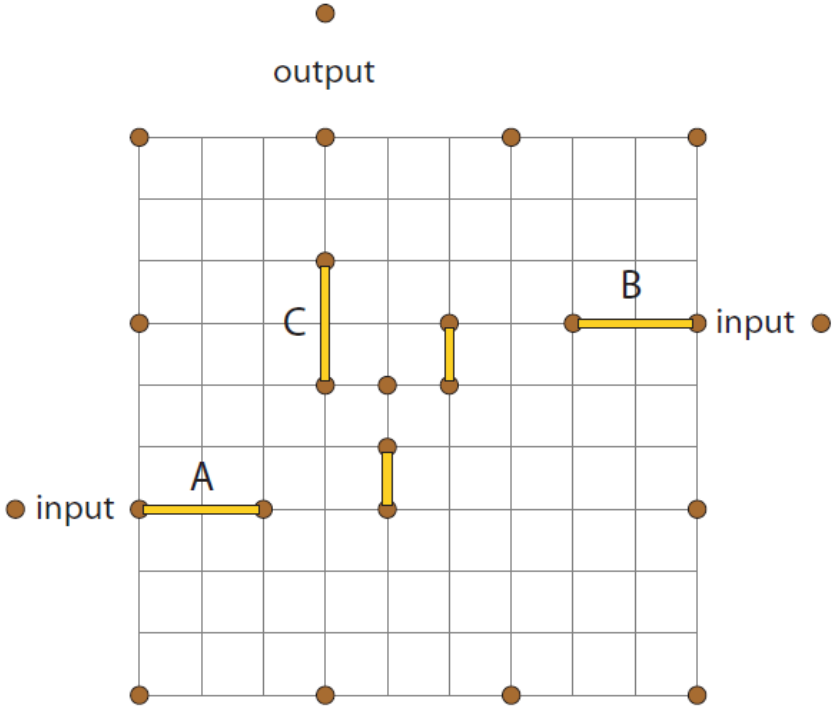
Hiker MOVES the medium plank and makes it across!

Plank Puzzles

[Hearn 2004]



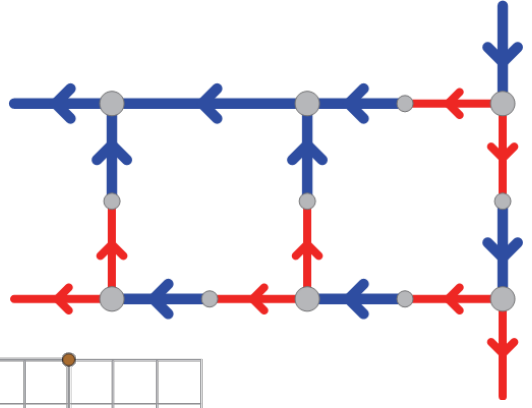
(a) AND



(b) OR

Plank Puzzles

[Hearn 2004]



End

Start

A

D

E

C

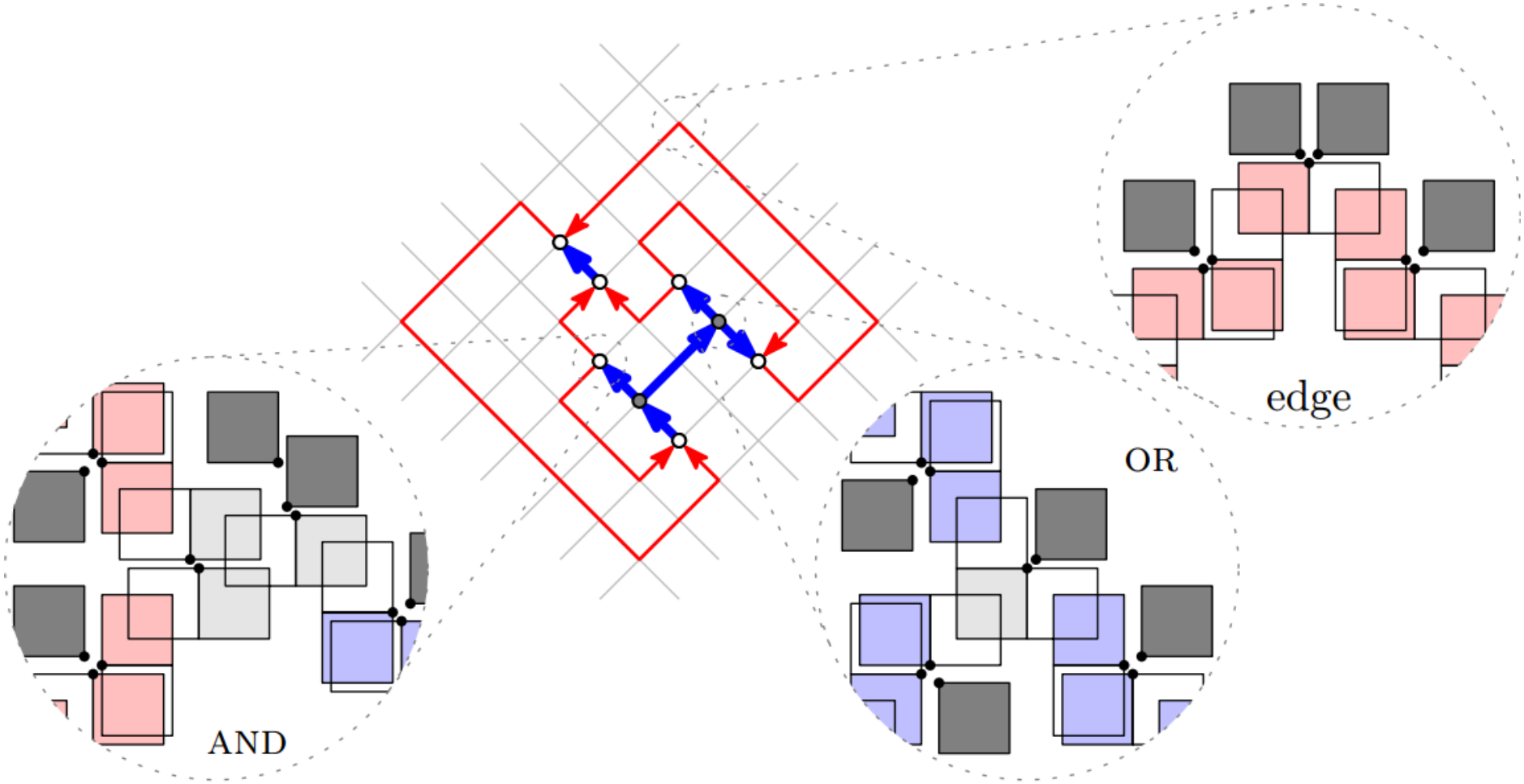
2

1

B

Dynamic Map Labeling

[Buchin & Gerrits 2013]



Partial Searchlight Scheduling

[Viglietta 2013]