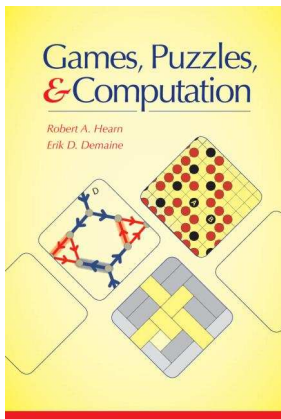


Game Complexity

Gadgets



Walter Kusters, Universiteit Leiden
www.liacs.leidenuniv.nl/~kusterswa/

IPA, Eindhoven; Friday, July 8, 2016

VN Detective en Thrillergids
17274 titels www.vnster.nl

thuis hulp pdf/csv uitgebreid zoeken op: titel auteur bladeren
mobiel contact

U bevindt zich hier: thuis — www.vnster.nl



Eerste Detectieve & Thrillergids

Sinds 1980 verblijft Vrij Nederland (VN) ons iedere zomer met een Detective en/Thrillergids. Deze bestaat onder meer uit een uitgebreide lijst met in het Nederlands verkrijgbare titels. Op 25 mei 2016 verscheen de 37ste gids [5*,H-P, z]. Voor de recensies leze men de papieren of digitale versie bij VN.

Er zijn in het Nederlands nog meer "spannende" boeken verschenen. Het streven is zoveel mogelijk titels in de lijst(en) op deze website op te nemen. De informatie is te vinden via verschillende zoekmogelijkheden, ook mobiel of via een app, en er kan gewoon gebladerd worden. Voor de duidelijkheid: er zijn hier geen boeken te koop!

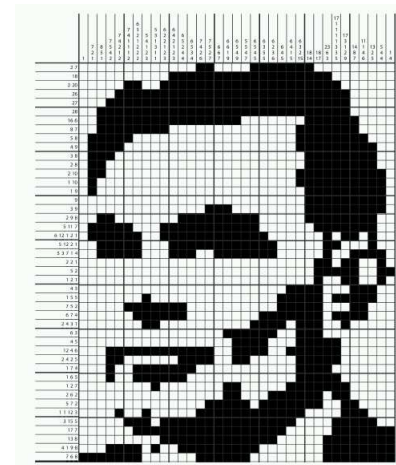
De navigatie is steeds in de kleur van het "submenu": [lichtblauw](#)—algemeen, [oranje](#)—zoeken, respectievelijk [donkerrood](#)—bladeren. Een en ander is tevens in PDF-formaat beschikbaar: 464 pagina's, 13 Mb, en ook in CSV-formaat. Voor vragen, opmerkingen of aanvullingen: neem gerust [contact](#) op. Of werk aan een eigen [collectie](#). Voor meer achtergronden raadplege men de [hulp-pagina's](#).



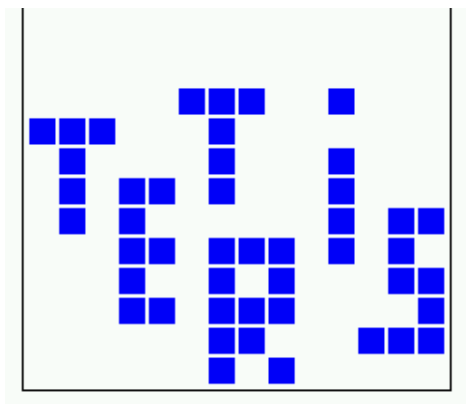
Vrij Nederland Detective en Thrillergids

mobiel contact

[link](#)

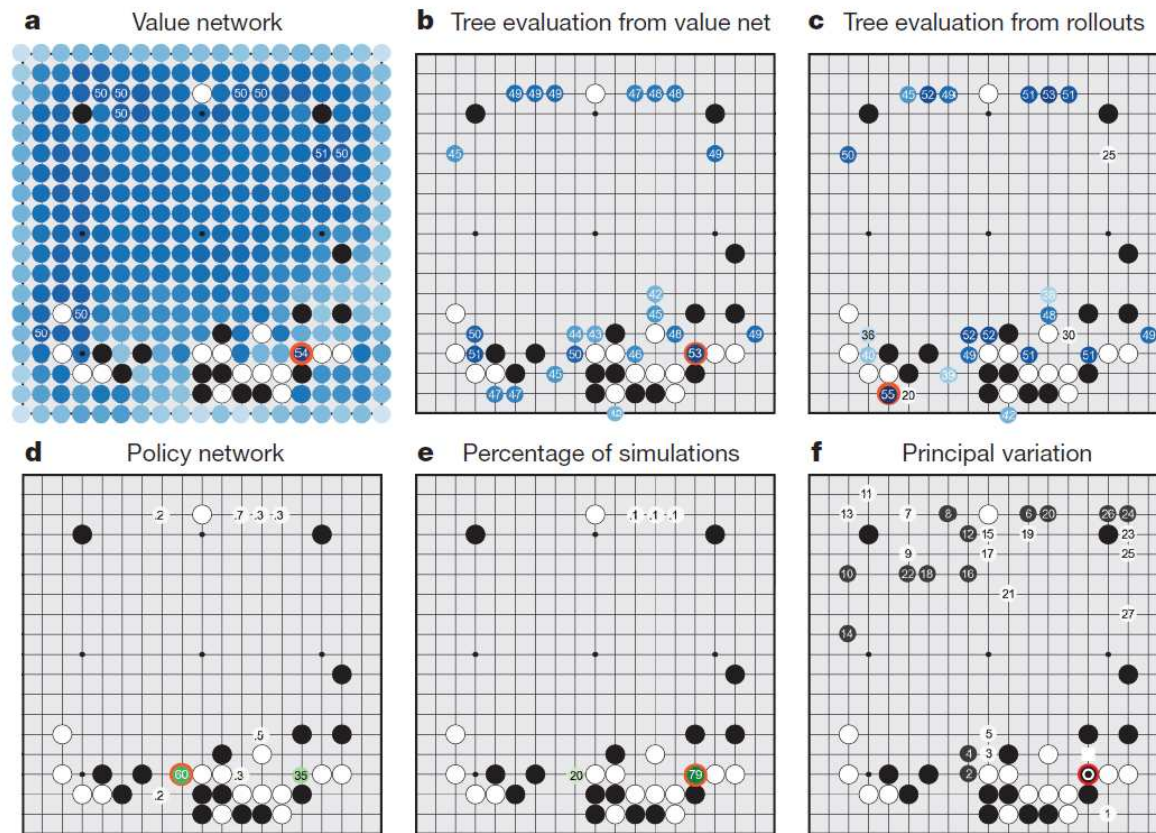


[link](#)



[link](#)

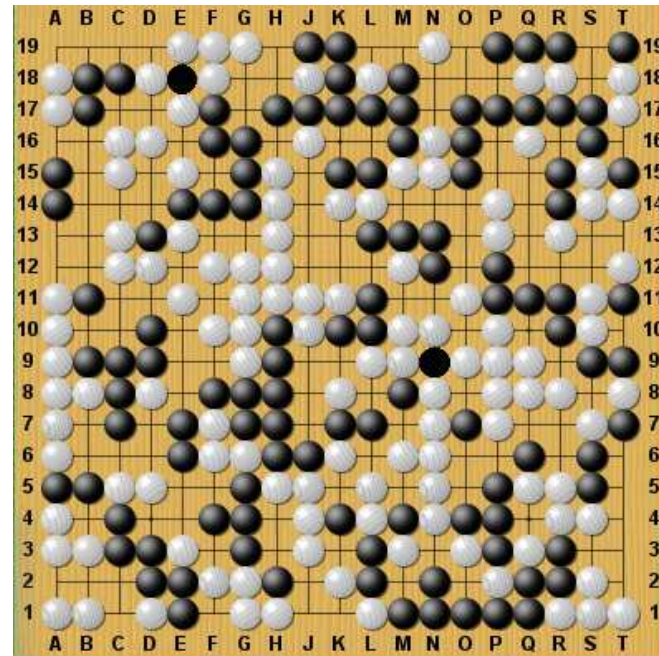
crime novels,
tomography
and Tetris



3.2016: AlphaGo(-ogle) defeats top Go player Lee Sedol

In 2016 John Tromp showed at CG2016 that there are

2081681993819799846
9947863334486277028
6522453884530548425
6394568209274196127
3801537852564845169
8519643907259916015
6281285460898883144
2712971531931755773
6620397247064840935



$\approx 2 \cdot 10^{170}$ legal positions in 19×19 Go, using DP & CPUs.

<https://tromp.github.io/go/legal.html>

Games

Jeopardy!

1000 POP. CULTURE	1000 OF COMMON	1000 YOUR PICKERS	1000 YOUR SCOPY	1000 FLY LINE AIRFARE	1000 NATIONAL PASTRIES
\$100	\$100	\$100	\$100	\$100	\$100
\$200	\$200	\$200	\$200	\$200	\$200
\$300	\$300	\$300	\$300	\$300	\$300
\$400	\$400	\$400	\$400	\$400	\$400
\$500	\$500	\$500	\$500	\$500	\$500

IN 2013 ROB FORD,
MAYOR OF THIS 4th-
LARGEST CITY IN N.
AMERICA, FIRST SAID
HE SMOKED WEED,
NOT CRACK...THEN
YES, OK, CRACK, TOO



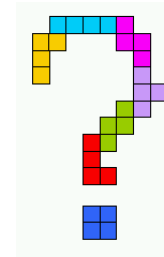
2011

What is
Toronto????



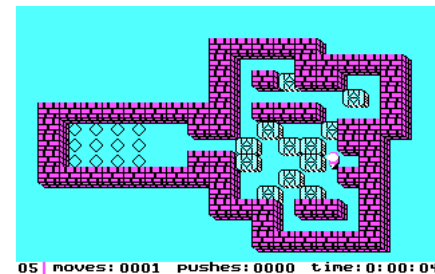
We study the complexity of games (puzzles, . . .). We want to make statements like

Tetris is NP-complete.



In order to do so, we examine **reductions** between appropriate games, with the help of **gadgets**.

Games studied include TipOver,
Plank puzzles, Sokoban→,
Rush Hour, Mahjonn, . . .

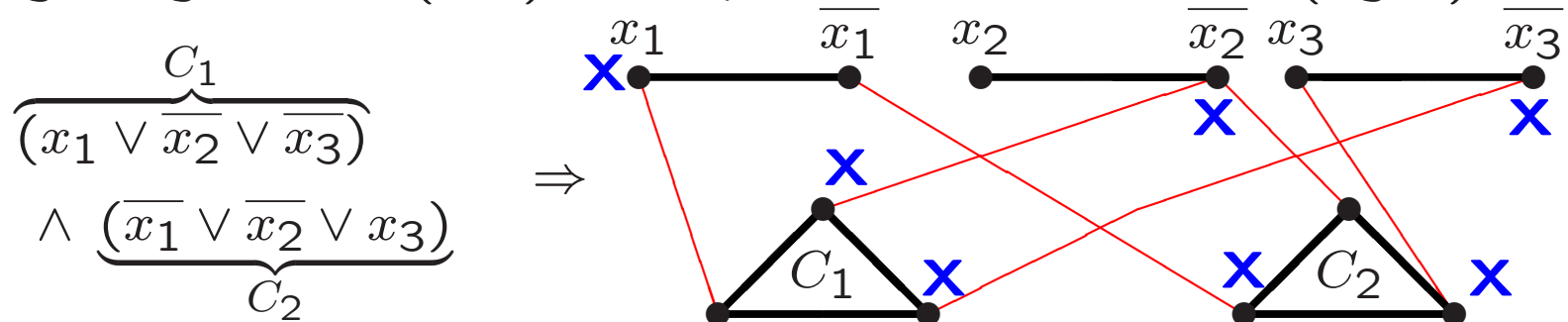


We want to **reduce** a known problem to a new one, for example, 3SAT to VC (so **V**ertex**C**over is NP-hard).

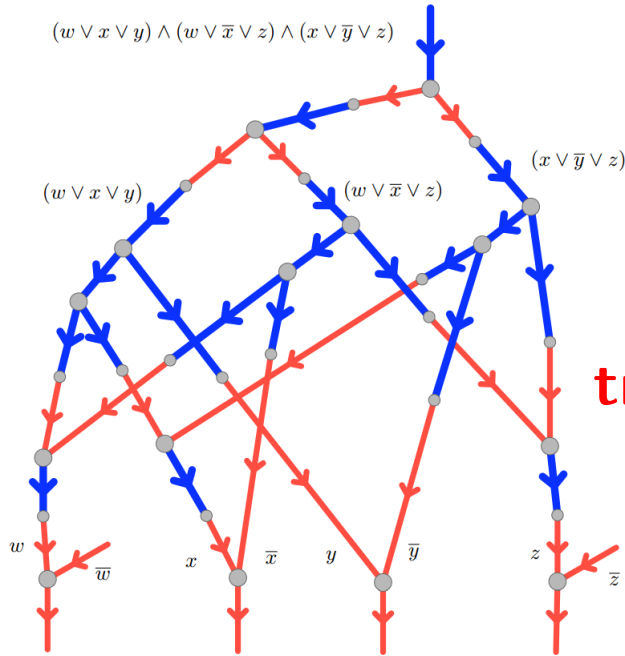
For every Boolean variable x_i we make a **variable gadget** (left) and for every clause C_j a **clause gadget** (right):



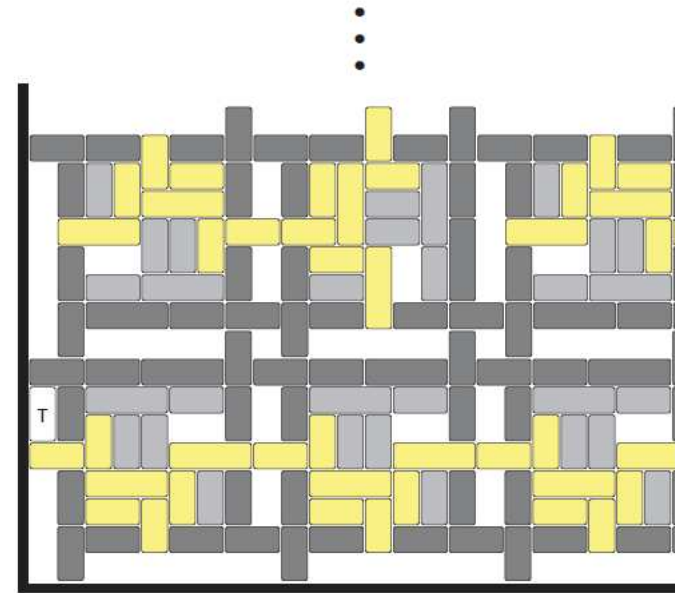
We connect these gadgets in the intuitive way; now satisfying assignments (left) correspond to vertex covers (right):



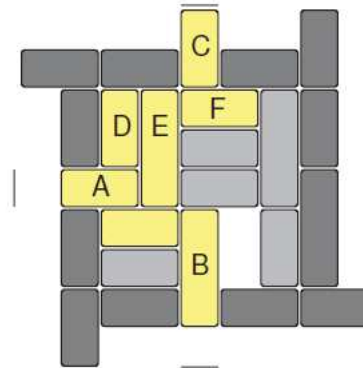
Satisfying assignment $x_1 = \text{true}$, $x_2 = x_3 = \text{false}$ gives a VC **X** of size $3 + 2 \cdot 2 = 7$, for 3 literals and 2 clauses.



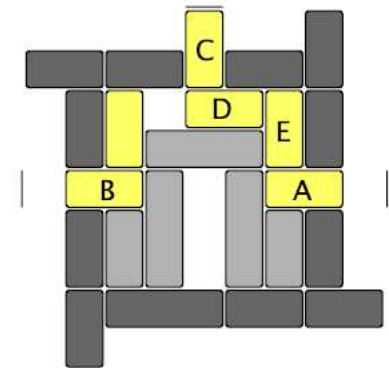
translate into



(a) Layout



(b) AND



(c) Protected OR

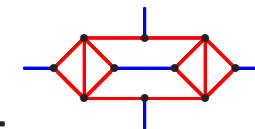
Suppose we want to show a game to be NP/PSPACE-hard (formally: some related (y/n)-decision problem Π).

For this purpose we produce a reduction from a known well-chosen *graph* game (formally: some related (y/n)-decision problem Π' , hopefully with planar graphs) to Π .



The less complicated Π' is, the better. If we are lucky, we only have to show how certain basic constructs are “emulated” by means of **gadgets**. Plus many details ...

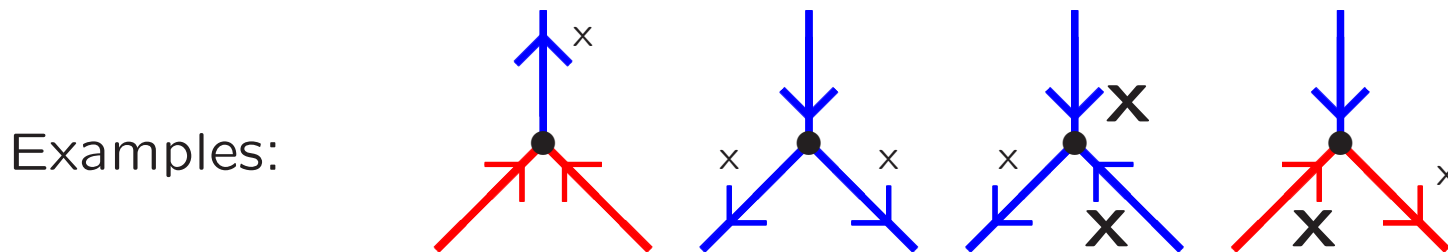
We also have **gadgets** to emulate certain (sub)graph behaviour in the graphs themselves.



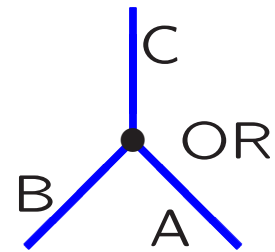
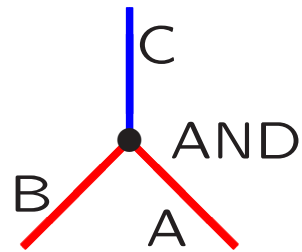
Constraint graphs consist of AND- and OR-nodes:



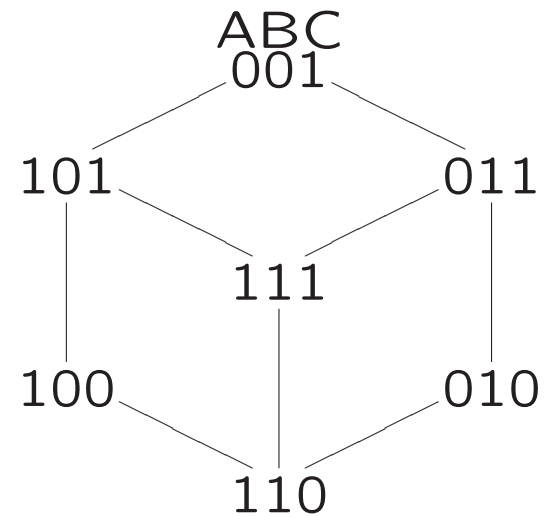
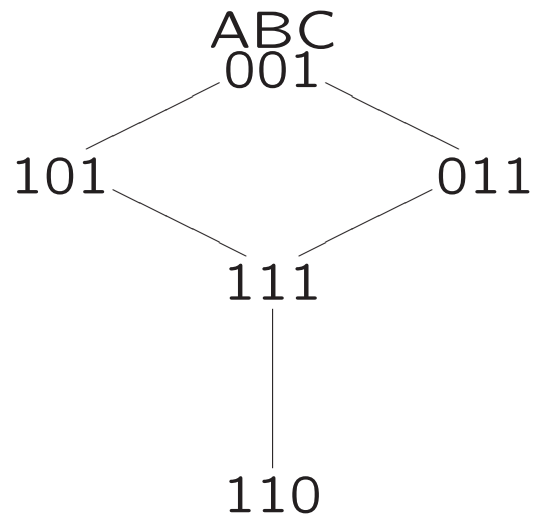
Edges are always directed such that every node = vertex receives a total input ≥ 2 , where incoming blue edges contribute 2 and incoming red edges 1.



An edge can be **reversed** if all total inputs remain ≥ 2 (**X**).

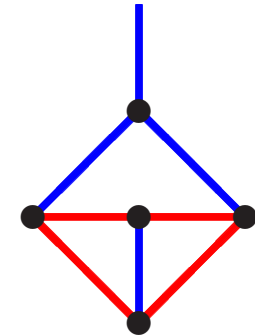


External behavior of these gadgets can be described by the statespaces below (where 1: points in; 0: points out):



We have several simple gadgets available:

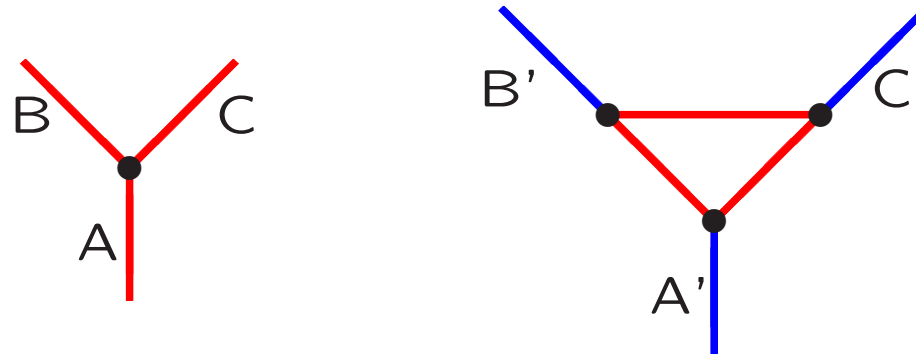
- free blue-edge terminator (FBET)
- *constrained blue-edge terminator* (CBET):
- free red-edge terminator (do we need this?)



Exercise: Explain the CBET (arrows? statespace?).

Exercise: Develop a FBET.

The *CHOICE*-vertex (left) can be emulated by the gadget on the right:



Exercise: Show that the emulation works.

Don't worry about the fact that A, B and C are all red or blue. What matters now is whether they point in or out.

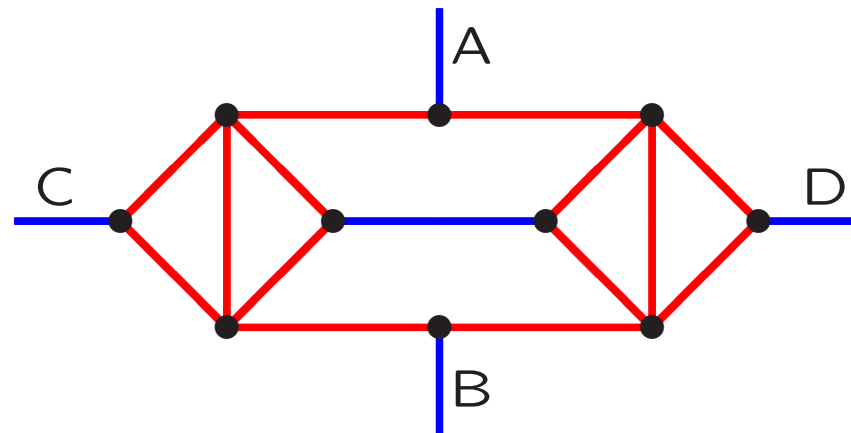
And in reality edges are always directed (have arrows)!

In many graphs we have (unavoidable) *edge crossings*.

We now want a gadget that can replace such a crossing. So assume that we have two crossing blue edges. (There is no node where the edges cross.)



If we have such a gadget, we need only emulate *planar graphs* in our reductions to specific games — and these are often planar (flat)!



Exercise: Show that A and B may not both point out.

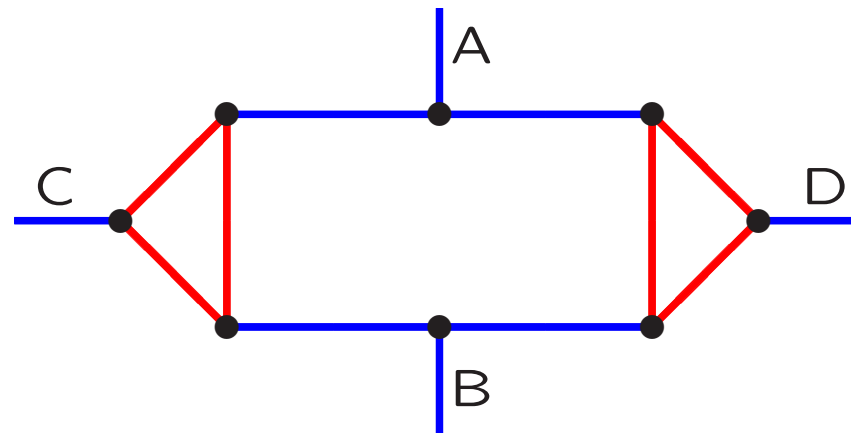
Exercise: Show there are $\leq 2^4 - 7 = 9$ states for ABCD.

Exercise: Show that this emulates two crossing edges.

Exercise: And if each edge may be reversed at most once?

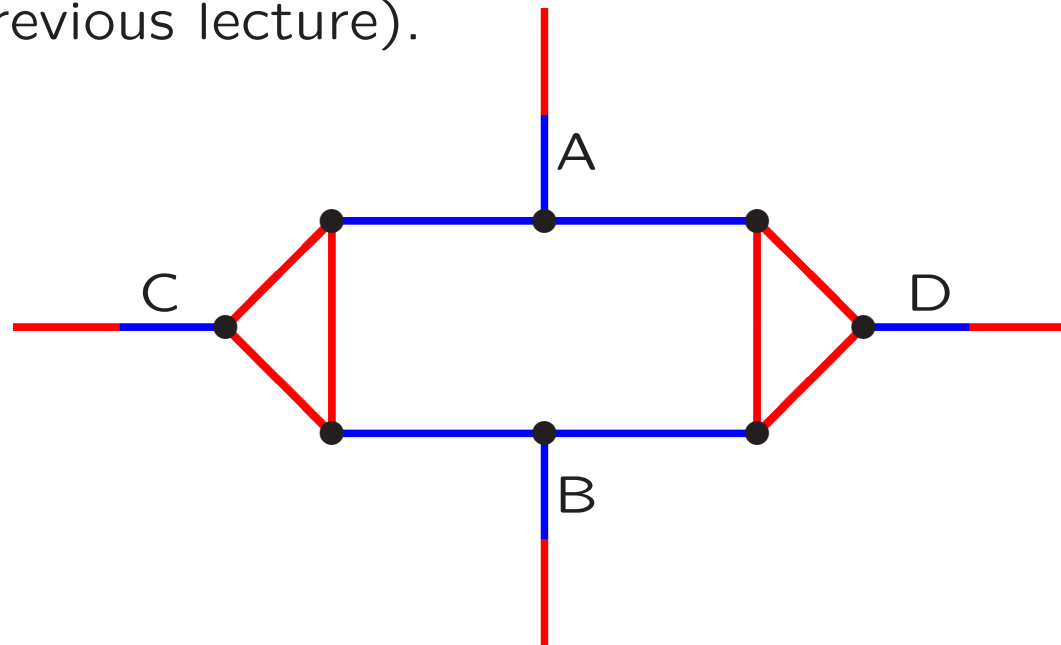
Wait a minute: did we just use “4-red-nodes” !?

The gadget below needs any 2 from A/B/C/D to go in:



Exercise: Show that this can replace a “4-reds-node”.

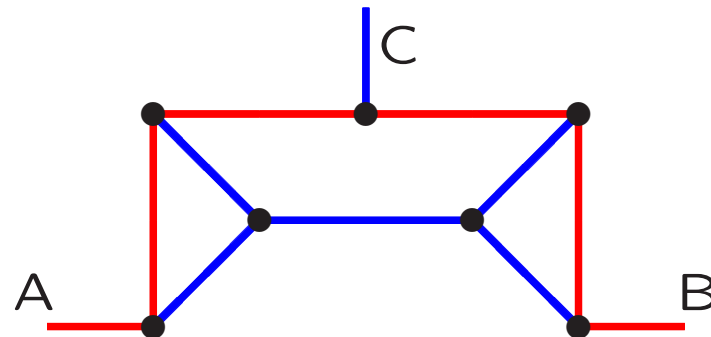
Remember that edges can “change color” on their way. The color is where the edges meet the vertices, so to speak (see previous lecture).



Exercise: Still OK if each edge may be reversed at most once?

In that case we (unfortunately) need a “race condition”.

For a *protected-OR-vertex* two of the three incident edges are special: they are *not both* allowed to be directed inward (by some outside force).



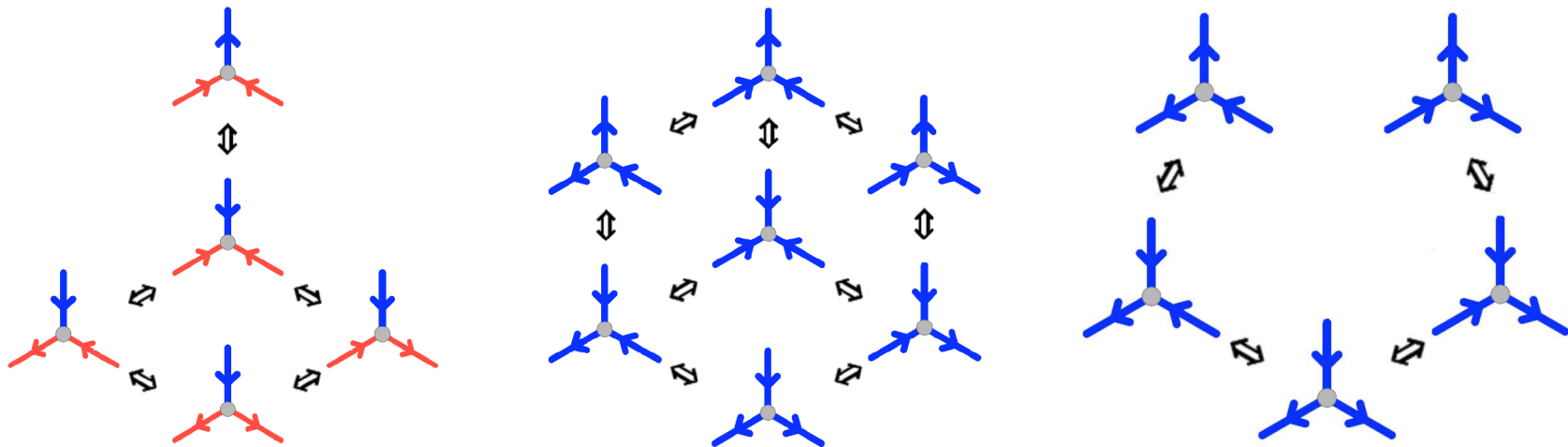
Exercise: Show that this emulates an OR-node.

(Remember again that A and B can “change to blue”.)

Exercise: Where are the protected-OR-nodes in the gadgets?

Exercise: Describe the statespace of a protected-OR-node.

The statespaces for AND, OR and protected-OR:



Reductions between problems concerning games are often based on simple gadgets ... and a lot of technique ... and peculiarities.

Thanks:

Erik Demaine & Bob Hearn (book) and Jan van Rijn.