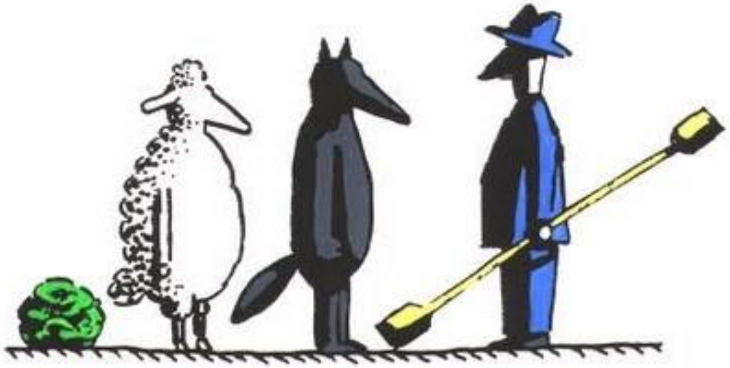


Cabbage, Goat, and Wolf

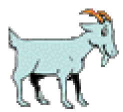
Algorithmics, NP-completeness, and a little Latin

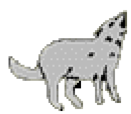
USF Math Club, Tampa FL, Feb 2016

Hendrik Jan Hoogeboom
Universiteit Leiden NL

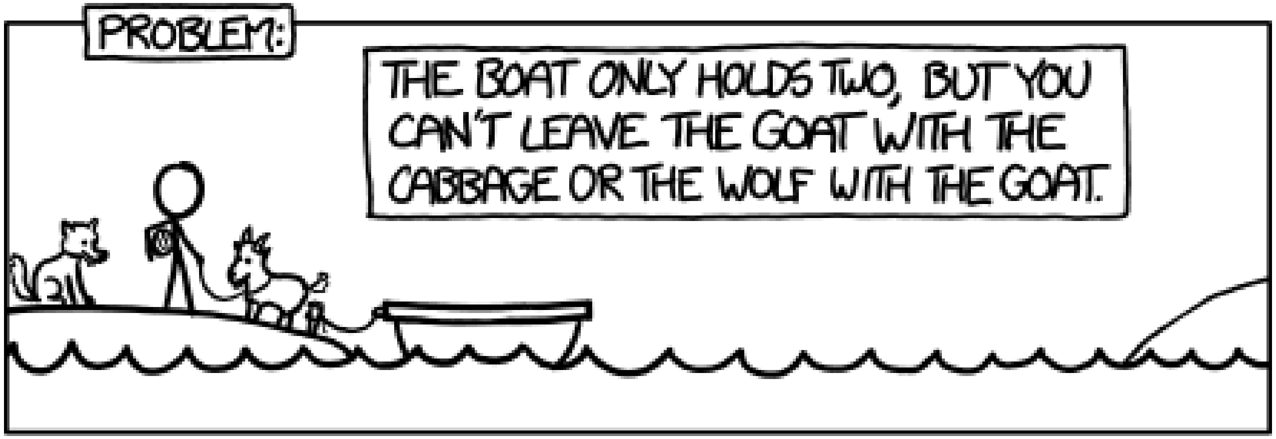
cabbage, goat, and wolf

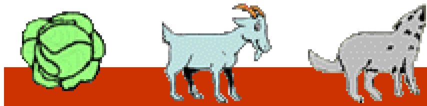
 algorithmics (problem solving)

 complexity

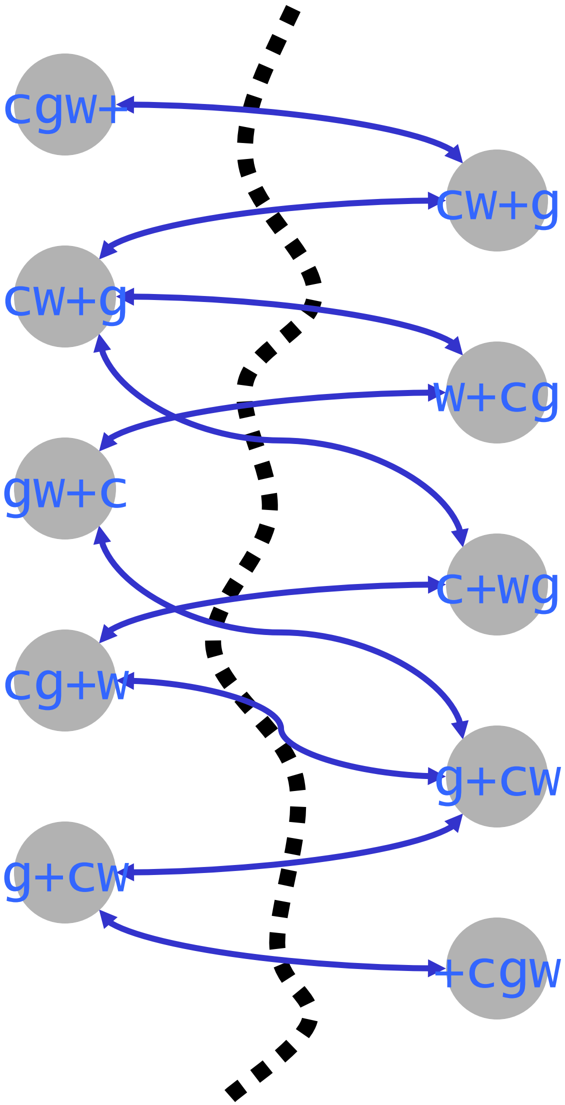
 natural computing

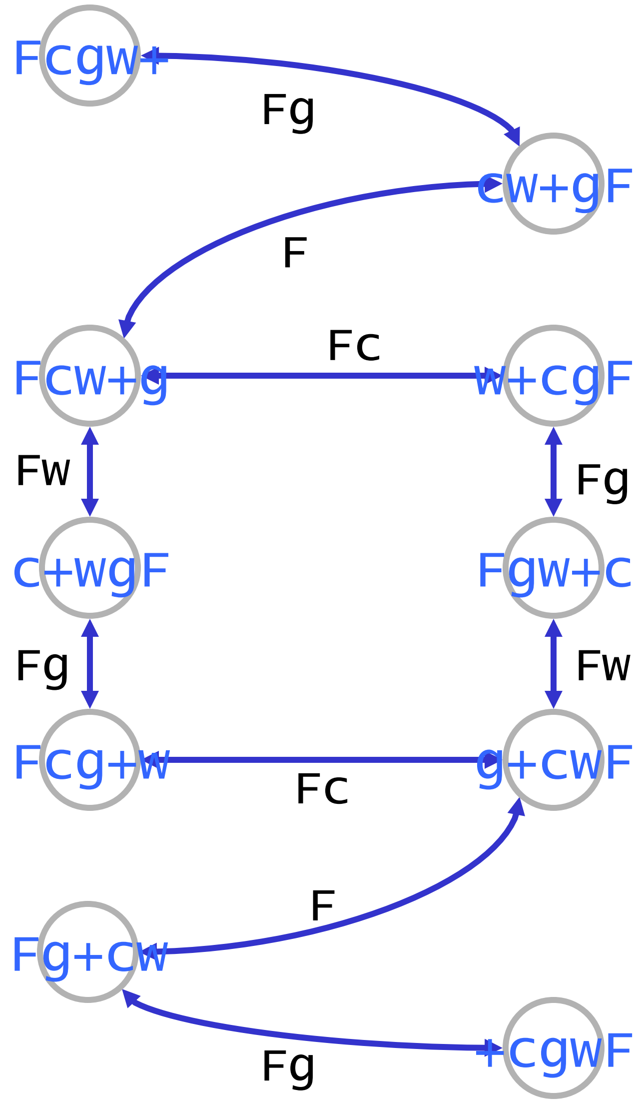
Algorithmics: a puzzle

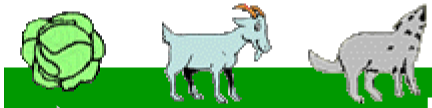

PROBLEM:
THE BOAT ONLY HOLDS TWO, BUT YOU CAN'T LEAVE THE GOAT WITH THE CABBAGE OR THE WOLF WITH THE GOAT.



cabbage, goat, wolf







Propositiones ad Acuendos Juvenes

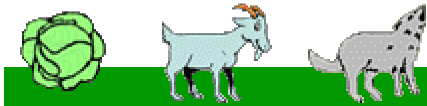
Alcuin of York (York ~735 - Tours 804)

XVIII. PROPOSITIO DE HOMINE ET CAPRA ET LVPO.

Homo quidam debebat ultra fluuium transferre lupum, capram, et fasciculum cauli. Et non potuit aliam nauem inuenire, nisi quae duos tantum ex ipsis ferre ualebat. Praeceptum itaque ei fuerat, ut omnia haec ultra illaesa omnino transferret. Dicat, qui potest, quomodo eis illaesis transire potuit?

Solutio
Simili namque tenore ducerem prius capram et dimitterem foris lupum et caulum. Tum deinde uenirem, lupumque transferrem: lupoque foris misso capram naui receptam ultra reducerem; capramque foris missam caulum transueherem ultra; atque iterum remigassem, capramque assumptam ultra duxissem. Sicque faciendo facta erit remigatio salubris, absque uoragine lacerationis.

http://en.wikipedia.org/wiki/Propositiones_ad_acuendos_juvenes



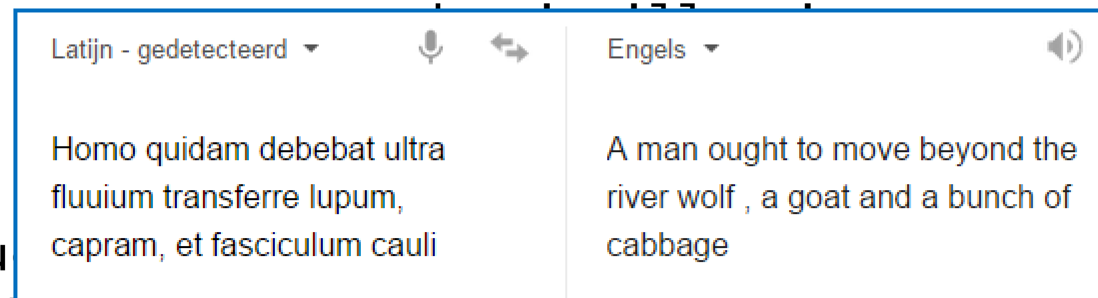
Propositiones ad Acuendos Juvenes

Alcuin of York (York ~735 - Tours 804)

XVIII. PROPOSITIO DE HOMINE ET CAPRA ET LVPO.

Homo quidam debebat ultra fluuium transferre lupum, capram, et fasciculum cauli. Et non potuit aliam nauem inuenire, nisi quae duos tantum ex ipsis ferre ualebat. Praeceptum itaque ei fuerat, ut omnia haec ultra illaesa omnino transferret. Dicat, qui

transire potuit?

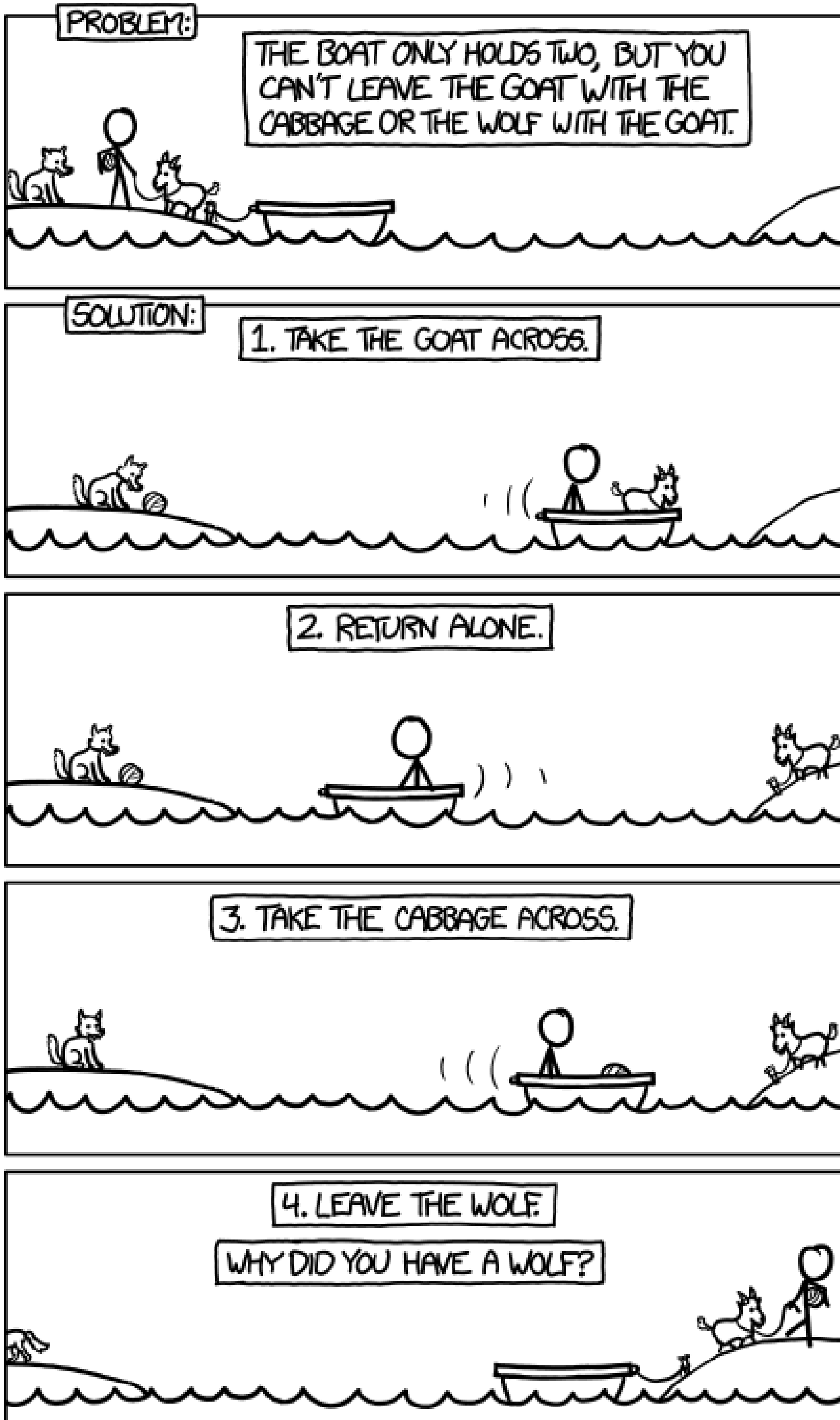


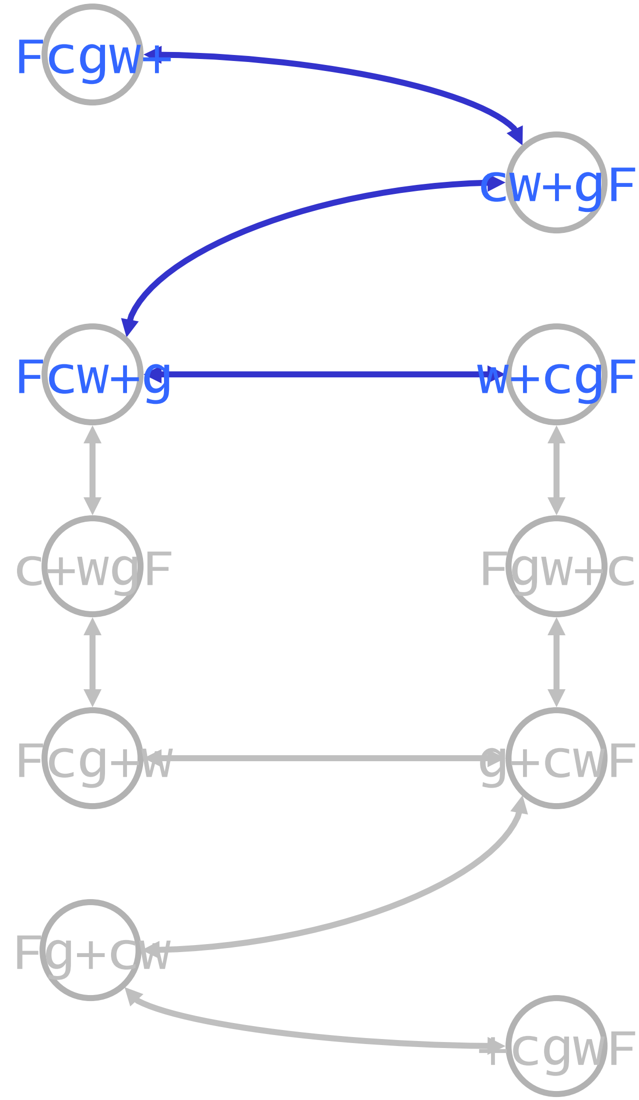
Solutio

Simili namque tenore du

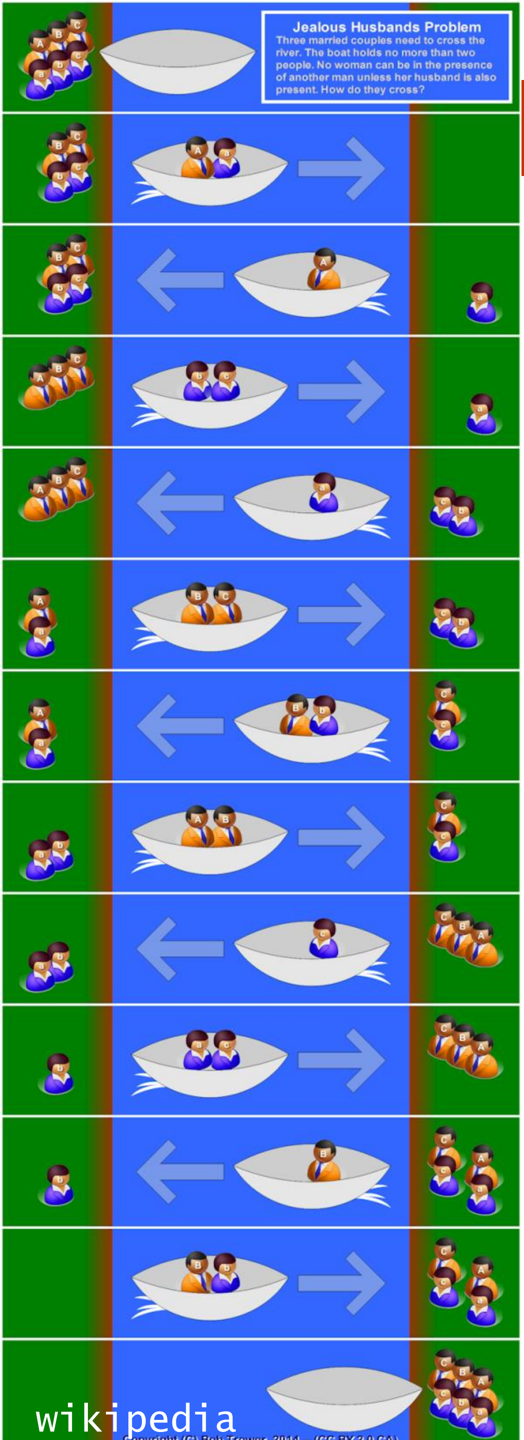
foris lupum et caulum. Tum deinde uenirem, lupumque transferrem: lupoque foris misso capram naui receptam ultra reducerem; capramque foris missam caulum transueherem ultra; atque iterum remigassem, capramque assumptam ultra duxissem. Sicque faciendo facta erit remigatio salubris, absque uoragine lacerationis.

Logic Boat





http://xkcd.com/1134/



jealous husbands

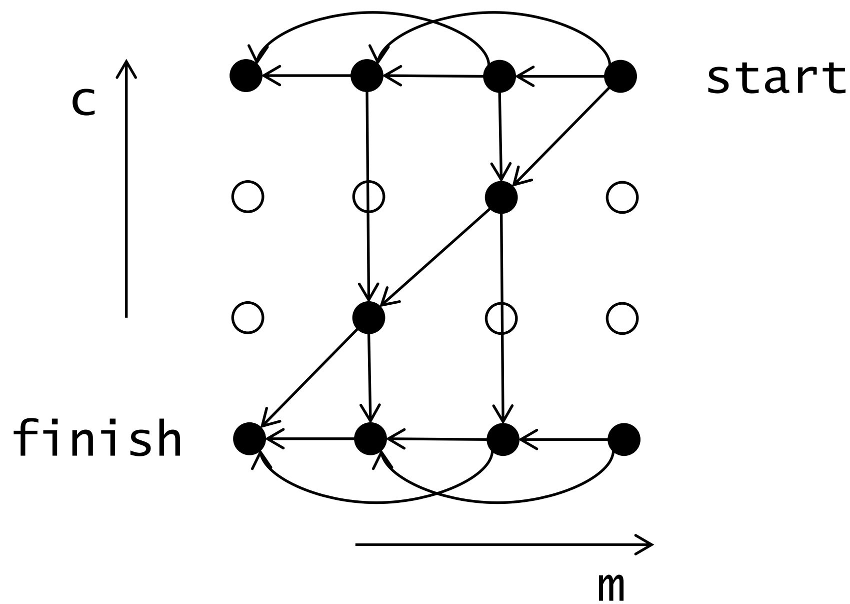
trip number	left bank	travel	right bank
(start)	Aa Bb Cc		
1	Ab Cc	Aa →	
2	Ab Cc	← A	a
3	A B C	bc →	a
4	A B C	← a	b c
5	Aa	BC →	b c
6	Aa	← Bb	Cc
7	a b	AB →	Cc
8	a b	← c	A B C
9	b	a c →	A B C
10	b	← B	Aa Cc
11		Bb →	Aa Cc
(finish)			Aa Bb Cc

missionaries and cannibals

3 missionaries and 3 cannibals; 2 passengers.

when the missionaries outnumber the cannibals the latter will be converted; how to get them safely over?

(c,m) (3-c,3-m)
c=0 or c≥m
c=3 or c≤m (other bank!)



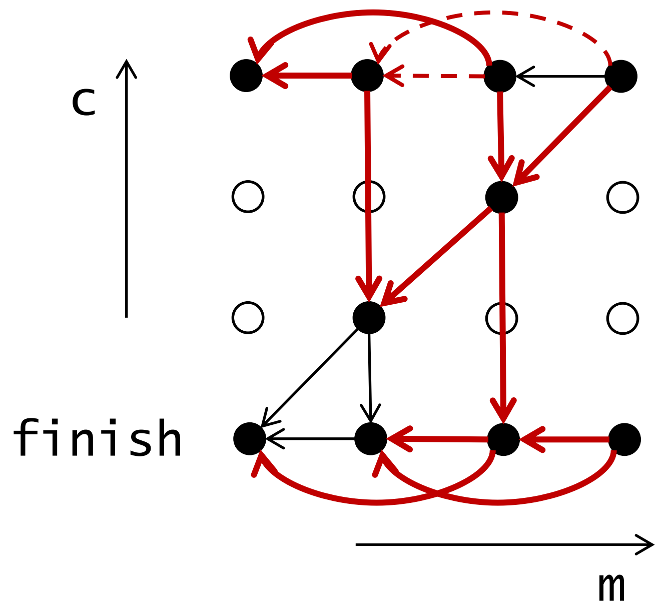
Graphical Solution of Difficult Crossing Puzzles
R.Fraley, K.L.Cooke, P.Detrick, Mathematics Magazine 9 (1966) 151-157

missionaries and cannibals

3 missionaries and 3 cannibals; 2 passengers.

when the missionaries outnumber the cannibals the latter will be converted; how to get them safely over?

(c,m) (3-c,3-m)
c=0 or c≥m
c=3 or c≤m (other bank!)

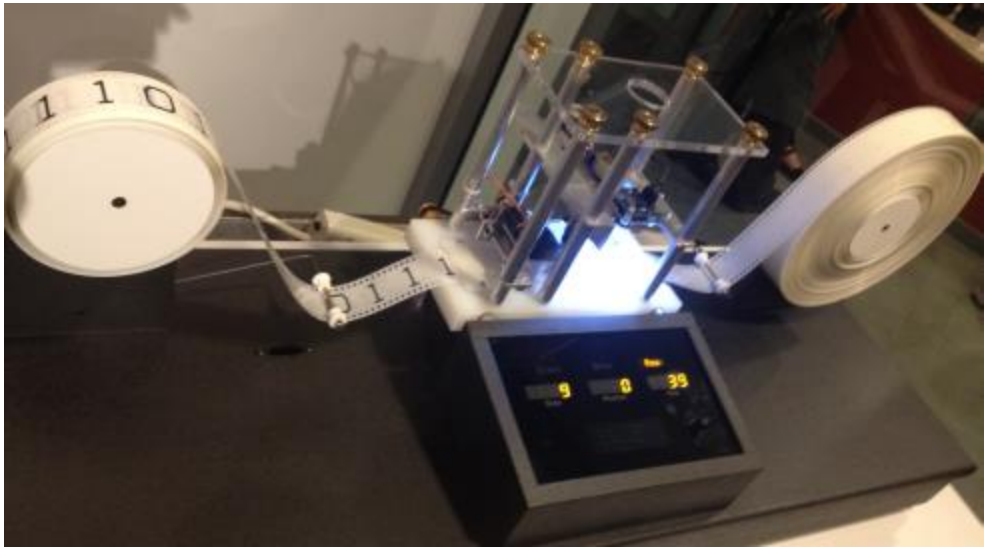


back and forth!
11 steps

Graphical Solution of Difficult Crossing Puzzles
R.Fraley, K.L.Cooke, P.Detrick, Mathematics Magazine 9 (1966) 151-157

complexity: NP-completeness

Turing machine



state	position						
r	ƀ	ƀ	$\boxed{1}$	1	0	1	ƀ
r	ƀ	ƀ	1	$\boxed{1}$	0	1	ƀ
r	ƀ	ƀ	1	1	$\boxed{0}$	1	ƀ
r	ƀ	ƀ	1	1	0	$\boxed{1}$	ƀ
r	ƀ	ƀ	1	1	0	1	$\boxed{\text{ƀ}}$
ℓ	ƀ	ƀ	1	1	0	$\boxed{1}$	ƀ
ℓ	ƀ	ƀ	1	1	$\boxed{0}$	0	ƀ
H	ƀ	ƀ	1	$\boxed{1}$	1	0	ƀ

time $1 \leq k \leq p(n)$

space $1 \leq i \leq p(n)$

(p,c,q,a,D) instruction
- p state
- c letter read
- q new state
- a letter written
- D direction move -1,0,+1

A.M. Turing (1936). On Computable Numbers, with an Application to the Entscheidungs problem. Proc London Math Soc (1937)

Emil Post. Finite Combinatory Processes—Formulation 1, J Symbolic Logic (1936)

dimensions

existential and *universal* states
computation = tree

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

AL AP APSPACE AEXPTIME

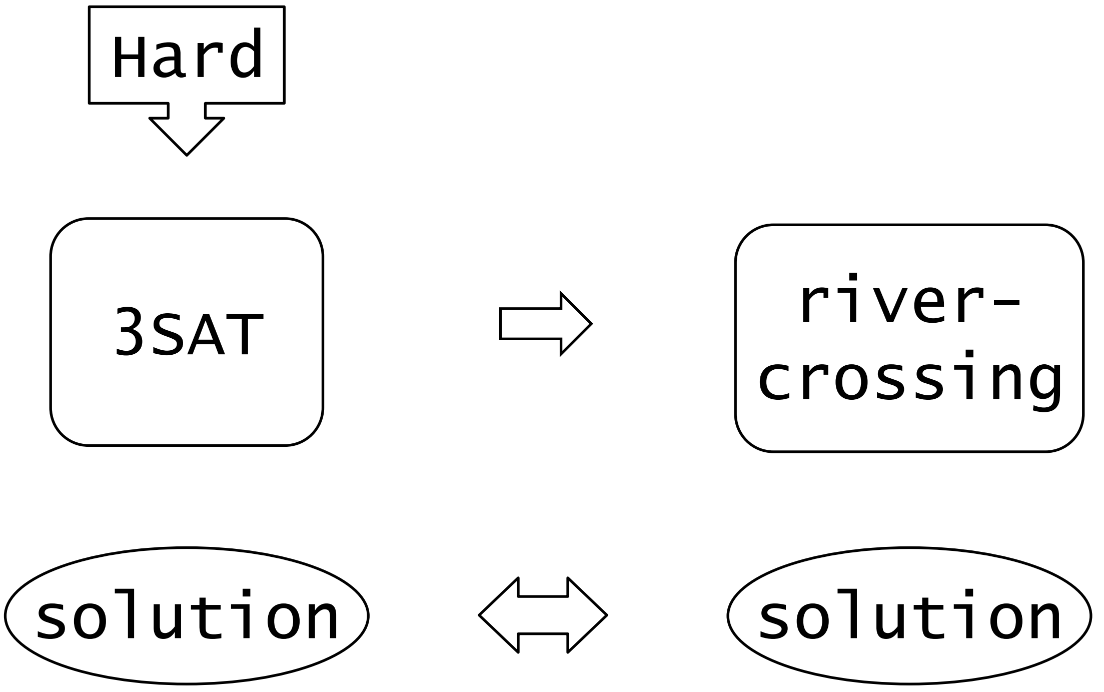
$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME} \subseteq \mathrm{NEXPTIME} \subseteq \mathrm{EXPSPACE}$

NPSPACE NEXPSPACE

Clay Institute
Millenium Prize

Chandra, Kozen, Stockmeyer. 'Alternation', J ACM 1981.

proving complexity



3SAT

literal
(negated) variable

$$\overbrace{(w \vee x \vee y)}^{\text{clause}} \wedge (w \vee \neg x \vee z) \wedge \ldots \wedge (x \vee \neg y \vee z)$$

3 conjunctive normalform

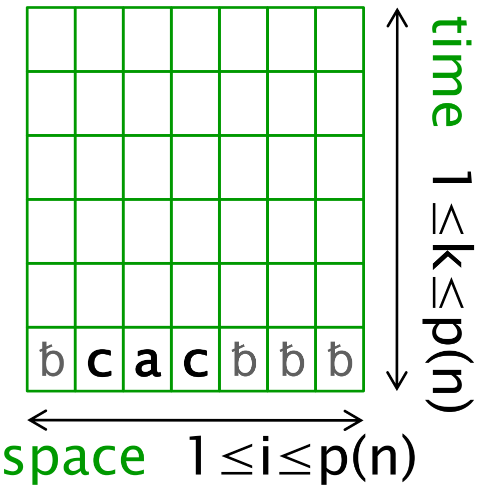
3SAT
given: given formula ϕ in 3CNF
question: is ϕ satisfiable?
(can we find a variable assignment making formula true)

Cook’71/Levin’73 3SAT is NP-complete

Cook/Levin

specify computation
at step k …

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

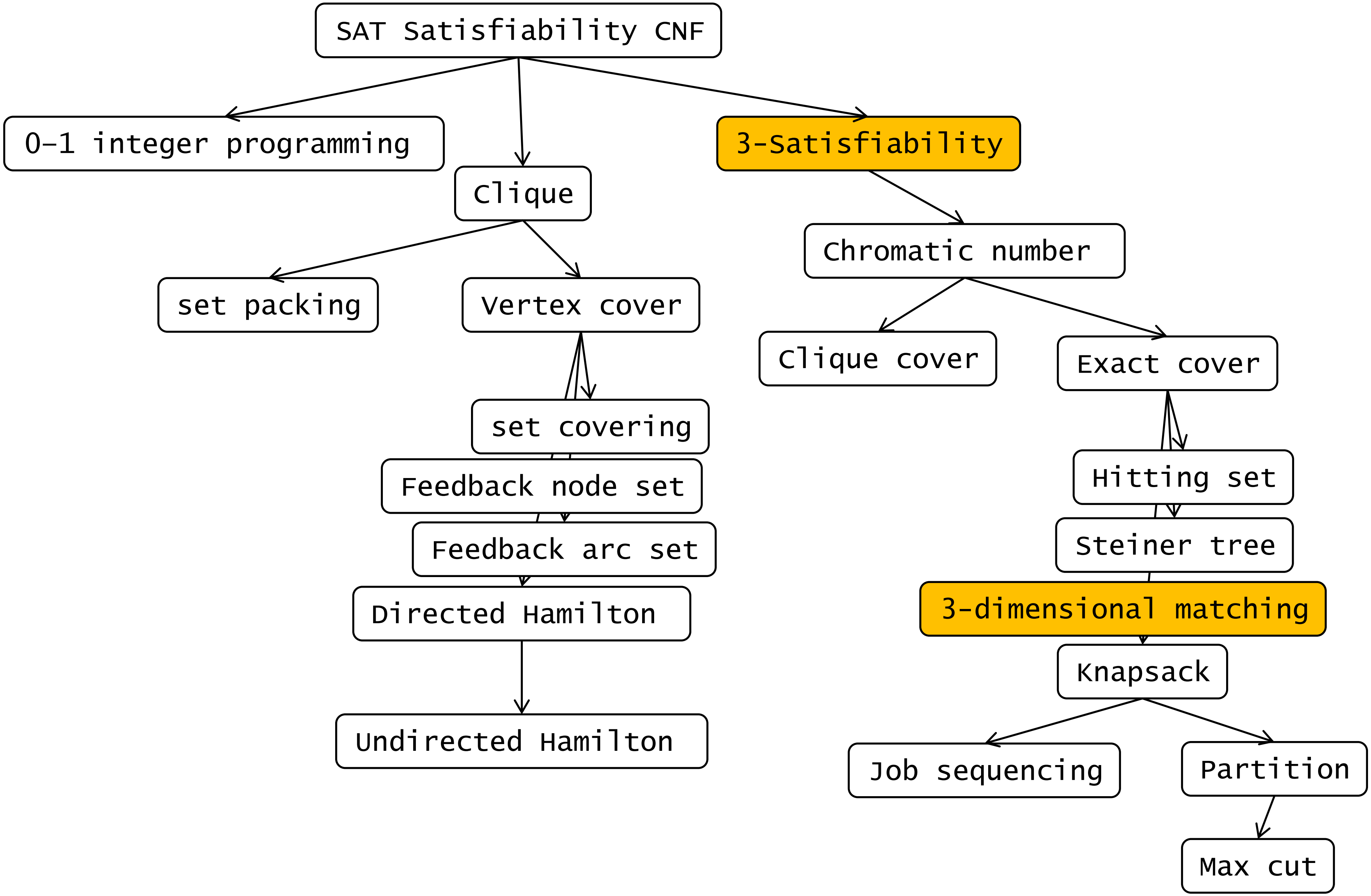


(p,a,p,a,0) for each p,a

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

Karp's 21 NP-complete problems



Richard M. Karp (1972) Reducibility Among Combinatorial Problems

‘generalized’ river crossing

D drivers
P passengers
c capacity boat
m max moves
forbidden combinations:
F_R right bank
F_L left bank
F_B boat

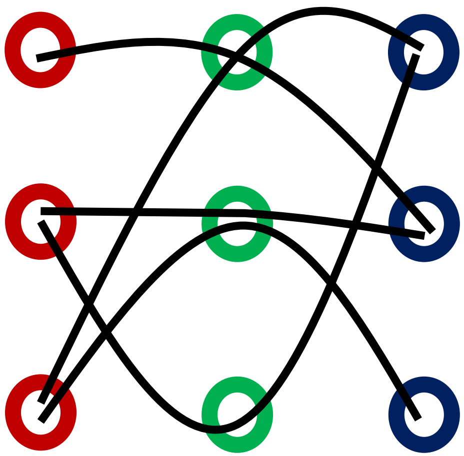
D = { Man }, P = { Wolf, Goat, Cabbage },
F_L = F_R = { {Wolf, Goat}, {Goat, Cabbage},
{Wolf, Goat, Cabbage} },
$F_B = \emptyset$,
c = 2, m = 100.

forbidden: otherwise polynomial, using graph traversal

Thm. [Ito etal.] Polynomial if $F_B = \emptyset$, and $m=\infty$.

Ito, Langerman, Yoshida. Generalized River Crossing Problems, Th. Comp. Sys. 2015

3 dim matching



given:
U,V,W, with |U|=|V|=|W|=k
and M ⊆ U x V x W

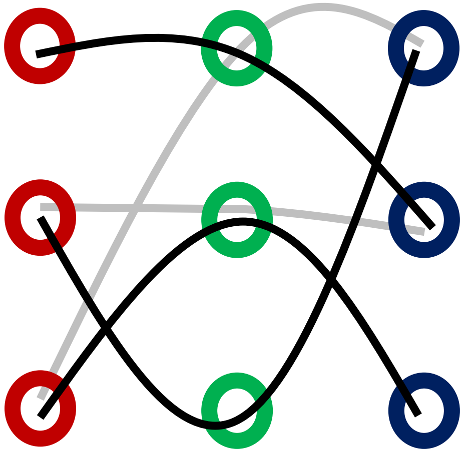
problem:
is there P⊆M with |P|=k
containing each element exactly once?

k=3

{a,b,c} x {p,q,r} x {x,y,z}

P = { (a,p,y), (b,q,y), (b,r,x), (c,p,x), (c,q,z) }

3 dim matching



given:
U,V,W, with |U|=|V|=|W|=k
and M ⊆ U x V x W

problem:
is there P⊆M with |P|=k
containing each element exactly once?

k=3

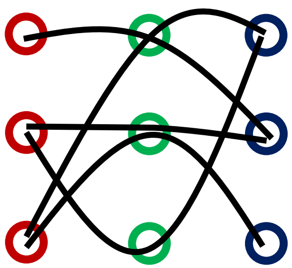
{a,b,c} x {p,q,r} x {x,y,z}

P = { (a,p,y), (b,q,y), (b,r,x), (c,p,x), (c,q,z) }

3D matching ➡ river crossing

U,V,W, with
|U|=|V|=|W|=k en
$M \subseteq U \times V \times W$

$D = U \cup \{s_1, s_2, s_3\}$, $P = V \cup W$, $c=3$,
$m=3k-1$, $F_R = \emptyset$, $F_L = \emptyset$,
$F_B{}^C =$
$\{\ \{u,v,w\} \mid (u,v,w) \in M\ \}$
$\cup\ \{\ \{s_1,s_2,s_3\}\ \}$



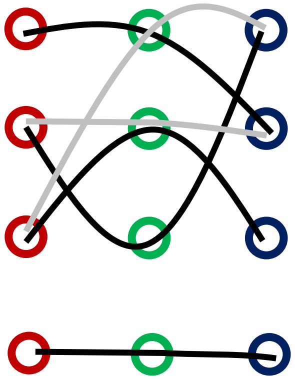
complement!

solution
(matching)

⟺

solution
(crossing)

reduction



matching

P =
{ (a,p,y),
(b,r,x),
(c,q,z),
(d,t,w) }

k=4 (even)

crossing

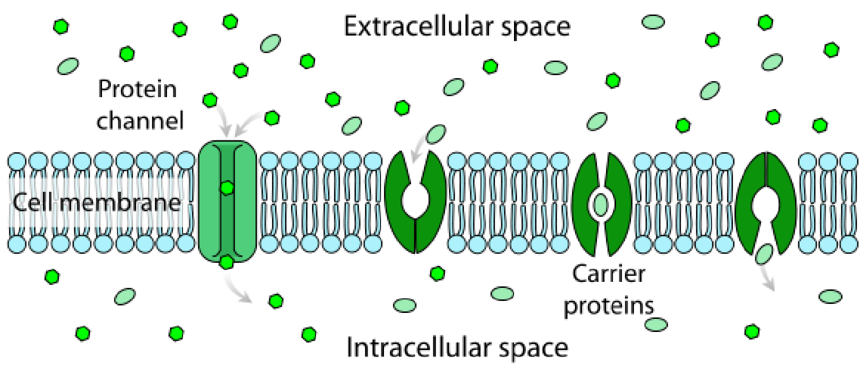
$\{s_1,s_2,s_3\} \rightarrow$
$\leftarrow s_1$
$\{a,p,y\} \rightarrow$
$\leftarrow s_2$
$\{b,r,x\} \rightarrow$
$\leftarrow s_3$
$\{s_1,s_2,s_3\} \rightarrow$
$\leftarrow s_1$
$\{c,q,z\} \rightarrow$
$\leftarrow s_2$
$\{d,s,w\} \rightarrow$
$\leftarrow s_3$
$\{s_1,s_2,s_3\} \rightarrow$

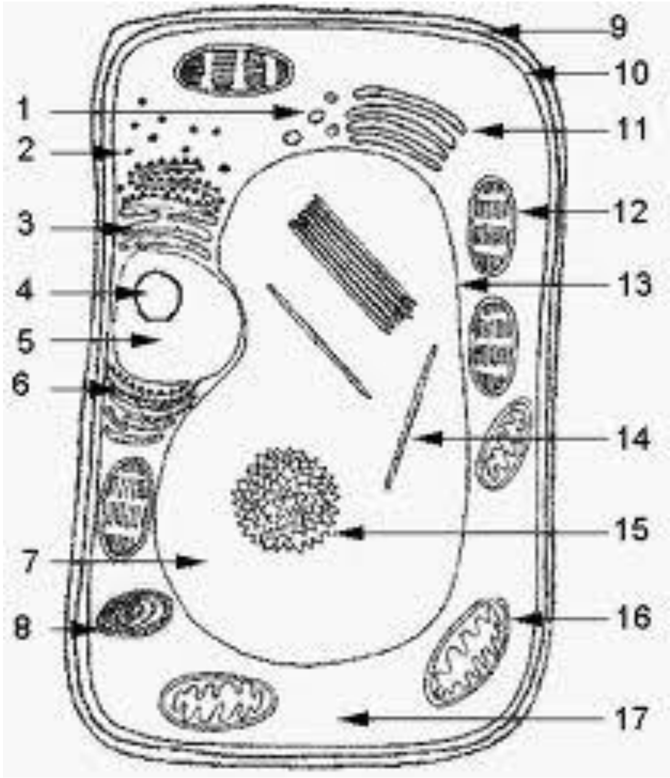
important

m=3k+1
3k+3 objects

Natural Computing: a model

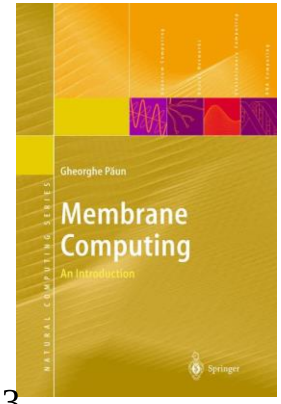
nature inspires

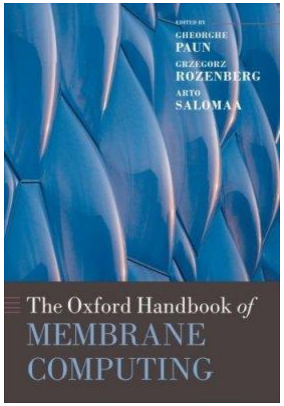


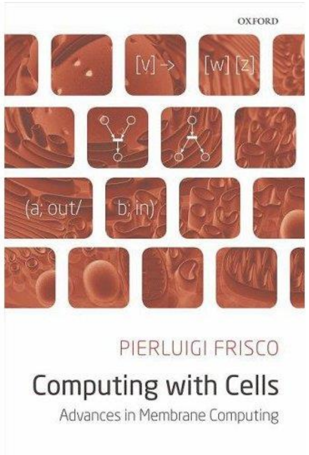


nested compartments
- information

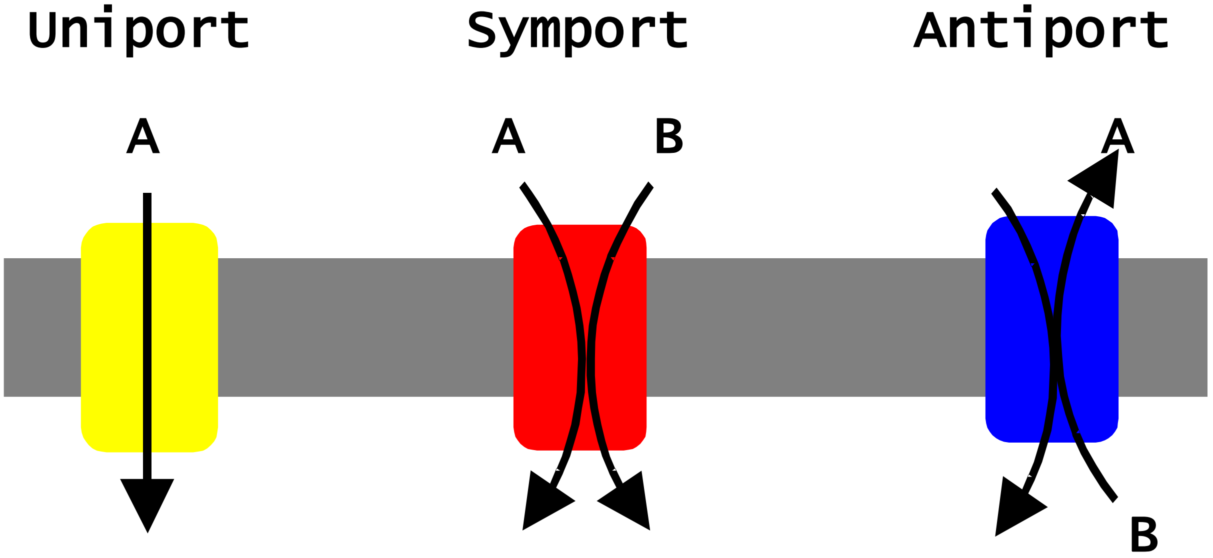
membranes - communication







simple system

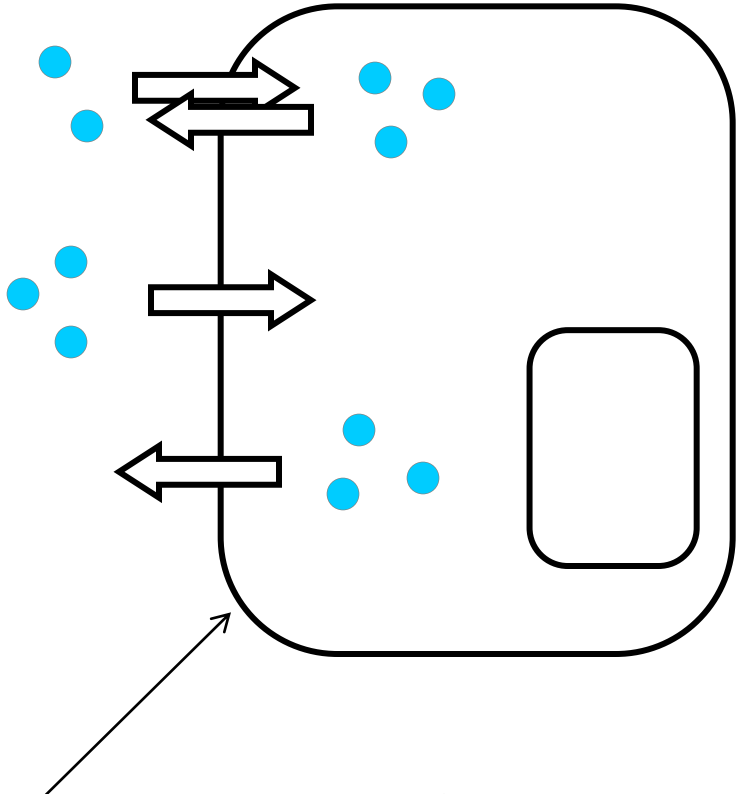


- only move together

- how many membranes?
- how many passengers?
- antiport necessary?

P systems with symport/antiport

Păun & Păun



rules in membrane

contents

- objects
  multiset symbols
  infinite supply
  in environment

rules

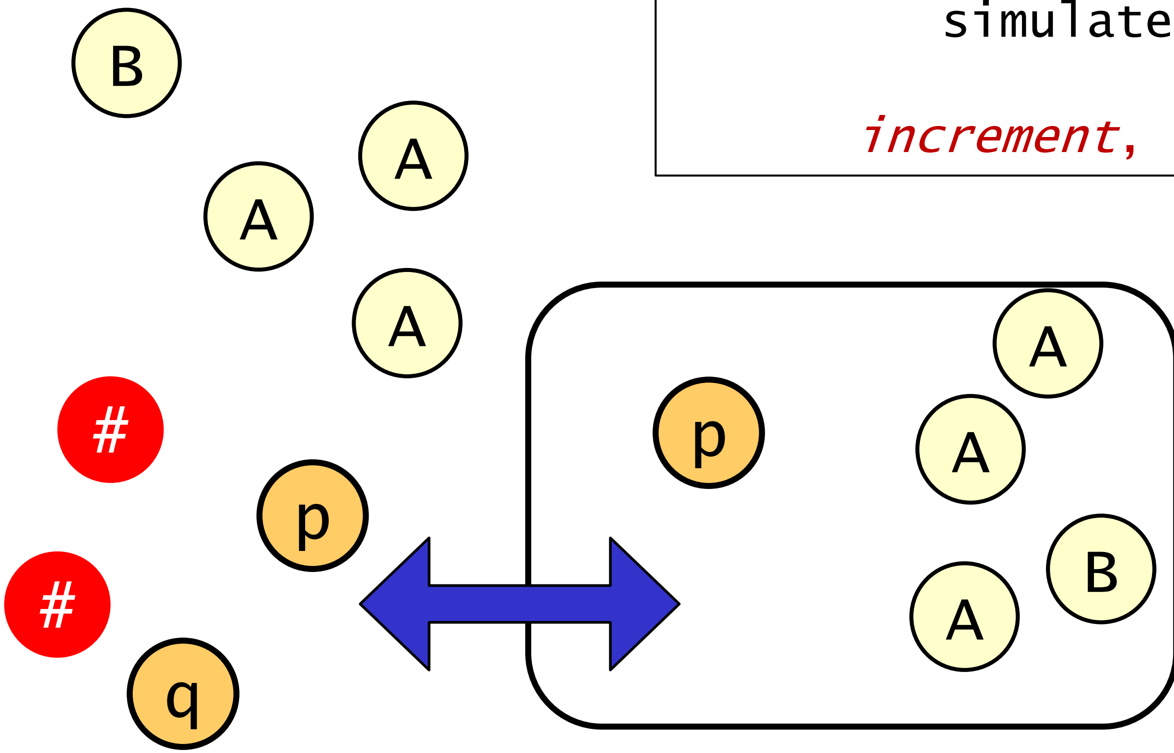
$(\; a_1 \ldots a_k \; \leftrightarrow \; b_1 \ldots b_\ell \;)$ antiport

$(\; a_1 \ldots a_k \; \rightarrow)$ symport (in)

$(\; \leftarrow \; a_1 \ldots a_k \;)$ (out) *sadly enough*

antiport single membrane will do

simulate a **counter automaton**
(=register machine)
increment, *decrement*, *zero test*



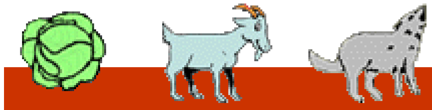
state p
counter A=3
B=1

(p→q, +A)
(p→q, -A)
(p→q, A=0)

counter aut

(qA ↔ p)
(q ↔ pA)
see next

antiport 'simulation'



conclusion

three person boat + forbidden combination:

back to where we started!

and this is the oldest ‘paper’ I ever cited:

References

AY99. Alcuin of York (735-804). *Propositiones ad Acuendos Iuvenes*, around 799.

Remark. The construction in the proof of Theorem 1 above reminds us of the classical wolf, cabbage, and goat problem. attributed to [AY99, Propositio XVIII][1]. The carrier v crossing the membrane echoes the little boat crossing the river, whereas the carrier ∂ models the conflicting presence of goat with either wolf or cabbage on the banks of the river (here the membrane).

[1] Propositio de Homine et Capra et Lupo. Homo quidam debebat ultra flavium transferre lupum, capram, et fasciculum cauli. Et non potuit aliam navem invenire, nisi

done

references

Stephen Cook. The complexity of theorem proving procedures, Proc. 3rd Ann. ACM STOC (1971) pp. 151-158

H.J. Hoogeboom. Carriers and Counters: P Systems with Carriers vs. (Blind) Counter Automata, LNCS 2450 (2003) pp. 140-151
DOI 10.1007/3-540-45005-X_12

Hiro Ito, Stefan Langerman, Yuichi Yoshida.
Generalized River Crossing Problems, Th. Comp. Sys. 56 (2015) 418-435
DOI 10.1007/s00224-014-9562-8