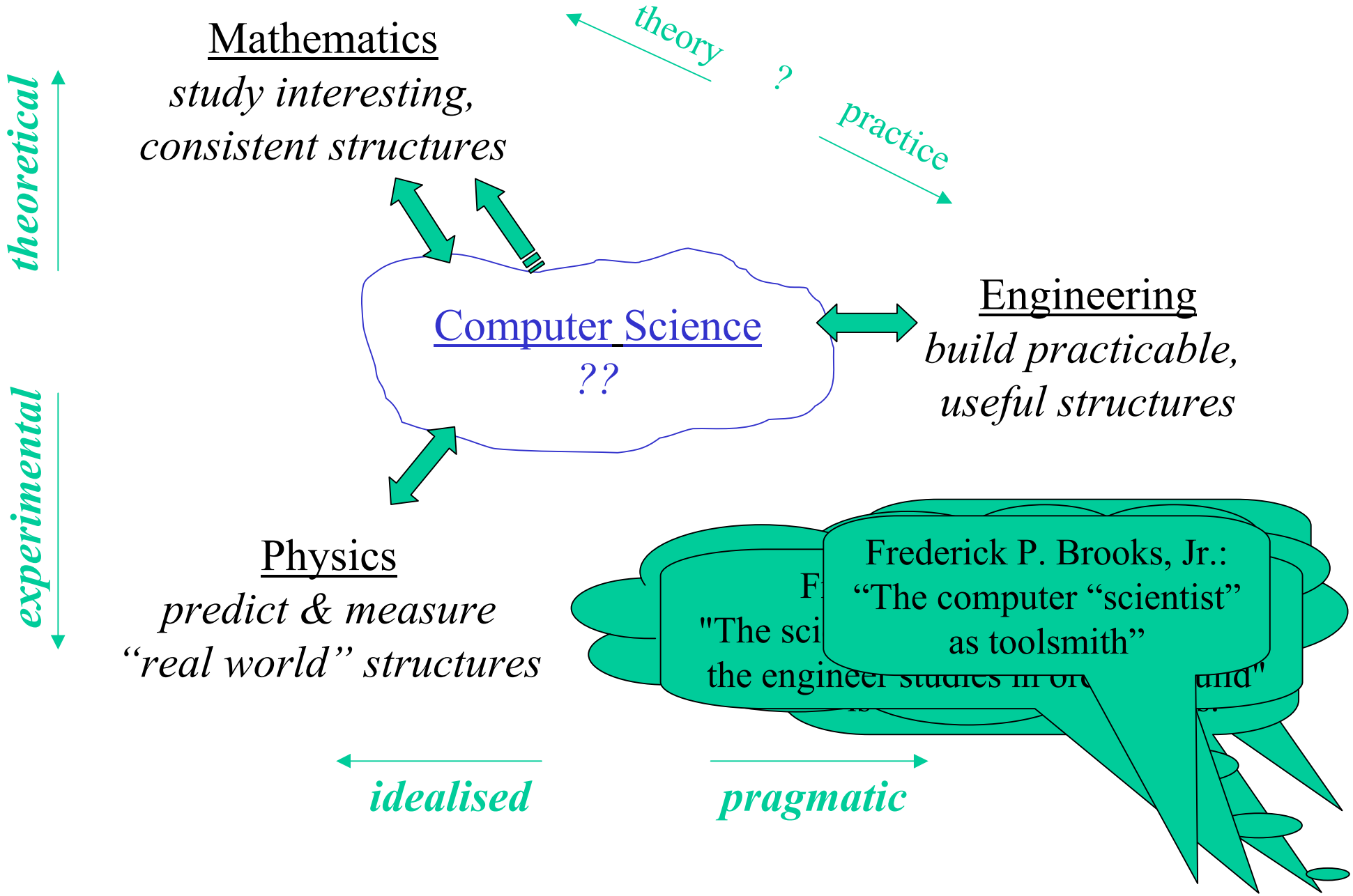
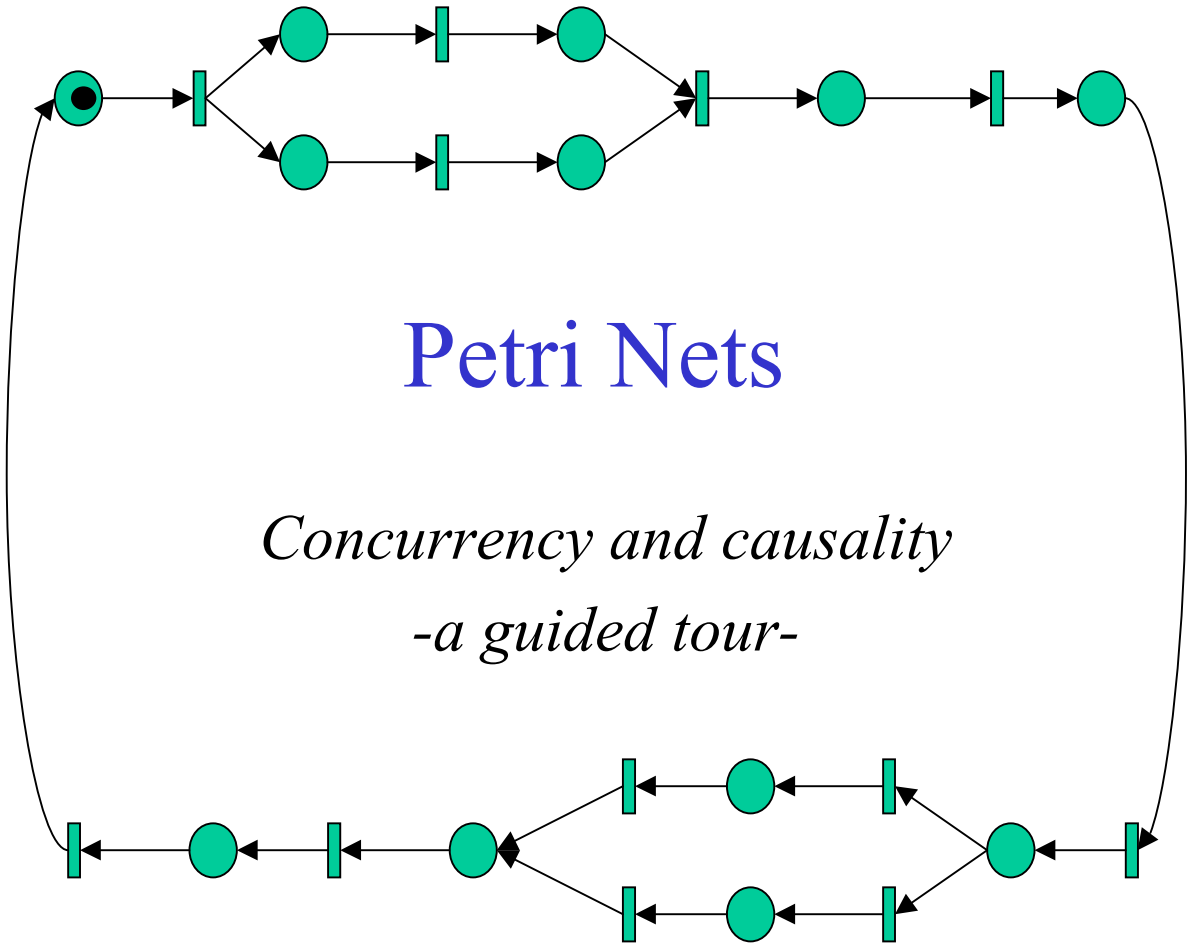

theory
?
practice
theoretical
experimental
Mathematics
study interesting,
consistent structures
Computer Science
??
Engineering
build practicable,
useful structures
Physics
predict & measure
"real world" structures
Frederick P. Brooks, Jr.:
"The computer "scientist"
as toolsmith"
idealised
pragmatic



Petri Nets

Concurrency and causality

-a guided tour-

Claus Reinke

Computing Lab, UKC

Problems with automata-based approaches

- automata are a theoretical and idealised model

- they reflect a Newtonian world-view:
  - *space & time as an absolute frame of reference*
  - *clockwork view of processes within this frame*

- 20th century developments in physics:
  - *special relativity (what is a clock? how fast can signals be?)*
  - *quantum physics (interactions by particle exchange)*

engineers build systems where the difference matters!

Simple models of complex worlds, or: “What is the problem?”

- Newtonian physics is a special case of more recent models.
  W*e hardly travel at relativistic speeds or operate in quantum dimensions, do we? So why not use the simpler model?*
- in models of existing systems, *automata imply an approximation*..

  (simpler, but applicable only if their assumptions hold)
- in designs of new systems, *automata involve over-specification*!

  (engineers have to implement the assumptions!)

Models can be unrealistic if they are too simple, and simplifying designs are harder to realise!

Petri's nets versus Petri nets, or: complex foundations for simple models

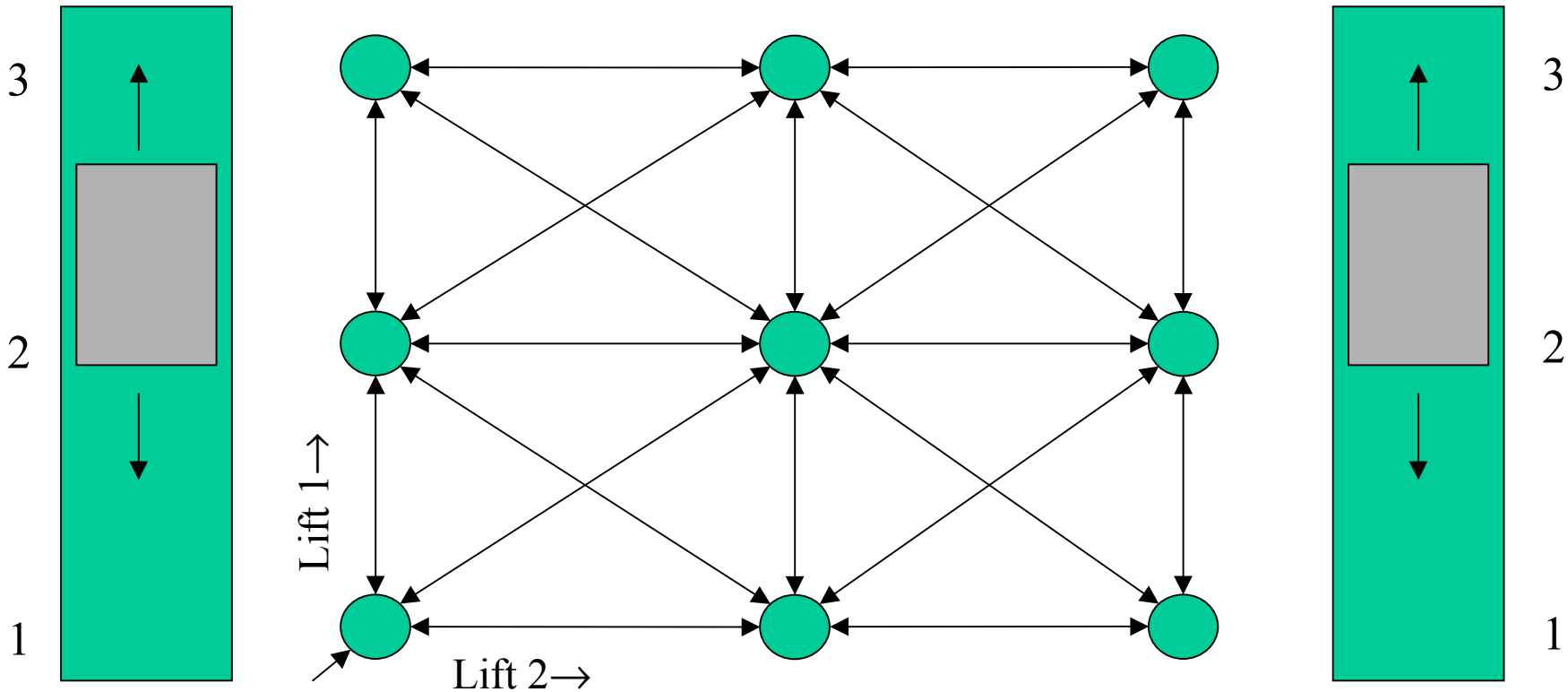
- For his nets, Carl Adam Petri has made an attempt to combine automata from theoretical CS, insights from physics, and pragmatic expertise from engineers:
  - *state is distributed, transitions are localised (space is relevant)*
  - *local causality replaces global time (time as a derived concept)*
  - *subsystems interact by explicit communication*

    (information transport is as relevant as information processing)

engineers can often ignore the background - Petri nets just work!

(but the background explains why things work, why concepts from other disciplines, such as logic, have been integrated into Petri nets so easily, and why foundational research has to continue)

From automata ...

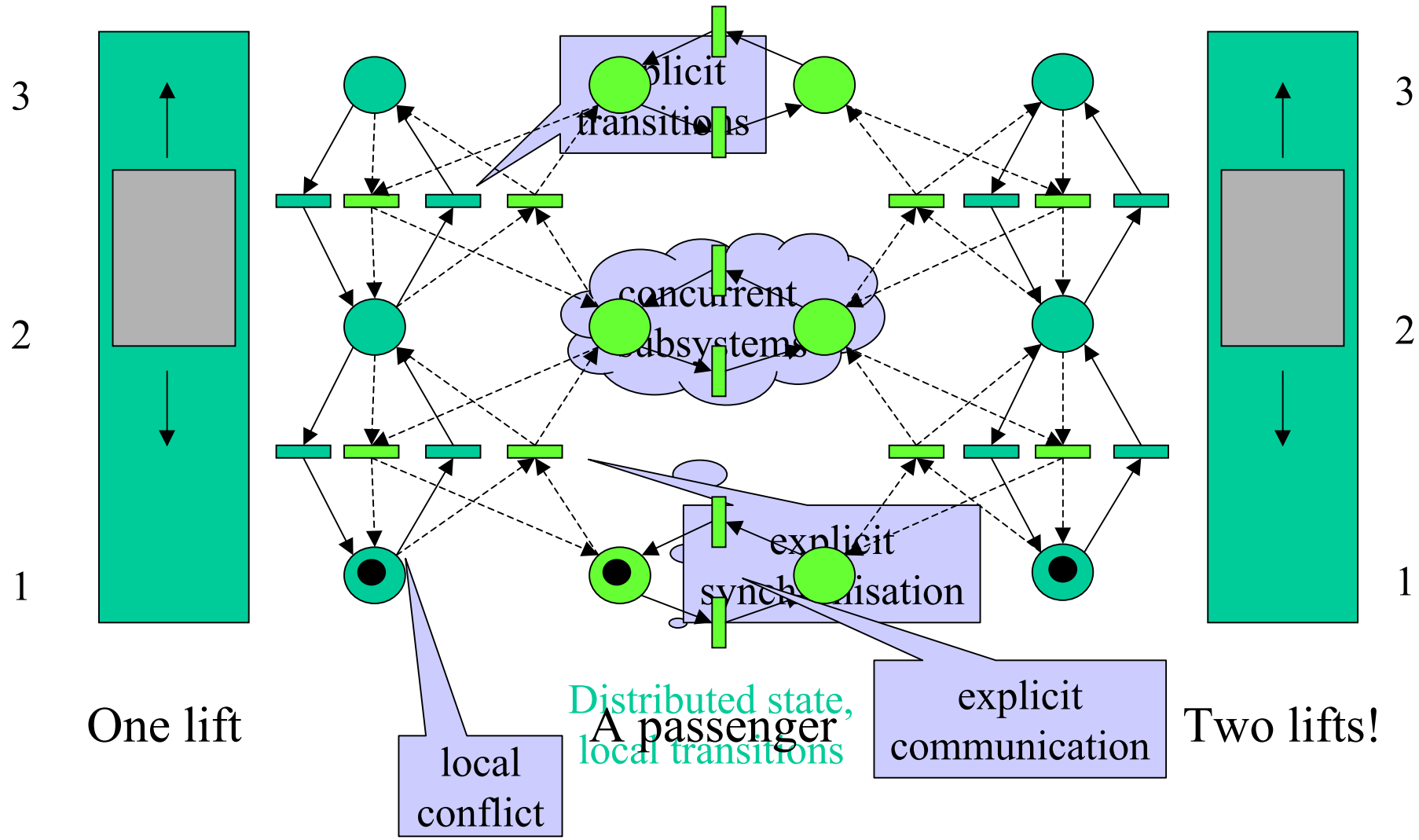


One lift

Product automaton
(⇒meta-level modelling)

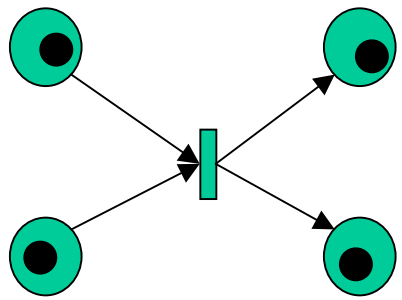
Two lifts?

… to Petri nets



Condition/event-systems (some terms)

- a subclass of Petri nets: state elements hold either one or no token
  - state elements represent conditions, which can be true or false
  - transition elements are represent local events
- an event is enabled if and only if
  - all its pre-conditions (connected by incoming arcs) are true
  - all its post-conditions (connected by outgoing arcs) are false
- an event occurrence negates its pre- and post-conditions
- events with overlapping pre- or post-conditions are in conflict, non-conflicting enabled events may occur concurrently;
- marking of the net: token distribution; c/e-system: c/e-net + marking; configurations: the possible markings of a c/e-net, cases of a c/e-system: the configurations reachable from initial marking (→case graph)



Automata are a <u>sequential</u> subclass of c/e-systems

exactly one condition is true,
each event has a single pre- and a single post-condition

An interlude on time

- timed automata measure time with a global clock (x happens *later than* y)
- implied total synchronisation of all local clocks
- execution of actions are assumed to be totally ordered (interleaving semantics)
- communication is synchronous by default
- time and actions proceed along two separate dimensions
  actions do not change time, sequences of actions can happen at the same time

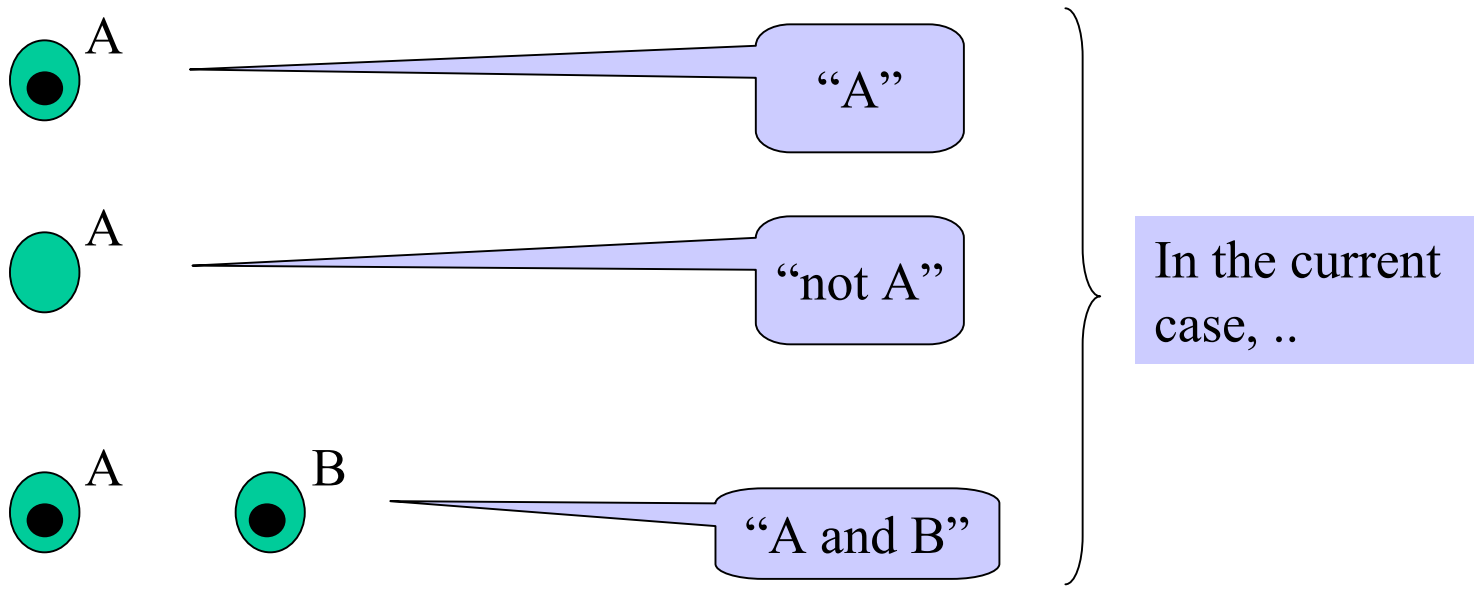
- nets replace global time by local causal dependencies (x happens *after* y)
- global and local clocks can be modelled, but need to be synchronised explicitly
- only a partial order of event occurrences is assumed (concurrency possible)
- communication is asynchronous by default
- events have zero duration, but correspond to advances at least in local time
  no changes outside time, and time is only measured in terms of changes

An ancient philosophical question: intervals or instances? The net answer: both!

- events roughly correspond to instances, conditions to open intervals
- all choices left to the modeller, continuity possible without real-valued time

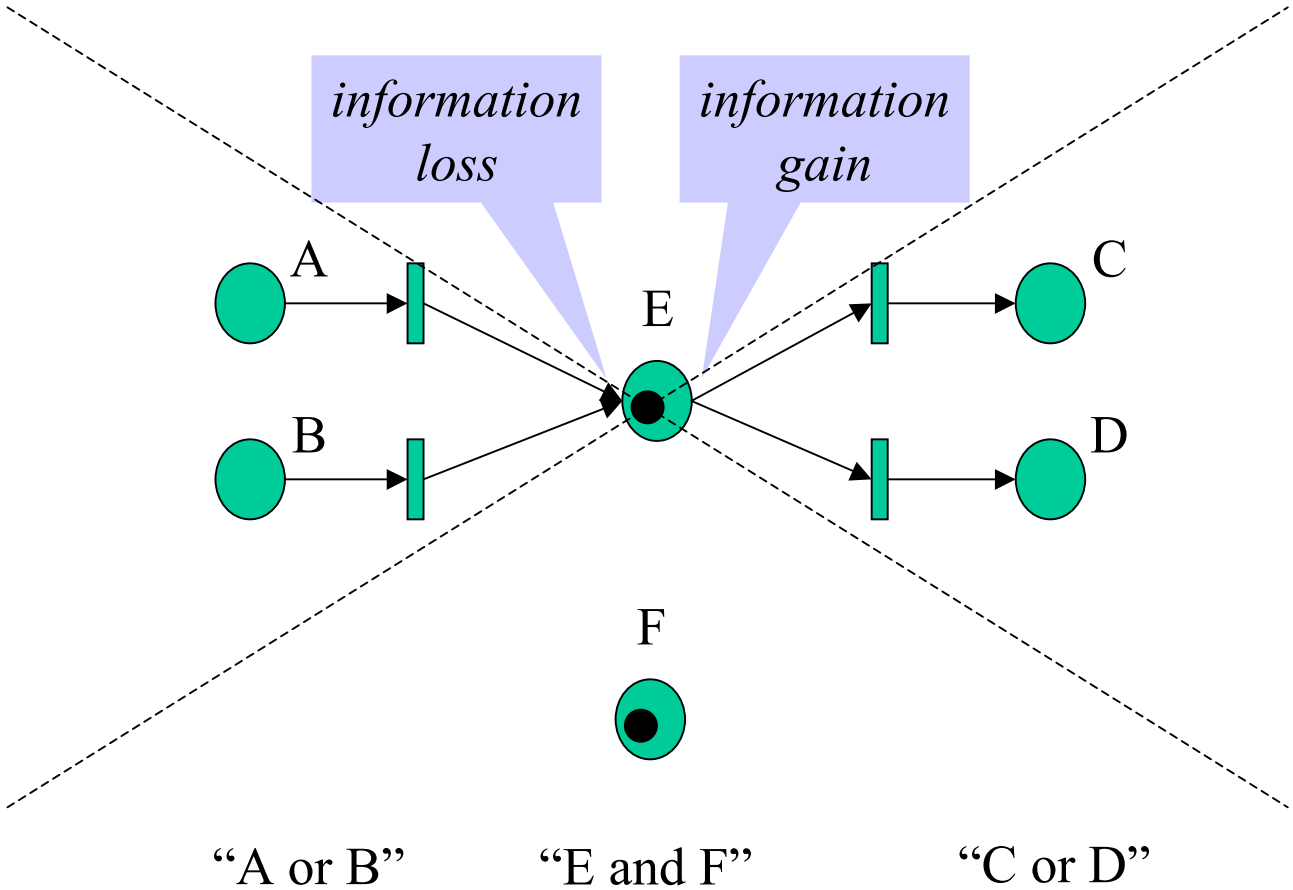
From automata to Petri nets, again, via Logic

- states in automata represent atomic propositions about system configurations
- in condition/event-systems, complex propositions about system configurations can be expressed as well:

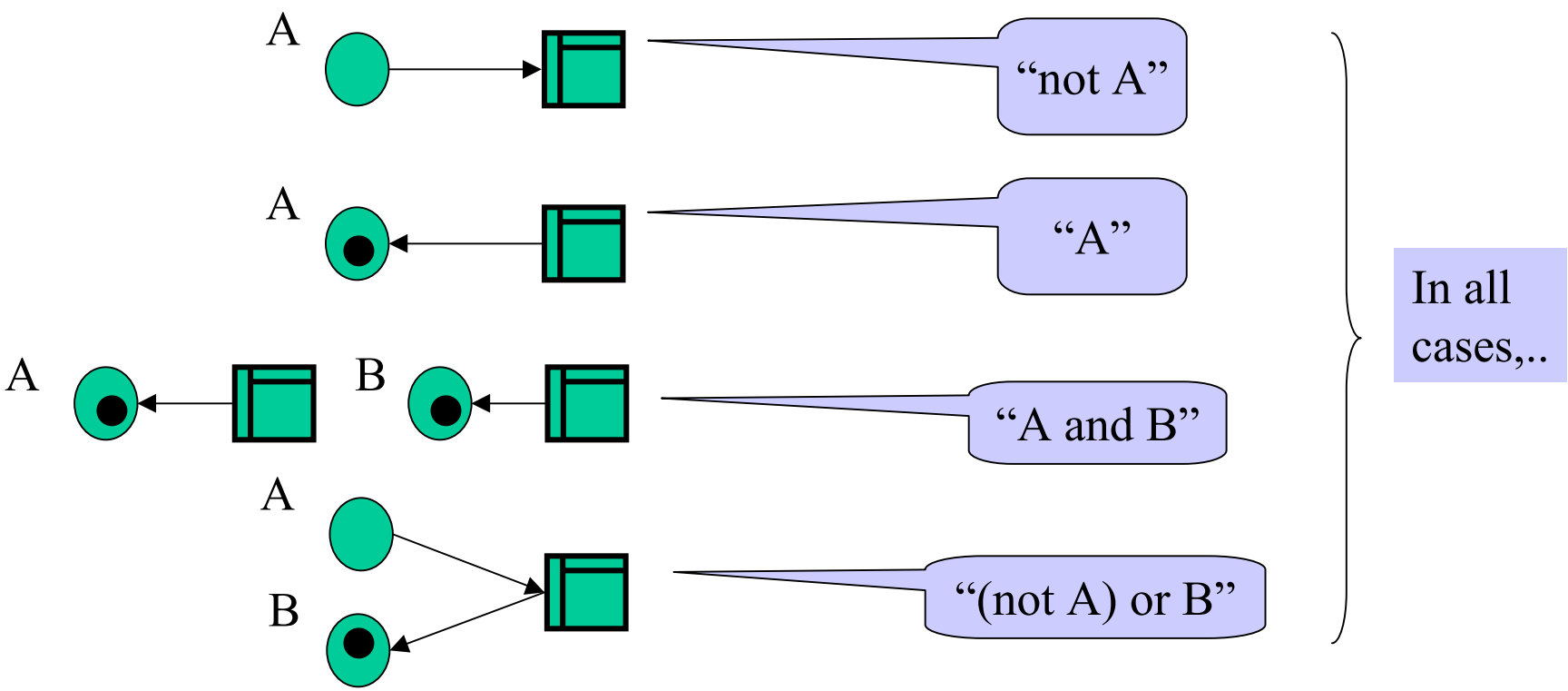


But valuations of propositions change dynamically, so..

No disjunction in the present case?



The Calculus of Facts
expressing system invariants in propositional logic



dead events (never enabled) ***system invariants (facts)***

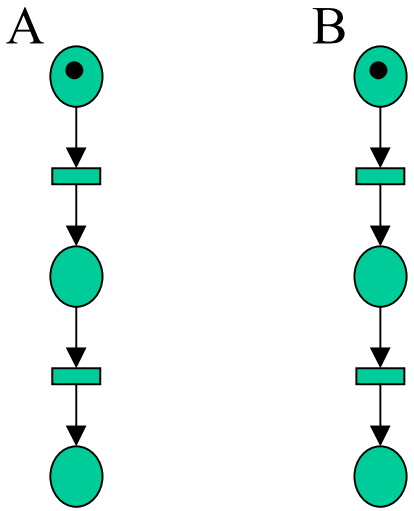
Reasoning, in terms of transformations of fact-nets, is also possible:

- expansion: adding conditions to facts
- resolution: eliminating bridges between facts

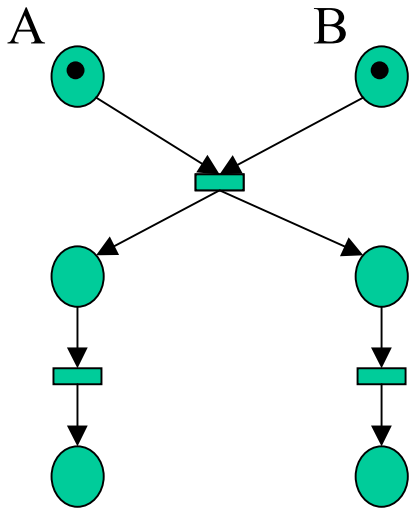
Petri nets: recap

- Petri nets are an extension of automata for concurrency and communication
- automata are a sequential sub-class of Petri nets
- main new features: distributed state and local transitions
- Petri nets have intrinsic connections to physics, logic, and engineering
- Petri himself has been working to found a theory of information processing, as a part of a science of computing, modelled after physics
- others have been busy just exploring the theoretical and practical possibilities:-)

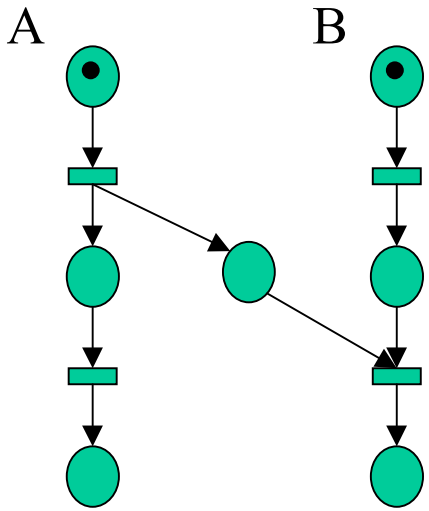
From FA to CPN - a micro-introduction



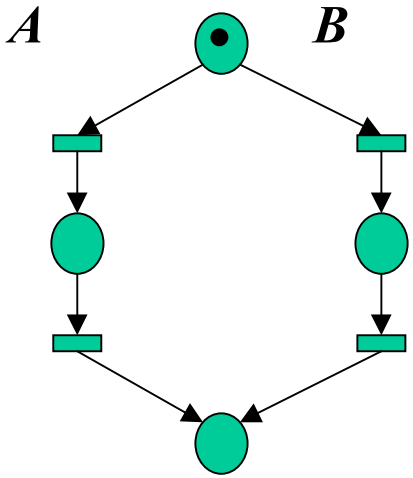
concurrency



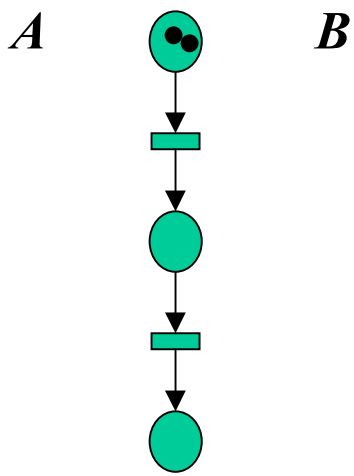
synchronisation



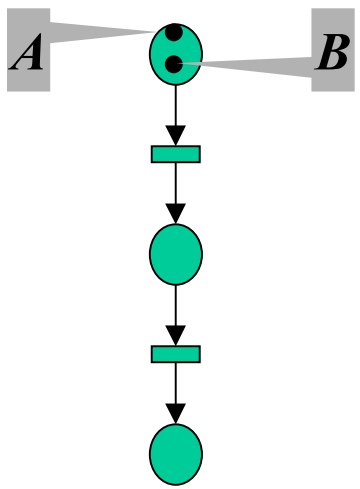
communication



conflict/choice



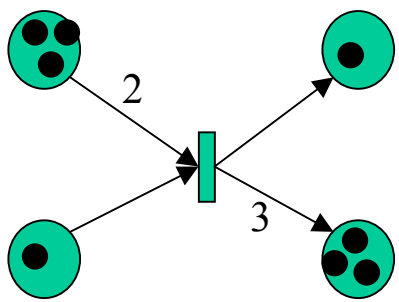
multiplicity/resources



individuality/data

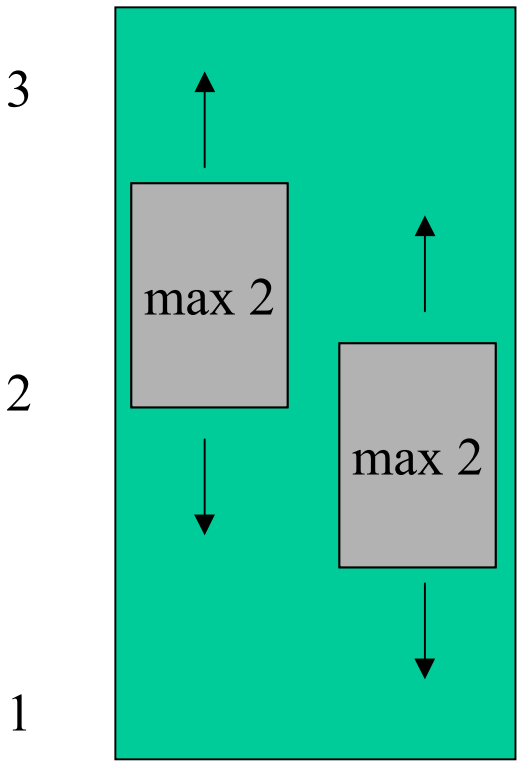
From conditions to resources

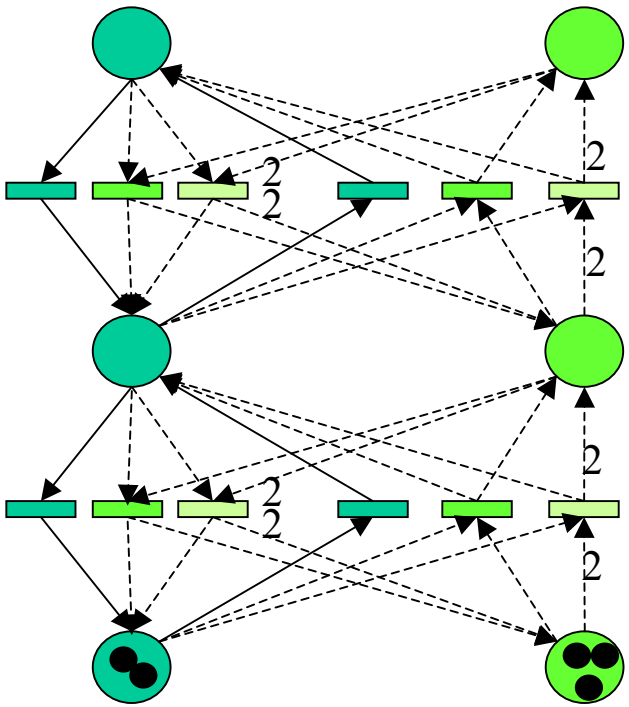
- c/e-systems model the flow of information, at a fundamental level (true/false)
- there are natural application areas for which the flow/transport of resources and the number of available resources is important (data flow, document-/workflow, production lines, communication networks, www, ..)
- place/transition-nets are a suitable generalisation of c/e-systems:
  - state elements represent places where resources (tokens) can be stored
  - transition elements represent local transitions or transport of resources
- a transition is enabled if and only if
  - sufficient resources are available on all its input places
  - sufficient capacities are available on all its output places
- a transition occurrence
  - consumes one token from each input place and
  - produces one token on each output place



(multiple arcs with the same source and target are abbreviated numerically)

Lifts again, with place/transition-nets



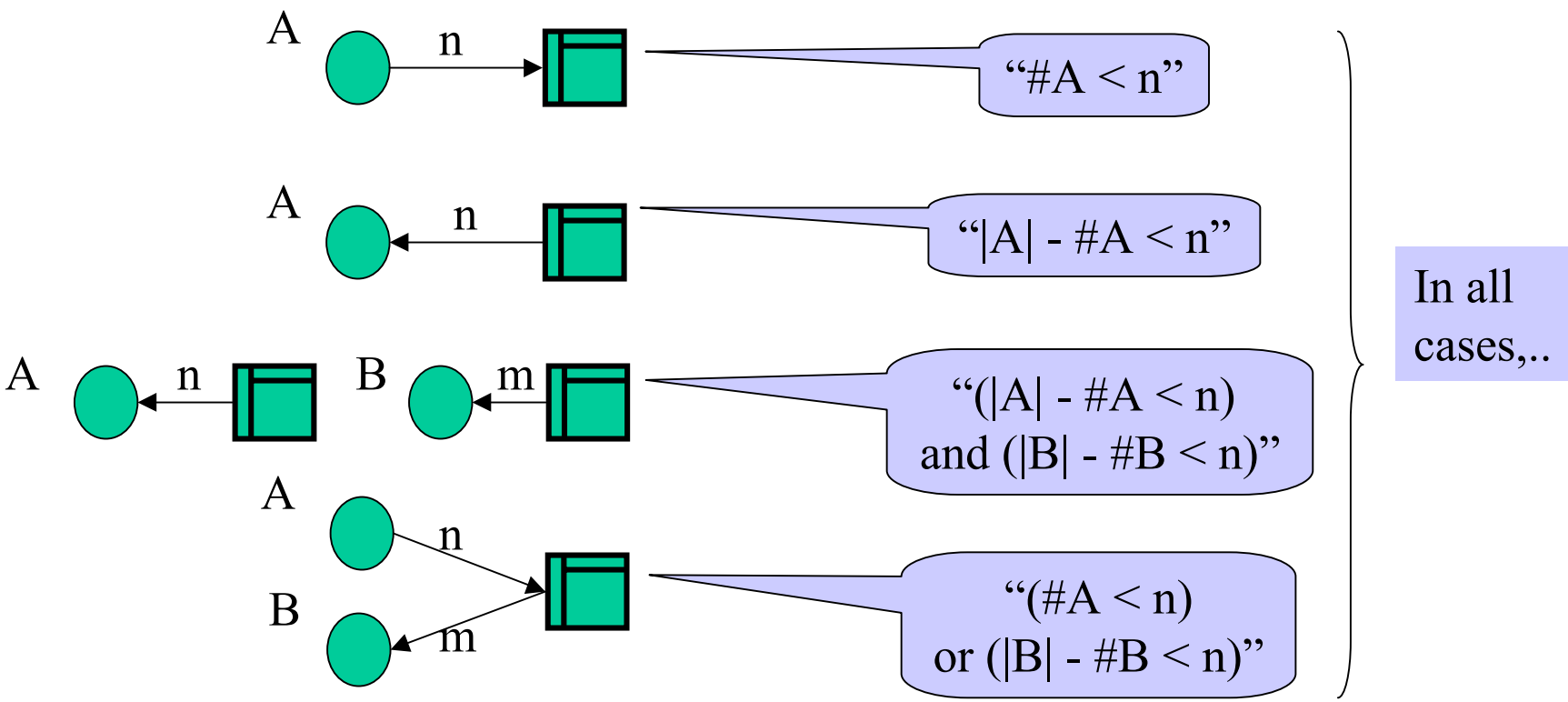


Two lifts

two nets,
folded in one

Three passengers

Numerical invariants, logically



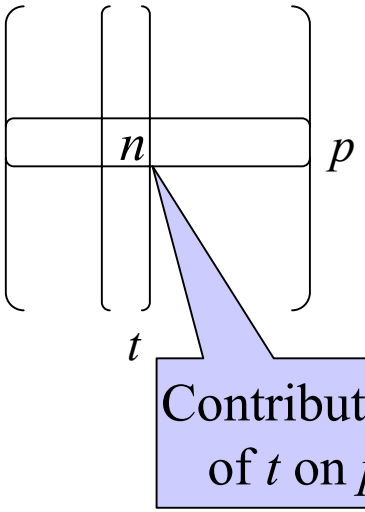
dead events (never enabled) ***system invariants (facts)***

- the logical approach can be carried over to place/transition-nets, using inequalities on the number of resources as elementary propositions
- but more specific, numerical techniques can be more effective

Numerical invariants, numerically

- incidence matrix C of a pure (no elementary loops) place/transition-net:

$$C_{p,t} := \begin{cases} -n, \text{ if there are } n \text{ arcs from } p \text{ to } t \\ +n, \text{ if there are } n \text{ arcs from } t \text{ to } p \\ 0, \text{ otherwise} \end{cases}$$



- marking vector m of a place/transition-net:

$$m_p := n, \text{if there are are } n \text{ token on place } p$$

- firing vector f of a multi-set of transition occurrences *(order not represented!)*:

$$f_t := n, \text{if transition } t \text{ fires } n \text{ times}$$

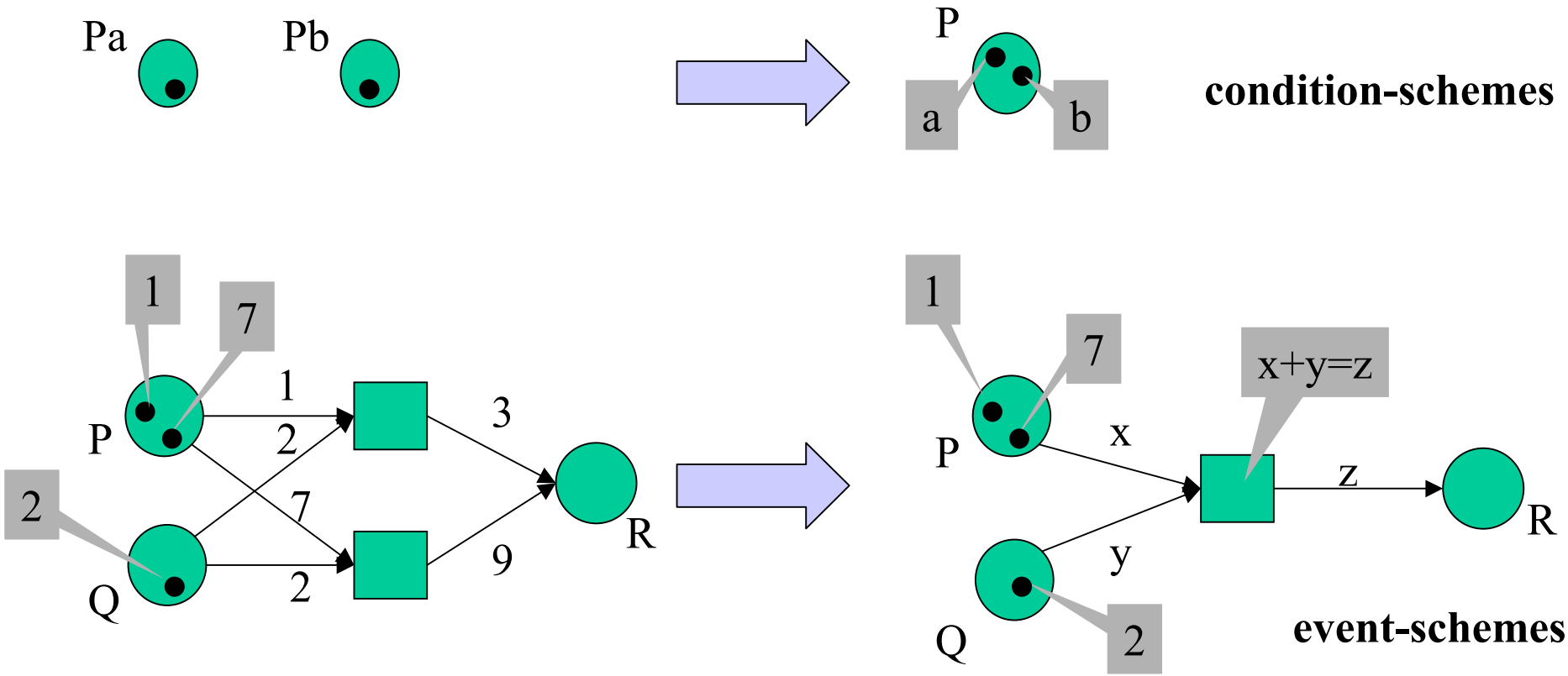
- weight vector i of a place invariant: set of places with constant (weighted) token sum

$$\forall m, m' : (m \mapsto m') \Rightarrow i^t \cdot m = \sum_p i_p * m_p = \sum_p i_p * m'_p = i^t \cdot m'$$

- necessary, though not sufficient condition for reachability: $m + C \cdot f^t = m'$

$$i^t \cdot m = i^t \cdot m' \Leftrightarrow i^t \cdot (m - m') = 0 \Rightarrow i^t \cdot C = 0$$

From propositions to predicates

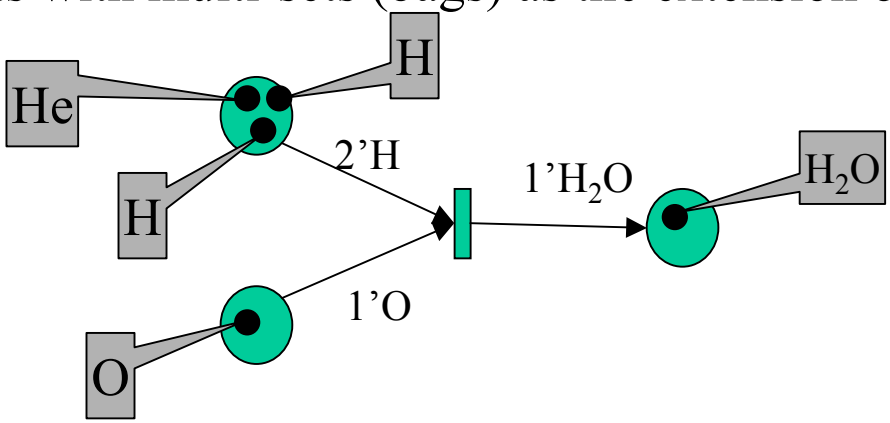


Predicate/transition-nets:

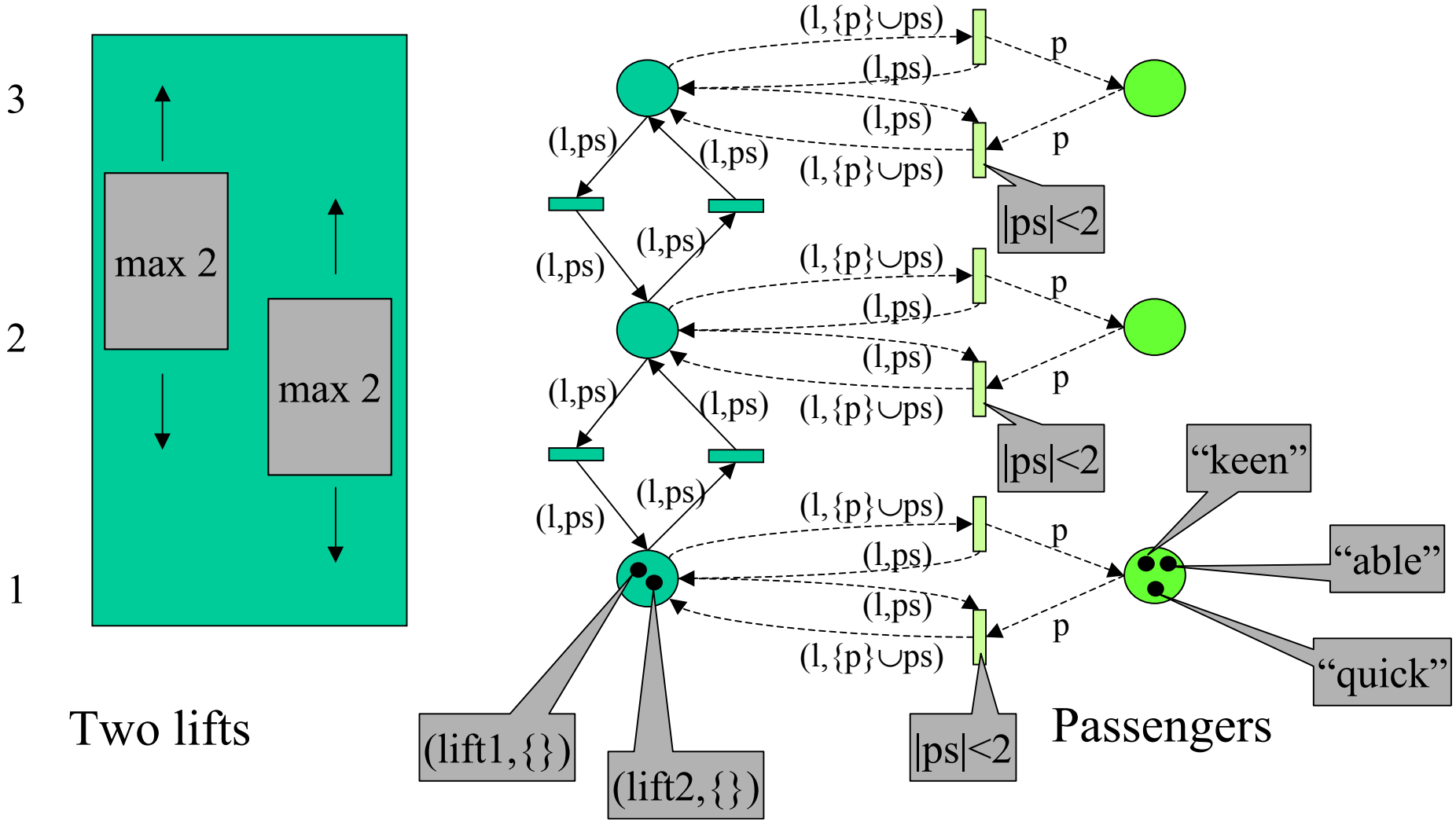
- individual tokens give the extension of predicates, which replace propositional conditions
- quantifiers and specifications in a predicate logic (+ some theory)
  permit the grouping of proposition-level events into predicate-level event-schemes

High-level Petri nets

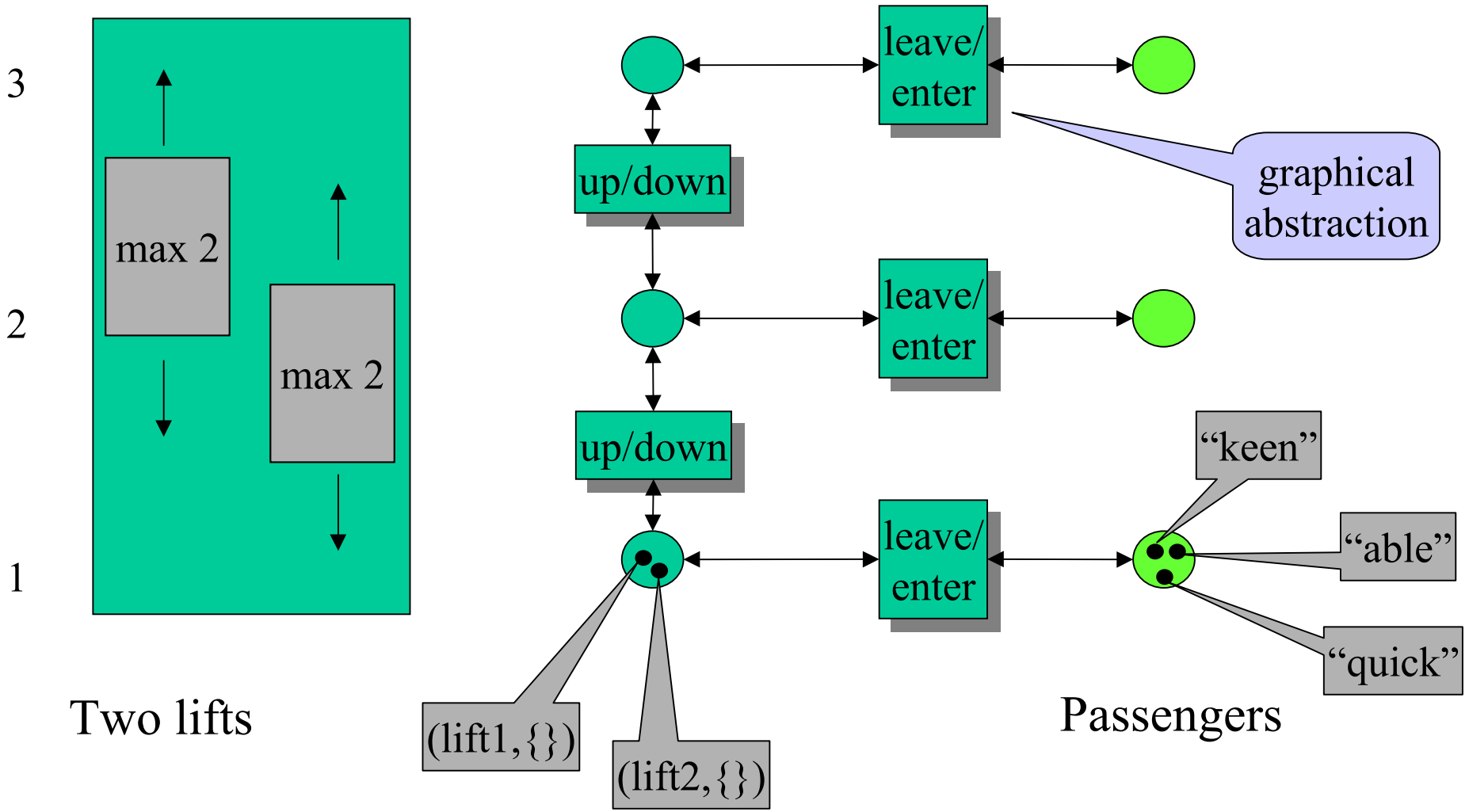
- proposed around 1980, various forms known, including predicate/transition-nets (PrT-nets), and coloured Petri nets (CPN)
- common features: individual tokens, net elements annotated in some textual inscription language (predicate logic, Standard ML, Java, ..), net semantics integrated with language semantics (variable bindings, guard predicates)
- integration of graphical specifications and textual inscriptions enable transformation between both within a single formalism
- multiplicity and individuality of tokens are orthogonal extensions:
  - for place/transition-nets, one might want to distinguish tokens
  - for PrT-nets, one might want indistinguishable tokens, replacing sets with multi-sets (bags) as the extension of predicates



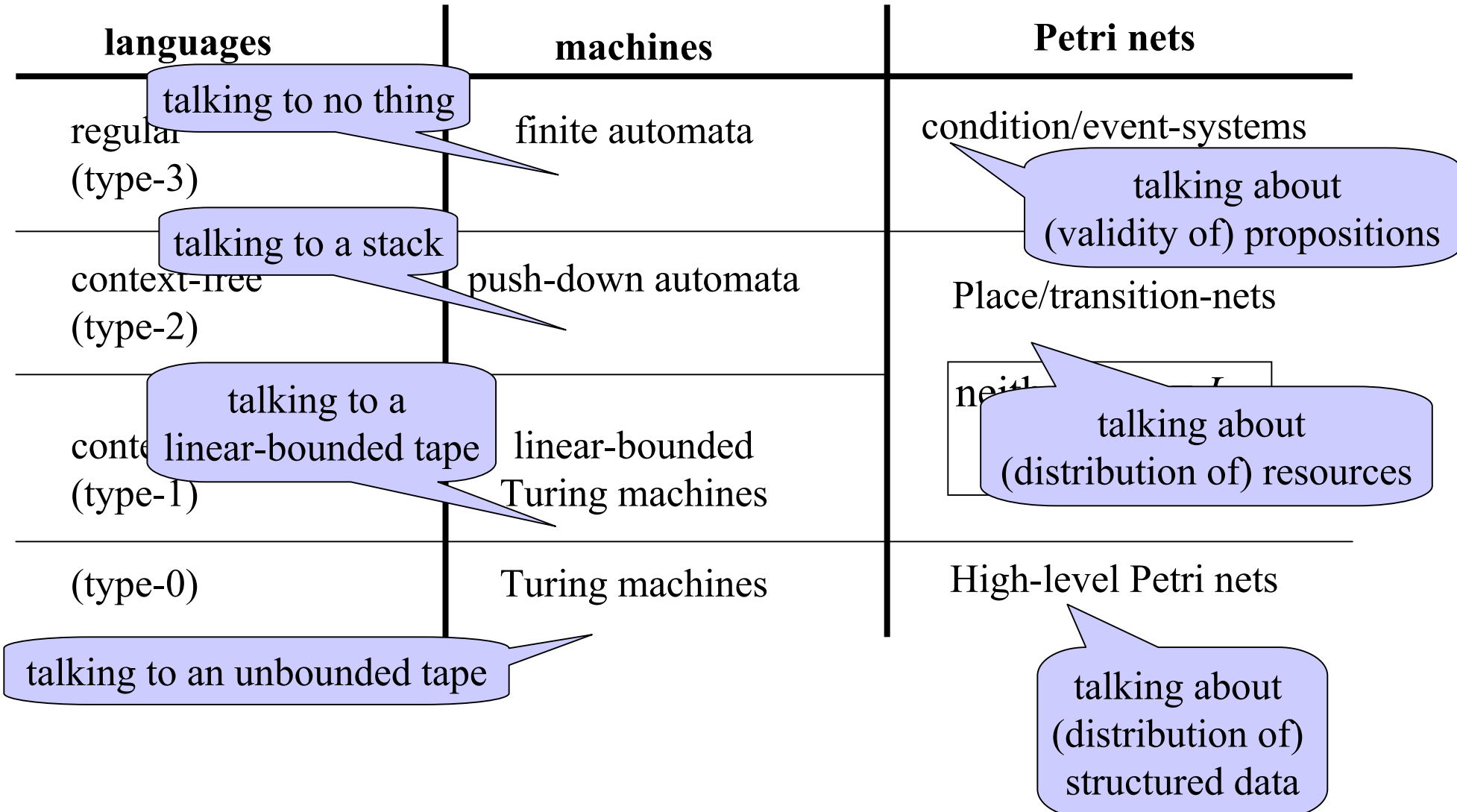
Lifts, yet again, with high-level nets



Lifts, yet again, with high-level nets



Chomsky hierarchy, machines and Petri nets
"Talking to" versus "Talking about"



The world of Petri Nets

- a way of thinking about concurrent and communicating systems
- straightforward generalisation of automata, but with concurrency in mind
- "pre-formal" graphical notation: intuitive use of formal methods
- good foundations in physics, engineering, logic, maths, theoretical CS
- the usual conflict between expressiveness and predictability/analysis
- wide range of applications, including communication networks in telecom and in computing, data-flow systems, workflow/office information systems,..
- still more work needed, both in domain-specific modifications and in foundations

Try them!-)

The world of Petri nets (events, bibliographies, research groups, tools,..)

http://www.daimi.aau.dk/PetriNets/

Carl Adam Petri's home page (cv, bibliography, laudatio)

http://www.informatik.uni-hamburg.de/TGI/mitarbeiter/profs/petri.html

References

The computer scientist as toolsmith II, Brooks F.P., Communications of the ACM 39(3):61-68, 1996
[ACM Allen Newell Award acceptance lecture]

Christopher Strachey's Oxford PRG motto: http://web.comlab.ox.ac.uk/oucl/about/philosophy.html
Carl Adam Petri's homepage: http://www.informatik.uni-hamburg.de/TGI/mitarbeiter/profs/petri_eng.html

Nets, Time and Space, C.A.Petri,Theoretical Computer Science, Special Volume on Petri Nets, 1996, No. 1-2, Vol. 153, Pages: 3-48
Concurrency and Continuity. Petri, C.A.; Smith, E., Lecture Notes in Computer Science Vol. 266:
Advances in Petri Nets 1987 / Rozenberg, G. (ed.) --- Springer Verlag, 1987, Pages: 273-292

The Calculus of Facts. Genrich, H.J.; Thieler-Mevissen, G. Mathematical Foundations of Computer Science 1976,
Mazurkiewicz, A.(ed.) --- Berlin, Heidelberg, New York: Springer-Verlag, 1976, Pages: 588-595

Hierarchies in Coloured Petri Nets. Huber, P.; Jensen, K.; Shapiro, R.M. Lecture Notes in Computer Science, Vol. 483;
Advances in Petri Nets 1990 / Rozenberg, G. (ed.) --- Berlin, Germany: Springer-Verlag, 1991, Pages: 313-341

Petri Net Theory and the Modeling of Systems J. L. Peterson, Prentice-Hall, N.J., 1981, ISBN: 0-13-661983-5

Petri Nets, An Introduction W. Reisig, EATCS, Monographs on Theoretical Computer Science, W.Brauer, G. Rozenberg, A. Salomaa (Eds.), Springer Verlag, Berlin, 1985.

Recommended Books and Papers on Coloured Petri Nets, Kurt Jensen**,** CPN Group**,** University of Aarhus, Denmark:
http://www.daimi.au.dk/~kjensen/papers_books/rec_papers_books.html

Further references (including introductory material): http://www.daimi.au.dk/PetriNets/bibl/