Petri Nets

Formal Methods

Lecture 9 (courtesy of Gabriel Eirea)

Farn Wang
Department of Electrical Engineering
National Taiwan University

Petri Nets

Reference:

- Tadao Murata. “Petri nets: Properties, Analysis and Applications.” Proc. of the IEEE, 77(4), 1989.
  - available on class website

Outline

- Introduction/History
- Transition enabling & firing
- Modeling examples
- Behavioral properties
- Analysis methods
- Liveness, safeness & reachability
- Analysis & synthesis of Marked Graphs
- Structural properties
- Modified Petri Nets

Introduction

- Petri Nets
  - concurrent, asynchronous, distributed, parallel, nondeterministic and/or stochastic systems
  - graphical tool
    - visual communication aid
  - mathematical tool
    - state equations, algebraic equations, etc
  - communication between theoreticians and practitioners

History

- **1962**: C.A. Petri's dissertation (U. Darmstadt, W. Germany)
- **1970**: Project MAC Conf. on Concurrent Systems and Parallel Computation (MIT, USA)
- **1975**: Conf. on Petri Nets and related Methods (MIT, USA)
- **1979**: Course on General Net Theory of Processes and Systems (Hamburg, W. Germany)
- **1980**: First European Workshop on Applications and Theory of Petri Nets (Strasbourg, France)
- **1985**: First International Workshop on Timed Petri Nets (Torino, Italy)

Applications

- **performance evaluation**
- **communication protocols**
- distributed-software systems
- distributed-database systems
- concurrent and parallel programs
- industrial control systems
- discrete-events systems
- multiprocessor memory systems
- dataflow-computing systems
- fault-tolerant systems
- etc, etc, etc

Definition

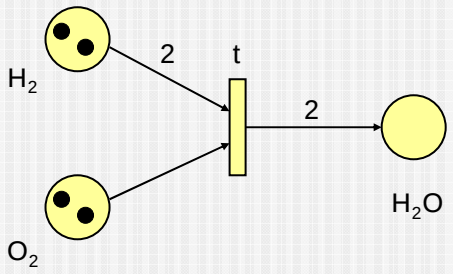
- Directed, weighted, bipartite graph
  - places
  - transitions
  - arcs (places to transitions or transitions to places)
  - weights associated with each arc
- Initial marking
  - assigns a non-negative integer to each place

Transition (firing) rule

- A transition t is enabled if each input place p has at least w(p,t) tokens
- An enabled transition may or may not fire
- A firing on an enabled transition t removes w(p,t) from each input place p, and adds w(t,p') to each output place p'

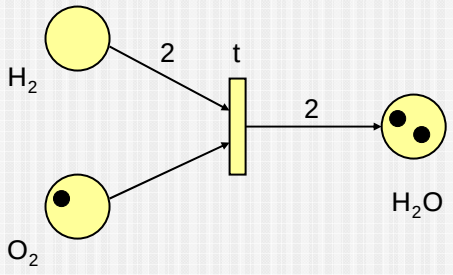
Firing example

$2H_2 + O_2 \rightarrow 2H_2O$



Firing example

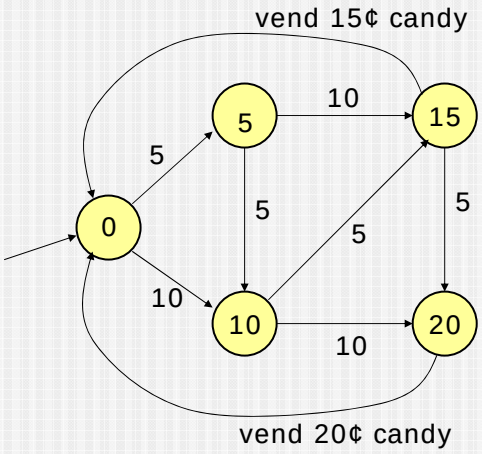
$2H_2 + O_2 \rightarrow 2H_2O$



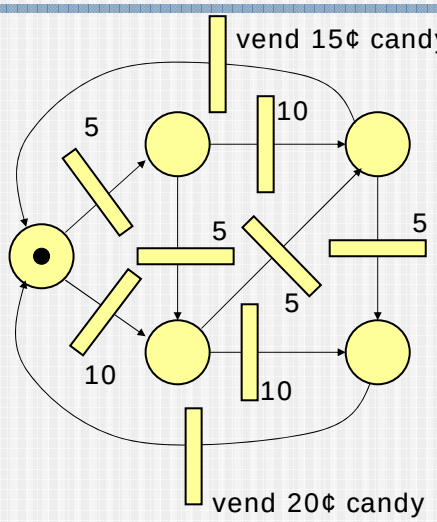
Some definitions

- **source transition**: no inputs
- **sink transition**: no outputs
- **self-loop**: a pair (p,t) s.t. p is both an input and an output of t
- **pure PN**: no self-loops
- **ordinary PN**: all arc weights are 1's
- **infinite capacity net**: places can accommodate an unlimited number of tokens
- **finite capacity net**: each place p has a maximum capacity K(p)
- **strict transition rule**: after firing, each output place can't have more than K(p) tokens
- **Theorem**: every pure finite-capacity net can be transformed into an equivalent infinite-capacity net

Modeling FSMs

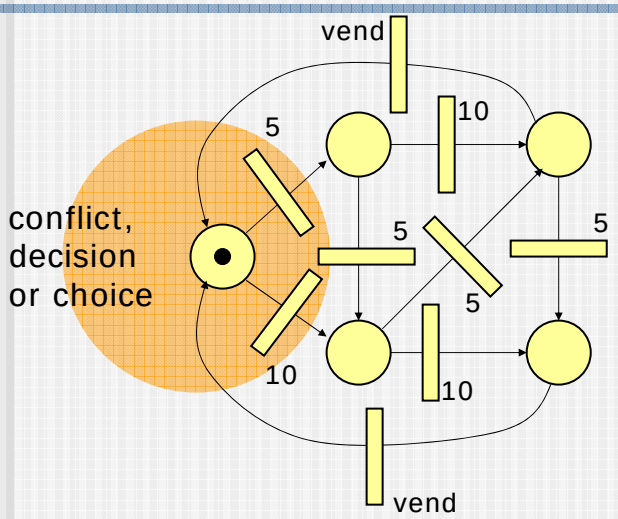


Modeling FSMs



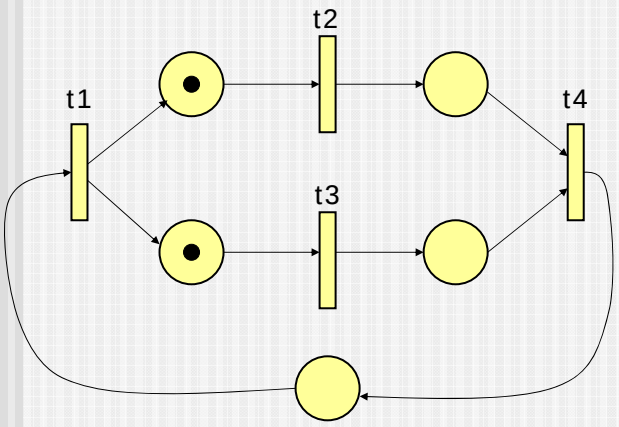
state machines:
each transition
has exactly
one input and
one output

Modeling FSMs



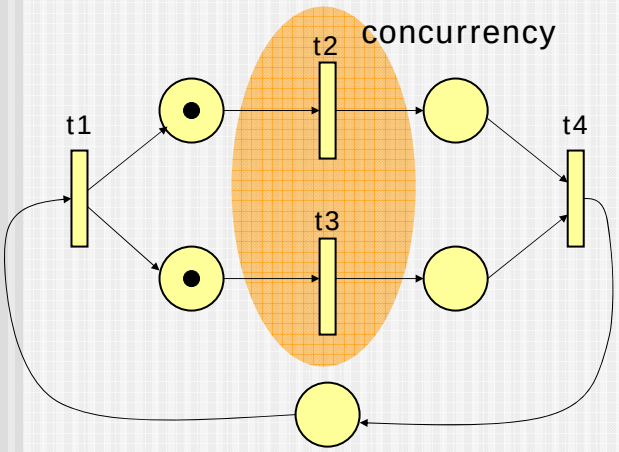
conflict,
decision
or choice

Modeling concurrency



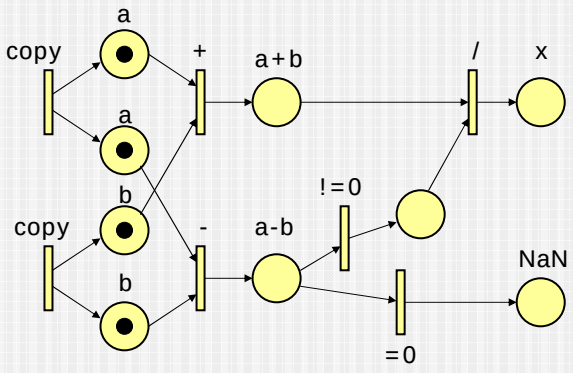
marked graph: each place has exactly one incoming arc and one outgoing arc.

Modeling concurrency

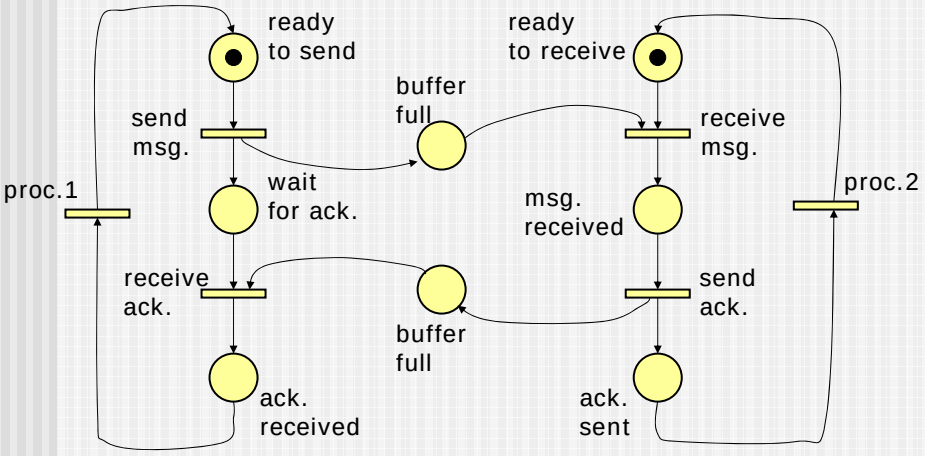


Modeling dataflow computation

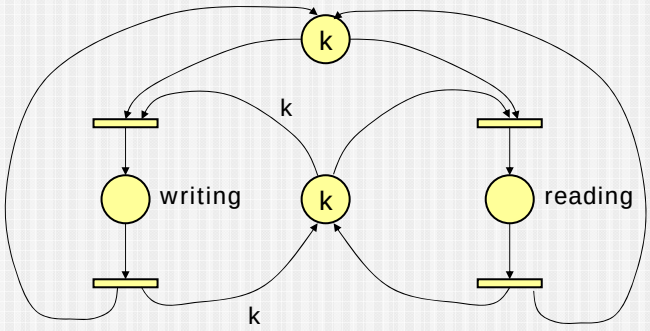
x = (a+b)/(a-b)



Modeling communication protocols



Modeling synchronization control



Behavioral properties (1)

- Properties that depend on the initial marking
- Reachability
  - Mn is reachable from M0 if exists a sequence of firings that transform M0 into Mn
  - reachability is decidable, but exponential
- Boundedness
  - a PN is bounded if the number of tokens in each place doesn't exceed a finite number k for any marking reachable from M0
  - a PN is safe if it is 1-bounded

Behavioral properties (2)

- Liveness
  - a PN is live if, no matter what marking has been reached, it is possible to fire any transition with an appropriate firing sequence
  - equivalent to deadlock-free
  - strong property, different levels of liveness are defined (L0=dead, L1, L2, L3 and L4=live)
- Reversibility
  - a PN is reversible if, for each marking M reachable from M0, M0 is reachable from M
  - relaxed condition: a marking M' is a home state if, for each marking M reachable from M0, M' is reachable from M

Behavioral properties (3)

- Coverability
  - a marking is coverable if exists M' reachable from M0 s.t. M'(p)>=M(p) for all places p
- Persistence
  - a PN is persistent if, for any two enabled transitions, the firing of one of them will not disable the other
  - then, once a transition is enabled, it remains enabled until it's fired
  - all marked graphs are persistent
  - a safe persistent PN can be transformed into a marked graph

Behavioral properties (4)

- Synchronic distance
  - maximum difference of times two transitions are fired for any firing sequence

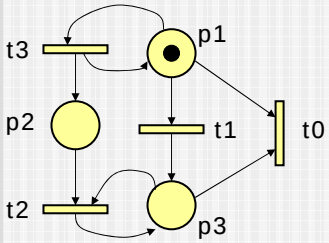
$$d_{12} = \max_{\sigma} \left| \overline{\sigma}(t_1) - \overline{\sigma}(t_2) \right|$$

  - well defined metric for condition/event nets and marked graphs
- Fairness
  - bounded-fairness: the number of times one transition can fire while the other is not firing is bounded
  - unconditional(global)-fairness: every transition appears infinitely often in a firing sequence

Analysis methods (1)

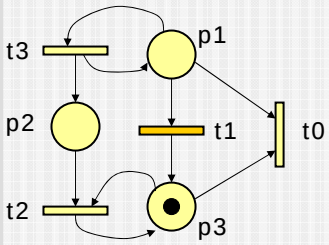
- Coverability tree
  - tree representation of all possible markings
    - root = M0
    - nodes = markings reachable from M0
    - arcs = transition firings
  - if net is unbounded, then tree is kept finite by introducing the symbol ω
  - Properties
    - a PN is bounded iff ω doesn't appear in any node
    - a PN is safe iff only 0's and 1's appear in nodes
    - a transition is dead iff it doesn't appear in any arc
    - if M is reachable form M0, then exists a node M' that covers M

Coverability tree example



M0=(100)

Coverability tree example

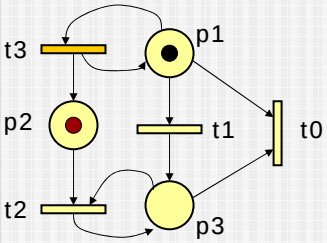


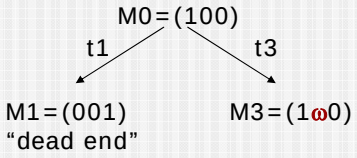
M0=(100)

t1

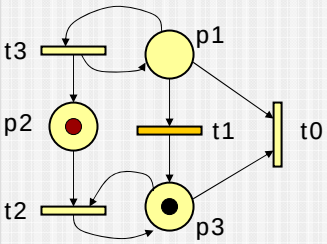
M1=(001)
“dead end”

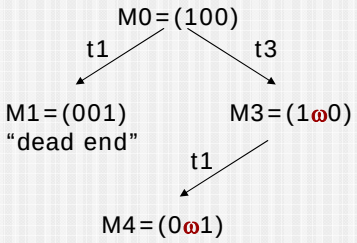
Coverability tree example



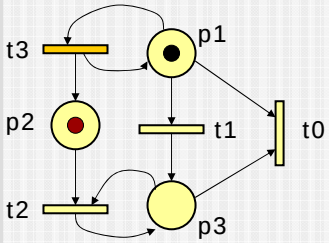


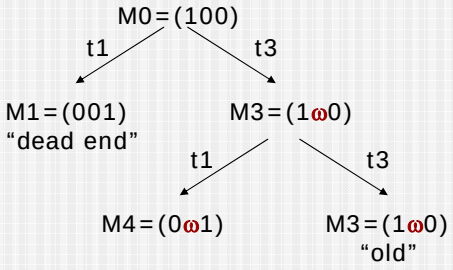
Coverability tree example



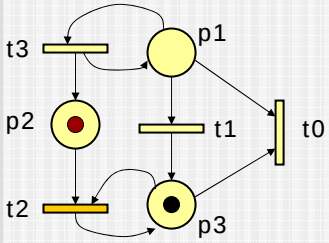


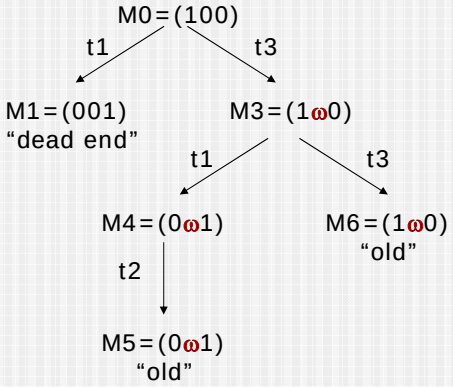
Coverability tree example



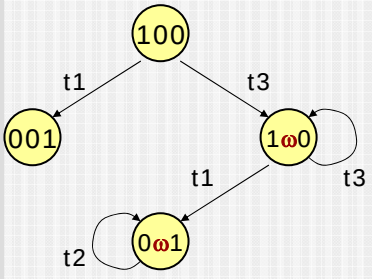


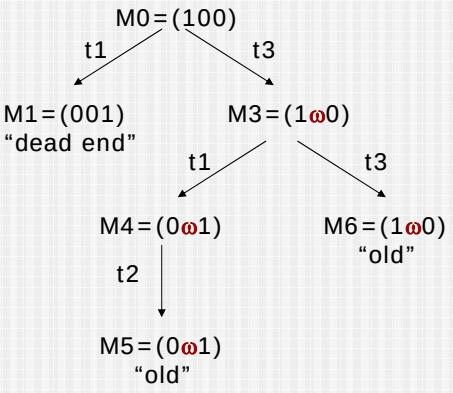
Coverability tree example





Coverability tree example





coverability graph **coverability tree**

Analysis methods (2)

- Incidence matrix
  - n transitions, m places, A is n x m
  - $a_{ij} = a_{ij}^{+} - a_{ij}^{-}$
  - a_{ij} is the number of tokens changed in place j when transition i fires once
- State equation
  - $M_k = M_{k-1} + A^T u_k$
  - $u_k = e_i$ unit vector indicating transition i fires

Necessary reachability condition

- M_d reachable from M_0, then

  $M_d = M_0 + A^T (u_1+u_2+\ldots+u_d)$

  $A^T x = \Delta M$

  then

  $\Delta M \in \text{range}(A^T)$

  $\Delta M \perp \text{null}(A)$

  $B_f \Delta M = 0$

  where the rows of B_f span null(A)

Analysis methods (3)

- Reduction rules that preserve liveness, safeness and boundedness
  - Fusion of Series Places
  - Fusion of Series Transitions
  - Fusion of Parallel Places
  - Fusion of Parallel Transitions
  - Elimination of Self-loop Places
  - Elimination of Self-loop Transitions
- Help to cope with the complexity problem

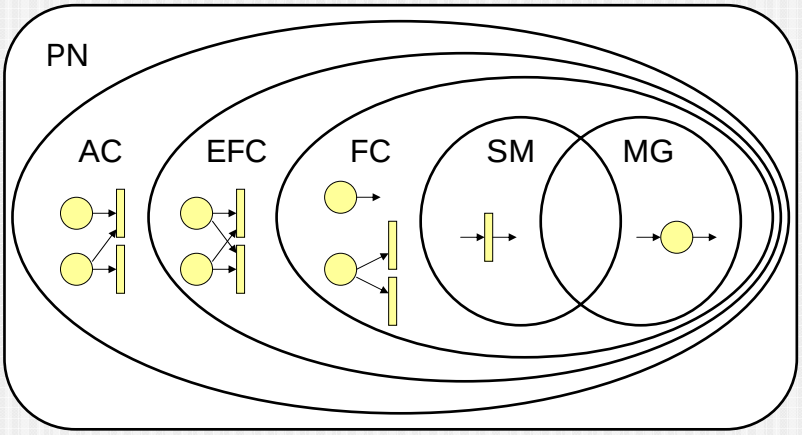
Subclasses of Petri Nets (1)

- Ordinary PNs
  - all arc weights are 1's
  - same modeling power as general PN, more convenient for analysis but less efficient
- State machine
  - each transition has exactly one input place and exactly one output place
- Marked graph
  - each place has exactly one input transition and exactly one output transition

Subclasses of Petri Nets (2)

- Free-choice
  - every outgoing arc from a place is either unique or is a unique incoming arc to a transition
- Extended free-choice
  - if two places have some common output transition, then they have all their output transitions in common
- Asymmetric choice (or simple)
  - if two places have some common output transition, then one of them has all the output transitions of the other (and possibly more)

Subclasses of Petri Nets (3)



Liveness and Safeness Criteria (1)

- general PN
  - if a PN is live and safe, then there are no source or sink places and source or sink transitions
  - if a connected PN is live and safe, then the net is strongly connected
- SM
  - a SM is live iff the net is strongly connected and M0 has at least one token
  - a SM is safe iff M0 has at most one token

Liveness and Safeness Criteria (2)

- MG
  - a MG is equivalent to a marked directed graph (arcs=places, nodes=transitions)
  - a MG is live iff M0 places at least one token on each directed circuit in the marked directed graph
  - a live MG is safe iff every place belongs to a directed circuit on which M0 places exactly one token
  - there exists a live and safe marking in a directed graph iff it is strongly connected

Liveness and Safeness Criteria (3)

- siphon S
  - every transition having an output place in S has an input place in S
  - if S is token-free under some marking, it remains token-free under its successors
- trap Q
  - every transition having an input place in Q has an output place in Q
  - if Q is marked under some marking, it remains marked under its successors

Liveness and Safeness Criteria (4)

- FC
  - a FC is live iff every siphon contains a marked trap
  - a live FC is safe iff it is covered by strongly-connected SM components, each of which has exactly one token at M0
  - a safe and live FC is covered by strongly-connected MG components
- AC
  - an AC is live if every siphon contains a marked trap

Reachability Criteria (1)

- acyclic PN
  - has no directed circuits
  - in an acyclic PN, Md is reachable from M0 iff exists a non negative integer solution to $A^T x = \Delta M$
- trap(siphon)-circuit net or TC (SC)
  - the set of places in every directed circuit is a trap(siphon)
  - in a TC (SC), Md is reachable from M0 iff (i) exists a non negative integer solution to $A^T x = \Delta M$, and (ii) the subnet with transitions fired at least once in x has no token-free siphons (traps) under M0 (Md)

Reachability Criteria (2)

- TCC (SCC) net
  - there is a trap (siphon) in every directed circuit
  - in a TCC, Md is reachable from M0 if (i) exists a non negative integer solution to $A^T x = \Delta M$, and (ii) every siphon in the subnet with transitions fired at least once in x has a marked trap under M0
  - in a SCC, Md is reachable from M0 if (i) exists a non negative integer solution to $A^T x = \Delta M$, and (ii) there are no token-free traps under Md in the subnet with transitions fired at least once in x

Reachability Criteria (3)

- forward(backward)-conflict-free net or FCF(BCF)
  - each place has at most one outgoing (incoming) arc
- nondecreasing(nonincreasing)-circuit net or NDC(NIC)
  - the token content in any directed graph is never decreased (increased) by any transition firing
- MG $\subset$ FCF $\subset$ NDC $\subset$ TC $\subset$ TCC
- MG $\subset$ BCF $\subset$ NIC $\subset$ SC $\subset$ SCC

Analysis of MGs

- reachability
  - in a live MG, Md is reachable from M0 iff $B_f \Delta M = 0$
  - in a MG, Md is reachable from M0 iff $B_f \Delta M = 0$ and the transitions that are fired don't lie on a token-free directed circuit
  - in a connected MG, a firing sequence leads back to the initial marking M0 iff it fires every transition an equal number of times
  - any two markings on a MG are mutually reachable iff the corresponding directed graph is a tree

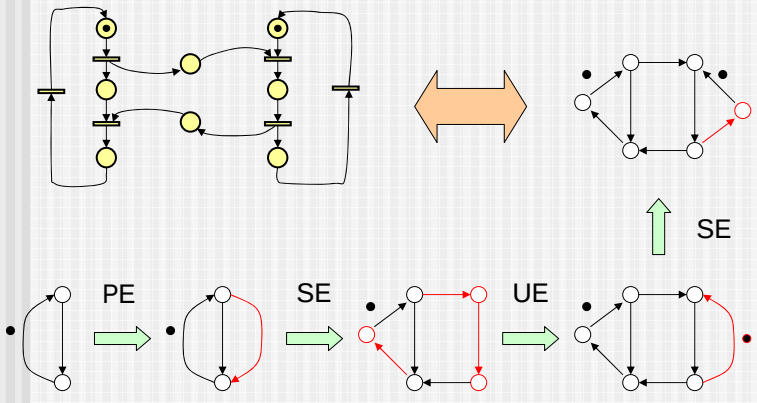
Synthesis of LSMGs (1)

- equivalence relation
  - M0~Md if Md is reachable from M0
  - $\rho(G)$ = number of equivalence classes of live-safe markings for a strongly connected graph G
  - we are interested in $\rho(G)=1$ (i.e., all markings are mutually reachable)
  - $\rho(G)=1$ iff there is a marking of G which places exactly one token on every directed circuit in G

Synthesis of LSMGs (2)

- $\rho(G)$ is invariant under operations
  - series expansion
  - parallel expansion
  - unique circuit expansion
  - V-Y expansion
  - separable graph expansion
- synthesis process can prescribe
  - liveness
  - safeness
  - mutual reachability
  - minimum cycle time
  - resource requirements

Synthesis of LSMGs (3)



Other synthesis issues (1)

- weighted sum of tokens
  - we are interested in finding the maximum and minimum weighted sum of tokens for all reachable markings
  - $\max \{M^T W \mid M \in R(M_0)\} =$ $\min \{M_0^T I \mid I \geq W, AI = 0\}$
  - $\min \{M^T W \mid M \in R(M_0)\} =$ $\max \{M_0^T I \mid I \leq W, AI = 0\}$

Other synthesis issues (2)

- token distance matrix T
  - tij is the minimum token content among all possible directed paths from i to j
  - useful to determine
    - firability (off-diagonal elements in a column >0)
    - necessity of firing (off-diagonal 0 entries)
    - synchronic distance (dij=tij+tji)
    - liveness
    - shortest firing sequence to enable a node(algorithm)
- maximum concurrency
  - algorithm to find a maximum set of nodes that can be fired concurrently at some marking

Other synthesis issues (3)

- Synchronic distance matrix D
  - $D = T + T^T$
  - D*D=D under Carre's algebra
  - given D, find a MG whose synchronic distance matrix is D
    - test distance condition
    - construct a tree
      - select nodes i0 with maximum distance
      - draw arcs to nodes jr with minimum distance to nodes i0
      - repeat until all arcs are drawn
    - replace each arc in the tree by a pair of oppositely directed arcs

Structural properties (1)

- properties that don't depend on the initial marking
- structural liveness
  - there exists a live initial marking
  - all MG are structurally live
  - a FC is structurally live iff every siphon has a trap
- controllability
  - any marking is reachable from any other marking
  - necessary condition: rank(A)=#places
  - for MG, it is also sufficient

Structural properties (2)

- structural boundedness
  - bounded for any finite initial marking
  - iff exists a vector y of positive integers s.t. $Ay \leq 0$
- (partial) conservativeness
  - a weighted sum of tokens is constant for every (some) place
  - iff exists a vector y of positive (nonnegative) integers s.t. $Ay = 0$

Structural properties (3)

- (partial) repetitiveness
  - every (some) transition occurs infinitely often for some initial marking and firing sequence
  - iff exists a vector x of positive (nonegative) integers s.t. $A^T x \geq 0$
- (partial) consistency
  - every (some) transition occurs at least once in some firing sequence that drives some initial marking back to itself
  - iff exists a vector x of positive (nonegative) integers s.t. $A^T x = 0$

Timed nets

- deterministic time delays introduced for transitions and/or places
- cycle time
  - assuming the net is consistent, τ is the time to complete a firing sequence leading back to the starting marking
  - delays in transitions
    - $\tau_{min} = \max\{y_k^T(A^-)^T D x / y_k^T M_0\}$
  - delays in places
    - $\tau_{min} = \max\{y_k^T D (A^+)^T x / y_k^T M_0\}$
  - timed MG
    - τ_{min} = max{total delay in C_k/M_0 (C_k)}

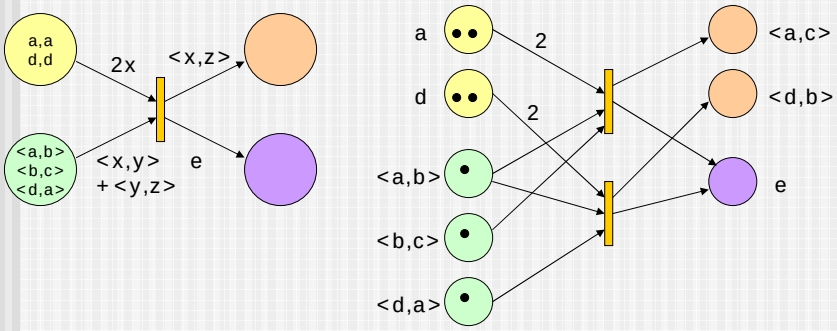
Stochastic nets

- exponentially distributed r.v. models the time delays in transitions
- the reachability graph of a bounded SPN is isomorphic to a finite Markov chain
- a reversible SPN generates an ergodic MC
  - steady-state probability distribution gives performance estimates
    - probability of a particular condition
    - expected value of the number of tokens
    - mean number of firings in unit time
- generalized SPN adds immediate transitions to reduce state space

High-level nets (1)

- they include
  - predicate/transition nets
  - colored PN
  - nets with individual tokens
- a HL net can be unfolded into a regular PN
  - each place unfolds into a set of places, one for each color of tokens it can hold
  - each transition unfolds into a set of transitions, one for each way it may fire

High-level nets (2)



High-level nets (3)

- logic program
  - set of Horn clauses
    $B \leftarrow A_1, A_2, \ldots, A_n$
    where A_i's and B are atomic formulae
    Predicate(arguments)
  - goal statement = sink transition
  - assertion of facts = source transition
  - can be represented by a high-level net
    - each clause is a transition
    - each distinct predicate symbol is a place
    - weights are arguments
  - sufficient conditions for firing the goal transition

Conclusions

- PNs have a rich body of knowledge
- PNs are applied succesfully to a broad range of problems
- analysis and synthesis results are available for subclasses of PNs
- there are several extensions of PNs
- much work remains to be done