The decidability frontier for Petri nets

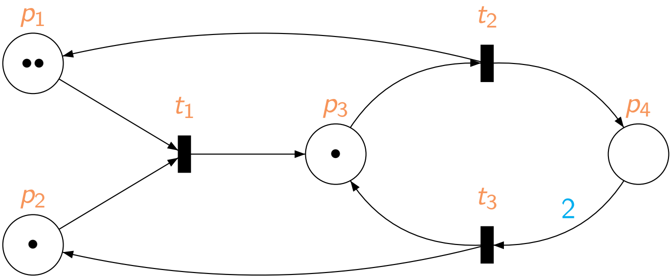
Madhavan Mukund

Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

Formal Methods Update Meeting
VIT Vellore
12 July 2011

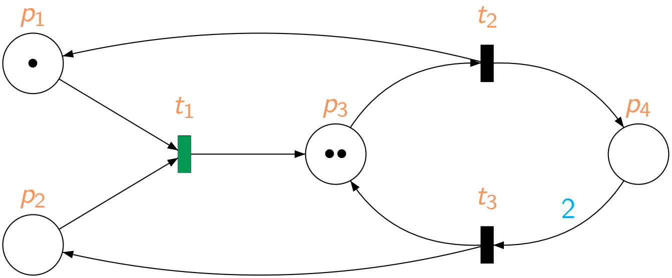
Petri nets

- A set P of places
- A set T of transitions
- Flow relation $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}_0$
- Initial marking $M_0 : P \rightarrow \mathbb{N}_0$
- Dynamics: "Token game"



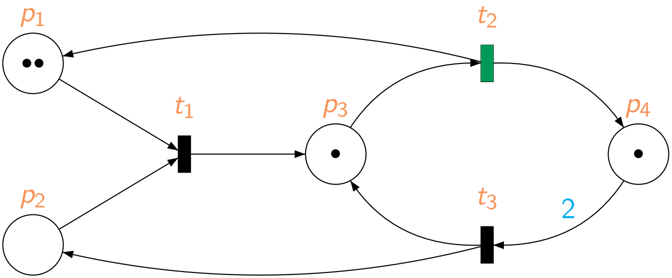
Petri nets

- A set P of places
- A set T of transitions
- Flow relation $F : (P \times T) \cup (T \times P) \to \mathbb{N}_0$
- Initial marking $M_0 : P \to \mathbb{N}_0$
- Dynamics: "Token game"



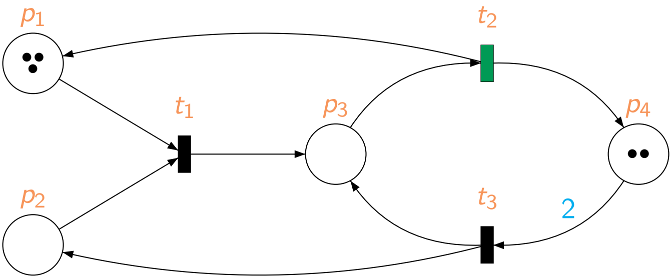
Petri nets

- A set P of places
- A set T of transitions
- Flow relation $F : (P \times T) \cup (T \times P) \to \mathbb{N}_0$
- Initial marking $M_0 : P \to \mathbb{N}_0$
- Dynamics: "Token game"



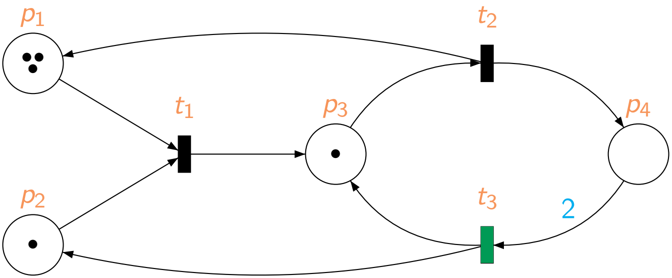
Petri nets

- A set P of places
- A set T of transitions
- Flow relation $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}_0$
- Initial marking $M_0 : P \rightarrow \mathbb{N}_0$
- Dynamics: "Token game"



Petri nets

- A set P of places
- A set T of transitions
- Flow relation $F : (P \times T) \cup (T \times P) \to \mathbb{N}_0$
- Initial marking $M_0 : P \to \mathbb{N}_0$
- Dynamics: "Token game"



Decision questions

- Reachability
  Is a marking M (exactly) reachable from M_0?
- Coverability
  Is a marking M coverable from M_0?
  - Can we reach M' such that for each p, $M'(p) \geq M(p)$
- Termination
  Is there an infinite execution?
- Boundedness
  Is the set of reachable markings finite
  - Is there a bound B such that no place has more than B tokens in any reachable marking?
- Place-boundedness
  For a given place p, is the number of tokens on p bounded in all reachable markings?

Decision questions . . .

- All these questions are decidable for “normal” Petri nets
  - Some proofs are easy (boundedness), others less so (reachability)
  - Classifying the computational complexity is a separate issue that we will not discuss

Decision questions . . .

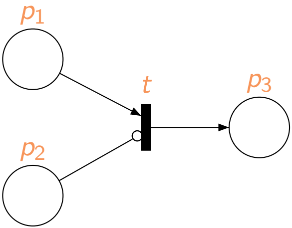
Inhibitor arcs

- Petri net places are like counters without test-for-zero

Decision questions . . .

Inhibitor arcs

- Petri net places are like counters without test-for-zero
- If we can check for the absence of tokens, everything becomes undecidable

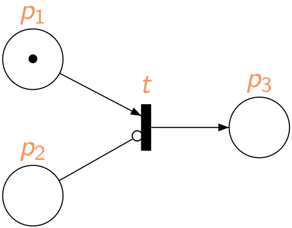


- t is enabled at M only if $M(p_1) > 0$ and $M(p_2) = 0$
- Two inhibitor arcs can simulate 2 counter machine

Decision questions . . .

Inhibitor arcs

- Petri net places are like counters without test-for-zero
- If we can check for the absence of tokens, everything becomes undecidable

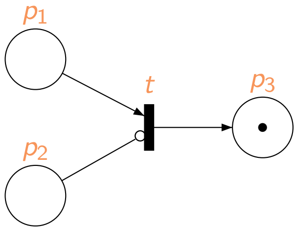


- t is enabled at M only if $M(p_1) > 0$ and $M(p_2) = 0$
- Two inhibitor arcs can simulate 2 counter machine

Decision questions ...

Inhibitor arcs

- Petri net places are like counters without test-for-zero
- If we can check for the absence of tokens, everything becomes undecidable



- t is enabled at M only if $M(p_1) > 0$ and $M(p_2) = 0$
- Two inhibitor arcs can simulate 2 counter machine

Boundedness

Karp-Miller reachability tree

- Start with the initial marking M_0
- Use BFS through space of reachable markings
  - Let M be a leaf node with t enabled at M such that $M \xrightarrow{t} M'$
  - Add M' as a new leaf if it does not already appear on the path from M_0 to M
  - Acceleration

    If $M' > M''$ for some marking on the path from M_0 to M, set $M'(p) = \omega$ wherever $M'(p) > M''(p)$

Dickson's lemma

A marking M over k places is a vector over $\mathbb{N}^k$

Given any infinite sequence of markings $M_1, M_2, \ldots$, there must exist positions i and j such that $i < j$ and $M_i \leq M_j$

Dickson's lemma

A marking M over k places is a vector over $\mathbb{N}^k$

Given any infinite sequence of markings $M_1, M_2, \ldots$, there must exist positions i and j such that $i < j$ and $M_i \leq M_j$

- Cannot have an infinite set of incomparable markings

The Karp-Miller tree

Boundedness and termination are decidable

The Karp-Miller tree is always finite, by Dickson's Lemma.

The given net is bounded iff ω does not appear in the tree.

The given net terminates if we can always expand all transitions fully in the tree.

- Never repeat a marking on any path
- Never apply acceleration

The Karp-Miller tree

Boundedness and termination are decidable

The Karp-Miller tree is always finite, by Dickson's Lemma.

The given net is bounded iff ω does not appear in the tree.

The given net terminates if we can always expand all transitions fully in the tree.

- Never repeat a marking on any path
- Never apply acceleration

The Karp-Miller tree, in fact, decides place-boundedness

Coverability

- For a set of markings S, $Pred(S)$ is the set of markings from where we can reach S
- If S is upward-closed, so is $Pred(S)$
- Any upward closed set S has a finite set of minimal elements $\{s_1, s_2, \ldots, s_k\}$ such that $S = \uparrow\{s_1, s_2, \ldots, s_k\}$—finite basis for S
- The set of markings that cover M is upward closed
- Iteratively compute a finite basis for $Pred(\uparrow M)$

What makes Petri net properties decidable?

- A set of incomparable markings must be finite
- Firing rule is compatible with marking order:

$$\begin{array}{ccc} M & \xrightarrow{t} & M' \\ \wedge\vert & & \\ M_1 & & \end{array}$$

What makes Petri net properties decidable?

- A set of incomparable markings must be finite
- Firing rule is compatible with marking order:

$$\begin{array}{ccc} M & \xrightarrow{t} & M' \\ \wedge| & & \wedge| \\ M_1 & \xrightarrow{t} & M_1' \end{array}$$

  - In fact $(M_1 - M) = (M_1' - M')$
  - Thus, $M < M_1$ implies $M' < M_1'$ — strict monotonicity

Well structured transition systems

Well quasi-order (wqo)

- $(X, \preceq)$, $\preceq$ is reflexive and transitive
- Given any infinite sequence $x_1, x_2, \ldots$ over X, there must exist positions i and j such that $i < j$ and $x_i \preceq x_j$

Well structured transition systems

Well quasi-order (wqo)

- $(X, \preceq)$, $\preceq$ is reflexive and transitive
- Given any infinite sequence $x_1, x_2, \ldots$ over X, there must exist positions i and j such that $i < j$ and $x_i \preceq x_j$

Note that this also rules out infinite descending chains.

Well structured transition systems

Well quasi-order (wqo)

- $(X, \preceq)$, $\preceq$ is reflexive and transitive
- Given any infinite sequence $x_1, x_2, \ldots$ over X, there must exist positions i and j such that $i < j$ and $x_i \preceq x_j$

Note that this also rules out infinite descending chains.

$(X, \rightarrow)$ is a well structured transition system if there is exists a wqo $(X, \preceq)$ such that $\rightarrow$ is compatible with $\preceq$

$$\begin{array}{ccc} x & \rightarrow & x' \\ \curlyvee\!| & & \curlyvee\!| \\ x_1 & \rightarrow & x_1' \end{array}$$

Well structured transition systems . . .

Concrete decision procedures for Petri nets can be lifted to WSTSs

- Karp-Miller tree generalize to finite reachability tree
  - Can use this to decide termination

Well structured transition systems . . .

Concrete decision procedures for Petri nets can be lifted to WSTSs

- Karp-Miller tree generalize to finite reachability tree
  - Can use this to decide termination
  - For boundedness, we need strict monotonicity

$$\begin{array}{ccc} x & \rightarrow & x' \\ \curlyvee & & \curlyvee \\ x_1 & \rightarrow & x_1' \end{array}$$

Well structured transition systems . . .

Concrete decision procedures for Petri nets can be lifted to WSTSs

- Karp-Miller tree generalize to finite reachability tree
  - Can use this to decide termination
  - For boundedness, we need strict monotonicity

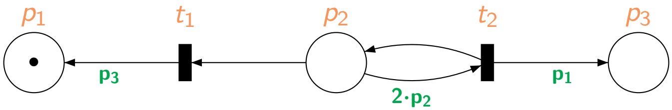
$$\begin{array}{ccc} x & \rightarrow & x' \\ \curlyvee & & \curlyvee \\ x_1 & \rightarrow & x'_1 \end{array}$$

- Backward saturation to compute coverability if the WSTS has an effective pred-basis

  Given a state $x \in X$, compute a finite basis for $Pred(\uparrow x)$

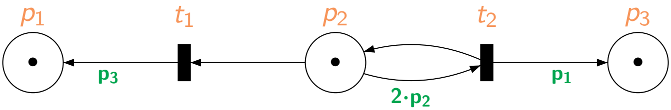
Generalized Petri nets

- Petri net with arc weights labelled by polynomials over places
  - Evaluate polynomial with respect to current marking
  - Resulting value determine whether a transition is enabled ...
  - ... and computes the effect of firing it.



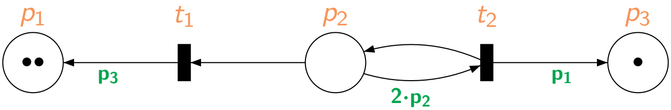
Generalized Petri nets

- Petri net with arc weights labelled by polynomials over places
  - Evaluate polynomial with respect to current marking
  - Resulting value determine whether a transition is enabled ...
  - ...and computes the effect of firing it.



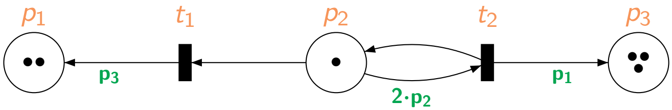
Generalized Petri nets

- Petri net with arc weights labelled by polynomials over places
  - Evaluate polynomial with respect to current marking
  - Resulting value determine whether a transition is enabled ...
  - ... and computes the effect of firing it.



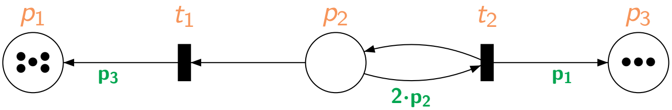
Generalized Petri nets

- Petri net with arc weights labelled by polynomials over places
  - Evaluate polynomial with respect to current marking
  - Resulting value determine whether a transition is enabled . . .
  - . . . and computes the effect of firing it.



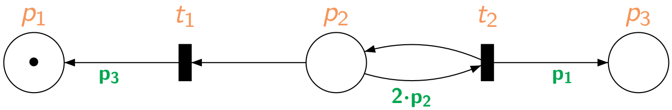
Generalized Petri nets

- Petri net with arc weights labelled by polynomials over places
  - Evaluate polynomial with respect to current marking
  - Resulting value determine whether a transition is enabled ...
  - ...and computes the effect of firing it.



Generalized Petri nets

- Petri net with arc weights labelled by polynomials over places
  - Evaluate polynomial with respect to current marking
  - Resulting value determine whether a transition is enabled ...
  - ... and computes the effect of firing it.



- Fibonacci net

  For odd k, marking $m_k = (fib(k+1), 0, fib(k))$

Generalizaed nets . . .

- All problems are undecidable in general
  - Subsume inhibitor arcs
  - To fire t_1, we need $2 \cdot M(p2)$ tokens at p_2
  - $M(p_2)$ must be 0!
- Subclasses clearly separate decision boundaries for reachability, coverability, termination, boundedness, place boundedness,

Decision problems for reset post-G nets

- Reset arc: $W(p, t) = p$
  - Resets (i.e., empties) input place p when t fires
- Transfer arc: $W(p, t) = p = W(t, p')$
  - Transfers contents of p to p'
- Post-G net: only output arcs are non-classical
- Double Petri net: Post G-net where $F(t, p) = p$ or $F(t, p) \in \mathbb{N}$.

  $F(t, p) = p$: doubling arc: doubles the marking of p

Reset Post-G nets

- Input arcs are either reset or classical
- Output arcs can have arbitrary polynomials

Reset Post-G nets

- Input arcs are either reset or classical
- Output arcs can have arbitrary polynomials

What's decidable

Coverability, termination

Reset post-G nets define WSTSs with effective pred-basis, but do not satisfy strict monotonicity.

Reset Post-G nets

- Input arcs are either reset or classical
- Output arcs can have arbitrary polynomials

What's decidable

Coverability, termination

Reset post-G nets define WSTSs with effective pred-basis, but do not satisfy strict monotonicity.

What's not

Boundedness

Reset post-G nets can "compute" polynomials. Complicated reduction from Hilbert's Tenth Problem.

Transfer nets

- All non-classical arcs are pairs that define transfers

Transfer nets

- All non-classical arcs are pairs that define transfers

What's decidable

Boundedness

Transfer nets define WSTSs with strict monotonicity.

Transfer nets

- All non-classical arcs are pairs that define transfers

What's decidable

Boundedness

Transfer nets define WSTSs with strict monotonicity.

What's not

Place-boundedness

- Simulate a reset post-G net N by a transfer net N'.
- Add a dummy place to N to get N'. Simulate resets by transferring tokens to this dummy place.
- N is unbounded iff some place other than the dummy place is unbounded in N'.

Post-G nets

- Input arcs are classical, only output arcs have extended weights

What's decidable

Place-boundedness

Post-G nets define WSTSs with strict monotonicity and an additional continuity condition required to compute place boundedness from the finite reachability tree.

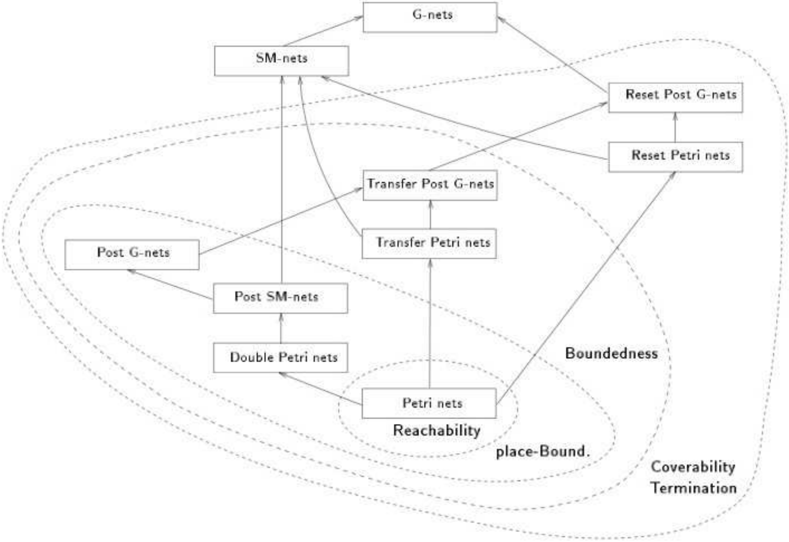
Reachability

Undecidability

Reachability is undecidable for double Petri nets, reset Petri nets and transfer Petri nets with two extended arcs.

Two extended arcs can simulate nets with inhibitor arcs.

What's decidable?



References

C. Dufourd, A. Finkel and P. Schnoebelen
Reset nets between decidability and undecidability
Proc ICALP, 1998

A. Finkel and P. Schnoebelen
Well-Structured Transition Systems Everywhere!
Theoretical Computer Science, 2002