An SMT-based Approach to Fair Termination Analysis

Javier Esparza, **Philipp J. Meyer**

Technische Universität München

Fair Termination Analysis

- Fair termination: No non-fair infinite execution sequence σ.
- PSPACE-complete for boolean programs.

Fair Termination Analysis

- Fair termination: No non-fair infinite execution sequence σ.
- PSPACE-complete for boolean programs.

SMT-Based Approach

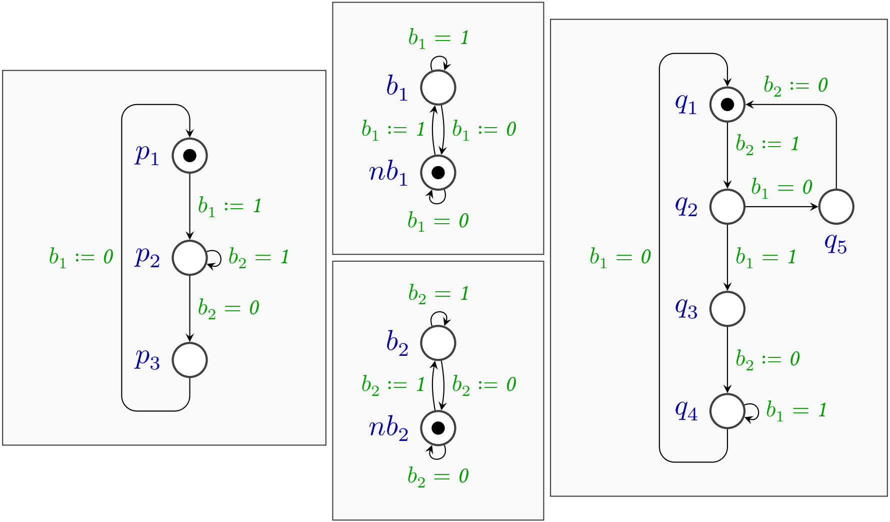
- Incomplete method based on reduction to feasibility of linear arithmetic constraints.
- Strengthened with refinement cycle which adds mixed linear and boolean constraints.
- Similar method previously applied for safety properties (An SMT-based Approach to Coverability Analysis, CAV14).

Lamport's 1-bit Algorithm for Mutual Exclusion

```
    procedure Process 1
    begin
      b1 := 0
      while true do
p1:     b1 := 1
p2:     while b2 = 1 do skip od
p3:     (* critical section *)
        b1 := 0
      od
    end
```

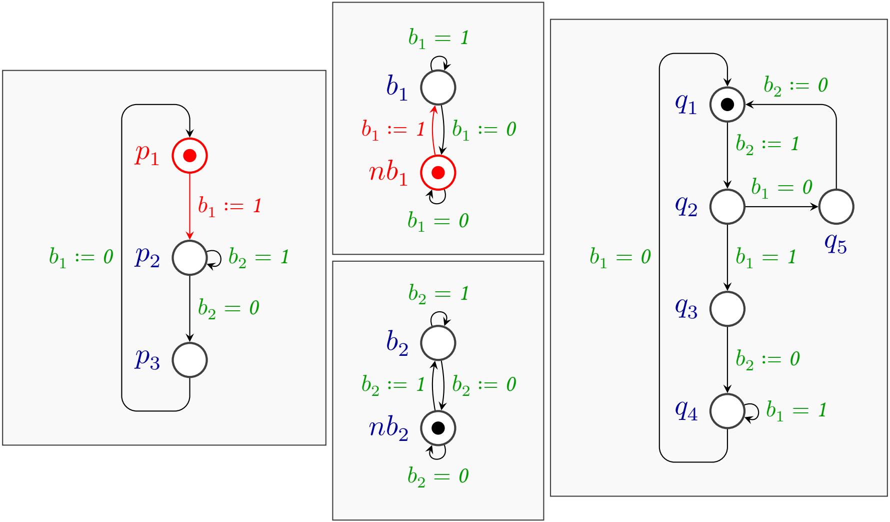
```
    procedure Process 2
    begin
      b2 := 0
      while true do
q1:     b2 := 1
q2:     if b1 = 1 then
q3:       b2 := 0
q4:       while b1 = 1 do skip od
          goto q1
        fi
q5:     (* critical section *)
        b2 := 0
      od
    end
```

Communicating Automata Model



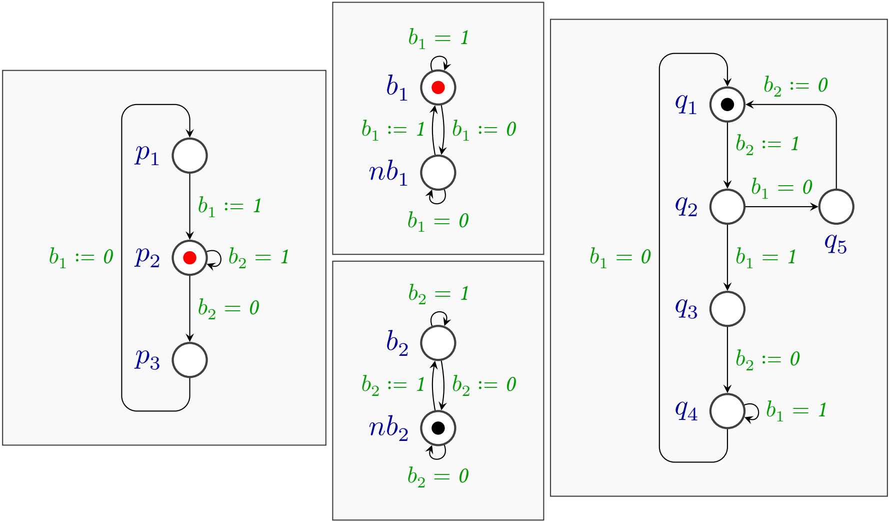
Property: If both processes are executed infinitely often, then the first process should enter the critical section (p_3) infinitely often.

Communicating Automata Model



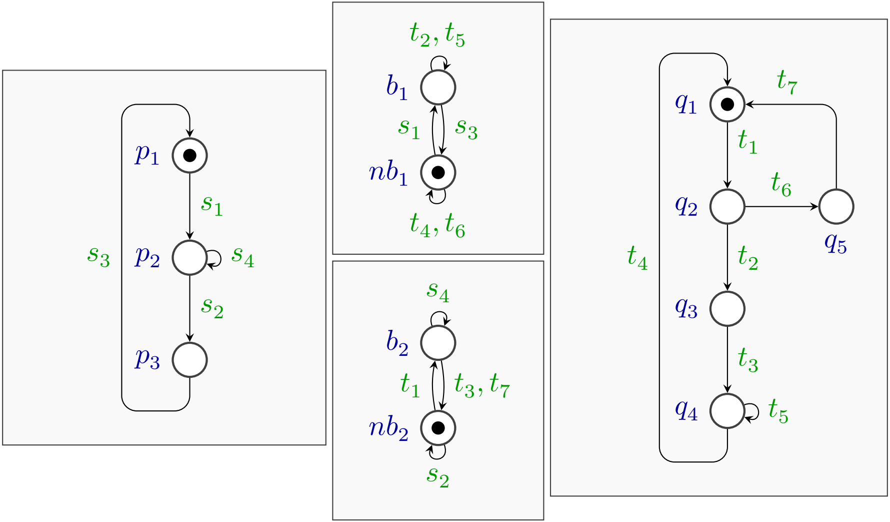
Property: If both processes are executed infinitely often, then the first process should enter the critical section (p_3) infinitely often.

Communicating Automata Model



Property: If both processes are executed infinitely often, then the first process should enter the critical section (p_3) infinitely often.

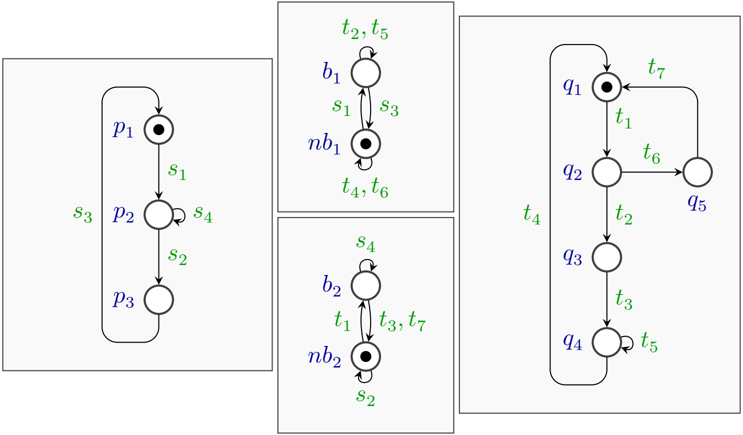
Abstract View of the Model



Property: For every infinite transition sequence σ, we have
$\varphi(\sigma) = \bigvee_{i=1}^{4}(s_i \in \inf(\sigma)) \wedge \bigvee_{i=1}^{7}(t_i \in \inf(\sigma)) \implies s_2 \in \inf(\sigma)$.

Loop Sequences

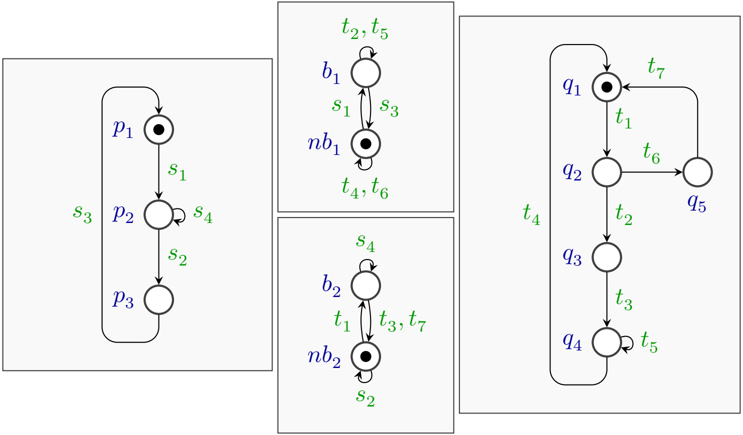
$$\{p_1, nb_1, nb_2, q_1\} \xrightarrow{t_1 t_6 t_7 s_1 t_1 t_2 t_3 s_2 t_5 s_3 t_4} \{p_1, nb_1, nb_2, q_1\}$$



Loop Sequences

$$\{p_1, nb_1, nb_2, q_1\} \xrightarrow{t_1 t_6 t_7 s_1 t_1 t_2 t_3 s_2 t_5 s_3 t_4} \{p_1, nb_1, nb_2, q_1\}$$

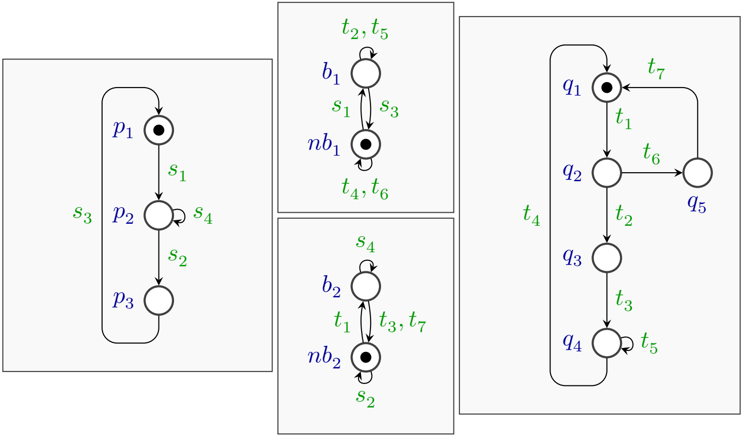
$$\#\sigma = \begin{pmatrix} \#t_1 & \#t_2 & \#t_3 & \#t_4 & \#t_5 & \#t_6 & \#t_7 & \#s_1 & \#s_2 & \#s_3 & \#s_4 \\ & & & & & & & & & & \end{pmatrix}$$



Loop Sequences

$$\{p_1, nb_1, nb_2, q_1\} \xrightarrow{t_1 t_6 t_7 s_1 t_1 t_2 t_3 s_2 t_5 s_3 t_4} \{p_1, nb_1, nb_2, q_1\}$$

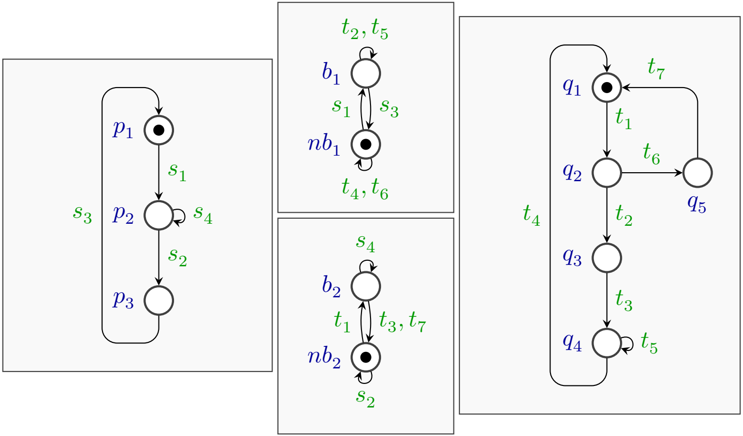
	$\#t_1$	$\#t_2$	$\#t_3$	$\#t_4$	$\#t_5$	$\#t_6$	$\#t_7$	$\#s_1$	$\#s_2$	$\#s_3$	$\#s_4$	
$\#\sigma = ($	2											$)$



Loop Sequences

$$\{p_1, nb_1, nb_2, q_1\} \xrightarrow{t_1 t_6 t_7 s_1 t_1 t_2 t_3 s_2 t_5 s_3 t_4} \{p_1, nb_1, nb_2, q_1\}$$

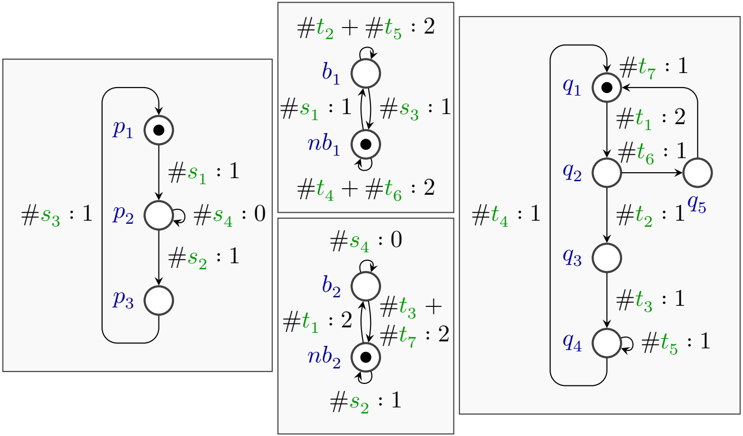
$$\#\sigma = \begin{pmatrix} \#t_1 & \#t_2 & \#t_3 & \#t_4 & \#t_5 & \#t_6 & \#t_7 & \#s_1 & \#s_2 & \#s_3 & \#s_4 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$



Loop Sequences

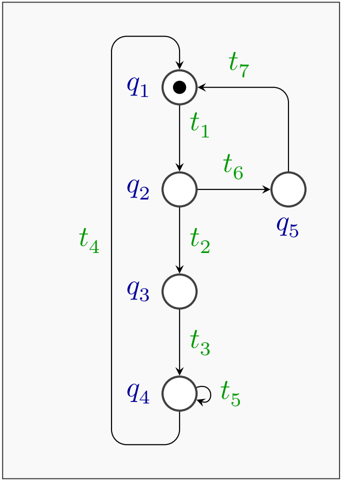
$$\{p_1, nb_1, nb_2, q_1\} \xrightarrow{t_1 t_6 t_7 s_1 t_1 t_2 t_3 s_2 t_5 s_3 t_4} \{p_1, nb_1, nb_2, q_1\}$$

	$\#t_1$	$\#t_2$	$\#t_3$	$\#t_4$	$\#t_5$	$\#t_6$	$\#t_7$	$\#s_1$	$\#s_2$	$\#s_3$	$\#s_4$
$\#\sigma =$	2	1	1	1	1	1	1	1	1	1	0



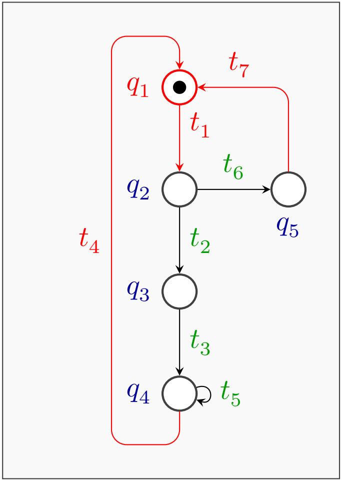
Necessary Condition for Loops

$$X = \begin{pmatrix} \overset{\#t_1}{t_1} & \overset{\#t_2}{t_2} & \overset{\#t_3}{t_3} & \overset{\#t_4}{t_4} & \overset{\#t_5}{t_5} & \overset{\#t_6}{t_6} & \overset{\#t_7}{t_7} & \overset{\#s_1}{s_1} & \overset{\#s_2}{s_2} & \overset{\#s_3}{s_3} & \overset{\#s_4}{s_4} \end{pmatrix}$$



Necessary Condition for Loops

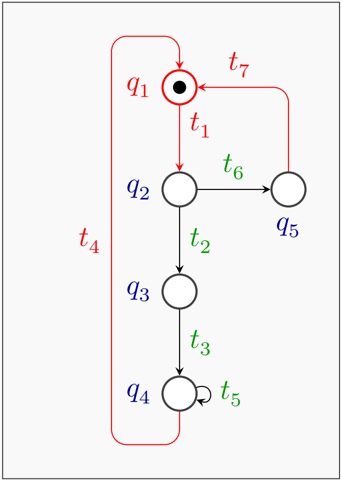
$$X = \begin{pmatrix} \#t_1 & \#t_2 & \#t_3 & \#t_4 & \#t_5 & \#t_6 & \#t_7 & \#s_1 & \#s_2 & \#s_3 & \#s_4 \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & s_1 & s_2 & s_3 & s_4 \end{pmatrix}$$



Necessary Condition for Loops

$$X = \begin{pmatrix} \overset{\#t_1}{t_1} & \overset{\#t_2}{t_2} & \overset{\#t_3}{t_3} & \overset{\#t_4}{t_4} & \overset{\#t_5}{t_5} & \overset{\#t_6}{t_6} & \overset{\#t_7}{t_7} & \overset{\#s_1}{s_1} & \overset{\#s_2}{s_2} & \overset{\#s_3}{s_3} & \overset{\#s_4}{s_4} \end{pmatrix}$$

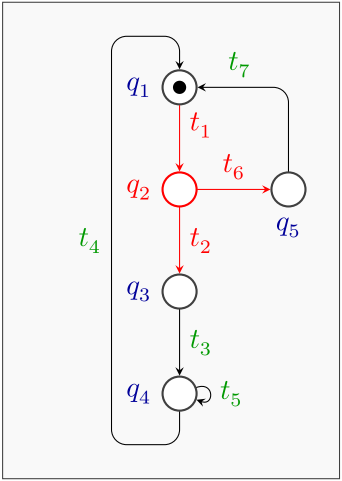
$q_1 : \quad t_4 + t_7 = t_1$



Necessary Condition for Loops

$$X = \begin{pmatrix} \#t_1 & \#t_2 & \#t_3 & \#t_4 & \#t_5 & \#t_6 & \#t_7 & \#s_1 & \#s_2 & \#s_3 & \#s_4 \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & s_1 & s_2 & s_3 & s_4 \end{pmatrix}$$

$q_1 : \quad t_4 + t_7 = t_1$

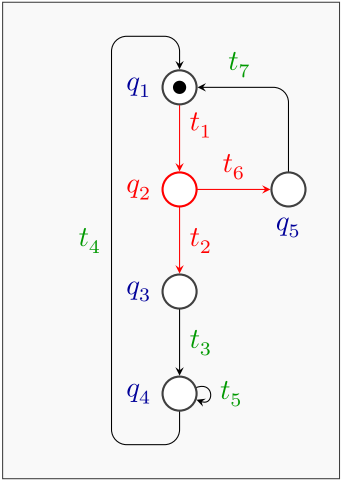


Necessary Condition for Loops

$$X = \begin{pmatrix} \#t_1 & \#t_2 & \#t_3 & \#t_4 & \#t_5 & \#t_6 & \#t_7 & \#s_1 & \#s_2 & \#s_3 & \#s_4 \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & s_1 & s_2 & s_3 & s_4 \end{pmatrix}$$

$$q_1: \quad t_4 + t_7 = t_1$$

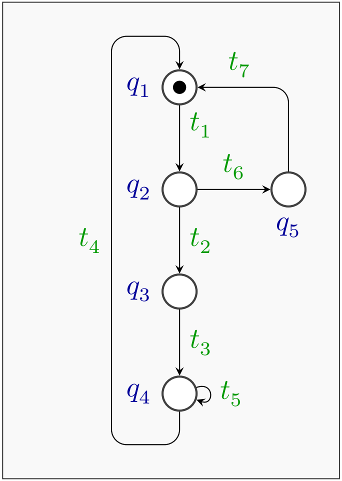
$$q_2: \quad t_1 = t_2 + t_6$$



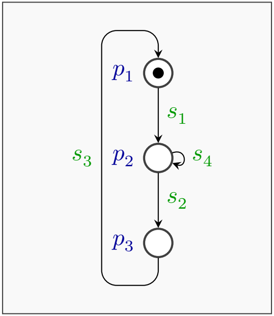
Necessary Condition for Loops

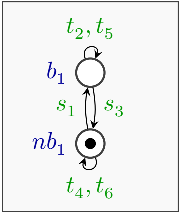
$$X = \begin{pmatrix} \#t_1 & \#t_2 & \#t_3 & \#t_4 & \#t_5 & \#t_6 & \#t_7 & \#s_1 & \#s_2 & \#s_3 & \#s_4 \\ t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & s_1 & s_2 & s_3 & s_4 \end{pmatrix}$$

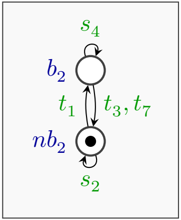
$$\begin{aligned} q_1 &: & t_4 + t_7 &= t_1 \\ q_2 &: & t_1 &= t_2 + t_6 \\ q_3 &: & t_2 &= t_3 \\ q_4 &: & t_3 &= t_4 \\ q_5 &: & t_6 &= t_7 \end{aligned}$$

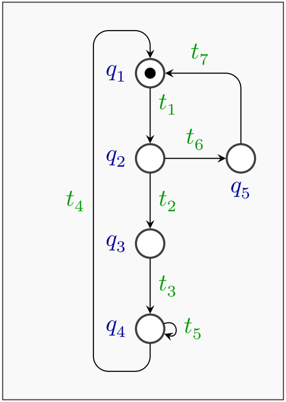


Necessary Condition for Loops









$$p_1: \quad s_3 = s_1$$
$$p_2: \quad s_1 = s_2$$
$$p_3: \quad s_2 = s_3$$

$$b_1: \quad s_1 = s_3$$
$$nb_1: \quad s_3 = s_1$$

$$b_2: \quad t_1 = t_3 + t_7$$
$$nb_2: \quad t_3 + t_7 = s_1$$

$$q_1: \quad t_4 + t_7 = t_1$$
$$q_2: \quad t_1 = t_2 + t_6$$
$$q_3: \quad t_2 = t_3$$
$$q_4: \quad t_3 = t_4$$
$$q_5: \quad t_6 = t_7$$

Termination Constraints

- Accumulate constraints in matrix form as $C \cdot X = 0$.
- If there is an infinite transition sequence σ, then the following constraints have a solution X:

$$\mathcal{C} :: \begin{cases} C \cdot X = 0 \\ X \geq 0 \\ X \neq 0 \end{cases}$$

- If the constraints have no solution, then the program is terminating.
- A solution X is *realizable* if there is a sequence σ with $\#\sigma = X$.

Fair Termination Constraints

- Fairness condition given by boolean formula φ over $t \in \inf(\sigma)$.
- If the program is not fairly terminating, then there is an infinite transition sequence σ satisfying $\sigma \models \neg\varphi$.
- Add constraint $\neg\varphi(X)$ to $\mathcal{C}$ for fair termination constraints.

Fairness for Lamport's Algorithm

$$\varphi(\sigma) = \bigvee_{i=1}^{4}(s_i \in \inf(\sigma)) \wedge \bigvee_{i=1}^{7}(t_i \in \inf(\sigma)) \implies s_2 \in \inf(\sigma)$$

$$\begin{aligned}\neg\varphi(X) = \; & (s_1 + s_2 + s_3 + s_4 > 0) \wedge \\ & (t_1 + t_3 + t_4 + t_5 + t_6 + t_7 > 0) \wedge \\ & (s_2 = 0)\end{aligned}$$

Fair Termination Constraints

$$
\begin{array}{llll}
s_3 = s_1 & t_4 + t_7 = t_1 & s_1 \geq 0 & t_1 \geq 0 \\
s_1 = s_2 & t_1 = t_2 + t_6 & s_2 \geq 0 & t_2 \geq 0 \\
s_2 = s_3 & t_2 = t_3 & s_3 \geq 0 & t_3 \geq 0 \\
 & t_3 = t_4 & & t_4 \geq 0 \\
 & t_6 = t_7 & & t_5 \geq 0 \\
s_1 = s_3 & t_1 = t_3 + t_7 & & t_6 \geq 0 \\
s_3 = s_1 & t_3 + t_7 = s_1 & & t_7 \geq 0
\end{array}
$$

$$s_1 + s_2 + s_3 + s_4 + t_1 + t_3 + t_4 + t_5 + t_6 + t_7 > 0$$

$$
\begin{array}{l}
(s_1 + s_2 + s_3 + s_4 > 0) \wedge \\
(t_1 + t_3 + t_4 + t_5 + t_6 + t_7 > 0) \wedge \\
(s_2 = 0)
\end{array}
$$

Fair Termination Constraints: Solution

$$X = \begin{pmatrix} \overset{t_1}{0} & \overset{t_2}{0} & \overset{t_3}{0} & \overset{t_4}{0} & \overset{t_5}{1} & \overset{t_6}{0} & \overset{t_7}{0} & \overset{s_1}{0} & \overset{s_2}{0} & \overset{s_3}{0} & \overset{s_4}{1} \end{pmatrix}$$

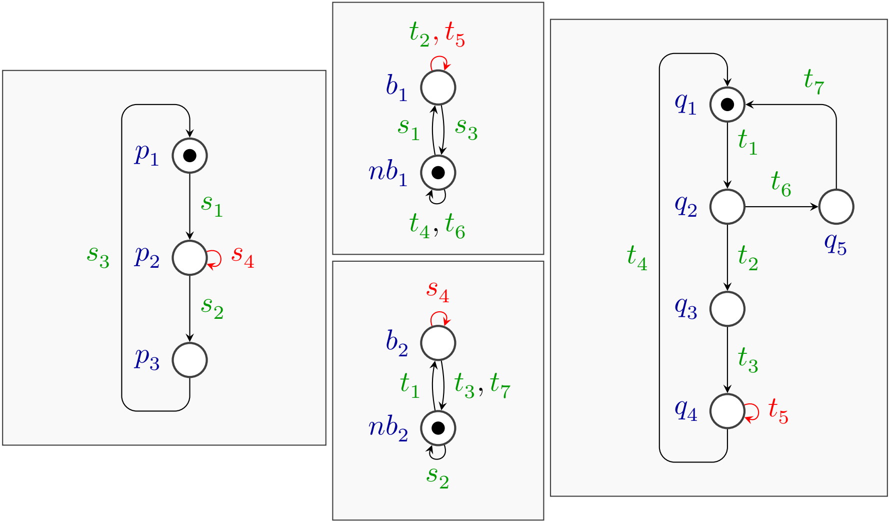
$$\begin{array}{llll}
s_3 = s_1 & t_4 + t_7 = t_1 & s_1 \geq 0 & t_1 \geq 0 \\
s_1 = s_2 & t_1 = t_2 + t_6 & s_2 \geq 0 & t_2 \geq 0 \\
s_2 = s_3 & t_2 = t_3 & s_3 \geq 0 & t_3 \geq 0 \\
 & t_3 = t_4 & & t_4 \geq 0 \\
 & t_6 = t_7 & & t_5 \geq 0 \\
s_1 = s_3 & t_1 = t_3 + t_7 & & t_6 \geq 0 \\
s_3 = s_1 & t_3 + t_7 = s_1 & & t_7 \geq 0
\end{array}$$

$$s_1 + s_2 + s_3 + s_4 + t_1 + t_3 + t_4 + t_5 + t_6 + t_7 > 0$$

$$(s_1 + s_2 + s_3 + s_4 > 0) \wedge$$
$$(t_1 + t_3 + t_4 + t_5 + t_6 + t_7 > 0) \wedge$$
$$(s_2 = 0)$$

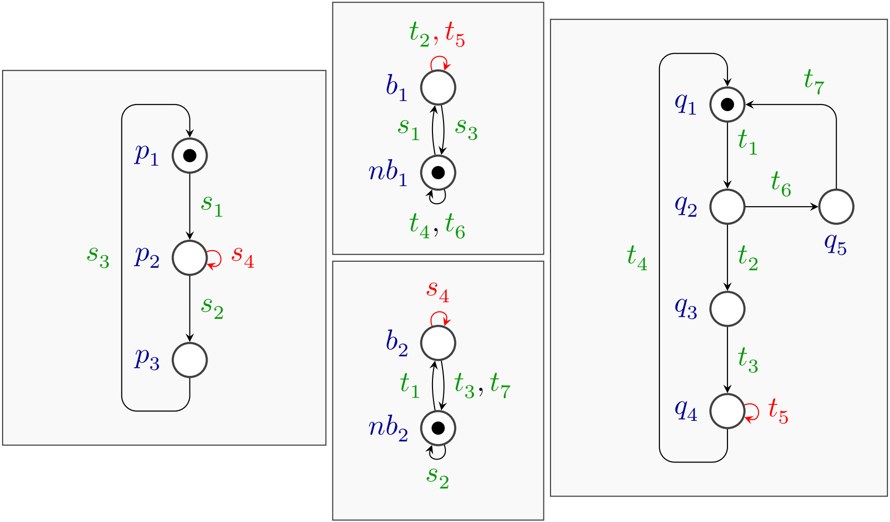
Fair Termination Constraints: Solution

$$X = \begin{pmatrix} \overset{t_1}{0} & \overset{t_2}{0} & \overset{t_3}{0} & \overset{t_4}{0} & \overset{t_5}{1} & \overset{t_6}{0} & \overset{t_7}{0} & \overset{s_1}{0} & \overset{s_2}{0} & \overset{s_3}{0} & \overset{s_4}{1} \end{pmatrix}$$



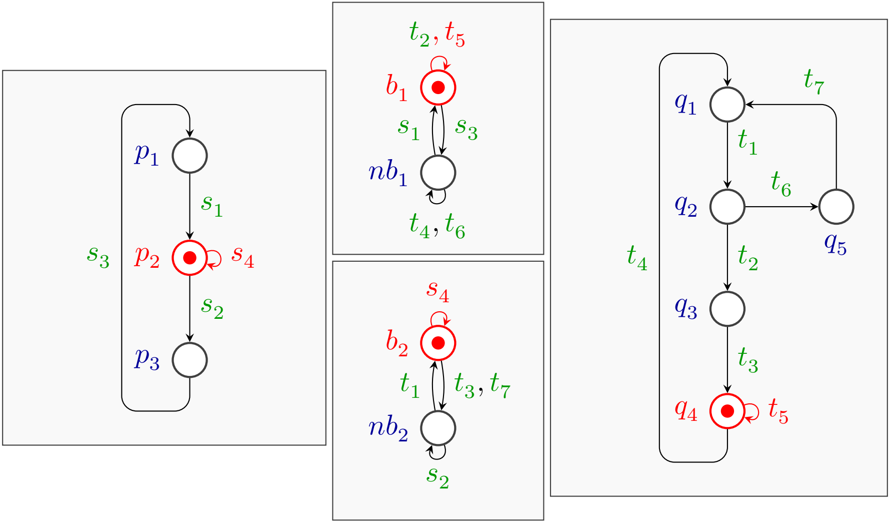
Solution realizable?

X realized by σ with $\inf(\sigma) = \{s_4, t_5\}$.



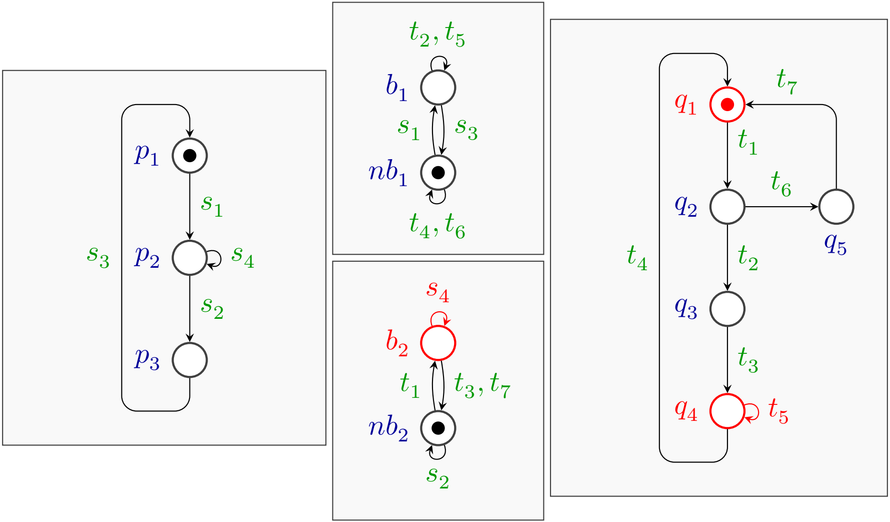
Solution realizable?

X realized by σ with $\inf(\sigma) = \{s_4, t_5\}$.



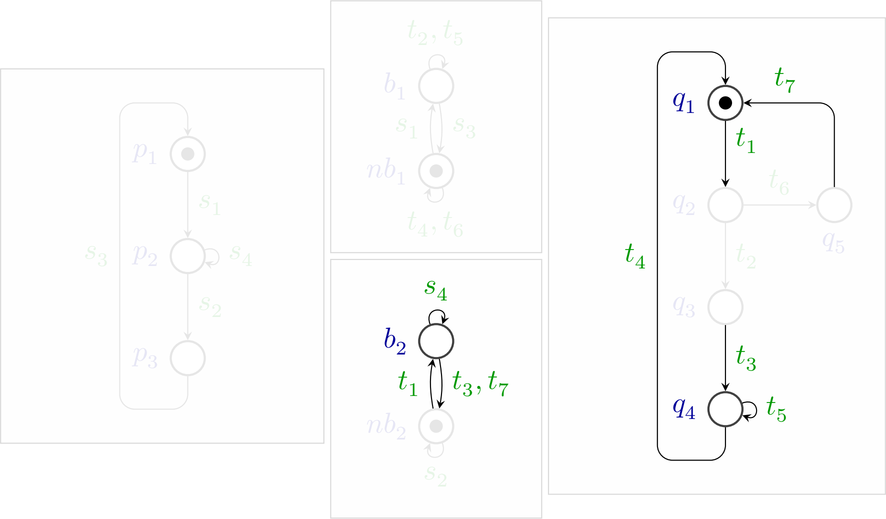
Refinement Component

q_1, q_4 and b_2 are in mutual exclusion.



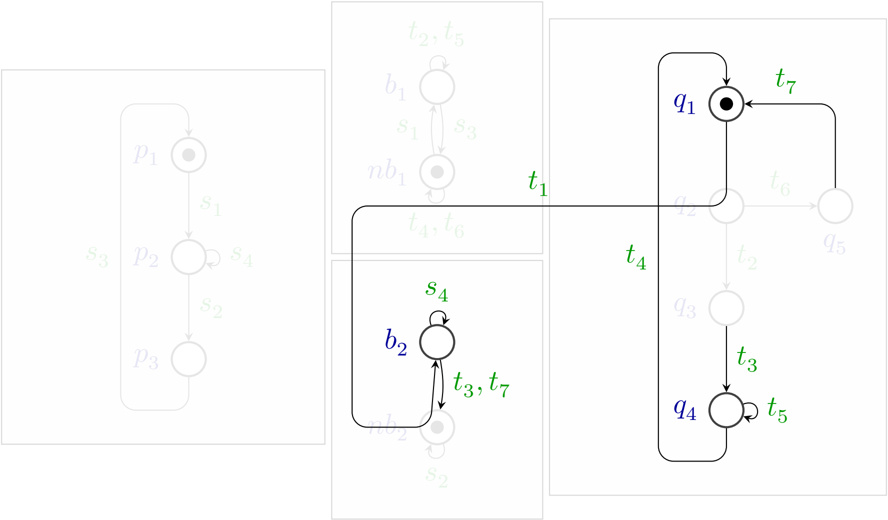
Refinement Component

q_1, q_4 and b_2 are in mutual exclusion.



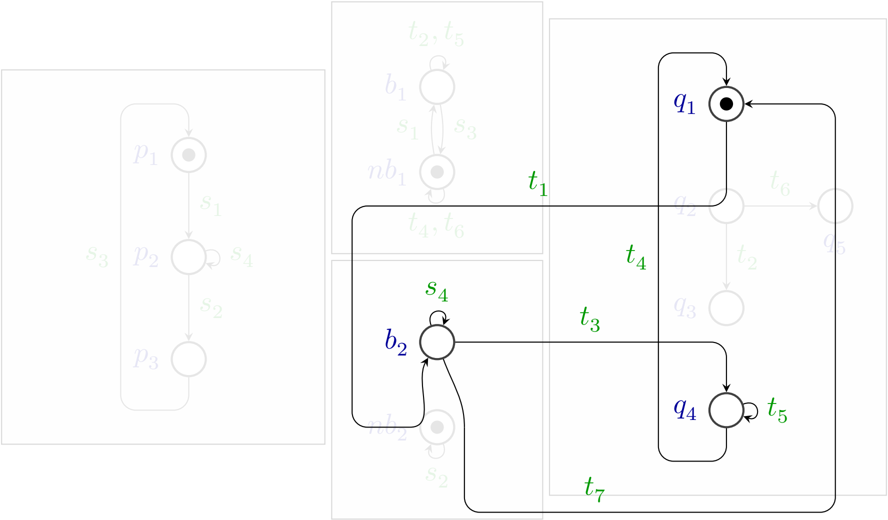
Refinement Component

q_1, q_4 and b_2 are in mutual exclusion.



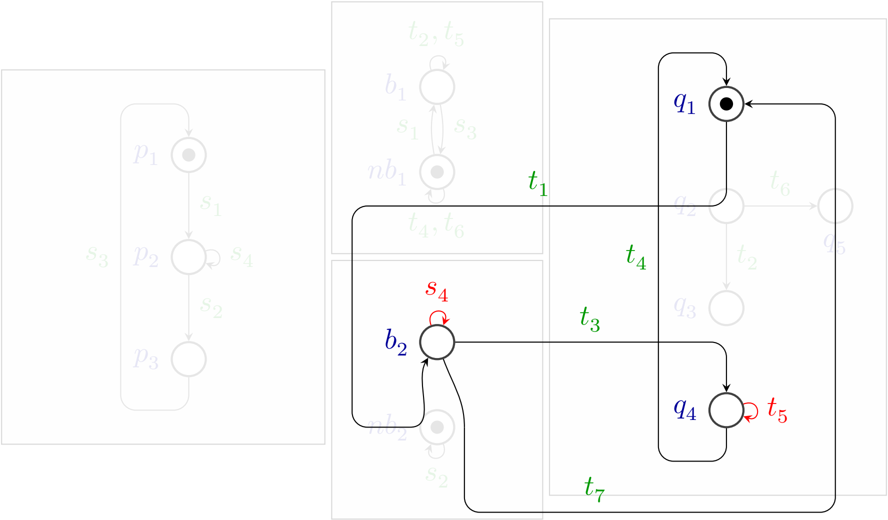
Refinement Component

q_1, q_4 and b_2 are in mutual exclusion.



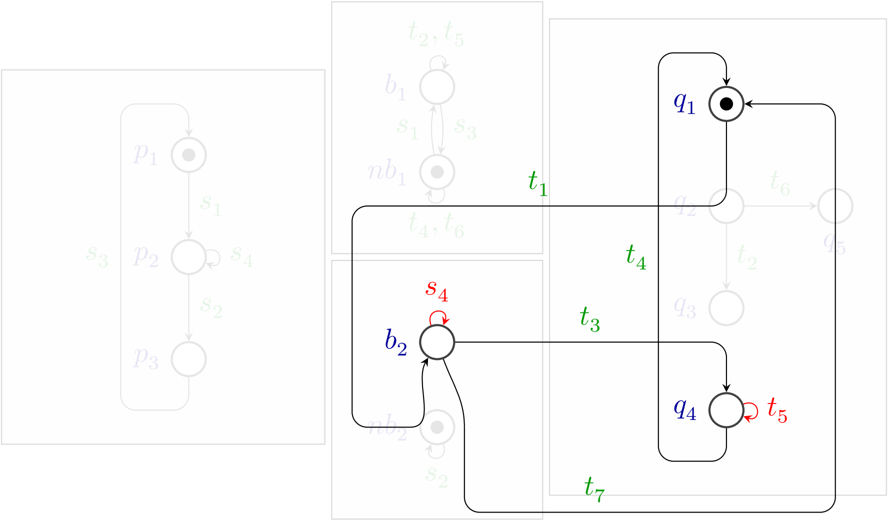
Refinement Constraint

X realized by σ with $\inf(\sigma) = \{s_4, t_5\}$.



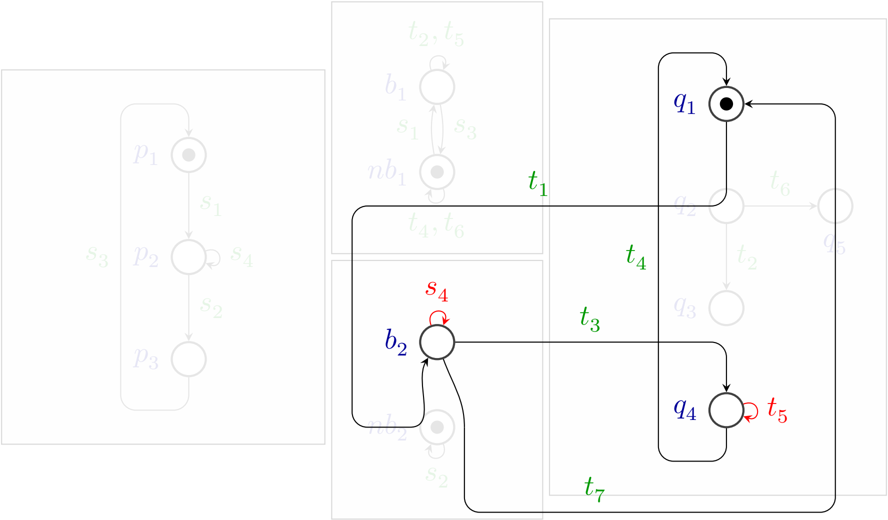
Refinement Constraint

X not realizable $\Rightarrow$ Generate refinement constraint δ.



Refinement Constraint

$$\delta = (s_4 = 0) \vee (t_5 = 0) \vee (t_1 + t_3 + t_4 + t_7 > 0)$$



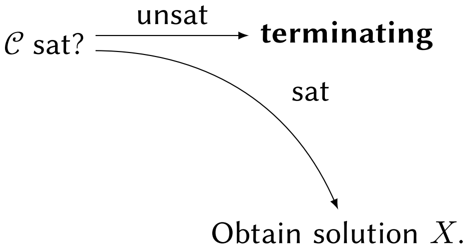
Refinement Loop

$\mathcal{C}$ sat?

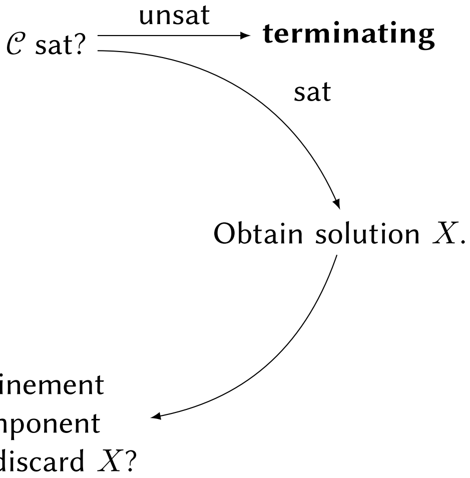
Refinement Loop

$\mathcal{C}$ sat? $\xrightarrow{\text{unsat}}$ **terminating**

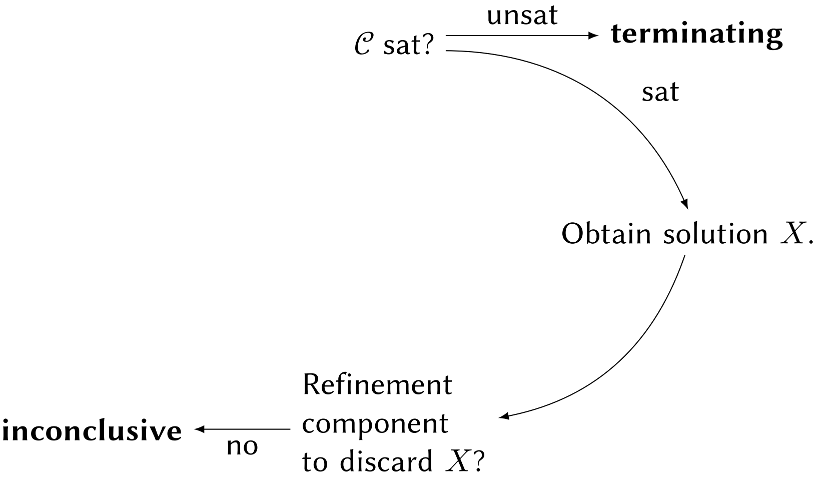
Refinement Loop



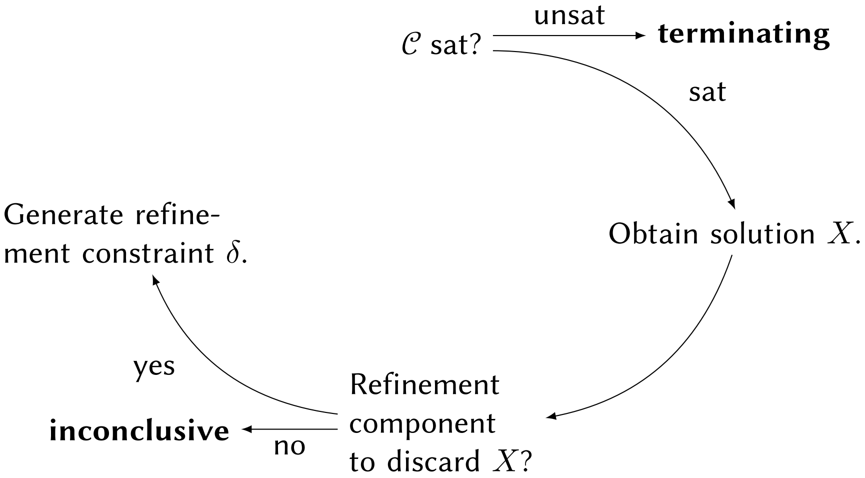
Refinement Loop



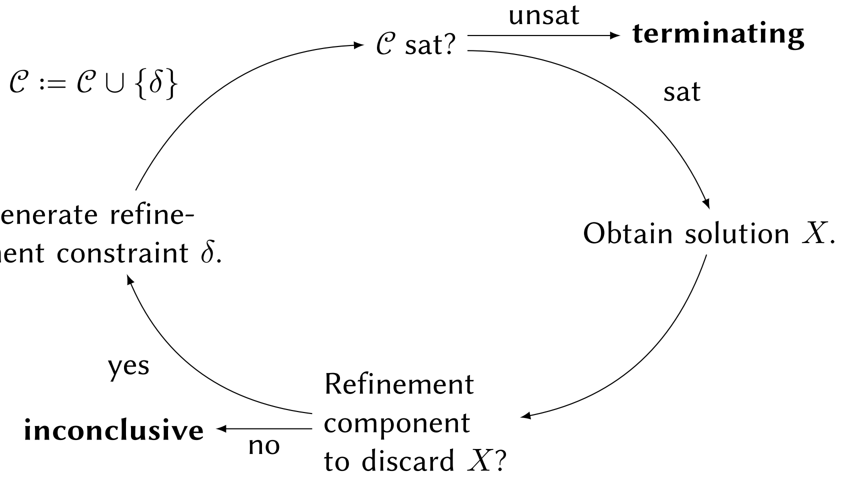
Refinement Loop



Refinement Loop



Refinement Loop

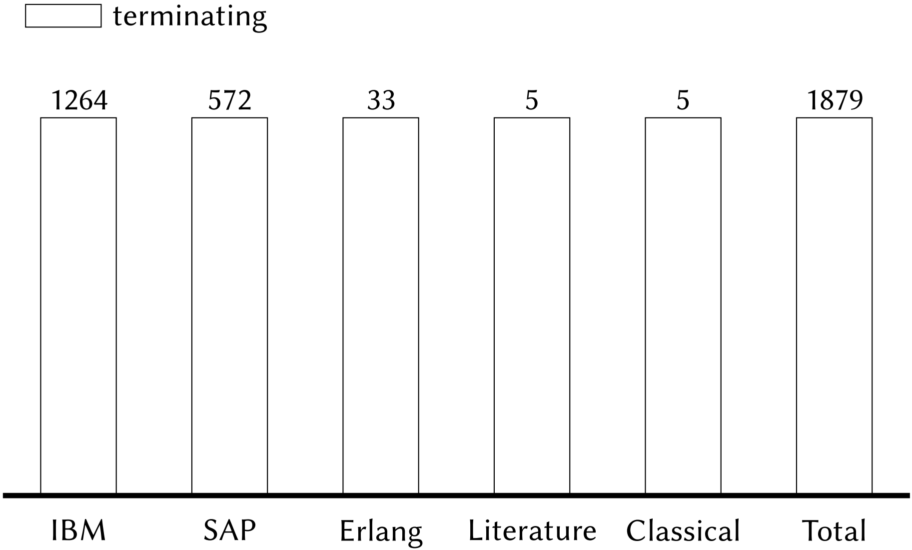


Experimental Evaluation

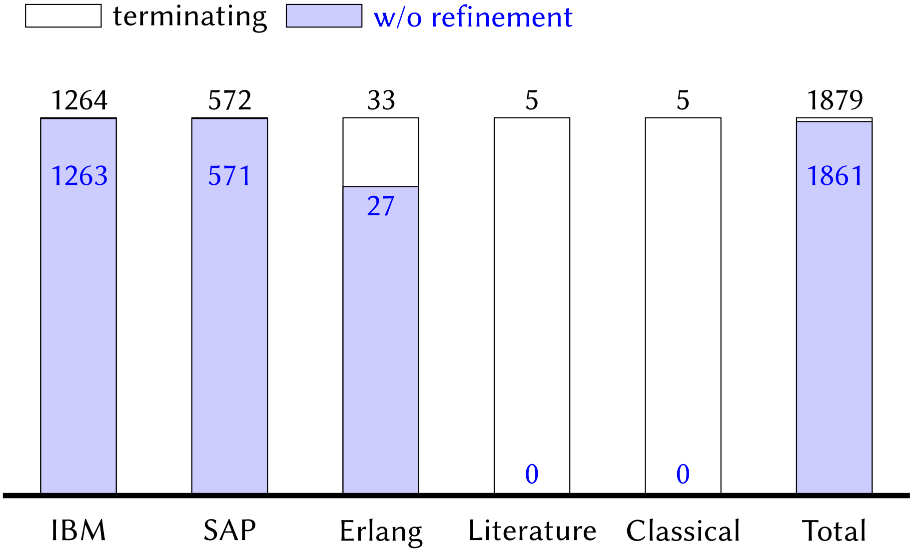
Benchmarks

- IBM/SAP — Workflow nets from business process models
  - 1976 examples
  - 1836 terminating
- Erlang — Models from the verification of Erlang programs
  - 50 examples, up to 66950 places and 213626 transitions
  - 33 terminating
- Literature — Selected examples from the literature
  - 5 examples, with unbounded variables
  - All terminating
- Classical — Classic asynchronous programs for mutual exclusion and distributed algorithms
  - 5 examples, scalable in number of processes
  - All fairly terminating

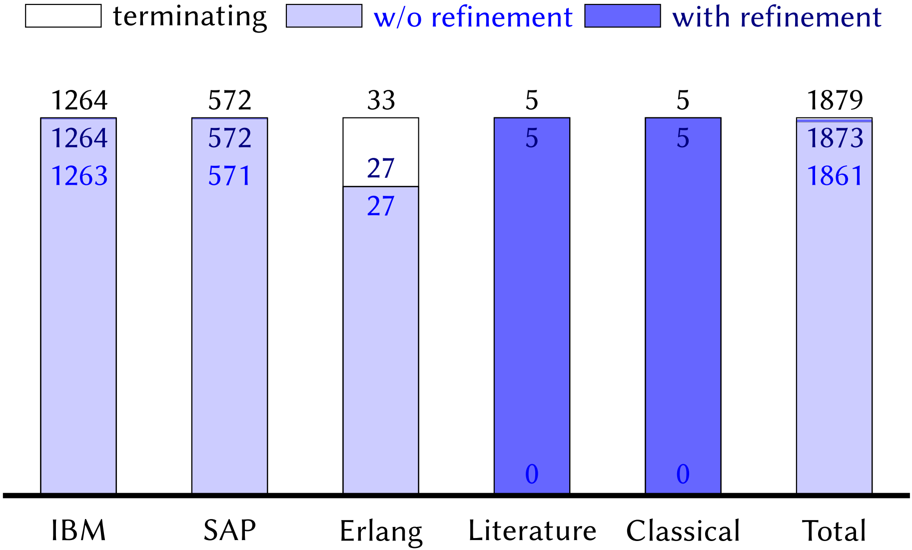
Rate of Success



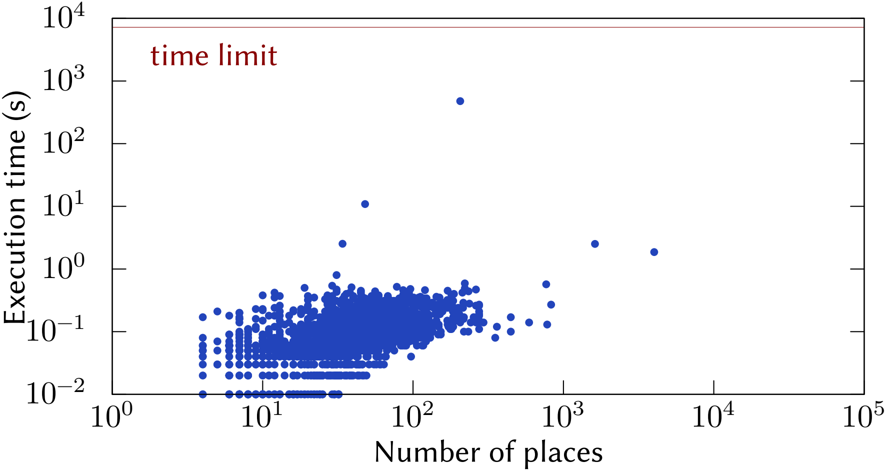
Rate of Success



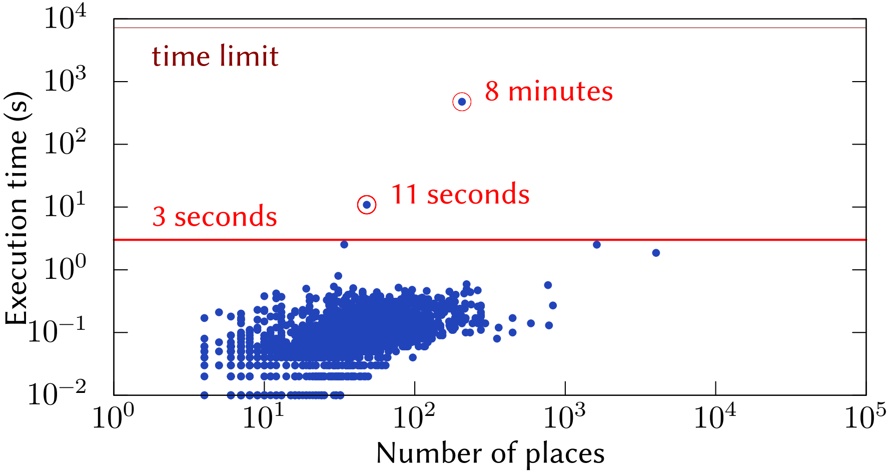
Rate of Success



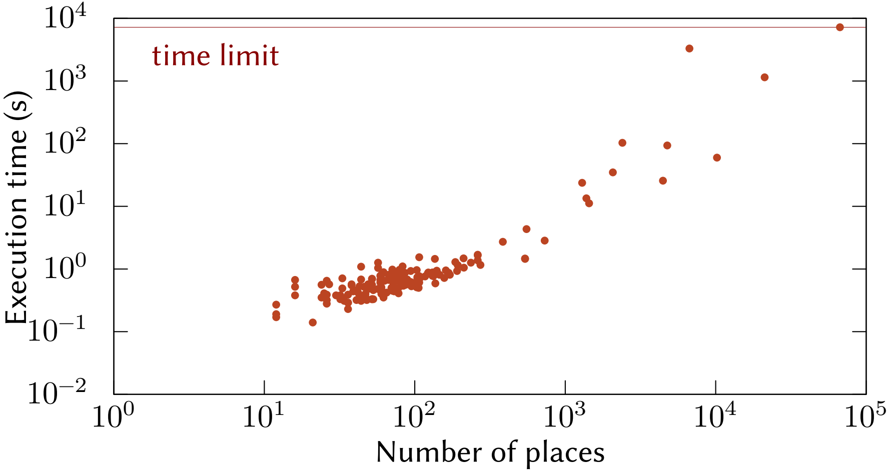
Performance on Positive Examples



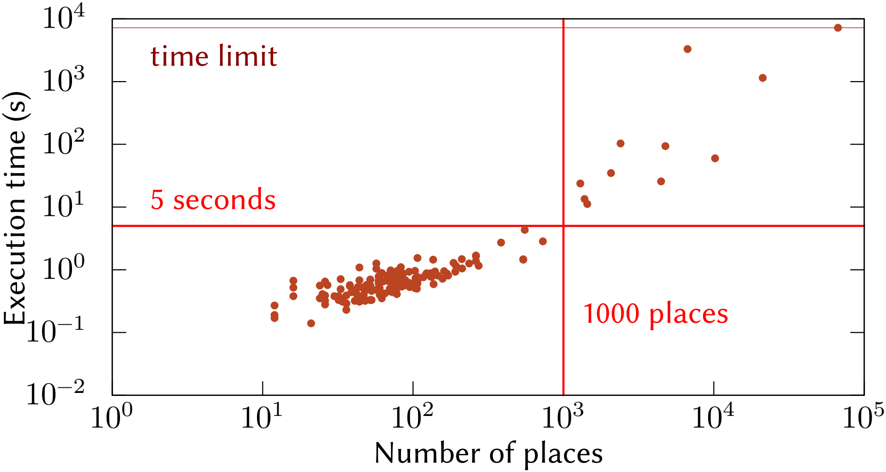
Performance on Positive Examples



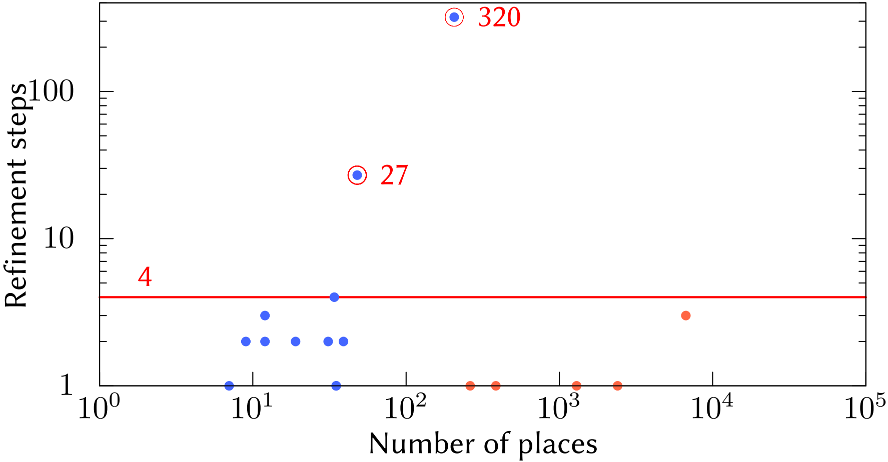
Performance on Negative Examples



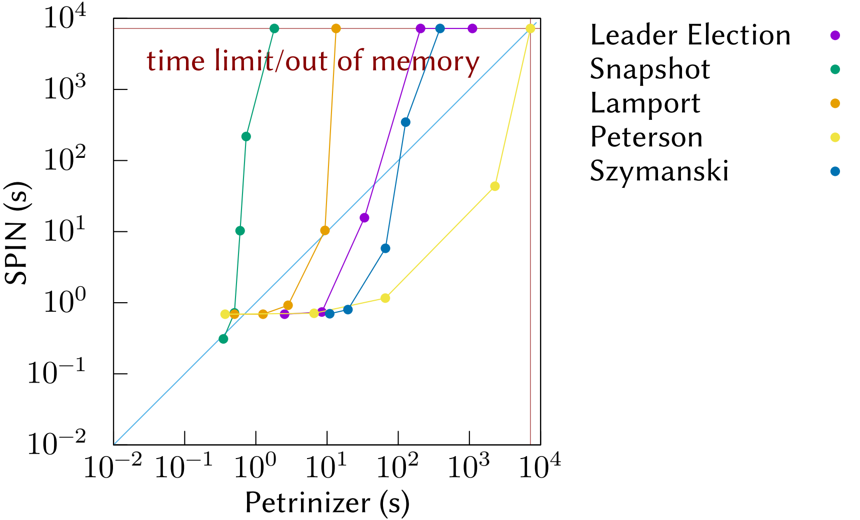
Performance on Negative Examples



Refinement Steps



Comparison with SPIN on Scaled Classical Suite



Summary

- Fast and effective technique for proving fair termination
- Incomplete, but high degree of completeness
- Large instances can be handled
- Constraints can be used as a certificate of fair termination