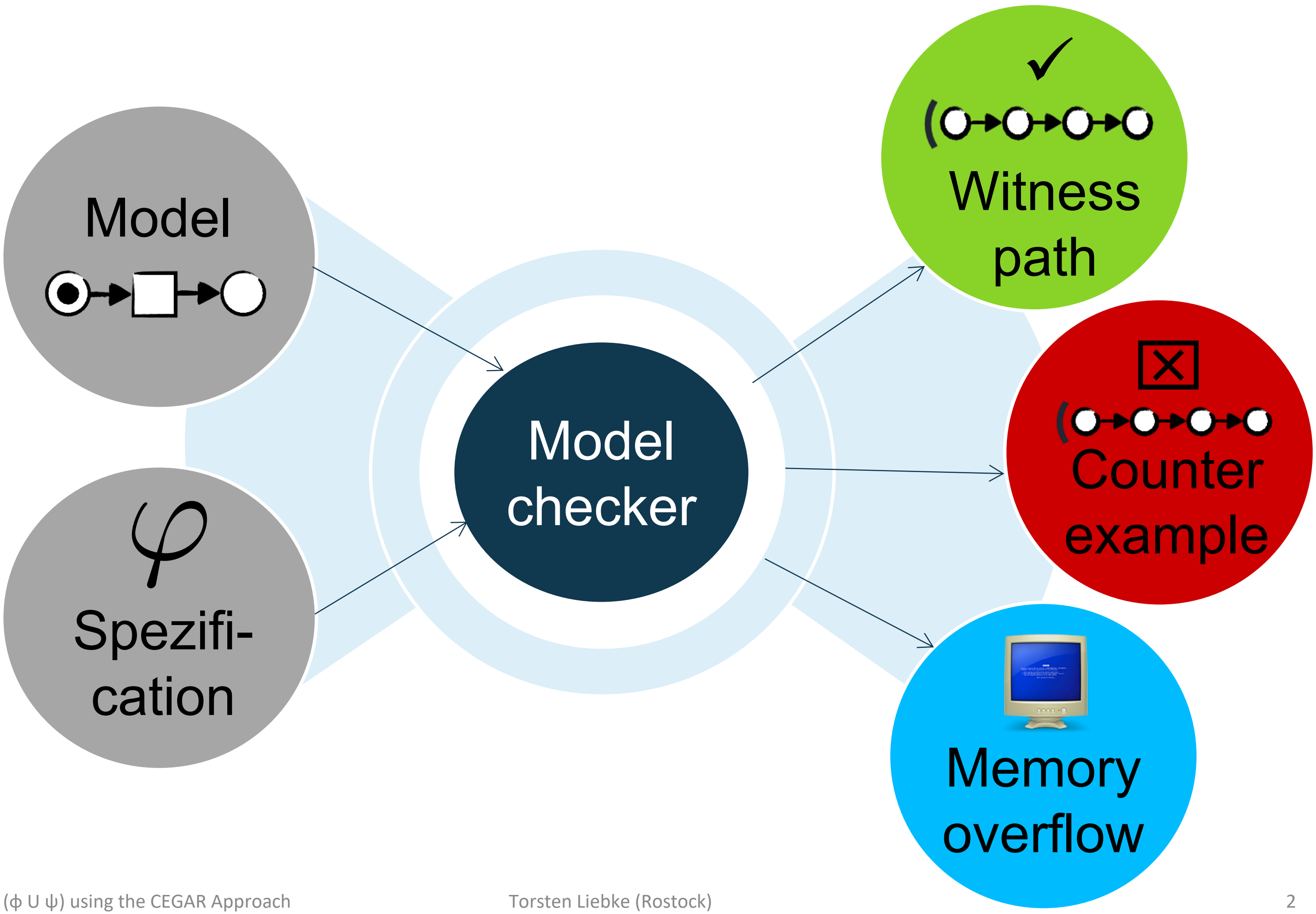
Solving E (ϕ U ψ) using the CEGAR Approach

Torsten Liebke and Karsten Wolf

University of Rostock

Model checking



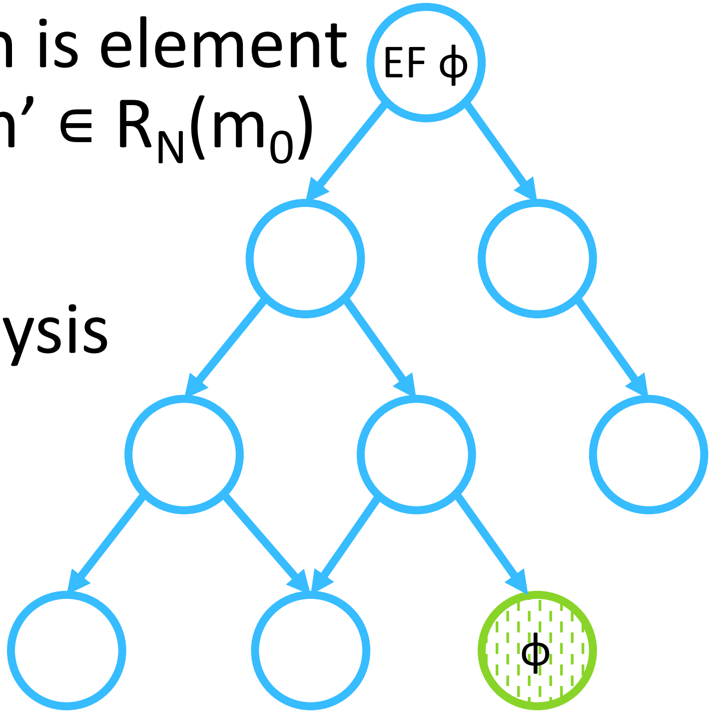
Reachability problem

Given: bounded Petri net $N = [P,T,F,W,m_0]$

Question: Exists marking m', which is element of the reachability graph $R_N(m_0)$: $m' \in R_N(m_0)$

Problem: state explosion

Solution approach: structural analysis



EF ϕ – Exists a path, where finally ϕ holds?

Success story for structural analysis for EF ϕ

In 2011 Harro Wimmel and Karsten Wolf:
Applying CEGAR to the Petri net state equation

Tool: *Sara,* won trophies in the MCC’2013

Sara

MCC’2013

Sara-integration in *LoLA* increased the performance from 75 % to 90 % in MCC’2016

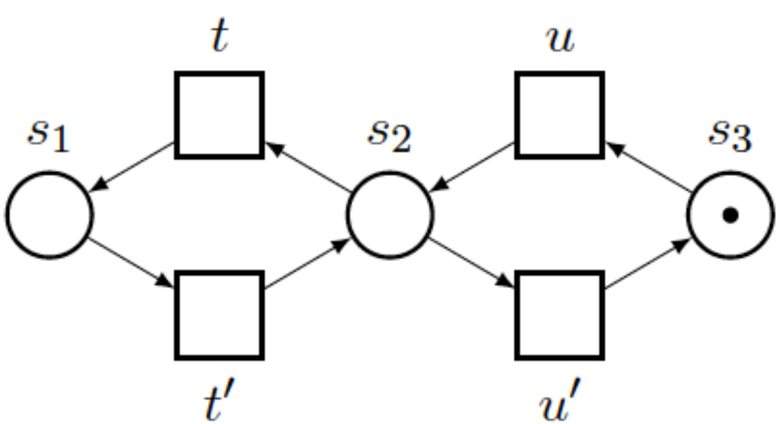
LoLA

MCC’2016

Petri net state equation

State equation: $m + C_N \cdot P(w) = m'$

- C_N = incidence matric, $P(w)$ = Parikh vector
- Necessary condition for reachability
- Integer Linear Programming (ILP) problem - can be solved with any ILP-solver



N

$$
\begin{array}{cc}
\begin{array}{cccc} t & t' & u & u' \end{array} & \\
\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} & \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array}
\end{array}
$$

C_N

State equation outcome

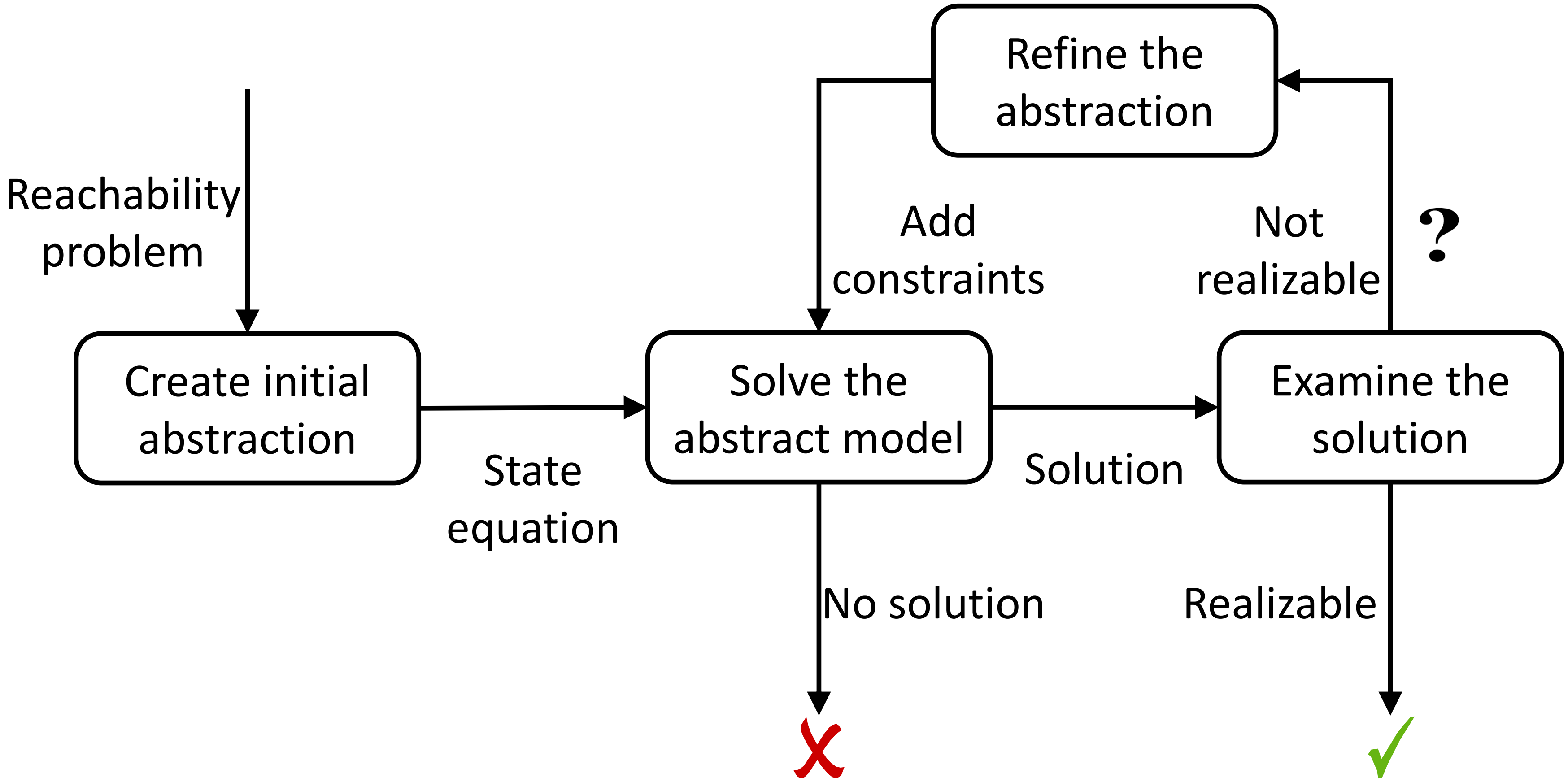
If the ILP problem is infeasible, the necessary condition is violated and the final marking is not reachable. ✗

If the ILP problem has a realizable solution, then the final marking is reachable. ✓

(realizable = firing sequence of the solution is executable)

If the ILP problem has an unrealizable solution, which is a counterexample, then the abstraction has to be refined. ?

Counter Example Guided Abstraction Refinement (CEGAR) approach

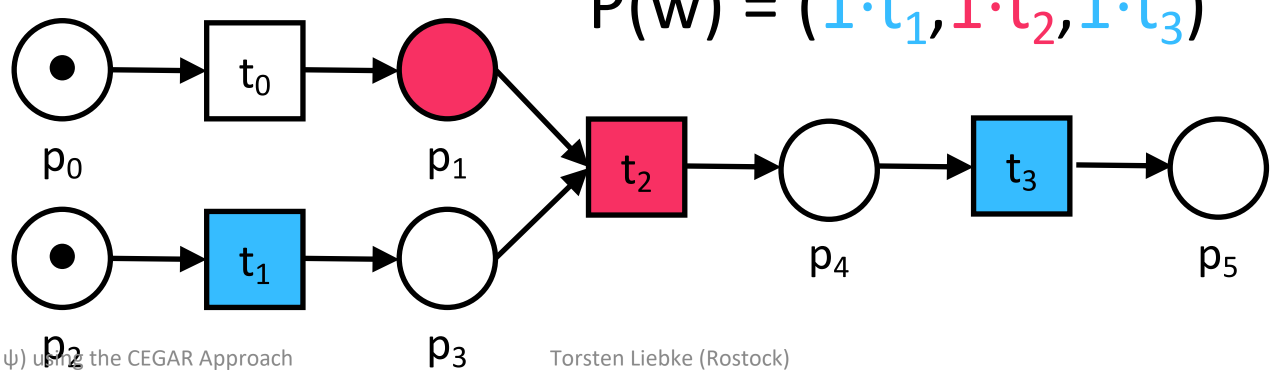


Not realizable

- **Given:** solution vector P(w) of the ILP-problem
- **Problem:** not all transitions of P(w) can fire

  => some places (called scapegoats)
  have not enough tokens
- **Solution:** we need to transfer or borrow tokens to fill the scapegoat places

$P(w) = (1 \cdot t_1, 1 \cdot t_2, 1 \cdot t_3)$



Refining the abstraction

Initial abstraction: state equation

Target function: minimize solution vector

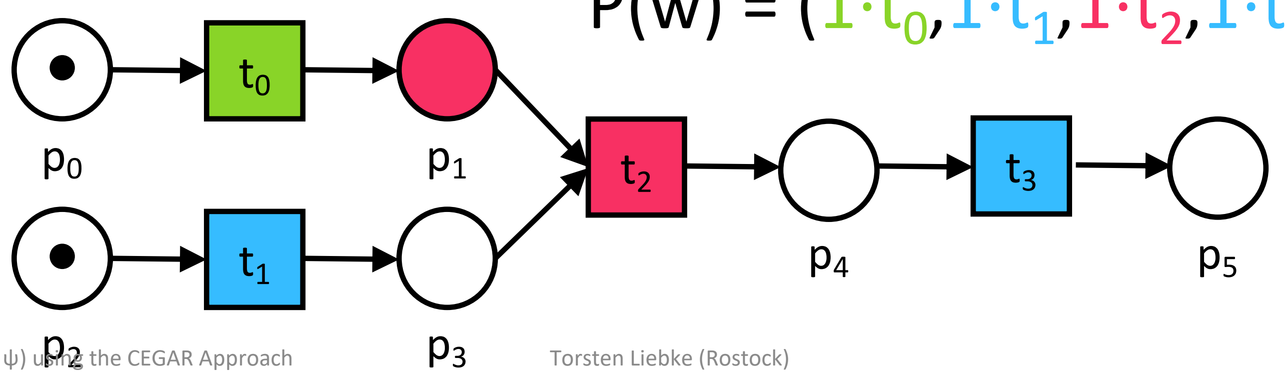
$\Rightarrow$ # of transition should be minimal

Not realizable:

- Add constraints to get a new solution
- Constraints: linear inequalities over transitions

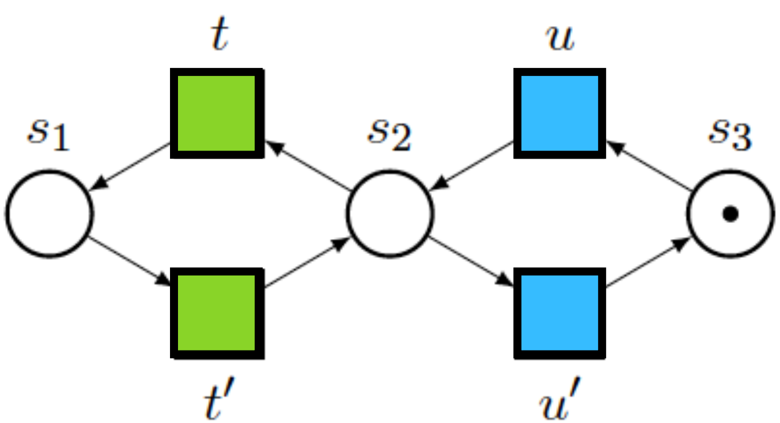
Add: $|t_0| > 0$

$P(w) = (1 \cdot t_0, 1 \cdot t_1, 1 \cdot t_2, 1 \cdot t_3)$



Borrowing tokens with transition-invariants

- P(w) is called a T-invariant if $C_N \cdot P(w) = 0$
- A fired T-invariant does not change the marking
- **T-invariant uu' can borrow tokens to the T-invariant tt'**



N

$$
\begin{array}{cc}
\begin{array}{cccc} t & t' & u & u' \end{array} & \\
\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} & \begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array}
\end{array}
$$

C_N

Adding constraints

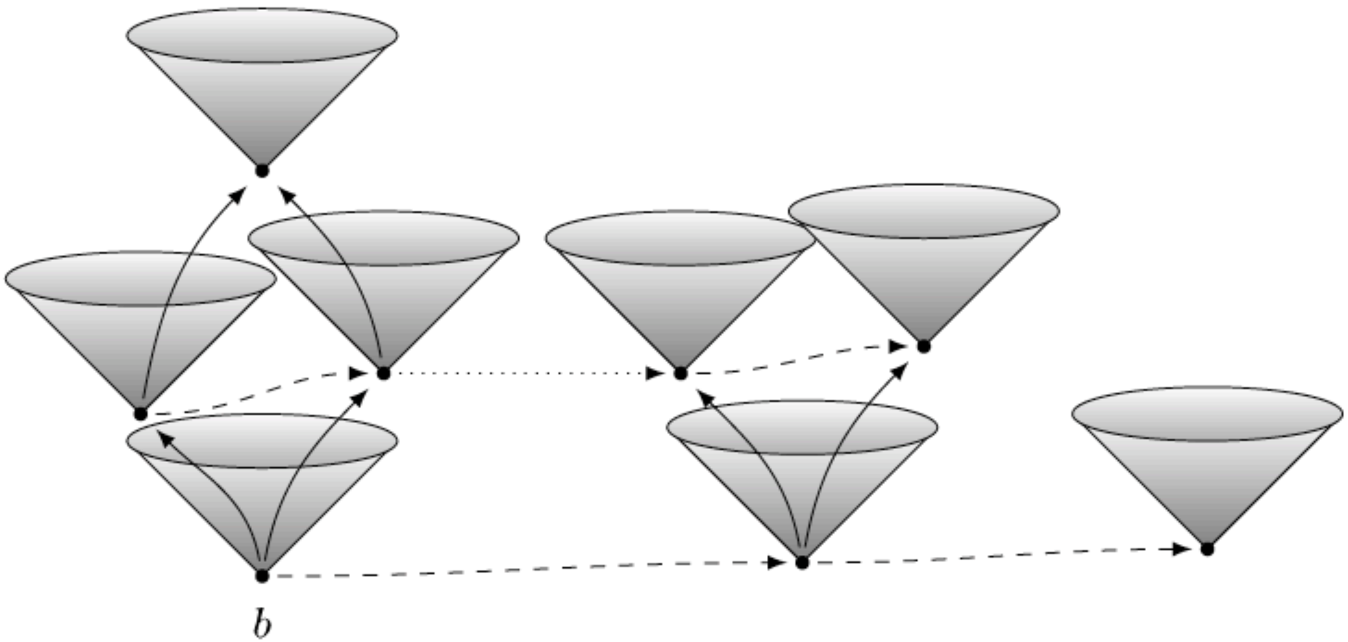
Jump constraints:

- Base solutions are pairwise incomparable
- Intend to generate a new base solution

Increment constraints:

- Generate a new non-base solution
- I.e., T-invariants are added
- interleaving with another sequence w may turn w from unrealizable to realizable

Solution Space



- Minimal solution b
- Black dots represent other solutions
- Dashed arrows are jumps
- Normal arrows are increment arrows (T-inv.)
- Cones are the linear solution spaces

Results of applying CEGAR to the Petri net state equation

- Finds positive and negative results
- Especially good for negative results

  $\Rightarrow$ Quite often and quite fast the ILP-problem becomes infeasible

  $\Rightarrow$ No need to explore the state space

Goal: apply this technique to other formulas

Applying structural analysis on different formulas

EF ϕ – already solved

Exists a path, where finally ϕ holds?

E (ϕ U ψ) – solved

Exists a path, where ϕ is true in every state along the path until a state is reached where ψ holds?

Simple CTL formulas

$(EX)^k$ ϕ – solved

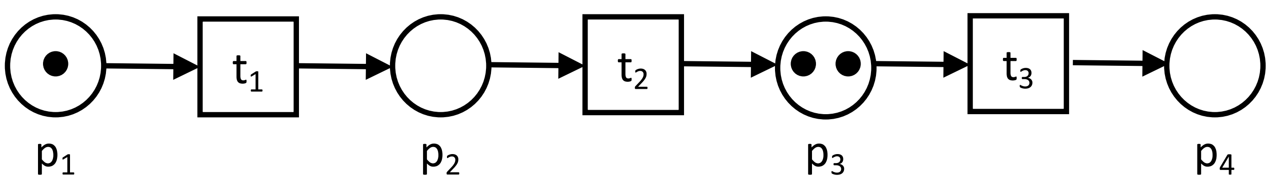
Exists a path, where ϕ holds in the k-th state?

EG ϕ – open

Exists a path, where ϕ holds in every state along the path?

Delta of a transition

- ϕ has the form $s = k_1p_1 + \ldots + k_np_n \leq k$
- Delta of a transition w.r.t. ϕ, is the effect of the transition regarding the truth value of ϕ.
- Formal: $\Delta_{t,\phi} = k_1C_N(p_1,t) + \ldots + k_nC_N(p_n,t)$

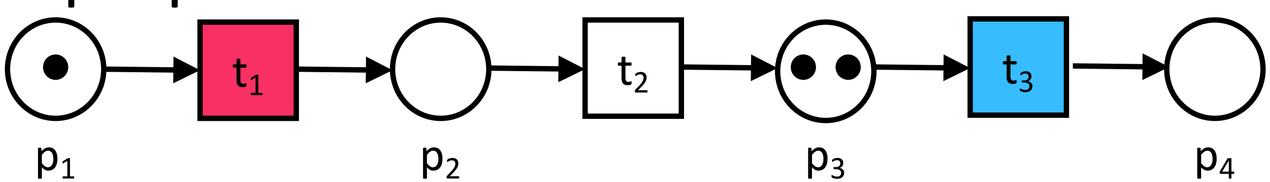


- $\phi = 2p_2 + 4p_3 \leq 5$
- $m_0 \not\models \phi$

Increasing / decreasing transitions

Transition t is called:

- Decreasing iff $\Delta_{t,\phi} > 0$; tendency to turn a true proposition into a false one
- Increasing iff $\Delta_{t,\phi} < 0$; tendency to turn a false proposition into a true one



- $\phi = 2p_2 + 4p_3 \leq 5$
- $m_0 \nvDash \phi$

- t_1 is decreasing; $\Delta_{t1,\phi} = 2$
- t_3 is increasing; $\Delta_{t3,\phi} = -4$

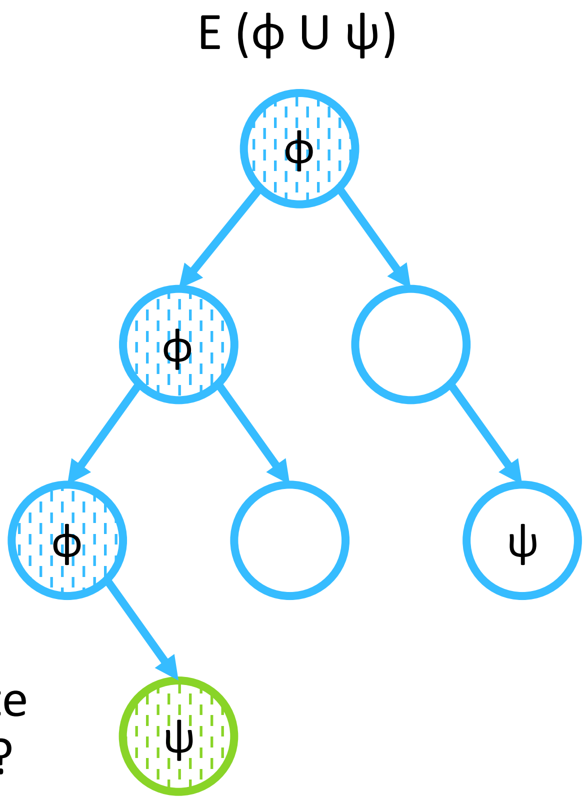
E (ϕ U ψ)

Idea:

E (ϕ U ψ) = EF ψ and keep ϕ true in every state along the path.

We only care about transitions, that can change ϕ: $T_\phi = \{t \in T \mid \Delta_{t,\phi} \neq 0\}$

E (ϕ U ψ) – Exists a path, where ϕ is true in every state along the path until a state is reached where ψ holds?



Add balance constraints

Balance constraints, ensure that ϕ is true after firing the complete ILP-solution.

- $\sum_{t \in T_\phi} \Delta_{t,\phi} \leq$ offset; offset w.r.t. the initial marking
- Only one ("last") transition is allowed to make ϕ false, when it makes ψ true at the same time.

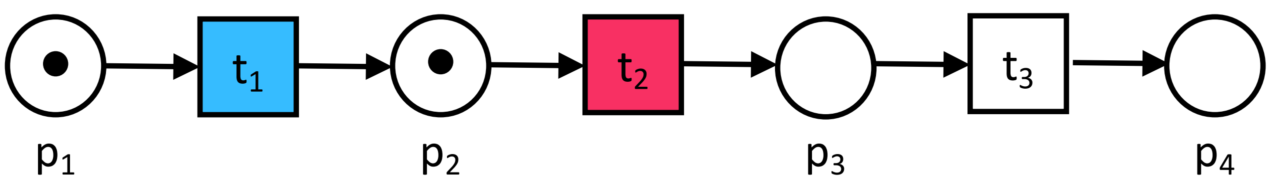
- Decreasing t
- Offset

- Increasing t

Example for balance constraints

E ($p_2 > 0$) U ($p_4 > 0$)

- Minimal solution is $t_2 t_3$
- t_2 is decreasing w.r.t. $p_2 > 0$
- Balance constraints adds t_1, which is increasing

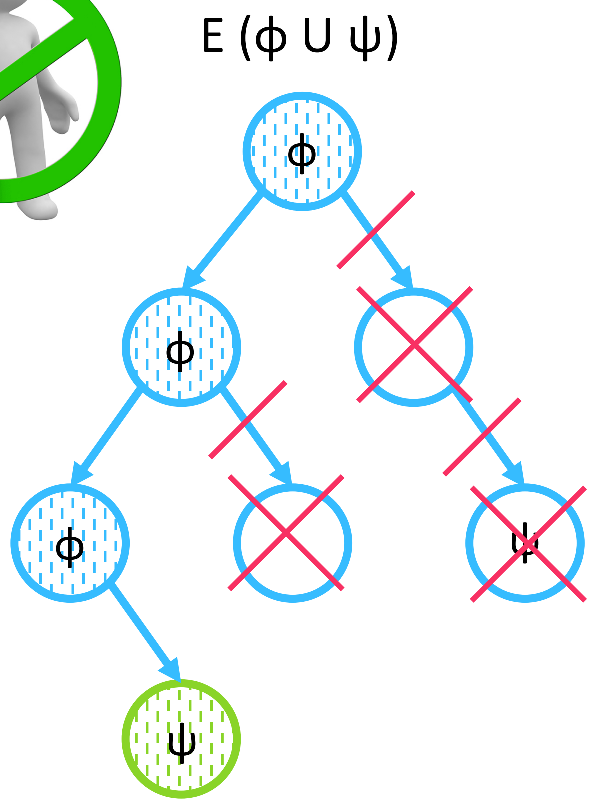


E ($p_2 > 0$) U ($p_4 > 0$) can be rewritten into the form $s \leq k$:

E ($-p_2 \leq -1$) U ($-p_4 \leq -1$)

Keeping ϕ true

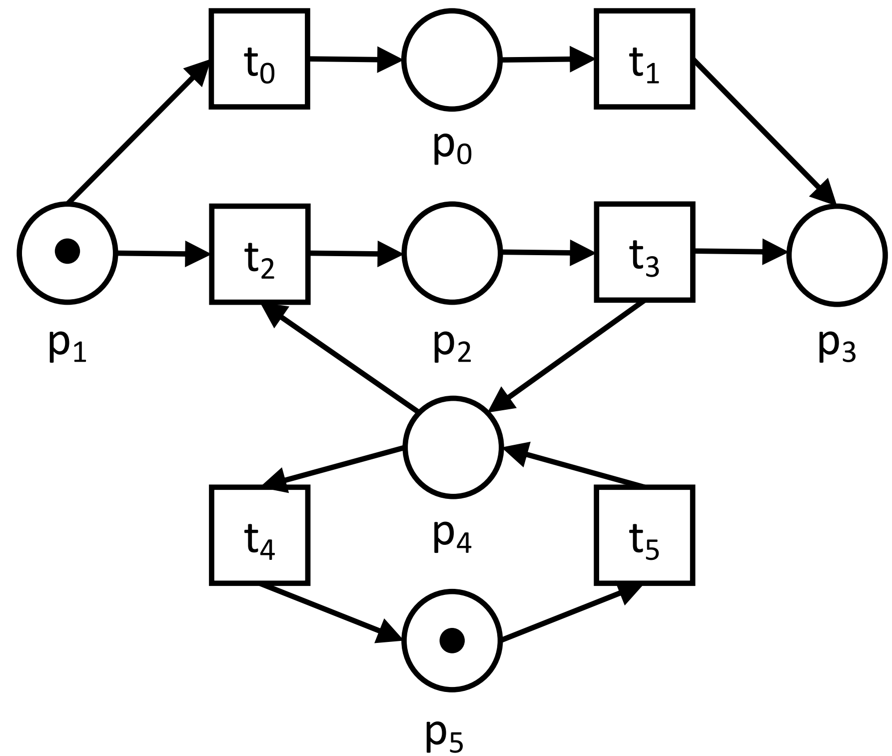
- After getting a solution P(w)
- We're looking for the maximal realizable firing sequence
- In the brute force tree we cut-off paths that violate ϕ



EF ψ guides the search and the balance constraints and the cut-off criterion are keeping ϕ true along the path.

Full example

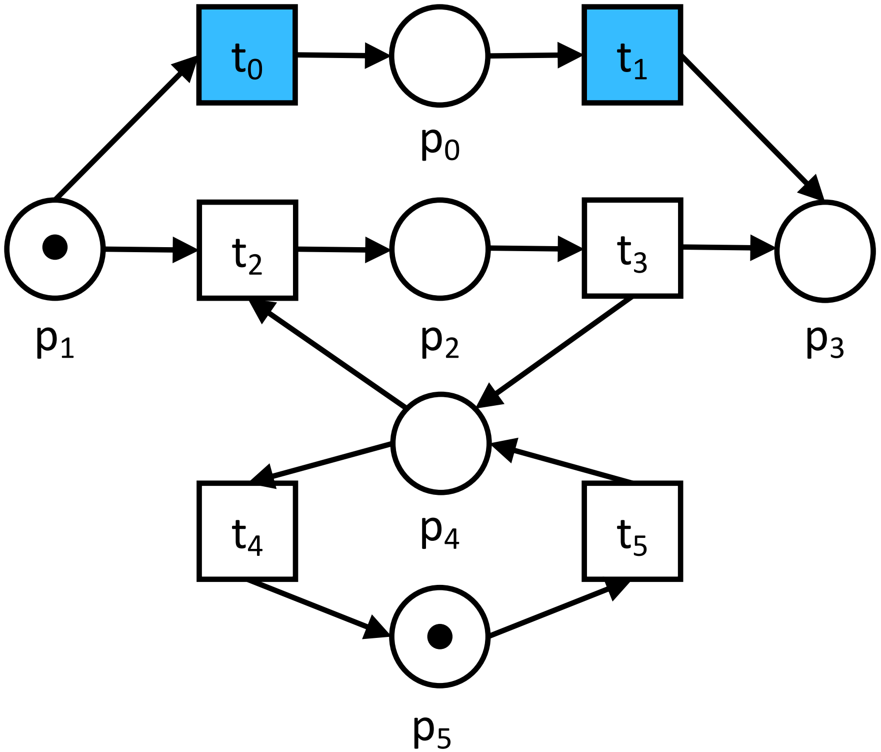
$E\ (p_1 + p_2 > 0)\ U\ (p_3 > 0)$



Full example

E $(p_1 + p_2 > 0)$ U $(p_3 > 0)$

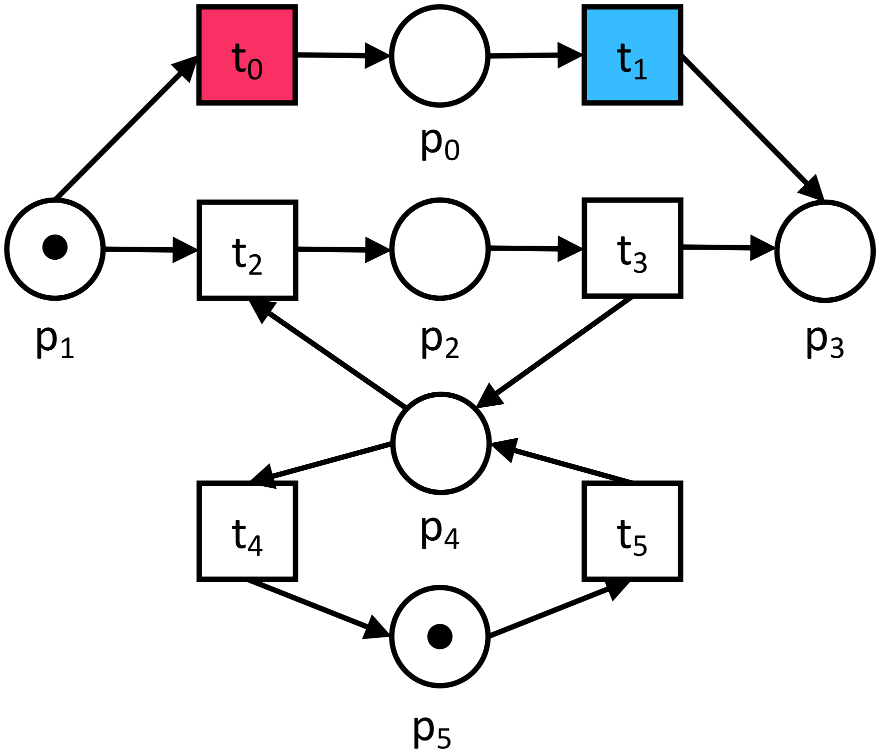
- Minimal solution: $t_0 t_1$



Full example

E $(p_1 + p_2 > 0)$ U $(p_3 > 0)$

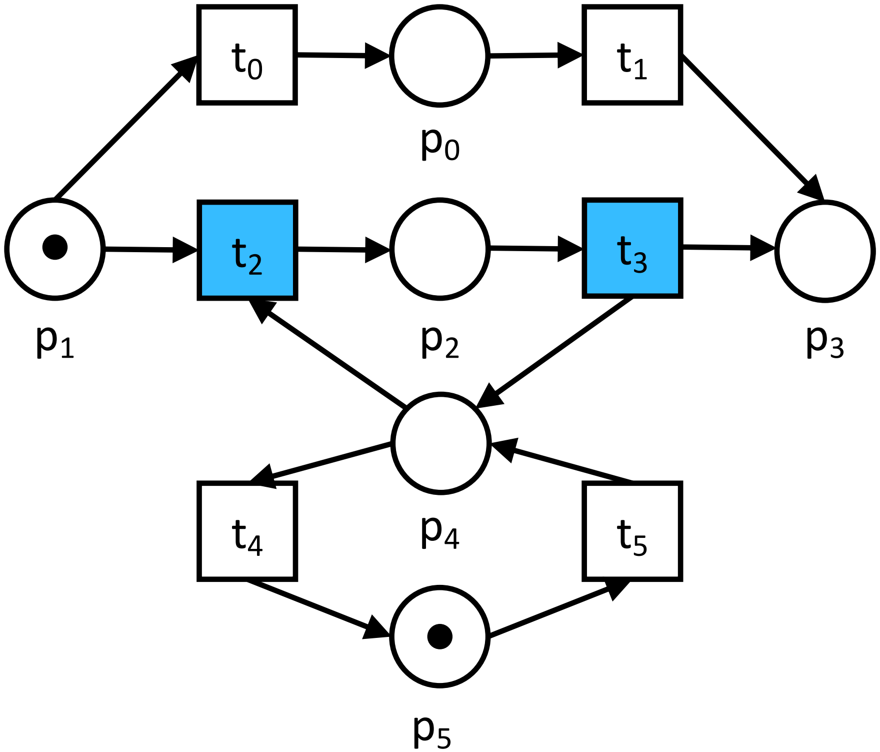
- Minimal solution: $t_0 t_1$
- Violates $(p_1 + p_2 > 0)$ and is cut off



Full example

E $(p_1 + p_2 > 0)$ U $(p_3 > 0)$

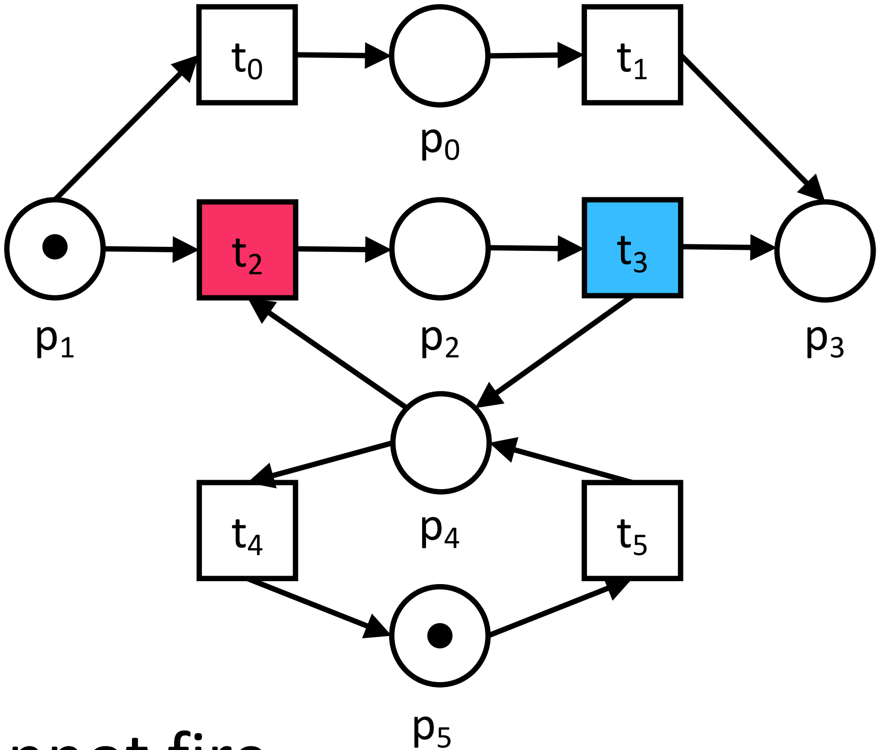
- Minimal solution: $t_0 t_1$
- Violates $(p_1 + p_2 > 0)$ and is cut off
- CEGAR: jump to next base solution: $t_2 t_3$



Full example

$E\ (p_1 + p_2 > 0)\ U\ (p_3 > 0)$

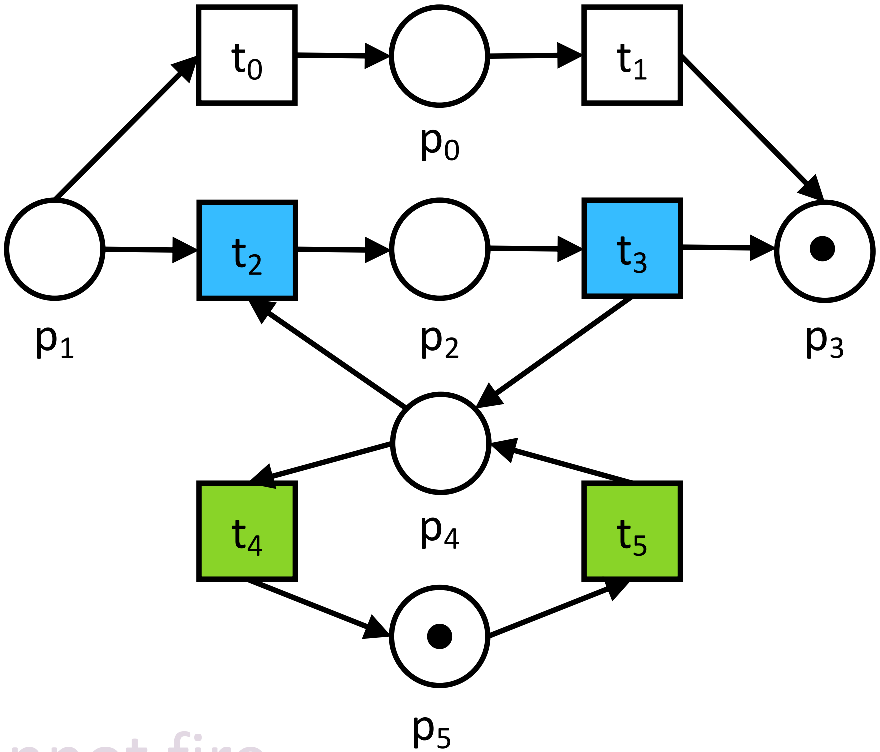
- Minimal solution: $t_0 t_1$
- Violates $(p_1 + p_2 > 0)$ and is cut off
- CEGAR: jump to next base solution: $t_2 t_3$
- Is only partial solution: t_2 cannot fire
- (also t_3 cannot fire)



Full example

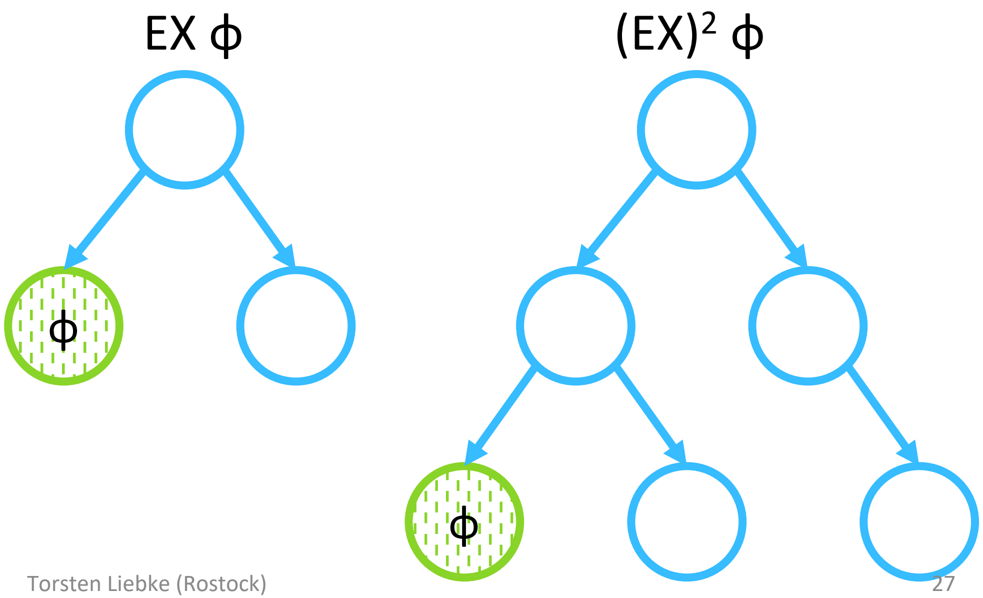
E $(p_1 + p_2 > 0)$ U $(p_3 > 0)$

- Minimal solution: t_0t_1
- Violates $(p_1 + p_2 > 0)$ and is cut off
- CEGAR: jump to next base solution: t_2t_3
- Is only partial solution: t_2 cannot fire
- CEGAR: increment solution with T-invariant t_4t_5
- Full solution: $t_5t_2t_3(t_4)$



$(EX)^k \phi$

- $(EX)^k$ ϕ – Exists a path, where ϕ holds in the k-th state?
- Add **length constraint**, which ensures, that the solution contains exactly k transitions:
- $\sum_{t \in T} P(w)|t| = k$



Future work

- Implementing it into LoLA
- We expect promising results, especially for negative results
- Could be a building brick
- Try to solve more complex formulas

Time for discussion!