Determining a Cut-off Size for the PGM example

- Aim: Show that there is k such that if a property holds for some instantiation, then it holds already for the instantiation with k copies.
- For PGM: For stuttering-free LTL safety properties speaking about the sender and one receiver, and about only one sequence number, it suffices to look at the model with two network elements and three receivers.
- For the untimed, abstract model.

Highlevel View

- Show that in the generic representation $T(\phi)$ of the backwards search for ϕ, it suffices to use k index variables.
- The generic representation $T(\phi)$ is a tree consisting of formulas, with the following property:

  $S(k') \models \phi \Longrightarrow$

  for the path π in $S(k')$ witnessing ϕ there is a path π' in $T(\phi)$ such that $\pi(i) \models \pi'(i)$ for all i.
- If $T(\phi)$ only uses k variables, then from the latter, it follows that if $k' > k$, then there is already a path π'' in $S(k)$ such that $\pi'' \models \phi$.
- It is possible to use additional termination conditions for the construction of $T(\phi)$ under the following condition: $T(\phi)$ without the additional termination condition is successful (ie. ϕ holds) iff $T(\phi)$ with additional termination condition is successful.

Construction of $T(\mathbf{EF}\, p)$

- Sets of states of the system are represented by index predicates: simple predicates over system variables that can be indexed by index variables or index terms. Index terms can be built up by taking prefix $pr(\sigma)$ (predecessors) or appending variables σj (successors).
- The set of states satisfying $\mathsf{EF}\, p$ is computed by least fixpoint computation for $F_{\mathsf{EF}\, p}(Q) = p \vee \widetilde{wp}.S.Q$,

  ($\widetilde{wp}.S.Q$ is the set of states that have a successor in Q),
- so the predicate representing this fixpoint is :

$$\mathsf{EF}\, p \Leftrightarrow p \vee \widetilde{wp}.S.p \vee \widetilde{wp}.S.(\widetilde{wp}.S.p) \vee \ldots$$

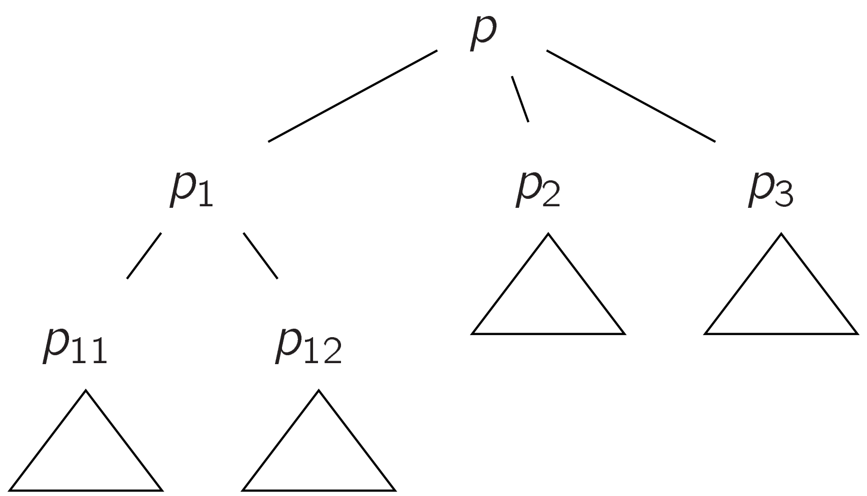
 Construction of $T(\mathrm{EF}\,p)$

- Building up this predicate is organized as a tree, where nodes are conjunctions of literals:

p

$\widetilde{wp}.S.p$

$\widetilde{wp}.S.p_1 \vee \widetilde{wp}.S.p_2 \vee \widetilde{wp}.S.p_3$
$= \widetilde{wp}.S.(p_1 \vee p_2 \vee p_3)$



- A node q is a leaf if:
  - There is another node q' such that $q \Rightarrow q'$.
  - $I \wedge q$ is satisfiable, where I is the predicate describing initial states. (➔ EF p holds, $T(\mathrm{EF}\,p)$ is successful).

Adaption of the tree construction to parameterized systems

- How to define $\widetilde{wp}.S.q$ for parameterized systems S?
- $\widetilde{wp}.S.p = \bigvee_{C \text{ comp. of S}} \widetilde{wp}.C.p$, so can compute $\widetilde{wp}$ componentwise.
- Assume q contains $t \neq t'$ for all index terms t' other than t in q. Then let $\widetilde{wp}.C[t].q$ be defined by the usual definition of $\widetilde{wp}$, considering index terms t, t' as constants.
- Consider the set $G(q)$ of conjunctions of inequations that describes all possibilities of index terms in q to be equal to $rep(g)$, some index term in q.

$$q \Leftrightarrow \bigvee_{g \in G(q)} (g \wedge q[g])$$

- Define:

$$\widetilde{wp}.S.q = \bigvee_{g \in G(q)} g \wedge \widetilde{wp}.C[\boldsymbol{rep}(g)].q[g]$$

- This is a uniform representation for all instantiations of the parameterized system:

  Let s be a state in the instantiation with n and v a valuation for the index variables in q.

  $s, v + w \models \widetilde{wp}.S.q$ iff there is a successor s' of s s.t. $s', v \models q$.

- Generic $T(\mathrm{EF}\, p)$:

  $S(k') \models \mathrm{EF}\, p \Longrightarrow$

  for the path π in $S(k')$ witnessing $\mathrm{EF}\, p$ there is a path π' in $T(\mathrm{EF}\, p)$ such that $\pi(i) \models \pi'(i)$ for all i.

Vertical parameterization in the PGM example

- It suffices to consider one layer of network elements:
- In $T(\mathrm{EF}\,p)$, nodes speaking about more network elements are redundant in the following sense:

1. For any satisfiable node $L' =$

$$p(\epsilon) \wedge q_1^0(0\sigma_1^0) \wedge \cdots \wedge q_{n_0}^0(0\sigma_{n_0}^0) \wedge \wedge \cdots q_0^k(k\sigma_0^k) \wedge \cdots \wedge q_{n_k}^k(k\sigma_{n_k}^k) \wedge \\ r_1^0(0\tau_1^0) \wedge \cdots \wedge r_m^k(k\tau_{m_k}^k)$$

   there is a satisfiable node $L =$

$$p(\epsilon) \wedge Q^0(0) \wedge \cdots Q^k(k) \wedge r_1^0(00) \wedge r_{m_0}^0(0m_0) \wedge \cdots \wedge r^k(k,0) \wedge \cdots \wedge r^k(km_k)$$

   such that
   $(q_1^i \wedge ..q_{n_0}^i)(i) \Longrightarrow Q^i(i)$.
2. Let I be the prediate describing initial states of network element:
   $I \wedge q_1^0, \ldots, I \wedge q_n^0$ satisfiable, then $I \wedge (q_1^0 \wedge ..q_{n_0}^0)$ satisfiable.

Conditions (1) and (2) can be proved by:

- considering which nodes containing new index variables are constructed during the tree construction
- observing that conjunction of these nodes are satisfiable
- observing that each of these node does not hold in an initial state of network element.

Network element, output nak

- in nak$(nr, j)(i_1)$
  - in nak$(nr, j')(i_1 i_2)$
  - $get(nr, I\_rep(i_1 i_2))$
    - in nak$(nr, j'')(i_1 i_2)$

- $get(nr, I\_rep(i_1))$
  - in nak$(nr, j)(i_1 i_2)$

Network element, output rdata:

- $\text{in rdata}(nr, trail)(pr(\sigma))$
 $\wedge(nr, last(\sigma)) \in I\_intf(pr(\sigma))$
  - $\text{in rdata}(nr, trail)(pr(\sigma))$
 $\wedge\text{in nak}(nr, last(\sigma))(pr(\sigma))$
    - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge(nr, last(pr(\sigma))) \in I\_intf(prpr(\sigma))$
 $\wedge\text{in nak}(nr, last(\sigma))(pr(\sigma))$
      - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge\text{in nak}(nr, last(pr(\sigma)))(prpr(\sigma))$
 $\wedge\text{in nak}(nr, last(\sigma))pr(\sigma)$
  - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge(nr, last(pr(\sigma))) \in I\_intf(prpr(\sigma))$
 $\wedge(nr, last(\sigma)) \in I\_intf(pr(\sigma))$
    - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge\text{in nak}(nr, last(pr(\sigma)))(prpr(\sigma))$
 $\wedge(nr, last(\sigma)) \in I\_intf(pr(\sigma))$
      - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge\text{in nak}(nr, last(\sigma))(pr(\sigma))$
      - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge get(nr, I\_rep(pr(\sigma)))$
 $\wedge(nr, last(\sigma)) \in I\_intf(pr(\sigma))$
        - $\text{in rdata}(nr, trail)(prpr(\sigma))$
 $\wedge\text{in nak}(nr, last(\sigma))(pr(\sigma))$

Horizontal parameterization

- Construction for $T(\phi)$ terminates for any safety property ϕ. (For every nr occurring in ϕ, no new nr').
- But we want to obtain a cut-off size:
- Consider safety formulas ϕ with atomic propositions built over atomic expressions

  in $\text{nak}(nr, i)(\epsilon)$, $nr \in rec\_nak(\epsilon)$, $nr \in l\_miss(\sigma)$, $nr \in l\_nr(\sigma)$, in $\text{rdata}(nr, trail)(\sigma)$, in $\text{odata}(nr, trail)(\sigma)$, in $\text{ncf}(nr)(\sigma)$:

  These are properties speaking about one receiver and only one message nr.
- For such properties, it suffices to consider the topology:

  Sender, one network element with two receivers and one network element with one receiver as successors.

$$\text{in nak}(nr, i)(\epsilon)$$

$$\text{in nak}(nr, i'')(i)$$

$$get(nr, I\_miss(i_1 i_2)) \wedge nr > RXW\_TR(i_1 i_2)$$

$$get(nr, I\_miss(i_1 i'')) \wedge nr > RXW\_TR(i_1 i'')$$

$$get(nr, I\_miss(i' i'')) \wedge nr > RXW\_TR(i' i'')$$

$$get(nr, l\_miss(i_1 i_2)) \wedge nr > RXW\_TR(i_1 i_2)$$

$$\begin{array}{c} \text{in odata}(nr', trail)(i_1) \\ \wedge RXW\_TR(i_1 i_2) < nr' \\ \wedge nr' > nr > trail \\ \wedge nr \notin l\_nr(i_1 i_2) \end{array}$$

$$\begin{array}{c} \text{in rdata}(nr', trail)(i_1 i_2) \\ \wedge RXW\_TR(i_1 i_2) < nr' \\ \wedge nr' > nr > trail \\ \wedge nr \notin l\_nr(i_1 i_2) \end{array}$$

$$\begin{array}{c} \text{in ncf}(nr)(i_1 i_2) \\ \wedge RXW\_TR(i_1 i_2) < nr \\ \wedge nr \notin l\_nr(i_1 i_2) \end{array}$$

$$\begin{array}{c} \text{in nak}(nr, i')(i_1) \\ \wedge RXW\_TR(i_1 i_2) < nr \\ \wedge nr \notin l\_nr(i_1 i_2) \end{array}$$

$$\begin{array}{c} get(nr, l\_miss(i_1 i')) \\ \wedge nr > RXW\_TR(i_1 i') \\ \wedge RXW\_TR(i_1 i_2) < nr \\ \wedge nr \notin l\_nr(i_1 i_2) \end{array}$$

$$
\begin{gathered}
\text{in ncf}(nr)(i_1 i') \\
\wedge nr > RXW\_TR(i_1 i') \\
\wedge RXW\_TR(i_1 i_2) < nr \\
\wedge nr \notin I\_nr(i_1 i_2) \\
\wedge nr \notin I\_nr(i_1 i') \\
X
\end{gathered}
$$

To Do

Can this method be applied for liveness properties?

Construction of $T(\mathrm{EG}\, p)$, but termination conditions are:

A node y is a leaf if:

- There is a predecessor x such that the path between x and y is fair and $I(x) \Rightarrow I(y)$.
- There is a node x such that $I(x) \Rightarrow I(y)$.

This makes it more difficult to argue that nodes with certain properties do not have to be considered.