Ultimate Automizer algorithm

Program $\mathcal{P}$ is correct because each error trace is infeasible, i.e. the inclusion $\mathcal{P} \subseteq \mathcal{A}_1 \cup \mathcal{A}_2$ holds.

Program / automaton $\mathcal{P}$ whose language is the set of error traces:

1. Alphabet: set of program statements
   $\Sigma = \{ p = 0, x < 0, x > 0, p = 0, n = 0, n > 0, x = 0, y = 0, x = y \}$
2. The language of $\mathcal{P}$ is the set of error traces.
3. In the first iteration, we analyze feasibility of the error trace $\tau_1 = \langle p = 0, x > 0, p = 0 \rangle$. $\tau_1$ is infeasible. Via interpolation, we obtain the following Hoare triples.
4. We construct the automaton $\mathcal{A}_1$ such that its language is the set of all traces whose infeasibility can be shown using the predicates $\text{true}$ and $\text{false}$.
5. Analogously, in the second iteration the automaton $\mathcal{A}_2$ is constructed.
6. We check the inclusion $\mathcal{P} \subseteq \mathcal{A}_1 \cup \mathcal{A}_2$ and conclude that each error trace is infeasible and hence $\mathcal{P}$ is correct.

### Automata-theoretic proof of program correctness

**Program** $\mathcal{P}$ **whose language is a set of infeasible traces**

1. **Definition** Given an automaton $A = (Q, \delta, q_{\text{init}}, Q_{\text{final}})$ over the alphabet of program statements, we call a mapping that assigns to each state $q \in Q$ a predicate $\varphi_q$ a Floyd-Hoare annotation for automaton $A$ if the following implications hold.

   $q, q' \in Q \land q \delta q' \Rightarrow \{ \varphi_q \} \{ \varphi_{q'} \}$ is a valid Hoare triple

2. **Theorem** If an automaton $A$ has a Floyd-Hoare annotation, then $A$ recognizes a set of infeasible traces.

### Interpolation with unsatisfiable cores

**Level 1: “interpolation” via strongest post**

- strongest post
- live variable analysis

**Level 2: interpolation via strongest post**

- strongest post
- live variable analysis
- unsatisfiable cores

**Level 3: interpolation via unsatisfiable cores**

- strongest post
- live variable analysis
- unsatisfiable cores

**Algorithm (for level 3)**

1. **Input:** infeasible trace $\mathbf{t}_1, \ldots, \mathbf{t}_n$ and unsatisfiable core $UC \subseteq \{ \mathbf{t}_1, \ldots, \mathbf{t}_n \}$
2. Replace each statement that does not occur in $UC$ by a skip statement or a havoc statement.
3. Compute sequence of predicates $\varphi_0, \ldots, \varphi_n$ iteratively using the strongest post predicate transformer. $sp$
4. Eliminate each variable from predicate $\varphi_i$ that is not live at position $i$ of the trace.
5. Output: sequence of predicates $\varphi_0, \ldots, \varphi_n$ which is a sequence of interpolants for the infeasible trace $\mathbf{t}_1, \ldots, \mathbf{t}_n$. 