Temporal Specification of Properties

Formula φ is $\mathcal{D}$-valid, denoted $\mathcal{D} \models \varphi$, if all initial states of $\mathcal{D}$ satisfy φ. Such a formula specifies a property of $\mathcal{D}$.

Following is a temporal specification of the main properties of program MUX-SEM.

$$y: \textbf{natural initially } y = 1$$

$$P_1 :: \left[\begin{array}{l} \ell_0 : \ \textbf{loop forever do} \\ \left[\begin{array}{ll} \ell_1 : & \textbf{Non-critical} \\ \ell_2 : & \textbf{request } y \\ \ell_3 : & \textbf{Critical} \\ \ell_4 : & \textbf{release } y \end{array} \right] \end{array} \right] \quad \| \quad P_2 :: \left[\begin{array}{l} m_0 : \ \textbf{loop forever do} \\ \left[\begin{array}{ll} m_1 : & \textbf{Non-critical} \\ m_2 : & \textbf{request } y \\ m_3 : & \textbf{Critical} \\ m_4 : & \textbf{release } y \end{array} \right] \end{array} \right]$$

- Mutual Exclusion – No computation of the program can include a state in which process P_1 is at ℓ_3 while P_2 is at m_3. Specifiable by the formula

$$\Box\, \neg(at\_\ell_3 \ \wedge \ at\_m_3)$$

- Accessibility for P_1 – Whenever process P_1 is at ℓ_2, it shall eventually reach it's critical section at ℓ_3. Specifiable by the formula

$$\Box\, (at\_\ell_2 \quad \rightarrow \quad \Diamond\, at\_\ell_3)$$

Model Checking

This is a process by which we algorithmically check that a given finite state FDS D satisfies its temporal specification φ. There are two approaches to this process:

- Enumerative (explicit state) approach, by which we construct a graph containing all the reachable states of the system, and then apply graph theoretic algorithms to its analysis.
- Symbolic approach, by which we continuously work with assertions which characterize sets of states.

Here, we will consider the symbolic approach. Note that every assertion over a finite-domain FDS can be represented as a boolean formula over boolean variables. We assume that a finite-state FDS is represented by such formulas, including the initial condition Θ and the bi-assertion ρ representing the transition relation.

We assume that we have an efficient representation of boolean assertions, and efficient algorithms for manipulation of such assertions, including all the boolean operations as well as existential and universal quantification. Note that, for a boolean variable b,

$$\exists b : \varphi(b) = \varphi(0) \ \vee \ \varphi(1) \qquad\qquad \forall b : \varphi(b) = \varphi(0) \ \wedge \ \varphi(1)$$

Also assume that we can efficiently check whether a given assertion is valid, i.e., equivalent to 1.

Successors and Their Transitive Closure

For an assertions $\varphi(V)$ and a bi-assertion $R(V, V')$, we define the existential successor predicate transformer:

$$\varphi \diamond R \quad = \quad unprime(\exists V : \varphi(V) \wedge R(V, V'))$$

Obviously

$$\|\varphi \diamond R\| \quad = \quad \{s \mid s \text{ is an } R\text{-successor of a } \varphi\text{-state}\}$$

For example

$$\begin{array}{lcl} (x = 0) \diamond (x' = x + 1) & = & unprime(\exists x : x = 0 \ \wedge \ x' = x + 1) \quad \sim \\ unprime(x' = 1) & \sim & x = 1 \end{array}$$

The immediate successor transformer can be iterated to yield the eventual successful transformer:

$$\varphi \diamond R^* \quad =$$
$$\varphi \ \vee \ \varphi \diamond R \ \vee \ (\varphi \diamond R) \diamond R \ \vee \ ((\varphi \diamond R) \diamond R) \diamond R \ \vee \ \cdots$$

Predecessors and Their Transitive Closure

For an assertions $\varphi(V)$ and a bi-assertion $R(V, V')$, we define the existential predecessor predicate transformer:

$$R \diamond \psi \quad = \quad \exists V' : R(V, V') \land \psi(V')$$

Obviously

$$\|R \diamond \varphi\| \quad = \quad \{s \mid s \text{ is an } R\text{-predecessor of a } \varphi\text{-state}\}$$

For example

$$(x' = x + 1) \diamond (x = 1) \quad = \quad \exists x' : x' = x + 1 \ \land \ x' = 1 \quad \sim \quad x = 0$$

The immediate predecessor transformer can be iterated to yield the eventual predecessor transformer:

$$R^* \diamond \varphi \quad =$$
$$\varphi \ \lor \ R \diamond \varphi \ \lor \ R \diamond (R \diamond \varphi) \ \lor \ R \diamond (R \diamond (R \diamond \varphi)) \ \lor \ \cdots$$

Fix-points

Let Ω be a set of elements. We consider set functions $f : 2^\Omega \to 2^\Omega$. For example, if $\Omega = \mathbb{N}$ we can define the set function $A + 1 = \{j + 1 \mid j \in A\}$ mapping a set A of naturals into the set of their successors. Similarly, we can define $A \ominus 1 = \{j \mid j + 1 \in A\}$.

A fix-point equation is an equation of the form

$$X = f(X),$$

where f is a set function and X is an unknown variable ranging over subsets of Ω. In a similar way, we can define fix-points inclusions of the forms $X \subseteq f(X)$ and $X \supseteq f(X)$.

Not every fix-point equation has a unique solution. Some, such as $X = \Omega - X$ have no solutions at all, while others, such as $X = X$ have many solutions. In fact every set $T \subseteq \Omega$ is a solution.

A set function f is called monotonic if $S_1 \subseteq S_2$ implies $f(S_1) \subseteq f(S_2)$. Restricting our attention to monotonic functions, the situation is much better.

The Knaster-Tarski Theorem

A set $T \subseteq \Omega$ is said to be a minimal solution of the fix-point equation $X = f(X)$, if T is a solution (i.e., $T = f(T)$), and T is a subset of any other solution $U = f(U)$.

Claim 1. *(Knaster-Tarski) For a monotonic set function f, the fix-point equation $X = f(X)$ has a unique minimal solution T.*

Furthermore, for the case that Ω is finite, T can be obtained as the union of the chain

$$T = \bigcup_{i=0,1,\ldots} f^i(\emptyset),$$

where $f^0(\emptyset) = \emptyset$, and $f^{i+1}(\emptyset) = f(f^i(\emptyset))$.

The minimal solution of the equation $X = f(X)$ is denoted by $\mu X.f(X)$.

For example:

$$\begin{array}{lcl} \mu X.(\{5\} \cup X+1) & = & \{5, 6, \ldots\} \\ \mu X.(\{5\} \cup X-1) & = & \{0..5\} \end{array}$$

Using fix-point notation, we can represent $\varphi \diamond R^*$ as $\mu\phi.(\varphi \vee \phi \diamond R)$ and $R^* \diamond \varphi$ as $\mu\phi.(\varphi \vee R \diamond \phi)$.

Maximal Fix-points

In a completely analogous way, we can define maximal fix-points. The appropriate variant of Knaster-Tarski will state that, for a monotonic function f, the equation $X = f(X)$ always has a unique maximal solution which, for a finite Ω is given by the intersection

$$\nu X.f(X) \quad = \quad \bigcap_{i=0,1,\ldots} f^i(\Omega)$$

An alternative approach to maximal fix-points defines

$$\nu X.f(X) \quad = \quad \overline{\mu X.\overline{f(\overline{X})}},$$

where $\overline{A} = \Omega - A$.

For example, defining $2 \cdot X = \{2x \mid x \in X\}$, we have

$$\nu X.(\{1\} \ \cup \ X \cap (2 \cdot X)) \quad = \quad \{2^n \mid n \geq 0\}$$

Model Checking Invariance Properties

Claim 2. [Invariance] *Property* $\square\, p$ *is valid over* FDS $\mathcal{D}$ *iff* $(\Theta_{\mathcal{D}} \diamond \rho^*_{\mathcal{D}}) \rightarrow p$ *is valid* *iff* $\Theta_{\mathcal{D}} \wedge (\rho^*_{\mathcal{D}} \diamond \neg p)$ *is invalid.*

As an example, we can use the following algorithm:

Algorithm INV $(\mathcal{D}, p)$: **assertion** — Check that FDS $\mathcal{D}$ satisfies $\square\, p$

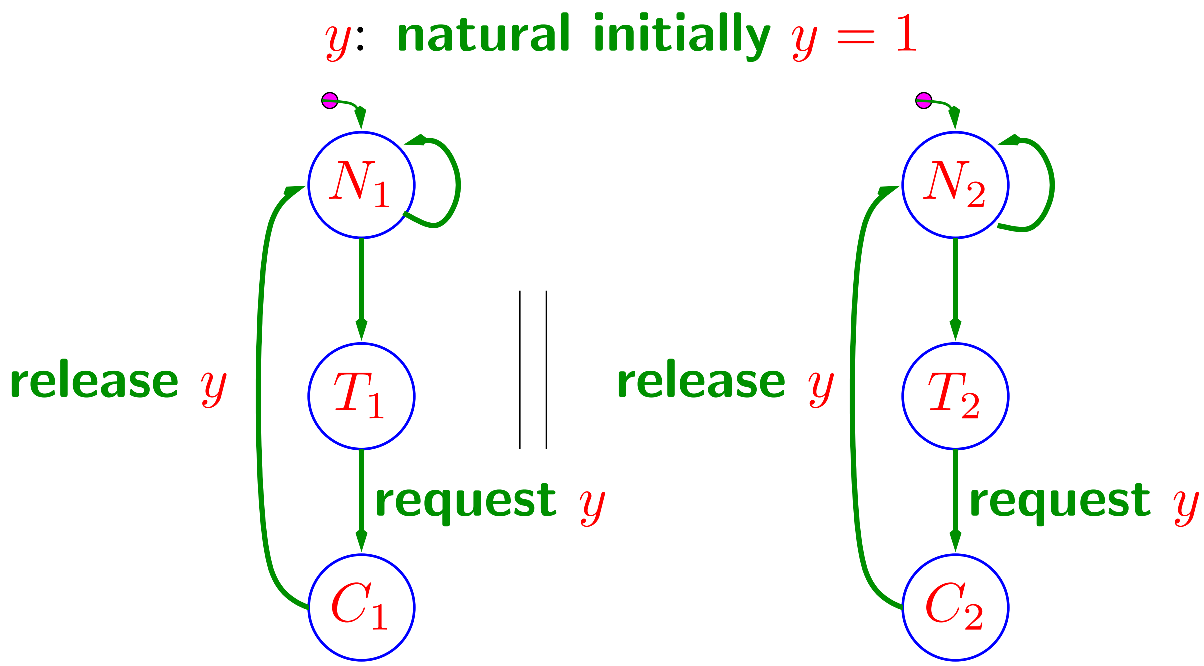
new, old : **assertion**

1. $old := 0$
2. $new := \neg p$
3. **while** $(new \neq old \wedge (\Theta_{\mathcal{D}} \wedge new = 0))$ **do**
   begin
4. $\quad old := new$
5. $\quad new := new \vee (\rho_{\mathcal{D}} \diamond new)$
   end
6. **return** $\Theta_{\mathcal{D}} \wedge new$

The algorithm returns an assertion characterizing all the initial states from which there exists a finite path leading to violation of p. It returns the empty (false) assertion iff $\mathcal{D}$ satisfies $\square\, p$.

Example: a Simpler MUX-SEM

Below, we present a simpler version of program MUX-SEM.



The semaphore instructions **request** y and **release** y respectively stand for

$$\langle \textbf{when } y = 1 \textbf{ do } y := 0 \rangle \quad \text{and} \quad y := 1.$$

Illustrate Backwards Exploration on MUX-SEM

We iterate as follows:

$$\varphi_0: \quad \pi_1 = C \wedge \pi_2 = C$$

$$\varphi_1: \quad \varphi_0 \vee \begin{pmatrix} \cdots \\ \vee\, \pi_1 = T \wedge y = 1 \wedge \pi_1' = C \wedge y' = 0 \\ \vee\, \pi_2 = T \wedge y = 1 \wedge \pi_2' = C \wedge y' = 0 \end{pmatrix} \diamond\, (\pi_1 = \pi_2 = C)$$

$$\sim$$

$$\pi_1 = \pi_2 = C \vee \pi_1 = T \wedge \pi_2 = C \wedge y = 1 \vee \pi_1 = C \wedge \pi_2 = T \wedge y = 1$$

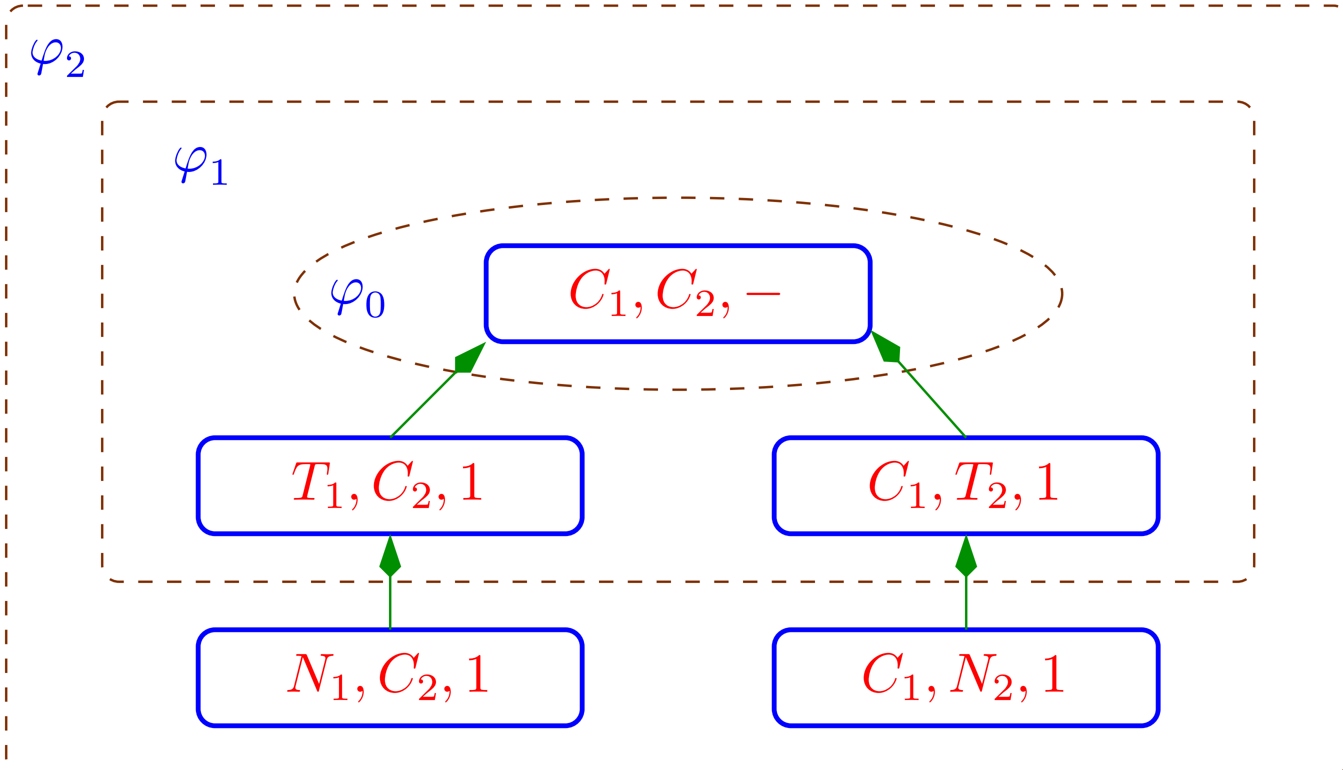
$$\varphi_2: \quad \varphi_1 \;\vee\; (\pi_1 = N \wedge \pi_2 = C \wedge y = 1 \vee \pi_1 = C \wedge \pi_2 = N \wedge y = 1)$$

$$\varphi_3: \quad \varphi_2 \;\vee\; (\pi_1 = C \wedge \pi_2 = C \wedge y = 0) \quad \sim \quad \varphi_2$$

The last equivalence is due to the general property $p \vee (p \wedge q) \sim p$.

If we intersect φ_3 with the initial condition $\Theta : \pi_1 = N \wedge \pi_2 = N \wedge y = 1$ we obtain 0 (false). We conclude that MUX-SEM satisfies $\square\, (\neg(\pi_1 = C \wedge \pi_2 = C))$.

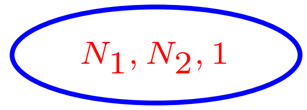
Symbolic Exploration Progresses in Layers



Illustrate Forward Exploration on MUX-SEM

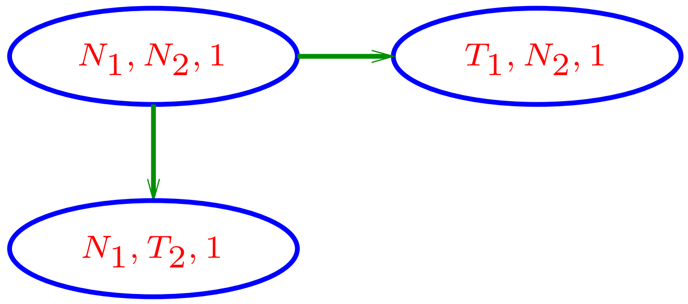
We iterate as follows:

φ_0 :



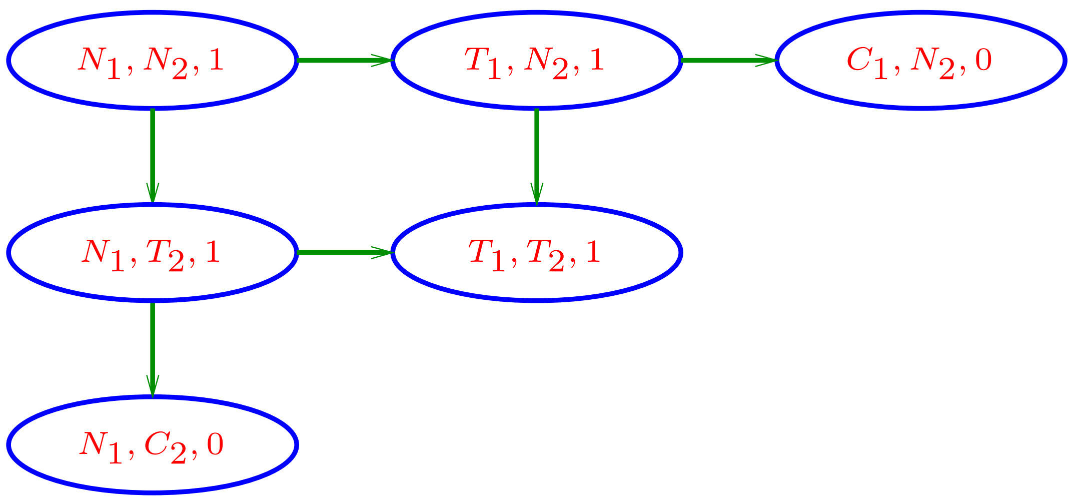
Iteration 1:

φ_1 :



Iteration 2:

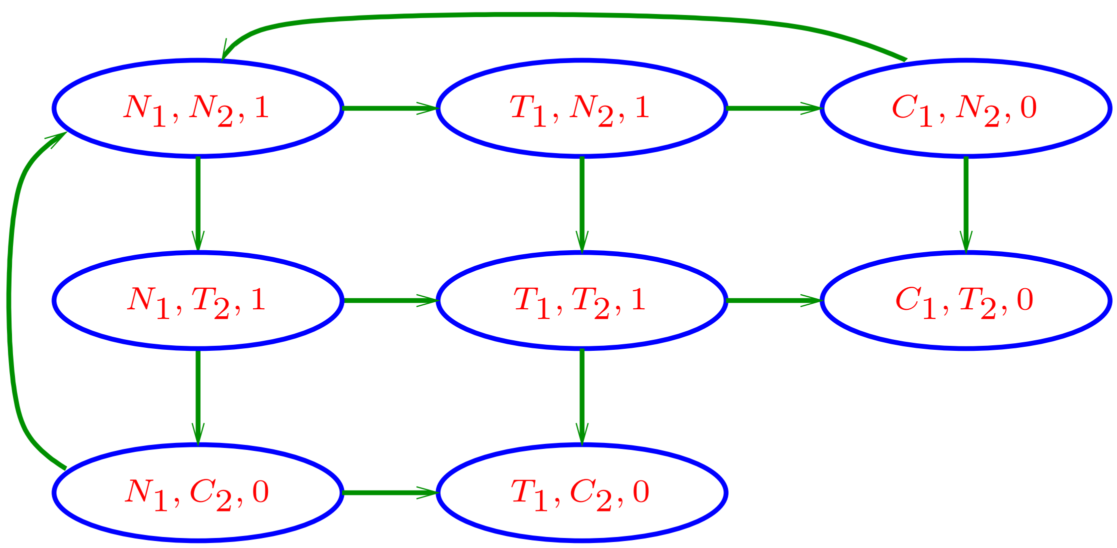
φ_2 :



Forward Exploration Continued

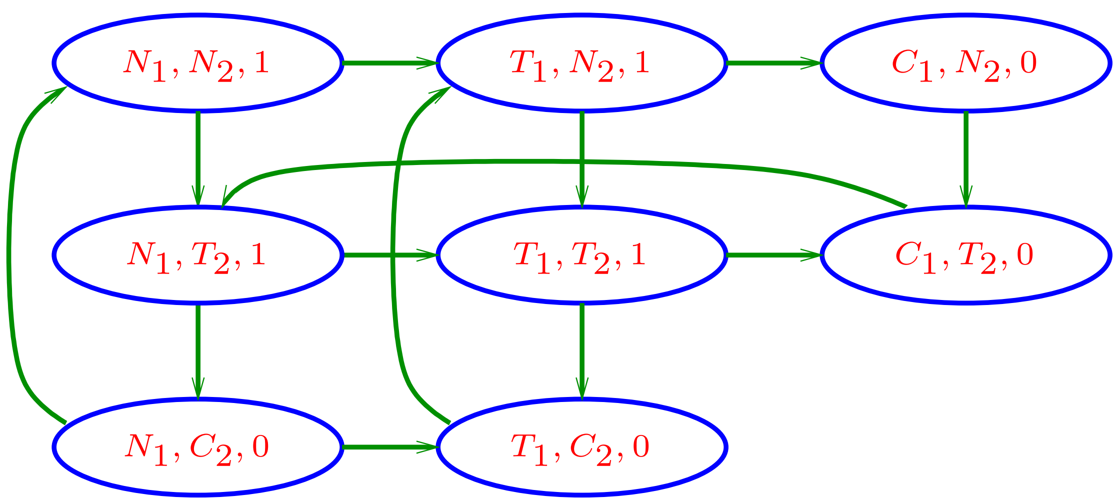
Iteration 3:

φ_3 :



Iteration 4 (Convergent):

φ_4 :



This last iteration has an empty intersection with $C_1 \ \wedge \ C_2$. We conclude $\square \ \neg(C_1 \ \wedge \ C_2)$.