Specification Language

Having described the computational model of clocked transition systems and the implementation language of timed SPL, it only remain to fix the specification language. Here we take good old LTL with the additional license of referring to all clocks in the system.

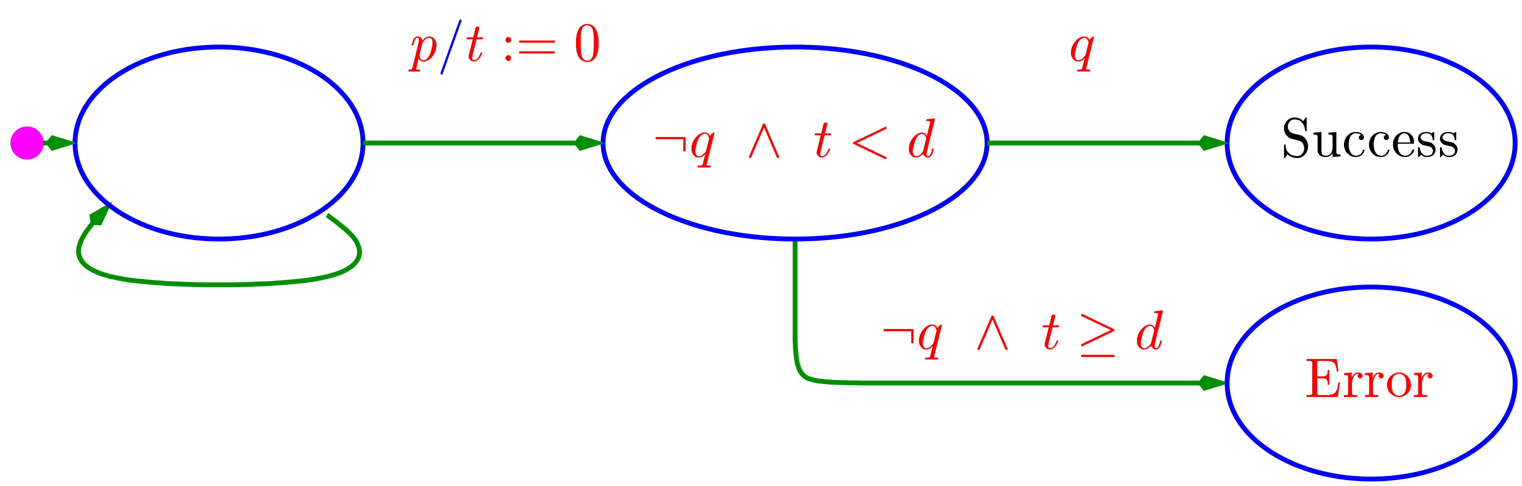
It is obvious how we can use LTL in order to specify untimed properties of timed systems. For timed properties, we will illustrate some of the most important ones:

- **Bounded Response**: Every p should be followed by an occurrence of q, not later than d time units.

This can be specified by each of the two following LTL formulas:

$$
\begin{array}{lcl}
p \ \wedge \ (T = t_0) & \Rightarrow & \diamondsuit\, (q \ \wedge \ T \leq t_0 + d) \\
p \ \wedge \ (T = t_0) & \Rightarrow & (T \leq t_0 + d)\ \mathcal{W}\ q
\end{array}
$$

In the case of model checking, the reference to the free variable t_0 is not convenient. Instead, we can augment the system with the following observer:



and then verify $\square\ \neg$Error.

Minimal Separation

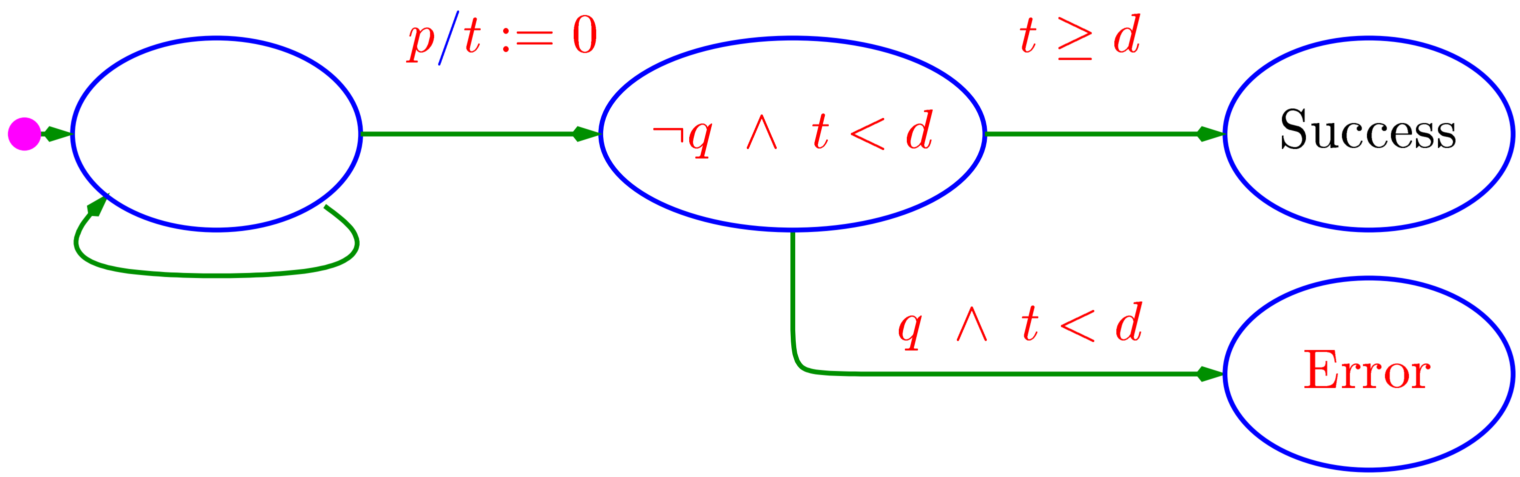
Another important timed property is:

- Minimal Separation: No q can occur earlier than d time units after an occurrence of p.

This can be specified by the following LTL formula:

$$p \ \wedge \ (T = t_0) \quad \Rightarrow \quad (\neg q)\, \mathcal{W}\, (T \geq t_0 + d)$$

Again, in the context of model checking, we can construct the following observer:



An Additional Example: Program UP-DOWN

Consider the following program UP-DOWN:

$$x, y: \textbf{ integer where } x = y = 0$$

$$\left[\begin{array}{l} \ell_0: \quad \textbf{while } x = 0 \textbf{ do} \\ \quad \ell_1: \ y := y + 1 \\ \ell_2: \quad \textbf{while } y > 0 \textbf{ do} \\ \quad \ell_3: \ y := y - 1 \\ \ell_4: \end{array}\right] \quad \| \quad \left[\begin{array}{l} m_0: \quad x := 1 \\ m_1: \end{array}\right]$$

$$- \ P_1 \ - \qquad\qquad - \ P_2 \ -$$

Assume we assign to it the time bounds $[1, 5]$. We wish to prove for this program the two properties:

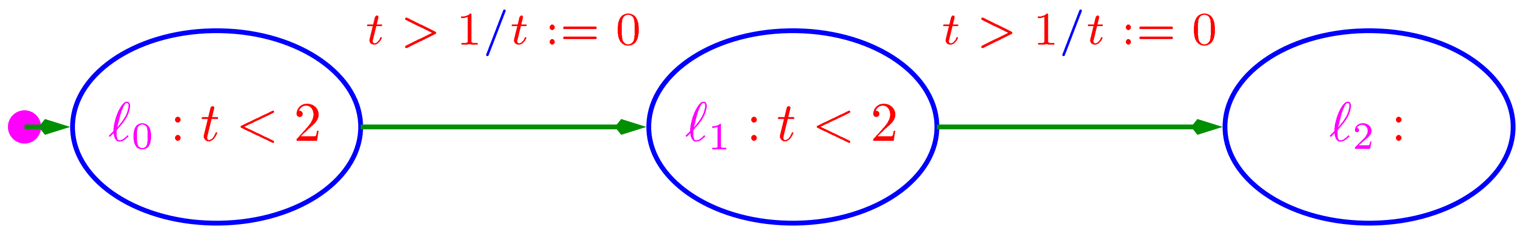
$\square \ (y + at\_\ell_1 \leq 3)$
$\diamond \ (at\_\ell_4 \ \wedge \ at\_m_1 \ \wedge \ T \leq 50)$

What is special about this program is that it contradicts the naive assumption that, in order to generate a behavior with the worst execution time, every process should proceed at the slowest pace possible. Here, in the initial steps, P_2 should proceed at its slowest pace, while P_1 should rush forward at maximal speed.

Files `updn.smv` and `updn.pf` are available on the course web page.

Dense Time

Obviously, the use of integer time may lead to distortions which can be sensed even with integer constraints. The system Φ_4



satisfies the property $\square\ (T \leq 3 \rightarrow at\_\ell_{0,1})$ under the integer-time model. However, under a dense-time model, the system can reach location ℓ_2 at time $T = 3$.

We conclude that, to reach a better precision, we must use dense time. The main problem is that the dense-time model no longer leads to finite-state systems.

Therefore, we will develop special methods which will enable us to deal with systems whose discrete part is finite state while its clocks vary over a dense domain. This leads us into the model of timed automata of [Alur & Dill].

Timed Automata

A timed automaton is a CTS with the following restrictions:

- The discrete variables range over finite domains.
- The time dependent component of the transition relations and the progress conditions, are formed as boolean combinations of inequalities of the form $t_i \sim c_i$ or $t_i - t_j \sim c_{ij}$ where $\sim \in \{<, \leq, >, \geq\}$ and c_i,c_{ij} are natural numbers.
- The only modifications to clocks by non-tick transitions are resets to 0.

Symbolic Representation

Recall that the state variables are partitioned into $V = D \cup C$. We assume that the discrete variables D range over finite domains. Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ be the set of different valuations that the variables in D can assume. For example, for system Φ_4, $\mathcal{D} = \{0, 1, 2\}$ are the three possible values that the single discrete variable π can assume. We can represent the transition relation as

$$\rho(D, C, D', C') \quad = \quad \bigvee_{d_i, d_j \in \mathcal{D}} D = d_i \ \wedge \ D' = d_j \ \wedge \ \rho_{ij}(C, C'),$$

where, for each $d_i, d_j \in \mathcal{D}$,

$$\rho_{ij}(C, C') \quad = \quad g_{ij}(C) \ \wedge \ C' = r_{ij}(C)$$

In this presentation, $g_{ij}(C)$ is a guard specifying a condition on the current values of the clocks under which a transition from d_i to d_j is allowed. The function r_{ij} is a reset function ensuring that, for each $t_k \in C$ either $r_{ij}(t_k) = 0$ or $r_{ij}(t_k) = t_k$. For example, for Φ_4,

$$\rho_{01} \quad = \quad \underbrace{t > 1}_{g_{01}} \wedge \underbrace{(t', T') = (0, T)}_{C' = r_{01}(C)}$$

The *tick* Transition

In a similar way, we can decompose the *tick* transition into the disjunction

$$\rho_{tick}(D,C,D',C') \quad = \quad \bigvee_{d_i \in \mathcal{D}} D = D' = d_i \ \land \ \rho^i_{tick}(C,C'),$$

where, for each $d_i \in \mathcal{D}$,

$$\rho^i_{tick}(C,C') \quad = \quad \exists \Delta > 0 : p_i(C+\Delta) \ \land \ C' = C + \Delta.$$

For example, for Φ_4,

$$\rho^0_{tick}(C,C'): \quad \exists \Delta > 0 : \underbrace{t+\Delta \leq 2}_{p_0} \ \land \ \underbrace{(t',T') = (t+\Delta, T+\Delta)}_{C'=C+\Delta}$$

A formula is called k-polyhedral if it is a boolean combination of atomic formulas of the forms $t_i \# c$ or $t_i - t_j \# c$, where the relation $\# \in \{<, \leq, >, \geq\}$ and $c \in \{0, \ldots, k\}$.

We restrict our attention to systems such that, for some $k \geq 0$, and each $d_i, d_j \in \mathcal{D}$, the guards $g_{ij}(C)$ and the progress conditions $p_i(C)$ are k-polyhedral.

An assertion $\varphi(D,C)$ is called k-admissible if there exists a decomposition

$$\varphi(D,C): \quad \bigvee_{d_i \in \mathcal{D}} D = d_i \ \land \ \psi_i(C)$$

such that each $\psi_i(C)$ is k-polyhedral.

The Main Result

The main result which is the basis for symbolic model-checking of dense-time systems is stated by

Claim 12. *Closure of k-admissible Assertions*
If φ is a k-admissible assertion, then so is its $\rho \vee \rho_{tick}$-predecessor.

In order to prove the claim, it is sufficient to show that if $\psi(C)$ is k-polyhedral, then so are its ρ_{ij}- and ρ^i_{tick}-predecessors, for every $d_i, d_j \in \mathcal{D}$.

The general computation of a predecessor is based on the formula:

$$\exists C' : \rho(C, C') \;\wedge\; \psi(C').$$

By expanding all formulas into DNF form and observing that existential quantification distributes over disjunctions, we see that it is sufficient to consider the case that ρ and ψ are conjunctions of k-atomic formulas.

Consider first the case that $\rho = \rho_{ij}(C, C')$. In that case, the predecessor is given by

$$\exists C' : g_{ij}(C) \;\wedge\; \psi(C') \;\wedge\; \bigwedge_{t_i \in C} t'_i = r_{ij}(t_i),$$

which can be simplified to

$$g_{ij}(C) \;\wedge\; \psi(r_{ij}(C))$$

Proof Continued

Next, consider the case that $\rho = \rho^i_{tick}(C, C')$. In this case, the predecessor is given by

$$\exists C' : \exists \Delta > 0 : p_i(C + \Delta) \ \wedge \ C' = C + \Delta \ \wedge \ \psi(C').$$

which can be simplified into

$$\exists \Delta > 0 : p_i(C + \Delta) \ \wedge \ \psi(C + \Delta)$$

Let us examine the effect that the replacement of C by $C + \Delta$ has on the various types of atomic formulas.

For formulas of the form $t_i - t_j \# c$, this replacement has no effect, because the addition of Δ is canceled.

A formula of the form $t_i \# c$ is changed into $t_i + \Delta \# c$, which can be rewritten as either $\Delta \prec c - t_i$ or $c - t_i \prec \Delta$, for $\prec \in \{<, \leq\}$. To obtain a uniform representation, we rewrite $\Delta > 0$ as $t_0 < \Delta$, where t_0 is an artificial clock having the constant value 0.

We form a new set of constraints S as follows:

- Each original constraint $t_i - t_j \# c$ is placed in S.

- For each pair of constraints $c_i - t_i \prec_i \Delta$ and $\Delta \prec_j c_j - t_j$, we place in S the constraint $c_i - c_j \prec t_i - t_j$ if $c_i \geq c_j$ or the constraint $t_j - t_i \prec c_j - c_i$ if $c_i < c_j$. In both cases, $\prec$ is taken to be strict ($<$) iff one of $\prec_i$ or $\prec_j$ is strict.

Finally, we substitute 0 for all occurrences of t_0. The conjunction of all constraints within S is the ρ^i_{tick}-predecessor of ψ. It is not difficult to see that this conjunction is k-polyhedral.

A Simplified Presentation

For the case that the time-progress condition has the form $p_i(t) : t^i \leq E_i$, we can simplify further the computation of the ρ-predecessor and ρ_{tick}-predecessor into:

$$\begin{array}{rl} \rho_{ij} \diamond \psi : & p_i(C) \ \wedge \ g_{ij}(C) \ \wedge \ \psi(r_{ij}(C)) \\ \rho^i_{tick} \diamond \psi : & \exists \Delta > 0 : \psi(C + \Delta) \end{array}$$

Thus, the time-progress condition p_i is moved from the computation of the ρ^i_{tick}-predecessor to the computation of the ρ_{ij}-predecessor.

Usually, we compute first $\varphi_i = \rho_{ij} \diamond \psi_j$ and then compute $\psi_i = \rho^i_{tick} \diamond \varphi_i$. This can be combined into a single computation $\psi_i = \rho^i_{tick} \diamond (\rho_{ij} \diamond \psi_j)$, given by

$$\psi_i : \quad \exists \Delta \geq 0 : p_i(C + \Delta) \ \wedge \ g_{ij}(C + \Delta) \ \wedge \ \psi_j(r_{ij}(C + \Delta))$$

This presupposes that, being in discrete state d_i, we let first time elapse for Δ time units, and then take the transition to discrete state d_j.

Working an Example

Let us apply this approach in order to check whether system Φ_4 can reach location ℓ_2 at time $T \leq 3$, thus violating the property $\square\ (T \leq 3 \rightarrow at\_\ell_{0,1})$ which is valid for Φ_4 under the integer-time model.

The goal state set is given by

$$\varphi_2^1: \quad at\_\ell_2 \ \wedge \ T \leq 3$$

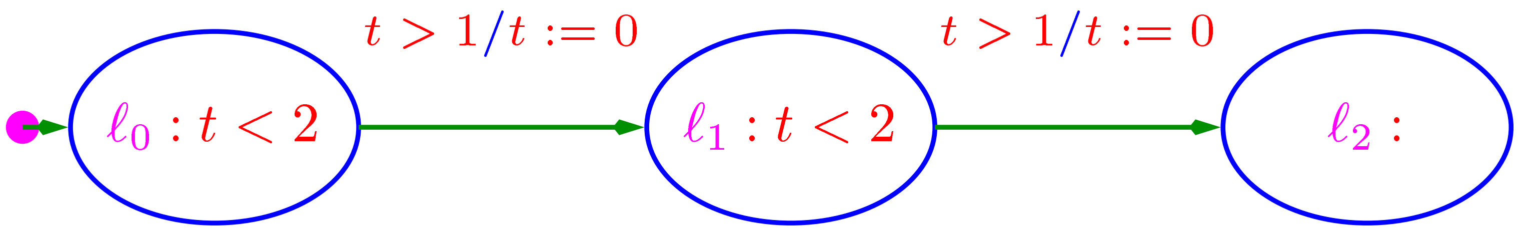
We first compute the predecessor of φ_2^1 by the $tick$ transition. This is computed by the formula $\exists \Delta > 0 : p_2(C{+}\Delta) \wedge \varphi_2^1(C{+}\Delta)$. Since p_2 the time-progress condition for location ℓ_2 is 1 (True), this simplifies first to $at\_\ell_2 \ \wedge \ \exists \Delta > 0 : T + \Delta \leq 3$ and then, finally to

$$\varphi_2^2: \qquad at\_\ell_2 \ \wedge \ T < 3$$

Taking the disjunction of φ_2^1 and φ_2^2 which share the same location, we obtain

$$\psi_2: \qquad at\_\ell_2 \ \wedge \ T \leq 3$$

Computation Continued



Next, we compute the predecessor of $\psi_2 : at\_\ell_2 \ \wedge \ T \leq 3$ along the discrete transition ρ_{12}. Using the combined formula for $\rho^i_{tick} \diamond (\rho_{12} \diamond \psi_2)$, we obtain

$$\psi_1 : \quad at\_\ell_1 \ \wedge \ \exists\Delta \geq 0 : 1 < t + \Delta \leq 2 \ \wedge \ T + \Delta \leq 3$$

As a first step in the Fourier-Motzkin elimination process, we rewrite the inequalities as:

$$\begin{array}{rcccl} 0 & \leq & \Delta & \leq & 3 - T \\ 1 - t & < & \Delta & \leq & 2 - t \end{array}$$

Eliminating Δ, we obtain

$$\psi_1 : \quad at\_\ell_1 \ \wedge \ T \leq 3 \ \wedge \ T - 2 < t \leq 2$$

Computing φ_0

The timed 01-predecessor of $\psi_1 : at\_\ell_1 \ \wedge \ T \leq 3 \ \wedge \ T - 2 < t \leq 2$ is computed as follows:

$$\begin{array}{rl} \psi_1(r_{01}(C+\Delta)): & at\_\ell_1 \ \wedge \ T + \Delta < 2 \\ \psi_0: & at\_\ell_0 \ \wedge \ \exists \Delta \geq 0 : T + \Delta < 2 \ \wedge \ 1 < t + \Delta \leq 2 \end{array}$$

Eliminating Δ, we obtain

$$\psi_0 : \quad at\_\ell_0 \ \wedge \ T < 2 \ \wedge \ T - 1 < t \leq 2$$

Since the initial condition $\Theta : at\_\ell_0 \ \wedge \ t = T = 0$ has a non-empty intersection with ψ_0, we conclude that Φ_4 has a computation reaching location ℓ_2 with $T \leq 3$. It follows that the property $\square \ (T \leq 3 \rightarrow at\_\ell_{0,1})$ is not valid for Φ_4 under the dense-time model.