並行分散計算特論 (12)

Shoji Yuen

2011/12/20

Timed regular language

A timed language L is *timed regular* if L is accepted by a timed Büchi automaton.

Examples of non-regular languages:

$$\{((ab)^\omega, \tau) | \tau_{4i-2} - \tau_{4i-3} < \tau_{4i} - \tau_{4i-1}\}$$
$$\{((a)^\omega, \tau) | \tau_i = 2i\}$$

No memory in time

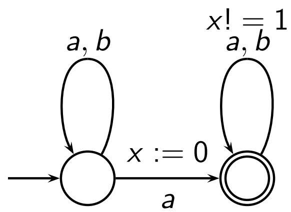
Closure property

Untimed case: A regular language is closed under *union*, *intersection*, *complement*, (*determinization*)

Timed case: Closed under *union*, *intersection*
Complementation: Generally No
Determinization: Generally No

Complementation

Timed Büchi automata not closed under complementation.



Sequences not accepted: $\exists i \exists j. i < j \wedge (a, \tau_i) \wedge \tau_j - \tau_i = 1$

Must check infinitely many pairs of (i, j) of either (a, τ_i) and (a, τ_j), or (a, τ_i) and (b, τ_j).

This cannot be implemented with finite states.

If complementation is closed

Language inclusion problem:

$$L_1 \subseteq L_2 \text{ is equivalent to } L_1 \cap \overline{L_2} = \emptyset$$

What is the subclass closed under complementation?

Verification

Check if the behavior of implementation satisfies the desied property.

Safety: A system does nothing wrong.
Liveness: A system does right thing.

Formal language approach:
Safety: $L(I) \subseteq L(S)$
Liveness: $L(I||S) \neq \emptyset$

Emptiness Checking

Set the state as a final state. Check if the language is empty or not.

Definition
For $\mathcal{A}$, check if its timed language $L(A)$ is empty or not.

Decidable in PSPACE complete.
Exponential wrt the number of clocks

Clock region

Time constraint: $\delta ::= x \leq c | c \leq x | \neg\delta | \delta_1 \wedge \delta_2$
where $c \in \mathcal{Q}$. $\mathcal{Q}$:Ratinal numbers.

Definition
$\nu \sim \nu'$ *if*

(1) $\forall x. \lfloor \nu(x) \rfloor = \lfloor \nu'(x) \rfloor$ *or* $(\nu(x) \geq c_x and \nu'(x) \geq c_x$

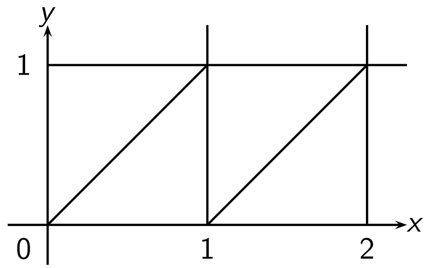
(2) $\forall x, y.$, *when* $\nu(x) \leq c_x$ *and* $\nu(y) \leq c_y$
$\text{fract}(\nu(x)) \leq \text{fract}(\nu(y))$ *iff* $\text{fract}(\nu'(x)) \leq \text{fract}(\nu'(y))$

(3) $\forall x.\text{fract}(\nu(x)) = 0$ *iff* $\text{fract}(\nu'(x)) = 0$

c_x : *biggest constant in δ of x*

Definition
Clock region: $I_c / \sim$ *where* $I_c = [C \rightarrow R_{\geq 0}]$

Clock region



$c_x = 2, c_y = 1$

6 corner points
14 open line segments
8 open regions

Rational numbers→Integers

For a region $\alpha, \alpha \models \delta$ if for all $\nu \in \alpha, \nu \models \delta$

upper bound of regions:$|C|! \cdot 2^{|C|} \cdot \prod_{x \in C}(2c_x + 2)$

c_x:maximum constant in time constraints of x

Region automaton

$R(\mathcal{A})$Region automaton of A

Definition

$\mathcal{A} = \langle \Sigma, S, S_0, C, E \rangle$, $R(\mathcal{A}) = \langle \Sigma, S_R, S_{R0}, E_R \rangle$

$$
\begin{aligned}
S_R &= \{(s, \alpha) | s \in S, \alpha \textit{ is a clock region}\} \\
S_{R0} &= \{(s_0, [\nu_0]) | s_0 \in S_0\} \\
E_R &= \{\langle (s, \alpha), a, (s', \alpha') \rangle | \langle s, s', a, \lambda, \delta \rangle \in E \\
& \quad \exists \alpha''. \alpha'' \textit{ is a time successor of } \alpha,\ \alpha'' \models \delta \textit{ and } \alpha' = [\lambda \to 0]
\end{aligned}
$$

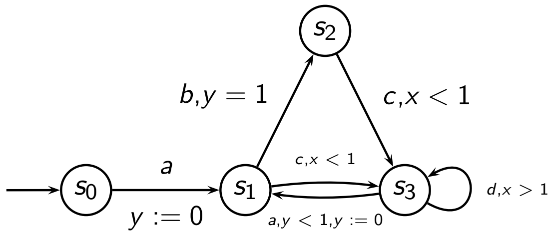
Time successors

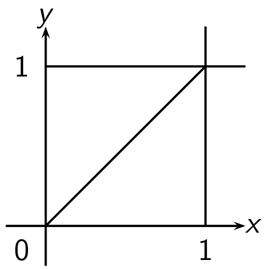
Definition

α' is a time successor of α if
$\forall \nu \in \alpha. \exists t \in R. \nu + t \in \alpha'$

For a time successor β of α, if $\alpha \models \delta$ implies $\beta \models \delta$, then β is identified with α.

Example





$\langle s_0, a, T, \{y\}, s_1 \rangle$:
time successor of $[x = y = 0]$:
$[0 < x = y < 1]$,$[x = y = 1]$,$[x > 1, y > 1]$
time successor:region reached by time passage

Region automata