並行分散計算特論 (14)

Shoji Yuen

2012/01/17

Logical Characterization

- Linear Temporal Logic
  Specification for the computation path
  Property description for a sequence of transitions
- Branching Temporal Logic
  Specification for the computation tree
  Property description for all possible tree of transitions

LTL states the properties about a path:
A state that satisfies p is always followed by a states that satisfies q

BTL sates the properties about branches:
All transitions from a state that satisfies p lead to the states that satisfy q

Propositional Temporal Logic

$$F \quad ::= \quad p \mid \neg F \mid F \wedge F \mid \bigcirc F \mid F \mathcal{U} F$$

Temporal operators

$$\begin{array}{rcl} \bigcirc\phi & : & \phi \text{ holds after the next transition.} \\ \phi_1 \mathcal{U} \phi_2 & : & \phi_1 \text{ holds until } \phi_2 \text{ holds.} \end{array}$$

$$\begin{array}{rcl} \sigma \models p & \text{iff} & p \in \sigma^0 \\ \sigma \models \neg\phi & \text{iff} & \sigma \not\models \phi \\ \sigma \models \phi_1 \wedge \phi_2 & \text{iff} & \sigma \models \phi_1 \text{ and } \sigma \models \phi_2 \\ \sigma \models \bigcirc\phi & \text{iff} & \sigma^1 \models \phi \\ \sigma \models \phi_1 \mathcal{U} \phi_2 & \text{iff} & \text{For some } j \geq 0. \sigma^j \models \phi_2 \\ & & \text{and } \sigma^k \models \phi_1 \text{ for all } 0 \leq k < j \end{array}$$

PTL specification

$$\Diamond\phi \equiv \mathtt{true}\ \mathcal{U}\phi \qquad \Box\phi \equiv \neg(\mathtt{true}\ \mathcal{U}\neg\phi)$$

$\Box(p \to \Diamond q)$: state p is always followed by state q.

Satisfiability of PTL-formulas is checked in PSPACE-complete.

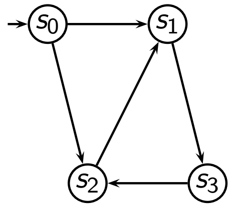
Model: Kripke structure $M = \langle S, S_0, \mu, E \rangle$

$$\begin{array}{rcl} S & : & \text{States} \\ S_0 \subseteq S & : & \text{Initial states} \\ \mu & : & S \to 2^{AP} \\ E \subseteq S \times S & : & \textit{Transitions} \end{array}$$

Check: Check if path(s) of M satisfy property ϕ.
Usually the paths beginning from an initial state.

Timed PTL

With specification when a transition may occur



ρ:$(S_0, I_0) \rightarrow (S_1, I_1) \rightarrow (S_2, I_2) \rightarrow$
I_i: interval I_i is adjacent to I_{i+1}

Interval:$[a, b]$,$[a, b)$,$[a, \infty)$,$(a, b]$,(a, b),(a, ∞)
[,] for closed intervals, (,) for open intervals

notation: ρ^t: State sequence after time t.

$$t \in I_i, \rho^t = (\sigma_i, I_i - t) \rightarrow (\sigma_{i+1, I_{i+1} - t}) \rightarrow \cdots$$

MITL(metric interval temporal logic)[Alur90]

MITL formula

$$F \ ::= \ p \mid \neg F \mid F \wedge F \mid F \ \mathcal{U}_I \ F$$

MITL semantics $\rho = (\overline{\sigma}, \overline{I})$:timed state sequence.

$$\begin{array}{rcl}
\rho \models p & \text{iff} & p \in \sigma^0 \\
\rho \models \neg\phi & \text{iff} & p \not\models \phi \\
\rho \models \phi_1 \wedge \phi_2 & \text{iff} & \rho \models \phi_1 \text{ and } \rho \models \phi_2 \\
\rho \models \phi_1 \ \mathcal{U}_I \phi_2 & \text{iff} & \text{for some } t \in I . \sigma^t \models \phi_2 \\
 & & \sigma^{t'} \models \text{for all } t' \in (0, t)
\end{array}$$

Derived operators

$$\Diamond_I \phi \equiv \texttt{true} \ \mathcal{U}_I \phi \qquad \Box_I \phi \equiv \neg \Diamond_I \neg \phi$$
$$\phi_1 \ {}_I\mathcal{U} \phi_2 \equiv \neg((\neg\phi_2) \ \mathcal{U}_I (\neg\phi_1))$$ ϕ_1 unless ϕ_2

What do these formulas mean?

$$p \wedge \Box_{\geq 0}(p \rightarrow (\neg) \ \mathcal{U}_{(0,\infty)} p).$$

$$\Box_{(0,\infty)}(p \rightarrow \Diamond_{[0,3)} q).$$

Rational time satisfiability

Definition

$(\overline{\sigma}, \overline{I})$ *is* rational *if the end-points of all intervals in* $\overline{I}$ *are rational.*

ρ *$\mathcal{Q}$-satifies* ϕ *iff* $\rho \models_{\mathcal{Q}} \phi$ *where all time variables range over only* $\mathcal{Q}$

A MITL formula ϕ *is* $\mathcal{Q}$*-satisfiable iff it is satisfiable.*

Undecidability[Alur90]

Proposition

If singular intervals are allowed as subscripts for temporal operators in MITL, then the satisfiability problem is undecidable.

Singular interval:$[a, a]$

If no singular interval is allowed,

$\Box(p \rightarrow \Diamond_{=5} q)$ is not in MITL
$\Box(p \rightarrow (\neg q)\ \mathcal{U}_{=5} q)$ is in MITL

Satisfiability

Theorem

The satisfiability of MITL (with no singlular interval) is EXPSPACE-complete.

Alur's algorithm:$O(2^{N \cdot K \cdot log(N \cdot K)})$

K: largest integer constant in ϕ+1
N: number of propositions, boolean connectives and temporal operators in ϕ.

Model checking TA with MITL

For a timed automaton A, check if $L_T(A) \subseteq L(\phi)$

Construct a TA that accepts $L(A) \cap L(B_{\neg\phi})$ and check its emptiness.

Construction of $A \times B_{\neg\phi}$ is exponential for $|\phi|$. Checking emptiness is PSPACE-complete.

A minor extension for timed automata to deal with intervals.

More on Timed Automata

Timed Safety Automata[Henzinger94]
Timed automata with state invariants.
State invariants:time constraints on states
All states are Büchi accepting.
Effective in checking safety properties. All safety properties can be checked in the finite prefix.

Zone construction[BengtssonYi04]
More efficient than regions Symbolic timed model checking.
Regions $\Rightarrow$ Zones (difference constrains)
Bounded Difference Matrix(BDM):Efficient data structure

Uppaal incorporates these effective verfication method.[BengtssonYi04]

Other timed model

Timed I/O automata [Kaynar, Lynch, Segara, Vaandrager05]

Process calculi with time

- Realtime ACP [Baeten91]
- Timed CCS [Yi90]
- Timed CSP [Schneider00]
- ATP [Nicolin, Sifaxis90]
- TPL [Regan, Hennessy95]

SOS characterization with time

OSOS with time tick [UlidowskiYuen04]

Summary

- Concurrent computation model
  A set of automata with synchronization
- Process calculus(CCS)
  Algebraic characterization based on bisimulations
- Timed extension
  Time as a synchronization measure
  A thoery of timed automaton and its extension

Await for time-aware application.