Alcino Cunha

SPECIFICATION AND MODELING

CTL MODEL CHECKING

COMPUTATION TREE LOGIC

SYNTAX

$$
\begin{aligned}
\phi \quad ::= \quad & p \\
& | \quad \top \qquad\quad | \qquad \bot \\
& | \quad \neg\phi \\
& | \quad \phi_1 \wedge \phi_2 \quad | \quad \phi_1 \vee \phi_2 \\
& | \quad \phi_1 \rightarrow \phi_2 \\
& | \quad \text{AG}\,\phi \qquad | \quad \text{EG}\,\phi \\
& | \quad \text{AF}\,\phi \qquad | \quad \text{EF}\,\phi \\
& | \quad \text{AX}\,\phi \qquad | \quad \text{EX}\,\phi \\
& | \quad \phi \,\text{AU}\, \psi \quad | \quad \phi \,\text{EU}\, \psi \\
& | \quad \phi \,\text{AR}\, \psi \quad | \quad \phi \,\text{ER}\, \psi
\end{aligned}
$$

SEMANTICS

- Defined over a model (transition system) M
- Technically, M should be a *Kripke structure* (S, I, R, L)
  - S is a finite set of states
  - $I \subseteq S$ is the set of initial states
  - $R \subseteq S \times S$ is a total transition relation
  - $L : S \rightarrow 2^A$ is a function that labels each state with the set of atomic propositions valid in that state (draw from domain A)
- A formula is valid iff it holds in all initial states

$$M \models \phi \quad \text{iff} \quad \forall s \in I \cdot M, s \models \phi$$

- A formula is valid in a state iff it holds in the (infinite) computation tree unrolled from that state

MINIMAL SYNTAX

- All CTL formulas can be expressed using $\top$, $\neg$, $\vee$, EX, EU, and EG

$$\bot \equiv \neg\top$$
$$\phi \wedge \psi \equiv \neg(\neg\phi \vee \neg\psi)$$
$$\phi \rightarrow \psi \equiv \neg\phi \vee \psi$$
$$AX\,\phi \equiv \neg\,EX\,\neg\phi$$
$$EF\,\phi \equiv \top\,EU\,\phi$$
$$AG\,\phi \equiv \neg\,EF\,\neg\phi$$
$$AF\,\phi \equiv \neg\,EG\,\neg\phi$$
$$\phi\,AR\,\psi \equiv \neg(\neg\phi\,EU\,\neg\psi)$$
$$\phi\,ER\,\psi \equiv EG\,\psi \vee (\psi\,EU\,(\phi \wedge \psi))$$
$$\phi\,AU\,\psi \equiv \neg(\neg\phi\,ER\,\neg\psi)$$

MODEL CHECKING

(UNBOUNDED) MODEL CHECKING

- Given a Kripke structure $M = (S, I, R, L)$ and a temporal formula ϕ, the goal of a model checking procedure is to find the set of all states in M that satisfy ϕ

$$\llbracket \phi \rrbracket_M \equiv \{s \in S \mid M, s \models \phi\}$$

EXPLICIT VS SYMBOLIC MODEL CHECKING

Explicit model checking

- Sets and transitions are encoded extensionally
- Semantics of temporal operators is implemented by graph traversals

$$M \models \phi \quad \text{iff} \quad I \subseteq \llbracket \phi \rrbracket_M$$

Symbolic model checking

- Sets and transitions are encoded intentionally by propositional formulas
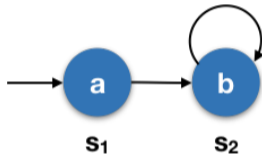- Semantics of temporal operators is implemented by fixpoint computations

$$M \models \phi \quad \text{iff} \quad I \rightarrow \llbracket \phi \rrbracket_M$$

EXPLICIT CTL MODEL CHECKING

EXPLICIT REPRESENTATION OF THE KRIPKE STRUCTURE

$$I = \{s_1\}$$
$$R = \{(s_1, s_2), (s_2, s_2)\}$$

PROPOSITIONAL CONNECTIVES AND NEXT

$$\llbracket p \rrbracket = \{s \in S \mid p \in L(s)\}$$
$$\llbracket \top \rrbracket = S$$
$$\llbracket \neg \phi \rrbracket = S - \llbracket \phi \rrbracket$$
$$\llbracket \phi \vee \psi \rrbracket = \llbracket \phi \rrbracket \cup \llbracket \psi \rrbracket$$
$$\llbracket \mathsf{EX}\,\phi \rrbracket = \{s \in S \mid \exists s' \in \llbracket \phi \rrbracket \cdot (s, s') \in R\} = R.\llbracket \phi \rrbracket$$

UNTIL

$$
\begin{aligned}
&[\![\phi \ \text{EU} \ \psi]\!] = \\
&\quad T \leftarrow [\![\psi]\!] \\
&\quad U \leftarrow [\![\psi]\!] \\
&\quad \textbf{while} \ T \neq \varnothing \\
&\quad\quad \textbf{choose} \ s \in T \\
&\quad\quad T \leftarrow T - \{s\} \\
&\quad\quad \textbf{for} \ t \in R.s \\
&\quad\quad\quad \textbf{if} \ t \notin U \wedge t \in [\![\phi]\!] \\
&\quad\quad\quad\quad U \leftarrow U \cup \{t\} \\
&\quad\quad\quad\quad T \leftarrow T \cup \{t\} \\
&\quad \textbf{return} \ U
\end{aligned}
$$

ALWAYS

- To determine $[\![EG\,\phi]\!]_M$ it suffices to restrict M to the states that satisfy ϕ

$$M_\phi = ([\![\phi]\!], I \cap [\![\phi]\!], R \cap ([\![\phi]\!] \times [\![\phi]\!]), L \cap ([\![\phi]\!] \times A))$$

- $M, s \models EG\ \phi$ iff $s \in [\![\phi]\!]$ and there exists a path in M_ϕ from s to some node in a *nontrivial strongly connected component* of M_ϕ
- A *Strongly Connected Component* (SCC) is a maximal subgraph where every node is reachable from every other
- A SCC is also *nontrivial* if it has at least one path (more than one node or one node with a self-loop)
- The nontrivial SCCs of M_ϕ can be computed efficiently with Tarjan's algorithm

$$\mathbf{scc}(M_\phi) \subseteq 2^{[\![\phi]\!]}$$

ALWAYS

$$
\begin{aligned}
&\llbracket \mathsf{EG}\, \phi \rrbracket = \\
&\quad T \leftarrow \cup_{C \in \mathbf{scc}(M_\phi)} \\
&\quad G \leftarrow T \\
&\quad \textbf{while } T \neq \varnothing \\
&\qquad \textbf{choose } s \in T \\
&\qquad T \leftarrow T - \{s\} \\
&\qquad \textbf{for } t \in R_\phi . s \\
&\qquad\quad \textbf{if } t \notin G \\
&\qquad\qquad G \leftarrow G \cup \{t\} \\
&\qquad\qquad T \leftarrow T \cup \{t\} \\
&\quad \textbf{return } G
\end{aligned}
$$

FAIRNESS

- A typical fairness constraint imposes φ to be recurrently true
- This constraint cannot be expressed in CTL
- CTL semantics must be adapted to handle fairness

$$M \models \overline{\phi} \quad \text{iff} \quad M \models \phi \text{ and } \varphi \text{ is recurrently true in } M$$

- To model check $\overline{\text{EG}\,\phi}$ it suffices to restrict the model to fair SCCs, namely those where φ is valid in some state
- Given that
  ▶ A path is fair iff any of its suffixes is fair
  ▶ $\overline{\text{EG}\,\top}$ holds in a state iff there is a fair path starting at that state
  to model check $\overline{\phi \text{ EU } \psi}$ it suffices to check $\overline{\phi} \text{ EU } (\overline{\psi} \wedge \overline{\text{EG}\,\top})$ instead
- Similarly for the remaining operators

SYMBOLIC CTL MODEL CHECKING

SYMBOLIC REPRESENTATION OF THE KRIPKE STRUCTURE
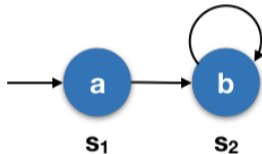
$$I = a \wedge \neg b$$
$$R = (a \wedge \neg b \wedge \neg a' \wedge b') \vee (\neg a \wedge b \wedge \neg a' \wedge b')$$

PROPOSITIONAL CONNECTIVES

$$[\![p]\!] = p$$
$$[\![\top]\!] = \top$$
$$[\![\neg\phi]\!] = \neg[\![\phi]\!]$$
$$[\![\phi \lor \psi]\!] = [\![\phi]\!] \lor [\![\psi]\!]$$

ALWAYS

- From the EG expansion rule

$$EG\,\phi \equiv \phi \land EX(EG\,\phi)$$

a greatest fixpoint algorithm can be directly obtained

$$[\![EG\,\phi]\!] = \nu Z . \phi \land [\![EX\,Z]\!]$$

$$
\begin{aligned}
&[\![EG\ \phi]\!] = \\
&\quad G \leftarrow \top \\
&\quad \textbf{repeat} \\
&\quad\quad G' \leftarrow G \\
&\quad\quad G \leftarrow [\![\phi]\!] \land [\![EX\,G]\!] \\
&\quad \textbf{until}\ G \equiv G' \\
&\quad \textbf{return}\ G
\end{aligned}
$$

UNTIL

- From the EU expansion rule

$$\phi \text{ EU } \psi \equiv \psi \vee (\phi \wedge \text{EX}(\phi \text{ EU } \psi))$$

a smallest fixpoint algorithm can be directly obtained

$$[\![\phi \text{ EU } \psi]\!] = \mu Z \,.\, \psi \vee (\phi \wedge [\![\text{EX } Z]\!])$$

$$[\![\phi \text{ EU } \psi]\!] =$$
$$\quad U \leftarrow \bot$$
$$\quad \textbf{repeat}$$
$$\quad\quad U' \leftarrow U$$
$$\quad\quad U \leftarrow [\![\psi]\!] \vee ([\![\phi]\!] \wedge [\![\text{EX } U]\!])$$
$$\quad \textbf{until } U \equiv U'$$
$$\quad \textbf{return } U$$

NEXT

- In symbolic model checking the transition relation *R* is a propositional formula with normal and primed (propositional) variables
- EX ϕ is valid in a state if there is some adjacent state where ϕ is valid

$$\llbracket \text{EX}\,\phi \rrbracket = \exists \overline{x}' \cdot R \wedge \llbracket \phi \rrbracket'$$

- $\llbracket \phi \rrbracket'$ is the formula obtained from $\llbracket \phi \rrbracket$ by replacing all variables by the respective primed version
- The existential quantifier can be eliminated by expansion

$$\exists x \cdot \phi \equiv \phi[\top/x] \vee \phi[\bot/x]$$

NEXT

$$R = (a \land \neg b \land \neg a' \land b') \lor (\neg a \land b \land \neg a' \land b')$$

$$
\begin{aligned}
\llbracket \text{EX } b \rrbracket &= \exists a', b' \cdot R \land b' \\
&= \exists a', b' \cdot R \\
&= \exists a' \cdot R[\top/b'] \lor R[\bot/b'] \\
&= \exists a' \cdot (a \land \neg b \land \neg a') \lor (\neg a \land b \land \neg a') \\
&= (a \land \neg b) \lor (\neg a \land b)
\end{aligned}
$$

$$
\begin{aligned}
\llbracket \text{EX } a \rrbracket &= \exists a', b' \cdot R \land a' \\
&= \exists a', b' \cdot (a \land \neg b \land \neg a' \land b' \land a') \lor \dots \\
&= \bot
\end{aligned}
$$

FAIRNESS

- To model check $\overline{\text{EG}\,\phi}$ under fairness constraint φ a different expansion rule (and fixpoint algorithm) is required

$$\overline{\text{EG}\,\phi} \equiv \overline{\phi} \wedge \text{EX}(\overline{\phi} \text{ EU } (\varphi \wedge \overline{\text{EG}\,\phi}))$$

- Likewise to explicit model checking, to model check $\overline{\phi} \text{ EU } \overline{\psi}$ symbolically it suffices to check $\overline{\phi} \text{ EU } (\overline{\psi} \wedge \overline{\text{EG}\,\top})$ instead
- Similarly for the remaining operators

EFFICIENCY

- Symbolic model checking requires procedures to check the validity and equivalence of propositional formulas
- These can be implemented efficiently using SAT or *Ordered Binary Decision Diagrams*
- In most situations symbolic model checking is much faster then explicit model checking