Alcino Cunha

SPECIFICATION AND MODELING

CTL MODEL CHECKING

Universidade do Minho & INESC TEC

2019/20

COMPUTATION TREE LOGIC

SYNTAX

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

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
  - $L : S \longrightarrow 2^A$ 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

$$
\begin{aligned}
\bot &\equiv \neg\top \\
\phi \wedge \psi &\equiv \neg(\neg\phi \vee \neg\psi) \\
\phi \rightarrow \psi &\equiv \neg\phi \vee \psi \\
\mathrm{AX}\,\phi &\equiv \neg\,\mathrm{EX}\,\neg\phi \\
\mathrm{EF}\,\phi &\equiv \top\ \mathrm{EU}\ \phi \\
\mathrm{AG}\,\phi &\equiv \neg\,\mathrm{EF}\,\neg\phi \\
\mathrm{AF}\,\phi &\equiv \neg\,\mathrm{EG}\,\neg\phi \\
\phi\ \mathrm{AR}\ \psi &\equiv \neg(\neg\phi\ \mathrm{EU}\ \neg\psi) \\
\phi\ \mathrm{ER}\ \psi &\equiv \mathrm{EG}\,\psi \vee (\psi\ \mathrm{EU}\ (\phi \wedge \psi)) \\
\phi\ \mathrm{AU}\ \psi &\equiv \neg(\neg\phi\ \mathrm{ER}\ \neg\psi)
\end{aligned}
$$

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

$$[\![\phi]\!]_M \equiv \{s \in S \mid M, s \models \phi\}$$

EXPLICIT VS SYMBOLIC MODEL CHECKING

Explicit model checking

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

$$M \models \phi \quad \text{iff} \quad I \subseteq [\![\phi]\!]_M$$

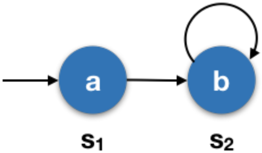
Symbolic model checking

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

$$M \models \phi \quad \text{iff} \quad I \longrightarrow [\![\phi]\!]_M$$

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

$$
\begin{aligned}
[\![p]\!] &= \{s \in S \mid p \in L(s)\} \\
[\![\top]\!] &= S \\
[\![\neg\phi]\!] &= S - [\![\phi]\!] \\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] \cup [\![\psi]\!] \\
[\![\mathsf{EX}\,\phi]\!] &= \{s \in S \mid \exists s' \in [\![\phi]\!] \cdot (s, s') \in R\}
\end{aligned}
$$

UNTIL

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

ALWAYS

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

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

- $M, s \models \mathrm{EG}\ \phi$ 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

$[\![\mathrm{EG}\,\phi]\!] =$

$T \longleftarrow \bigcup_{C \in \mathbf{scc}(M_\phi)}$

$G \longleftarrow T$

while $T \neq \emptyset$

  choose $s \in T$

  $T \longleftarrow T - \{s\}$

  for $t \in R_\phi.s$

    if $t \notin G$

      $G \longleftarrow G \cup \{t\}$

      $T \longleftarrow T \cup \{t\}$

return G

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

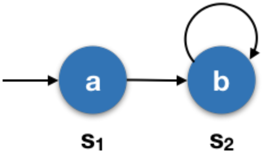
$$M \models \overline{\phi} \quad \text{iff} \quad M \models \phi \text{ and } \varphi \text{ is recurrently true in } M$$

- To model check $\overline{\mathsf{EG}\,\phi}$ 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
  - $\overline{\mathsf{EG}\,\top}$ holds in a state iff there is a fair path starting at that state

  to model check $\overline{\phi\ \mathsf{EU}\ \psi}$ it suffices to check $\overline{\phi}\ \mathsf{EU}\ (\overline{\psi} \wedge \overline{\mathsf{EG}\,\top})$ 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

$$
\begin{aligned}
[\![p]\!] &= p \\
[\![\top]\!] &= \top \\
[\![\neg\phi]\!] &= \neg[\![\phi]\!] \\
[\![\phi \vee \psi]\!] &= [\![\phi]\!] \vee [\![\psi]\!]
\end{aligned}
$$

ALWAYS

- From the EG expansion rule

$$\mathsf{EG}\,\phi \equiv \phi \wedge \mathsf{EX}(\mathsf{EG}\,\phi)$$

a greatest fixpoint algorithm can be directly obtained

$$[\![\mathsf{EG}\,\phi]\!] = \nu Z\,.\,\phi \wedge [\![\mathsf{EX}\,Z]\!]$$

$[\![\mathsf{EG}\ \phi]\!] =$

$G \longleftarrow \top$

repeat

$G' \longleftarrow G$

$G \longleftarrow [\![\phi]\!] \wedge [\![\mathsf{EX}\,G]\!]$

until $G \equiv G'$

return G

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

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

$[\![\phi \text{ EU } \psi]\!] =$

$U \longleftarrow \bot$

repeat

$U' \longleftarrow U$

$U \longleftarrow [\![\psi]\!] \vee ([\![\phi]\!] \wedge [\![\text{EX}\, U]\!])$

until $U \equiv U'$

return U

NEXT

- In symbolic model checking the transition relation R is a propositional formula with normal and primed (propositional) variables
- $\mathsf{EX}\,\phi$ is valid in a state if there is some adjacent state where ϕ is valid

$$[\![\mathsf{EX}\,\phi]\!] = \exists \bar{x}' \cdot R \wedge [\![\phi]\!]'$$

- $[\![\phi]\!]'$ is the formula obtained from $[\![\phi]\!]$ 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

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

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

$$\begin{aligned}
[\![\mathrm{EX}\, a]\!] &= \exists a', b' \cdot R \land a' \\
&= \exists a', b' \cdot (a \land \neg b \land \neg a' \land b' \land a') \lor \ldots \\
&= \bot
\end{aligned}$$

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 $\overline{\phi\ \text{EU}\ \psi}$ 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