Crash course – Verification of Finite Automata
CTL model-checking

Exercise session - 03.12.2015
Romain Jacob
Reminders

- **Objective:**
  
  Verify properties over DES models
  
  Formal method $\Rightarrow$ Absolute guarantees!

- **Problem:**
  
  Combinatorial explosion
  
  $\Rightarrow$ Huge amount of states, computationally intractable

- **Solution:**
  
  Work with sets of states
  
  $\Rightarrow$ Symbolic Model-Checking
  
  $\Rightarrow$ Tool: (O)BDDs
Reminders

**Sets**
- \( A \)
- \( s \in A \)

\[ \sigma(s) = x_1 \overline{x}_0 = (1,0) \text{ and } \psi_A = x_1 + x_0 \Rightarrow s \models \psi_A ? \]

**Boolean functions/Characteristic functions**
- \( \psi_E = 1 \)
- \( \psi_A = f \)
- \( \psi_B = g \)
- \( \psi_{A \cap B} = f \cdot g \)

**Example:**
\[ \sigma(s) = x_1 \overline{x}_0 = (1,0) \text{ and } \psi_A = x_1 + x_0 \Rightarrow s \models \psi_A ? \]

**BBD representation of Boolean functions**

**Equivalence between sets and Boolean equations**

\[ f : x_1 + \overline{x}_1 x_2 + \overline{x}_2 \overline{x}_3 \]

Fall \( x_1 = 0 \)
\[ f |_{x_1=0} : x_2 + \overline{x}_2 \overline{x}_3 \]
- Fall \( x_2 = 0 \)
  \[ f |_{x_1=0, x_2=0} : \overline{x}_3 \]
- Fall \( x_2 = 1 \)
  \[ f |_{x_1=0, x_2=1} : 1 \]

Fall \( x_1 = 1 \)
\[ f |_{x_1=1} : 1 \]

\[ f : x_1 \]

\[ x_1 \]

\[ x_2 \]

\[ x_3 \]

\[ 0 \]

\[ 1 \]
Today’s menu

1. Reachability of states

2. Comparison of automata

3. Formulation and verification of CTL properties

Can be formulated as reachability problems
Reachability of states

- **Idea:** Fairly simple
  1. Start from the initial set of states,
  2. Compute all states you can transition to in one hop (one transition),
     \[ \rightarrow \text{The successor states,} \]
  3. Join the two sets,
  4. Iterate from 2. until you reach a fix point.

Done!

- *Is this guarantee to terminate?*
Reachability of states

- **Idea:** Fairly simple
  1. Start from the initial set of states,
  2. Compute all states you can transition to in one hop (one transition),
     → The successor states,
  3. Join the two sets,
  4. Iterate from 2. until you reach a fix point.
    Done!

- *Is this guarantee to terminate?*
  → Only if you have a finite model!!

- *How do we formalize this?*
Formalization of reachable states

\[ \delta : X \subseteq E \rightarrow X' \subseteq E \]

\[ q \mapsto q' \]

\[ q \in X \iff \exists q' \in X', \delta(q, q') \text{ is defined} \]

\[ \psi_\delta(q, q') = 1 \]

\[ \overline{q} \notin X \iff \forall q' \in X', \delta(q, q') \text{ is defined} \]

\[ \forall q' \in X, \psi_\delta(q, q') = 0 \]
Formalization of reachable states

\[ \delta : X \subseteq E \rightarrow X' \subseteq E \]

\[ q \mapsto q' \]

What is \( Q' \)?

\[ q' \in Q' \Rightarrow q' \in X' \Rightarrow \exists q \in X, \psi_\delta(q, q') = 1 \]

Not sufficient!

We also need that \( q \) belongs to \( Q \):

\[ q' \in Q \Leftrightarrow \psi_Q(q) = 1 \]
Formalization of reachable states

\[\delta : X \subseteq E \rightarrow X' \subseteq E\]

\[q \mapsto q'\]

What is \(Q'\)?

\[q' \in Q' \iff \exists q \in X, \psi_Q(q) = 1 \text{ and } \psi_\delta(q, q') = 1\]

\[\iff \exists q \in X, \psi_Q(q) \cdot \psi_\delta(q, q') = 1\]

\[Q' = \text{Suc}(Q, \delta) = \{q' \mid \exists q \in X, \psi_Q(q) \cdot \psi_\delta(q, q') = 1\}\]
Formalization of reachable states

$\delta : X \subseteq E \rightarrow X' \subseteq E$

$q \mapsto q'$

$Q' = Suc(Q, \delta) = \{q' \mid \exists q \in X, \psi_Q(q) \cdot \psi_\delta(q, q') = 1\}$

$\iff \psi_{Q'} = \psi_Q \cdot \psi_\delta$

$Q_R$: set of reachable states

$Q_R = Q_0 \cup \bigcup_{i \geq 0} Suc(Q_i, \delta)$

$\iff \psi_{Q_R} = \psi_{Q_0} \sum_{i \geq 0} \psi_{Q_i} \cdot \psi_\delta$

Again, finite union if finite model
Comparison of automata

Two automata are equivalent if the following condition is true:

\[
\psi_Y(y_1, y_2) = (\exists q_1, q_2 : \psi_Q(q_1, q_2) \cdot \psi_{\omega_1}(q_1, y_1) \cdot \psi_{\omega_2}(q_2, y_2))
\]

\[
\psi_\delta(q_1, q_2, q_1', q_2') = (\exists u : \psi_{\omega_1}(u, q_1, q_1') \cdot \psi_{\omega_2}(u, q_2, q_2'))
\]

\[
\psi_Q(q_1, q_2)
\]

\[
\psi_Y(y_1, y_2) = (\exists q_1, q_2 : \psi_Q(q_1, q_2) \cdot \psi_{\omega_1}(q_1, y_1) \cdot \psi_{\omega_2}(q_2, y_2))
\]

The automata are not equivalent if the following term is true:

\[
\exists y_1, y_2 : \psi_Y(y_1, y_2) \cdot (y_1 \neq y_2)
\]

- Get rid of inputs
- Compute \( Q_R \)
- Deduce reachable outputs
- Test for equivalence

Don’t compare states!
Formulation of CTL properties

Based on atomic propositions ($\phi$) and quantifiers

- $A\phi \rightarrow \text{"All } \phi\text{"}$, $\phi$ holds on all paths
- $E\phi \rightarrow \text{"Exists } \phi\text{"}$, $\phi$ holds on at least one path
- $X\phi \rightarrow \text{"NeXt } \phi\text{"}$, $\phi$ holds on the next state
- $F\phi \rightarrow \text{"Finally } \phi\text{"}$, $\phi$ holds at some state along the path
- $G\phi \rightarrow \text{"Globally } \phi\text{"}$, $\phi$ holds on all states along the path
- $\phi_1 U \phi_2 \rightarrow \text{"\phi_1 Until } \phi_2\text{"}$, $\phi_1$ holds until $\phi_2$ holds

Proper CTL formula: $\{A,E\} \{X,F,G,U\} \phi$

Quantifiers go by pair, you need one of each.
Formulation of CTL properties

$\text{EF } \phi : \text{“There exists a path along which at some state } \phi \text{ holds.”}$

$q \models \phi \quad q \models \text{EF } \phi$
Formulation of CTL properties

$AF \phi : \text{“On all paths, at some state } \phi \text{ holds .”}$

$q \models \phi \quad q \models AF \phi$
Formulation of CTL properties

AG $\phi$ : “On all paths, for all states $\phi$ holds.”
Formulation of CTL properties

$EG \phi$ : “There exists a path along which for all states $\phi$ holds.”
Formulation of CTL properties

$E \phi U \Psi$ : “There exists a path along which $\phi$ holds until $\Psi$ holds.”
Formulation of CTL properties

$A\phi U \Psi$ : “On all paths, $\phi$ holds until $\Psi$ holds.”
Formulation of CTL properties

$AX \phi$ : “On all paths, the next state satisfies $\phi$.”

$EX \phi$ : “There exists a path along which the next state satisfies $\phi$.”
Formulation of CTL properties

$\text{AG EF } \phi$ : “On all paths, for all states, there exists a path along which at some state $\phi$ holds.”

$q \models AG \ EF \phi$

$q \models \phi$

$r$
Formulation of CTL properties

\[ \text{AG } \phi \equiv \neg \text{EF } \neg \phi \]
\[ \text{AF } \phi \equiv \neg \text{EG } \neg \phi \]
\[ \text{EF } \phi \equiv \neg \text{AG } \neg \phi \]
\[ \text{EG } \phi \equiv \neg \text{AF } \neg \phi \]

“On all paths, for all states \( \phi \) holds.”
\[ \equiv \]

“There exists no path along which at some state \( \phi \) doesn’t hold.”

\[ \ldots \]

**Remark:** There exists other temporal logics

→ LTL (Linear Tree Logic)
→ CTL* = \{CTL,LTL\}
→ ...
Verification of CTL properties

- Convert the property into a reachability problem
  - Start from states in which the property holds;
  - Compute all predecessor states (same as for successors, but with reverse the transition function) for which the property still holds true;
  - If initial states set is a subset, the property is satisfied by the model.

- Computation specifics are described in the lecture slides.
So... what is Model-Checking exactly?

Algorithm

- **Input**
  - A DES model, \( M \)
    - Finite automata,
    - Petri nets,
    - Kripke machine, ...
  - A logic property, \( \phi \)
    - CTL,
    - LTL, ...

- **Output**
  - \( M \models \phi \) ?
  - A diagnosis trace showing that the property does not hold!!
Crash course – Verification of Finite Automata
CTL model-checking

Your turn to work!

Slides online on my webpage:
http://people.ee.ethz.ch/~jacobr/
Comparison of Finite Automata

a) Express the characteristic function of the transition relation for both automaton, $\psi_r(x, x', u)$.

$$\psi_A(x_A, x'_A, u) = x_A x'_A u + x_A x'_A \overline{u} + x_A x'_A u + x_A x'_A \overline{u}$$

$$\psi_B(x_B, x'_B, u) = x_B x'_B u + x_B x'_B \overline{u} + x_B x'_B u + x_B x'_B \overline{u}$$
Comparison of Finite Automata

b) Express the joint transition function, $\psi_f$.

$$\psi_f(x_A, x_A', x_B, x_B') = (\exists u : \psi_A(x_A, x_A', u) \cdot \psi_B(x_B, x_B', u))$$

$$\psi_f(x_A, x_A', x_B, x_B')$$

$$= (\overline{x_A}x_A' + x_Ax_A') \cdot (\overline{x_B}x_B' + x_Bx_B') +$$

$$= (\overline{x_A}x_A' + x_Ax_A') \cdot (\overline{x_B}x_B' + x_Bx_B')$$

$$= \overline{x_A}x'_A \overline{x_B}x'_B + \overline{x_A}x'_A x_Bx'_B + x_Ax'_A \overline{x_B}x'_B + x_Ax'_A x_Bx'_B +$$

$$= \frac{\overline{x_A}x'_A x_Bx'_B}{2} + \frac{x_Ax'_A x_Bx'_B}{2} + \frac{x_Ax'_A \overline{x_B}x'_B}{2} + \frac{x_Ax'_A x_B \overline{x_B}x'_B}{2} + \frac{x_Ax'_A \overline{x_B} \overline{x_B}x'_B}{2} + \frac{x_Ax'_A x_B \overline{x_B} \overline{x_B}x'_B}{2} + \frac{x_Ax'_A \overline{x_B} \overline{x_B} \overline{x_B}x'_B}{2} + \frac{x_Ax'_A x_B \overline{x_B} \overline{x_B} \overline{x_B}x'_B}{2} +$$
Comparison of Finite Automata

c) Express the characteristic function of the reachable states, $\psi_X(x_A, x_B)$.

$$\psi_X(x_A, x_B) = \overline{x_A}x_B$$

$$\psi_{X_1} = \overline{x_A}x_B + \overline{x_A}x_B + x_Ax_B$$

$$\psi_{X_2} = \overline{x_A}x_B + \overline{x_A}x_B + x_Ax_B$$

$$= \psi_{X_1}$$

$\rightarrow$ the fix-point is reached!

$$\psi_X = \overline{x_A}x_B + \overline{x_A}x_B + x_Ax_B$$
d) Express the characteristic function of the reachable output, $\psi_Y(x_A, x_B)$.

$$\psi_{g_A} = \overline{x_A y_A} + x_A y_A$$
$$\psi_{g_B} = \overline{x_B y_B} + x_B y_B$$

and

$$\psi_X = \overline{x_A x_B} + x_A \overline{x_B} + x_A x_B$$

$$\psi_Y(y_A, y_B) = (\exists (x_A, x_B) : \psi_X \cdot \psi_{g_A} \cdot \psi_{g_B})$$
$$= y_A y_B + \overline{y_A y_B} + \overline{y_A y_B}$$
Comparison of Finite Automata

e) Are the automata equivalent? Justify with a simple calculus.

\[ \psi_Y (y_A, y_B) = y_A y_B + \overline{y_A} \overline{y_B} + \overline{y_A} y_B \]

and \( (y_A \neq y_B) = \overline{y_A} y_B + y_A \overline{y_B} \)

implies \( \psi_Y \cdot (y_A \neq y_B) \neq 0 \)

→ Automata are not equivalent.
Temporal Logic

i. $\text{EF } a$
   \[ Q = \{0, 1, 2, 3\} \]

ii. $\text{EX AX } a$

iii. $\text{EF ( } a \text{ AND EX NOT}(a) \text{ )}$
Temporal Logic

i. EF a
   \[ Q = \{0, 1, 2, 3\} \]

ii. EX AX a
    \[ Q = \{1, 2\} \]

iii. EF ( a AND EX NOT(a) )
Temporal Logic

i. EF a
   \[ Q = \{0, 1, 2, 3\} \]

ii. EX AX a
   \[ Q = \{1, 2\} \]

iii. EF ( a AND EX NOT(a) )
   \[ Q = \{0, 1, 2, 3\} \]
Temporal Logic

- Trick: \( \text{AF } Z \ \text{NOT}(\text{EG NOT}(Z)) \)

Require: \( \psi_Z, \psi_f \)

```plaintext
current = NOT(\( \psi_Z \))
next = current AND (EXISTS(\( \psi_f \) AND current))
while next != current do
    current = next;
    next = current AND (EXISTS(\( \psi_f \) AND current));
end while
return \( \psi_{AFZ} = \text{NOT}(current) \);
```

\( \triangleright \) Equivalence in term of sets:

\( \triangleright X_0 \)

\( \triangleright X_1 = X_0 \cap \text{Pre}(X_0, f) \)

\( \triangleright X_i 
eq X_{i-1} \)

\( \triangleright X_i = X_{i-1} \cap \text{Pre}(X_{i-1}, f) \)

\( \triangleright X_f = \text{EG NOT}(Z) \)

\( \triangleright X_f \models \text{AF } Z = \text{NOT}(\text{EG NOT}(Z)) \)
Crash course – Verification of Finite Automata
CTL model-checking

See you next week!

Slides online on my webpage:
http://people.ee.ethz.ch/~jacobr/