CS 267: Automated Verification

Lectures 4: $\mu$-calculus

Instructor: Tevfik Bultan
\(\mu\)-Calculus

\(\mu\)-Calculus is a temporal logic which consist of the following:

- Atomic properties AP
- Boolean connectives: \(\neg\), \(\land\), \(\lor\)
- Precondition operator: EX
- Least and greatest fixpoint operators: \(\mu y . \mathcal{F} y\) and \(\nu y . \mathcal{F} y\)
  - \(\mathcal{F}\) must be syntactically monotone in \(y\)
    - meaning that all occurrences of \(y\) in within \(\mathcal{F}\) fall under an even number of negations
μ-Calculus

• μ-calculus is a powerful logic
  – Any CTL* property can be expressed in μ-calculus

• So, if you build a model checker for μ-calculus you would handle all the temporal logics we discussed: LTL, CTL, CTL*

• One can write a μ-calculus model checker using the basic ideas about fixpoint computations that we discussed
  – However, there is one complication
    • Nested fixpoints!
Mu-calculus Model Checking Algorithm

eval(f : mu-calculus formula) : a set of states

- case: $f \in \text{AP}$ return $\{s \mid L(s,f)=\text{true}\}$;
- case: $f \equiv \neg p$ return $S - \text{eval}(p)$;
- case: $f \equiv p \land q$ return $\text{eval}(p) \cap \text{eval}(q)$;
- case: $f \equiv p \lor q$ return $\text{eval}(p) \cup \text{eval}(q)$;
- case: $f \equiv \text{EX } p$ return $\text{EX}(\text{eval}(p))$;
Mu-calculus Model Checking Algorithm

eval(f)

...  
case: f ≡ \mu y . g(y)
    y := False;
    repeat {
        y\textsubscript{old} := y;
        y := eval(g(y));
    } until y = y\textsubscript{old}
    return y;
Mu-calculus Model Checking Algorithm

eval(f)

...  
  case: f \equiv \forall y . g(y)  
       y := True;  
       repeat {  
           y_{old} := y;  
           y := eval(g(y));  
       } until y = y_{old}  
  return y;
Nested Fixpoints

• Here is a CTL property
  \[ \text{EG EF } p = \forall y . \left( \mu z . p \lor \text{EX } z \right) \land \text{EX } y \]
  – The fixpoints are not nested.
  – Inner fixpoint is computed only once and then the outer fixpoint is computed
  – Fixpoint characterizations of CTL properties do not have nested fixpoints

• Here is a CTL* property
  \[ \text{EGF } p = \forall y . \mu z . \left( p \lor \text{EX } z \right) \land \text{EX } y \]
  – The fixpoints are nested.
  – Inner fixpoint is recomputed for each iteration of the outer fixpoint
Nested Fixpoint Example

EF p

0 |= EG EF p

EG EF p = \( \forall y . (\mu z . p \lor \text{EX } z) \land \text{EX } y \)

EF p fixpoint

\( F_1(\emptyset) = \{1\} \)
\( F_1^2(\emptyset) = \{0,1\} \)
\( F_1^3(\emptyset) = \{0,1\} \)

EG EF p = \{0\}

EGF p = \( \forall y . \mu z . ((p \lor \text{EX } z) \land \text{EX } y) \)

nested fixpoint

<table>
<thead>
<tr>
<th>( F_3 )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,0</td>
<td>{0,1,2}</td>
<td>( \emptyset )</td>
</tr>
<tr>
<td>0,1</td>
<td></td>
<td>{1}</td>
</tr>
<tr>
<td>0,2</td>
<td></td>
<td>{0,1}</td>
</tr>
<tr>
<td>0,3</td>
<td></td>
<td>{0,1}</td>
</tr>
<tr>
<td>1,0</td>
<td>{0,1}</td>
<td>( \emptyset )</td>
</tr>
<tr>
<td>1,1</td>
<td></td>
<td>( \emptyset )</td>
</tr>
<tr>
<td>2,0</td>
<td></td>
<td>( \emptyset )</td>
</tr>
<tr>
<td>2,1</td>
<td></td>
<td>( \emptyset )</td>
</tr>
<tr>
<td>3,0</td>
<td></td>
<td>( \emptyset )</td>
</tr>
</tbody>
</table>

EGF p = \( \emptyset \)