Bisimulation equivalence: general idea

- \( M, M' \) bisimilar if they have ‘corresponding executions’
  - to each step of \( M \) there is a corresponding step of \( M' \)
  - to each step of \( M' \) there is a corresponding step of \( M \)

- Bisimilar models satisfy same CTL* properties

- Bisimilar: same truth/falsity of model properties

- **Simulation** gives property-truth preserving abstraction 
  (see later)
Let $R : S \rightarrow S \rightarrow \mathbb{B}$ and $R' : S' \rightarrow S' \rightarrow \mathbb{B}$ be transition relations.

$B$ is a bisimulation relation between $R$ and $R'$ if:

$B : S \rightarrow S' \rightarrow \mathbb{B}$

$\forall s s'. B s s' \Rightarrow \forall s_1 \in S. R s s_1 \Rightarrow \exists s'_1. R' s' s'_1 \land B s_1 s'_1$

(to each step of $R$ there is a corresponding step of $R'$)

$\forall s s'. B s s' \Rightarrow \forall s'_1 \in S. R' s' s'_1 \Rightarrow \exists s_1. R' s s_1 \land B s_1 s'_1$

(to each step of $R'$ there is a corresponding step of $R$)
Bisimulation equivalence: definition and theorem

- Let $M = (S, S_0, R, L)$ and $M' = (S', S'_0, R', L')$

- $M \equiv M'$ if:
  - there is a bisimulation $B$ between $R$ and $R'$
  - $\forall s_0 \in S_0. \exists s'_0 \in S'_0. B s_0 s'_0$
  - $\forall s'_0 \in S'_0. \exists s_0 \in S_0. B s_0 s'_0$
  - there is a bijection $\theta : AP \to AP'$
  - $\forall s \ s'. B s s' \Rightarrow L(s) = L'(s')$

- Theorem: if $M \equiv M'$ then for any CTL* state formula $\psi$:
  $M \models \psi \iff M' \models \psi$

- See Q14 in the Exercises
Abstraction

- Abstraction creates a simplification of a model
  - separate states may get merged
  - an abstract path can represent several concrete paths
- $M \preceq \overline{M}$ means $\overline{M}$ is an abstraction of $M$
  - to each step of $M$ there is a corresponding step of $\overline{M}$
  - atomic properties of $M$ correspond to atomic properties of $\overline{M}$
- Special case is when $\overline{M}$ is a subset of $M$ such that:
  - $\overline{M} = (\overline{S}_0, \overline{S}, \overline{R}, \overline{L})$ and $M = (S_0, S, R, L)$
    - $\overline{S} \subseteq S$
    - $\overline{S}_0 = S_0$
    - $\forall s \ s' \in \overline{S}. \ R \ s \ s' \iff R \ s \ s'$
    - $\forall s \in \overline{S}. \ L \ s = L \ s$
  - $\overline{S}$ contain all reachable states of $M$
    - $\forall s \in \overline{S}. \forall s' \in S. \ R \ s \ s' \Rightarrow s' \in \overline{S}$
- All paths of $M$ from initial states are $\overline{M}$-paths
  - hence for all CTL formulas $\psi$: $\overline{M} \models \psi \Rightarrow M \models \psi$
Recall JM1

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: IF LOCK=0 THEN LOCK:=1; 0: IF LOCK=0 THEN LOCK:=1;</td>
<td></td>
</tr>
<tr>
<td>1: X:=1; 1: X:=2;</td>
<td></td>
</tr>
<tr>
<td>2: IF LOCK=1 THEN LOCK:=0; 2: IF LOCK=1 THEN LOCK:=0;</td>
<td></td>
</tr>
<tr>
<td>3: 3:</td>
<td></td>
</tr>
</tbody>
</table>

◮ Two program counters, state: \((pc_1, pc_2, lock, x)\)

\[
S_{JM1} = [0..3] \times [0..3] \times \mathbb{Z} \times \mathbb{Z}
\]

\[
R_{JM1} (0, pc_2, 0, x) \quad (1, pc_2, 1, x) \quad R_{JM1} (pc_1, 0, 0, x) \quad (pc_1, 1, 1, x)
\]

\[
R_{JM1} (1, pc_2, lock, x) \quad (2, pc_2, lock, 1) \quad R_{JM1} (pc_1, 1, lock, x) \quad (pc_1, 2, lock, 2)
\]

\[
R_{JM1} (2, pc_2, 1, x) \quad (3, pc_2, 0, x) \quad R_{JM1} (pc_1, 2, 1, x) \quad (pc_1, 3, 0, x)
\]

◮ Assume \(\text{NotAt11} \in L_{JM1}(pc_1, pc_2, lock, x) \iff \neg ((pc_1 = 1) \land (pc_2 = 1))\)

◮ Model \(M_{JM1} = (S_{JM1}, \{(0, 0, 0, 0)\}, R_{JM1}, L_{JM1})\)

◮ \(S_{JM1}\) not finite, but actually \(lock \in \{0, 1\}, x \in \{0, 1, 2\}\)

◮ Clear by inspection that \(M_{JM1} \preceq \overline{M}_{JM1}\) where:

\[
\overline{M}_{JM1} = (\overline{S}_{JM1}, \{(0, 0, 0, 0)\}, \overline{R}_{JM1}, \overline{L}_{JM1})
\]

◮ \(\overline{S}_{JM1} = [0..3] \times [0..3] \times [0..1] \times [0..3]\)

◮ \(\overline{R}_{JM1}\) is \(R_{JM1}\) restricted to arguments from \(\overline{S}_{JM1}\)

◮ \(\text{NotAt11} \in \overline{L}_{JM1}(pc_1, pc_2, lock, x) \iff \neg ((pc_1 = 1) \land (pc_2 = 1))\)

◮ \(\overline{L}_{JM1}\) is \(L_{JM1}\) restricted to arguments from \(\overline{S}_{JM1}\)
Simulation relations

Let $R : S \rightarrow S \rightarrow B$ and $\overline{R} : \overline{S} \rightarrow \overline{S} \rightarrow B$ be transition relations.

$H$ is a simulation relation between $R$ and $\overline{R}$ if:

- $H$ is a relation between $S$ and $\overline{S}$ – i.e. $H : S \rightarrow \overline{S} \rightarrow B$

- to each step of $R$ there is a corresponding step of $\overline{R}$ – i.e.:
  \[ \forall s \bar{s}. \ H s \bar{s} \Rightarrow \forall s' \in S. \ R s s' \Rightarrow \exists \bar{s}' \in \overline{S}. \ \overline{R} \bar{s} \bar{s}' \land H \bar{s}' \bar{s}' \]

Also need to consider abstraction of atomic properties

- $H_{AP} : AP \rightarrow \overline{AP} \rightarrow B$

- details glossed over here
Simulation preorder: definition and theorem

- Let $M = (S, S_0, R, L)$ and $\overline{M} = (\overline{S}, \overline{S_0}, \overline{R}, \overline{L})$

- $M \preceq \overline{M}$ if:
  - there is a simulation $H$ between $R$ and $\overline{R}$
  - $\forall s_0 \in S_0. \exists \overline{s_0} \in \overline{S_0}. H \ s_0 \ \overline{s_0}$
  - $\forall s \ \overline{s}. H \ s \ \overline{s} \Rightarrow L(s) = \overline{L(s)}$

- ACTL is the subset of CTL without E-properties
  - e.g. $\text{AG AF}p$ – from anywhere can always reach a $p$-state

- Theorem: if $M \preceq \overline{M}$ then for any ACTL state formula $\psi$: $\overline{M} \models \psi \Rightarrow M \models \psi$

- If $\overline{M} \models \psi$ fails then cannot conclude $M \models \psi$ false
Example (Grumberg)

\[ H \text{ a simulation} \]

\[ H \text{ RED STOP} \land H \text{ YELLOW GO} \land H \text{ GREEN GO} \]

\[ H_{AP} : \{r, y, g\} \rightarrow \{r, yg\} \rightarrow \mathbb{B} \]

\[ H_{AP} r r \land H_{AP} y yg \land H_{AP} g yg \]

\[ \overline{M} \models AG AF \neg r \text{ hence } M \models AG AF \neg r \]

\[ \text{but } \neg (\overline{M} \models AG AF r) \text{ doesn’t entail } \neg (M \models AG AF r) \]

\[ \llbracket AG AF r \rrbracket_{\overline{M}}(STOP) \text{ is false} \]

(consider \( \overline{M} \)-path \( \pi' \) where \( \pi' = STOP.GO.GO.GO.\ldots \))

\[ \llbracket AG AF r \rrbracket_{M}(RED) \text{ is true} \]

(abstract path \( \pi' \) doesn’t correspond to a real path in \( M \))
CEGAR

- Counter Example Guided Abstraction Refinement

- Lots of details to fill out (several different solutions)
  - how to generate abstraction
  - how to check counterexamples
  - how to refine abstractions

- Microsoft SLAM driver verifier is a CEGAR system
Temporal Logic and Model Checking – Summary

- Various property languages: LTL, CTL, PSL (Prior, Pnueli)
- Models abstracted from hardware or software designs
- Model checking checks $M \models \psi$ (Clarke et al.)
- Symbolic model checking uses BDDs (McMillan)
- Avoid state explosion via simulation and abstraction
- CEGAR refines abstractions by analysing counterexamples
- Triumph of application of computer science theory
  - two Turing awards, McMillan gets 2010 CAV award
  - widespread applications in industry
<table>
<thead>
<tr>
<th>Topic</th>
<th>Slides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to models</td>
<td>1 - 9</td>
</tr>
<tr>
<td>Atomic properties</td>
<td>10</td>
</tr>
<tr>
<td>Trees and paths</td>
<td>11 - 12</td>
</tr>
<tr>
<td>Examples of properties</td>
<td>13 - 16</td>
</tr>
<tr>
<td>Reachability</td>
<td>17</td>
</tr>
<tr>
<td>Introduction to model checking</td>
<td>18 - 26</td>
</tr>
<tr>
<td>Symbolic model checking</td>
<td>27 - 32</td>
</tr>
<tr>
<td>Disjunctive partitioning of BDDs</td>
<td>33 - 35</td>
</tr>
<tr>
<td>Generating counter-examples</td>
<td>36 - 42</td>
</tr>
<tr>
<td>Introduction to temporal logic</td>
<td>43 - 45</td>
</tr>
<tr>
<td>Linear Temporal Logic (LTL)</td>
<td>46 - 58</td>
</tr>
<tr>
<td>Computation Tree Logic (CTL)</td>
<td>59 - 75</td>
</tr>
<tr>
<td>CTL model checking</td>
<td>75 - 83</td>
</tr>
<tr>
<td>History of model checking</td>
<td>84</td>
</tr>
<tr>
<td>Expressibility of LTL and CTL</td>
<td>57 - 58, 85 - 87</td>
</tr>
<tr>
<td>CTL*</td>
<td>88 - 90</td>
</tr>
<tr>
<td>Fairness</td>
<td>91 - 92</td>
</tr>
<tr>
<td>Propositional modal $\mu$-calculus</td>
<td>93</td>
</tr>
<tr>
<td>Sequential Extended Regular Expressions (SEREs)</td>
<td>94 - 95</td>
</tr>
<tr>
<td>Assertion Based Verification (ABV) and PSL</td>
<td>96 - 107</td>
</tr>
<tr>
<td>Dynamic verification: event semantics</td>
<td>108 - 117</td>
</tr>
<tr>
<td>Bisimulation</td>
<td>118 - 120</td>
</tr>
<tr>
<td>Abstraction</td>
<td>121 - 125</td>
</tr>
<tr>
<td>Counterexample Guided Abstraction Refinement (CEGAR)</td>
<td>126</td>
</tr>
<tr>
<td>Summary</td>
<td>127</td>
</tr>
</tbody>
</table>
THE END