Bisimulation, Simulation Relation and Linear-Time Logic

Barbara Jobstmann
Kripke Structure

\[ K = (S, T, s_0, A, L) \]
\[ S = \{a, b, c\} \]
\[ T = \{(a, b), (b, a), (b, c), (c, c), (c, a)\} \]
\[ s_0 = a \]
\[ A = \{p, q\} \]
\[ L(a) = \{} \]
\[ L(b) = \{p\} \]
\[ L(c) = \{q\} \]
Bisimulation Relation

- **Definition**: Given two Kripke structures K and K' with A=A', a relation $B \subseteq S \times S'$ is a **bisimulation relation** if $(s,s') \in B$ implies
  1) $L(s) = L(s')$
  2) if $(s,t) \in T$ then $\exists t' \in S'$. $(s',t') \in T'$ and $(t,t') \in B$
  3) if $(s',t') \in T'$ then $\exists t \in S$. $(s,t) \in T$ and $(t,t') \in B$

- **Definition**: K and K' are **bisimilar** if there exists a bisimulation relation $B$ among them and $(s_0,s_0') \in B$. 

Barbara Jobstmann
Bisimulation Relation

• **Theorem:** Given K and K' with A=A'. Let f: $2^{S \times S'} \rightarrow 2^{S \times S'}$ be the function defined by

\[
f(Z) = \{(s, s') \in Z \mid \begin{align*}
L(s) &= L'(s) \\
\forall t \in S. (s, t) \in T \rightarrow \\
\exists t' \in S'. (s', t') \in T' \land (t, t') \in Z \}
\]

There exists a maximal bisimulation relation $B \subseteq S \times S'$, and $B = \nu Z. f(Z)$.
Bisimulation Relation

• **Proof:** $f$ is defined over the finite lattice $2^{S \times S'}$ and it is monotonic in $Z$. Furthermore if $B$ is a bisimulation relation, then $B = f(B)$ must hold, and vice versa (every fixpoint of $f$ is a bisimulation relation). Tarki's theorem tells us that the maximal bisimulation relation exist and can be computed by iterating $f$ starting from $S \times S'$.
Algorithm

1) Keep two sets of pairs of states:
   B (which eventually will be the bisim. rela.)
   A (pairs of B that need to be processed)

2) Initialize B and A to include
   all pairs \((s,s') \in S \times S'\) s.t. \(L(s) = L(s')\)

3) While A is not empty do the following.
   1) Remove one pair \((s,s')\) from A
   2) Check last two conditions of definition for \((s,s')\)
   3) If one condition does not hold, remove \((s,s')\)
      from B, and add to A all pairs \((r,r') \in B\)
      s.t. \((r,s) \in T\) or \((r',s') \in T'\)
Example
Bisimularity and CTL

- **Theorem** [Browne, Clarke, Grumberg, 1988] Two Kripke structure are bisimilar if and only if they satisfy the same CTL properties.

```
Bisimilar? Distinguishing formula?
```
Simulation

• **Definition:** Given K and K' with $A \subseteq A'$, a relation $R \subseteq S \times S'$ is a *simulation relation* if $(s,s') \in R$ implies
  1) $L(s) = L(s') \cap A$
  2) if $(s,t) \in T$ then
     \[ \exists t' \in S'. (s',t') \in T, (t,t') \in R \]

• **Definition:** Structure K' *simulates* structure K if there exists a simulation relation $R \subseteq S \times S'$ s.t. $(s_0,s_0') \in R$. 

Barbara Jobstmann
Simulation Equivalent

• If K simulate K' and K' simulate K, then K and K' are *simulation equivalent*.

• Two bisimilar structures are simulation equivalent, but the converse is not always true.
Example

K and K' are simulation equivalent but not bisimilar!
Branching versus Linear Time

• In CTL, every point in time has more than one possible future, e.g.,
  – in EF p, we say there is a future (a path) in which p eventually holds
  – in AF p, we say for all possible futures p eventually holds
Branching-Time View of Systems

Behavior of a system is a tree
Branching-Time View of Systems

Behavior of a system is a tree

Atomic Propositions

{p}

{q}

...
Behavior of a system is a set of paths = language
Behavior of a system is a set of paths = language
Example

- Vending machines
- $\text{AG}(25c \rightarrow (\text{EX coffee} \land \text{EX tea}))$
Linear Temporal Logic

• LTL Syntax (p...atomic proposition)
  \( \phi := p \mid \phi \land \phi \mid \neg \phi \mid X\phi \mid \phi U \phi \mid \phi R \phi \)
  – Boolean operators
  – Temporal operators
    \( X\phi \) ...next operator
    \( \phi U \psi \) ...until operator
    \( \phi R \psi \) ...release operator
  – Standard abbreviations:
    false = p \land \neg p
    true = \neg false
    \( \phi \lor \psi = \neg (\neg \phi \land \neg \psi) \)
Abbreviations

• $F \varphi$ (eventually) = true $U \varphi$
• $G \varphi$ (globally) = false $R \varphi$

• Similar meaning to CTL operators but without path quantifiers $A$ and $E$, since we talk about one path at a time.
• Temporal operators are allowed to follow each other, e.g., $G F p$, meaning?
Semantics

• Defined over infinite paths
• Given finite set of atomic propositions $A$, alphabet $\Sigma = 2^A$
paths $\pi$ are in $\Sigma^\omega$
• If $\pi = \pi_0 \pi_1 \pi_2 \ldots$ is an infinite path, we denote the suffix starting at $\pi_i$ by $\pi^i$
Semantics

- Suppose $\pi = \pi_0 \pi_1 \pi_2 \ldots$, then
  - $\pi \models true$,
  - $\pi \not\models false$,
  - $\pi \models \varphi$ iff $\varphi$ in $A$ and $\varphi$ in $\pi_0$
  - $\pi \not\models \neg \varphi$ iff $\pi \models \varphi$
  - $\pi \models \varphi \land \psi$ iff $\pi \models \varphi$ and $\pi \models \psi$
  - $\pi \models X\varphi$ iff $\pi^1 \models \varphi$
Semantics

- Suppose $\pi = \pi_0 \pi_1 \pi_2 \ldots$, then
  - $\pi \models \phi U \psi$ iff there exists $i \geq 0$ s.t. $\pi^i \models \psi$ and for all $0 \leq j < i$, $\pi^j \models \phi$
  - $\pi \models \phi R \psi$ iff for all $i \geq 0$ $\pi^i \models \psi$ or there exists $j \geq 0$ s.t. $\pi^j \models \phi$ and for all $0 \leq i \leq j$, $\pi^i \models \psi$

- Applying this semantics to $F$ and $G$ gives:
  - $\pi \models F \phi$ iff there exists $i \geq 0$ s.t. $\pi^i \models \phi$
  - $\pi \models G \phi$ iff for all $i \geq 0$. $\pi^i \models \phi$

- Kripke structure $K \models \phi$ iff for all paths $\pi$ of $K$, $\pi \models \phi$
## Example

| Trace   | \{q\} | \{q\} | \{p\} | \{q,p\} | \{q\} | \{q\} | ...
|---------|--------|--------|--------|--------|--------|--------|--------|
| p       | F      | F      | T      | T      | F      | F      | ...
| q       | T      | T      | F      | T      | T      | T      | ...
| X p     | F      | T      | T      | F      | F      | F      | ...
| X X p   | T      | T      | F      | F      | F      | F      | ...
| q U p   | T      | T      | T      | T      | T      | F      | F      | ...
| F p     | T      | T      | T      | T      | T      | F      | F      | ...
| G F p   | F      | F      | F      | F      | F      | F      | F      | ...
| G q     | F      | F      | F      | T      | T      | T      | T      | ...
| F G q   | T      | T      | T      | T      | T      | T      | T      | ...
Example

\[
\begin{align*}
&K \vDash p \ ? \\
&K \vDash q \ ? \\
&K \vDash XXq \ ? \\
&K \vDash Fp \ ? \\
&K \vDash Fq \ ? \\
&K \vDash GFp \ ? \\
&K \vDash G(p \rightarrow Fq) \ ?
\end{align*}
\]
Example

- $K \models p$ - True
- $K \models q$ - False
- $K \models XXq$ - False
- $K \models Fp$ - True
- $K \models Fq$ - False
- $K \models GFp$ - False
- $K \models G(p \rightarrow Fq)$ - False
Consider a system with
- two Boolean input variables $r_1, r_2$ and
- two Boolean output variables $g_1, g_2$

Desired properties
- $g_1$ and $g_2$ are never true at the same time
- Whenever $r_1$ is true then $g_1$ is true within the next three steps
- Every request ($r_2$) is eventually granted by raising the grant signal $g_2$.
- $r_1$ is true before $g_1$ is true.
Consider a system with two Boolean input variables $r_1, r_2$ and two Boolean output variables $g_1, g_2$.

Desired properties:
- $g_1$ and $g_2$ are never true at the same time.
- Whenever $r_1$ is true then $g_1$ is true within the next three steps.
- Every request ($r_2$) is eventually granted by raising the grant signal $g_2$.
- $r_1$ is true before $g_1$ is true.

English ↔ Formula:

- $G(\neg g_1 \vee \neg g_2)$
- $G(r_1 \rightarrow (g_1 \vee X(g_1) \vee X(X(g_1))))$
- $G(r_2 \rightarrow F(g_2))$
- $\neg g_1 U r_2$ or $r_1 R \neg g_1$ (different meaning)

Barbara Jobstmann