Software Design, Modelling and Analysis in UML

Lecture 14: Hierarchical State Machines II

2012-01-17

Prof. Dr. Andreas Podelski, Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:
- Putting It All Together: ODs define initial states
- Hierarchical State Machines: kind, region
- Initial pseudostate, final state

This Lecture:
- Educational Objectives: Capabilities for following tasks/questions.
  - What does this hierarchical State Machine mean? What may happen if I inject this event?
  - What is: AND-State, OR-State, pseudo-state, entry/exit/do, final state, ...

- Content:
  - Composite states
  - Legal state configuration
  - Lca, depth, ...
  - Exit/Entry, internal transitions
  - History and others
In a sense, composite states are about **abbreviation**, **structuring**, and **avoiding redundancy**.

Idea: in Tron, for the Player’s Statemachine, instead of

[Diagram of a state machine with states labeled `n`, `w`, `e`, `s`, `X/`, `resigned`, and transitions between them.]
**Composite States**

and instead of

write

**Recall: Syntax**

translates to

\[
\{(\{\text{top}, \text{st}\}, \{s, \text{st}\}, \{s_1, \text{st}\}\{s_1', \text{st}\}\{s_2, \text{st}\}\{s_2', \text{st}\}\{s_3, \text{st}\}\{s_3', \text{st}\}\},
\{\text{top} \mapsto \{s\}, \text{s} \mapsto \{\{s_1, s_1'\}, \{s_2, s_2'\}, \{s_3, s_3'\}\}, s_1 \mapsto \emptyset, s_1' \mapsto \emptyset, \ldots\}\}
\]

\[
\{\text{top} \mapsto \{s\}, \text{s} \mapsto \{\{s_1, s_1'\}, \{s_2, s_2'\}, \{s_3, s_3'\}\}, s_1 \mapsto \emptyset, s_1' \mapsto \emptyset, \ldots\}\}
\]

\[
\text{region} : (\ast) \rightarrow \mathcal{E} \cup \mathcal{S} \times \mathcal{E} \times \mathcal{S} \\
\rightarrow, \psi, \text{annot}) \quad \Psi (\ast) \rightarrow \{s_1, s_2, s_3\}
\]
Syntax: Fork/Join

• For brevity, we always consider transitions with (possibly) multiple sources and targets, i.e.
  \[ \psi : (\rightarrow) \rightarrow (2^S \setminus \emptyset) \times (2^S \setminus \emptyset) \]

• For instance,

  \[ \begin{array}{c}
  s_1 \\
  \downarrow s_2 \\
  \downarrow s_3 \\
  \end{array} \rightarrow \begin{array}{c}
  \text{tr[gd/act]} \\
  \end{array} \rightarrow \begin{array}{c}
  s_4 \\
  \downarrow s_5 \\
  \downarrow s_6 \\
  \end{array} \]

  translates to

  \[ \langle S, \text{kind, region}, \{t_1\}, \{t_1 \mapsto (\{s_2, s_3\}, \{s_5, s_6\})\}, \{t_1 \mapsto (\text{tr, gd, act})\} \rangle \]

• Naming convention: \[ \psi(t) = (\text{source}(t), \text{target}(t)). \]

---

Composite States: Blessing or Curse?

- what may happen on \( E \)?
- what may happen on \( E, F \)?
- can \( E, G \) kill the object?
- ...

---
**Composite States: Blessing or Curse?**

**States:**
- what are legal state configurations?
- what is the type of the implicit st attribute?

**Transitions:**
- what are legal transitions?
- when is a transition enabled?
- what effects do transitions have?

**States: st, (Legal) State Configurations**

- The type of st is from now on a set of states, i.e. \( st : 2^S \)
- A set \( S_1 \subseteq S \) is called (legal) state configurations if and only if
  - \( \text{top} \in S_1 \), and
  - For each state \( s \in S_1 \), that has a non-empty region \( \emptyset \neq R \in \text{region}(s) \), exactly one (non pseudo-state) child of \( s \) is in \( S_1 \), i.e.
    \[
    \{ s \in R \mid \text{kind}(s) \in \{ \text{st}, \text{fin} \} \} \cap S_1 = 1.
    \]

**Examples:**

- \( S_0 = \{ \text{top}, s_1, s_2 \} \) in \( \text{LEGAL} \) (or \( \text{false} \))
- \( S_1 = \{ \text{top}, s_1, s_2, s_3 \} \) NOT LEGAL, but \( \text{kind}(s_1) \neq \text{true} \)
- \( S_2 = \{ \text{top}, s_2 \} \) NOT LEGAL, no child of \( s_1 \)
- \( S_3 = \{ \text{top}, s_2 \} \) NOT LEGAL, not exactly one
- \( S_4 = \{ \text{top}, s_3 \} \) NOT LEGAL
- \( S_5 = \{ \text{top}, s_3 \} \) NOT LEGAL, and \( s_2 \in S_1 \)
- \( S_6 = \{ \text{top}, s_2, s_3 \} \) NOT LEGAL, and \( s_2 \in S_1 \)
- \( S_7 = \{ \text{top}, s_1, s_2, s_3 \} \) NOT LEGAL, and \( s_2 \in S_1 \)

- \( \text{child}(s_2) = \{ s_3, s_5 \} \)
Towards Transitions: A Partial Order on States

The substate- (or child-) relation induces a partial order on states:
- \( \text{top} \leq s \), for all \( s \in S \),
- \( s \leq s' \), for all \( s' \in \text{child}(s) \),
- transitive, reflexive, antisymmetric,
- \( s' \leq s \) and \( s'' \leq s \) implies \( s' \leq s'' \) or \( s'' \leq s' \).

Least Common Ancestor and Ting

- The least common ancestor is the function \( \text{lca} : 2^S \rightarrow S \) such that
  - The states in \( S_1 \) are (transitive) children of \( \text{lca}(S_1) \), i.e.
    \[ \text{lca}(S_1) \leq s, \text{ for all } s \in S_1 \subseteq S, \]
  - \( \text{lca}(S_1) \) is minimal, i.e. if \( \hat{s} \leq s \) for all \( s \in S_1 \), then \( \hat{s} \leq \text{lca}(S_1) \)
- **Note:** \( \text{lca}(S_1) \) exists for all \( S_1 \subseteq S \) (last candidate: top).
Least Common Ancestor and Ting

- Two states $s_1, s_2 \in S$ are called **orthogonal**, denoted $s_1 \perp s_2$, if and only if
  - they are unordered, i.e. $s_1 \not\leq s_2$ and $s_2 \not\leq s_1$, and
  - they live in different regions of an AND-state, i.e.
    $$\exists s, \text{region}(s) = \{S_1, \ldots, S_n\}, 1 \leq i \neq j \leq n : s_1 \in \text{child}(S_i) \land s_2 \in \text{child}(S_j),$$

- A set of states $S_1 \subseteq S$ is called **consistent**, denoted by $\downarrow S_1$, if and only if for each $s, s' \in S_1$,
  - $s \leq s'$, or
  - $s' \leq s$, or
  - $s \perp s'$.
Legal Transitions

A hierarchical state-machine \((S, \text{kind}, \text{region}, \rightarrow, \psi, \text{annot})\) is called well-formed if and only if for all transitions \(t \in \rightarrow,\)

1. source and destination are consistent, i.e. \(\downarrow \text{source}(t)\) and \(\downarrow \text{target}(t)\).
2. source (and destination) states are pairwise unordered, i.e.
   - \(\text{for all } s, s' \in \text{source}(t)\) (\(\in \text{target}(t)\)), \(s \perp s'\).
3. the top state is neither source nor destination, i.e.
   - \(\text{top} \notin \text{source}(t) \cup \text{source}(t)\).
4. Recall: final states are not sources of transitions.

Example:

CLAIM: \((\text{ii}) \Rightarrow (\iota)\)
References


