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
Composite States

(formalisation follows [Damm et al., 2003])
Composite States

- In a sense, composite states are about abbreviation, structuring, and avoiding redundancy.

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

![Diagram of Tron Player's Statemachine](image)

write

![Diagram of Resigned State](image)
Composite States

and instead of

\[ \text{fast}N \]

\[ \text{fast} \]

\[ \text{slow} \]

write
Recall: Syntax

translates to

\[
\begin{align*}
\{(top, st), (s, st), (s_1, st)(s_1', st)(s_2, st)(s_2', st)(s_3, st)(s_3', st)\},
\end{align*}
\]
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,

\[ \psi(t) = (\text{source}(t), \text{target}(t)). \]

*Special Case:* 

\[ \begin{array}{c}
\text{SPECIAL CASE: } \\
S_1 \xrightarrow{t_2} S_2
\end{array} \]

maps to: \( t_2, \{ t_2 \rightarrow (S_2, S_2), t_2 \rightarrow \text{ann} \} \)

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

**Diagram:** 

- Fork/Join diagram illustrating transitions and states.
- **Annotations:** 
  - **Joint:** Indicates state transitions.
  - **Fork:** Indicates state splits.
  - **Join:** Indicates state merges.
  - **Annot:** Indicates additional annotations.
Composite States: Blessing or Curse?

- what may happen on $E$?
- what may happen on $E$, $F$?
- can $E$, $G$ kill the object?
- ...

[Diagram of composite states with nodes and transitions labeled with $s_1$, $s_2$, $s_3$, $s_4$, $s_5$, $s_6$, $s_7$, $s_8$ and actions $E$, $F$, $G$.]
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?

- what may happen on $E$?
- what may happen on $E$, $F$?
- can $E$, $G$ kill the object?
- ...

![Diagram of states and transitions]
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
  - with 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.
    \[
    |\{ s \in R \mid \text{kind}(s) \in \{ st, \text{fin}\} \} \cap S_1 | = 1.
    \]

- Examples:

\[
\begin{array}{c|c|c|c}
\hline
s & s_1 & s_2 & s_3 \\
\hline
S_4 = \{ \text{top}, s_1, s_2 \} & \text{is LEGAL (or \( \{ s_3 \} \))} \\
\hline
S_1 = \{ s \} & \text{NOT LEGAL: top missing} \\
S_2 = \{ s, \text{top} \} & \text{NOT LEGAL: no child of s in} \ \ S_1 \\
S_3 = \{ s, \text{top}, s_1, s_3 \} & \text{NOT LEGAL: not exactly one} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\hline
s & s_1 & s_2 & s_3 \\
\hline
S_1 & s_1' & s_2' & s_3' \\
\hline
\text{child}(s) = \{ s_1, s_2, \ldots, s_3 \} & \text{NOT LEGAL, and} \\
S_7 = \{ \text{top}, s_1, s_2, s_3 \} & s_3 \text{ or } s_3' & \text{NOT LEGAL} \\
S_5 = \{ \text{top}, s_1, s_2 \} & \text{NOT LEGAL} \\
S_3 = \{ \text{top}, s_1, s_2, s_3 \} & \text{is LEGAL} \\
(\text{or \( \{ s_1, s_2, s_3 \} \) for short}) \\
\end{array}
\]
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 $s'' \leq s'$. 
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).
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
- $s \perp s'$.

**CLAIM:** $\forall S_1 \subseteq S$ • $S_1$ is legal state config. $\implies S_1$ is consistent
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 \( t \in \rightarrow, \)

(i) • source and destination are consistent, i.e. \( \downarrow \text{source}(t) \) and \( \downarrow \text{target}(t), \)

(ii) • source (and destination) states are pairwise unordered, i.e.

   • forall \( s, s' \in \text{source}(t) \ (\in \text{target}(t)), \ s \perp s', \)

(iii) • the top state is neither source nor destination, i.e.

   • \( \text{top} \notin \text{source}(t) \cup \text{source}(t). \)

• Recall: final states are not sources of transitions.

Example:

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


