Lecture 9: Live Sequence Charts

SOFTWARETECHNIK / SOFTWARE-ENGINEERING

2016-06-06

Prof. Dr. Andreas Podelski,  Dr. Bernd Westphal
Albert-Ludwigs-Universität Freiburg, Germany

Topics:
- Requirements Engineering: Content
- Excursion: Symbolic Büchi Automata
- LSC Semantics (Cuts, Firedsets, Automaton Construction, Full LSC)
- Pre-Charts
- Requirements Engineering with scenarios
- Strengthening scenarios into requirements
- Software specification
- Software specification and requirements engineering
- LSCs vs. Software
- Scenarios and tests

Requirements Engineering Wrap-Up
A Symbolic Büchi Automaton (TBA) is a tuple \( B = (C_B, Q, q_{ini}, \rightarrow, Q_F) \) where

- \( C_B \) is a set of atomic propositions,
- \( Q \) is a finite set of states,
- \( q_{ini} \in Q \) is the initial state,
- \( \rightarrow \subseteq Q \times \Phi(C_B) \times Q \) is the finite transition relation.
- Each transition \((q, \psi, q') \in \rightarrow\) from state \( q \) to state \( q' \) is labelled with a formula \( \psi \in \Phi(C_B) \).
- \( Q_F \subseteq Q \) is the set of fair (or accepting) states.

Definition. Let \( B = (C_B, Q, q_{ini}, \rightarrow, Q_F) \) be a TBA and \( w = \sigma_1, \sigma_2, \sigma_3, \ldots \in (\Phi(C_B) \rightarrow B)_{\omega} \) an infinite word, each letter is a valuation of \( \Phi(C_B) \).

An infinite sequence \( \bar{\upsilon} = q_0, q_1, q_2, \ldots \in Q_{\omega} \) of states is called a run of \( B \) over \( w \) if and only if

- \( q_0 = q_{ini} \),
- for each \( i \in \mathbb{N}_0 \) there is a transition \((q_i, \psi_i, q_{i+1}) \in \rightarrow\) s.t. \( \sigma_i \models \psi_i \).
Definition. Let $((L, \preceq, \sim), I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)$ be an LSC body.

A non-empty set $\emptyset \neq C \subseteq L$ is called a cut of the LSC body iff

1. $C$ is downward closed, i.e. $\forall l, l' \in L \cdot l' \in C \land l \preceq l' \Rightarrow l \in C$,
2. $C$ is closed under simultaneity, i.e. $\forall l, l' \in L \cdot l' \in C \land l \sim l' \Rightarrow l \in C$,
3. $C$ comprises at least one location per instance line, i.e. $\forall I \in I \cdot C \cap I \neq \emptyset$.

The temperature function is extended to cuts as follows:

$\Theta(C) = \{\text{hot}, \text{cold}\}$, if $\exists l \in C \cdot (\not\exists l' \in C \cdot l \prec l') \land \Theta(l) = \text{hot}$

otherwise, that is, $C$ is hot if and only if at least one of its maximal elements is hot.

Cut Examples

A Successor Relation on Cuts

The partial order $\preceq$ and the simultaneity relation $\sim$ of locations induce a direct successor relation on cuts of an LSC body as follows:

Definition. Let $C \subseteq L$ be a cut of LSC body $((L, \preceq, \sim), I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)$.

A set $\emptyset \neq F \subseteq L$ of locations is called fired-set $F$ of cut $C$ if and only if

1. $C \cap F = \emptyset$ and $C \cup F$ is a cut, i.e. $F$ is closed under simultaneity,
2. all locations in $F$ are direct $\prec$-successors of the front of $C$, i.e. $\forall l \in F \exists l' \in C \cdot l' \prec l \land (\not\exists l'' \in C \cdot l' \prec l'')$,
3. locations in $F$, that lie on the same instance line, are pairwise unordered, i.e. $\forall l \neq l' \in F \cdot (\exists I \in I \cdot \{l, l'\} \subseteq I) \Rightarrow l \not\preceq l' \land l' \not\preceq l$,
4. for each asynchronous message reception in $F$, the corresponding sending is already in $C$, i.e. $\forall (l, E, l') \in \text{Msg} \cdot l' \in F \Rightarrow l \in C$.

The cut $C' = C \cup F$ is called direct successor of $C$ via $F$, denoted by $C \Rightarrow F C'$. 
Live Sequence Charts (if well-formed)
• have an abstract syntax.
• From an abstract syntax, mechanically construct its TBA.
• A universal LSC is satisfied by a software S if and only if all words induced by the computation paths of S are accepted by the LSC's TBA.
• An existential LSC is satisfied by a software S if and only if there is a word induced by a computation path of S which is accepted by the LSC's TBA.

Pre-charts allow us to specify:
• anti-scenarios (“this must not happen”),
• activation interactions.

Method:
• discuss (anti-)scenarios with customer,
• generalise into universal LSCs and re-validate.

References