Software Engineering
Errata for the Course Slides

May 27, 2020

Note: The following ‘repairs’ apply to the slides of 2018 as distributed with the recordings. Most errors have been fixed with the 2019-slides, yet we assume it to be harder to match the 2019-slides against the English 2018-recordings (that we need because we are in an English language season) than to provide a consistent set of recordings, slides, and errata.

Contents

Lecture 4: Software Project Management (NEW) 1
  Slide 35, ‘Building Blocks Can Be Arbitrarily Complicated’ 1

Lecture 9: Scenarios & Use Cases 1
  Slide 46, ‘Language of LSC Body: Example’ 1

Lecture 10: Live Sequence Charts & RE Wrap-Up 2
  Slide 8, ‘Loop Condition’ 2
  Slide 10, ‘Example’ and Slide 5, ‘Language of LSC Body: Example’ 2
  Slide 40, ‘LSC Semantics with Pre-Chart’ 2

Lecture 12: Structural Software Modelling II 2
  Slide 28, ‘More Interesting Example’ 2

Lecture 4: Software Project Management (NEW)

Slide 35, ‘Building Blocks Can Be Arbitrarily Complicated’

The first item in the list to the right of the diagram needs to read:

If a test detected an error in $M$,

Lecture 9: Scenarios & Use Cases

Slide 46, ‘Language of LSC Body: Example’

Lecture 10: Live Sequence Charts & RE Wrap-Up

Slide 8, ‘Loop Condition’

The message aspect of the loop condition (first bullet point) needs to read

\[
\psi_{\text{Msg}}(q) = \neg \bigvee_{1 \leq i \leq n, \psi \in \text{Msg}(q_i \setminus q)} \psi \land \big(\text{strict} \implies \bigwedge_{\psi \in \mathcal{E}_{q_i} \cap \text{Msg}(\mathcal{L})} \neg \psi \big)
\]

that is, in the non-strict case, the loop accepts all letters where none of the messages of any successor cut is sent or received.

Slide 10, ‘Example’ and Slide 5, ‘Language of LSC Body: Example’

The loop condition of state \(q_6\) needs to read

\[
\neg (G_{I_2,I_1} \lor G_{I_2,I_1})
\]

and the progress condition from \(q_4\) to \(q_6\) needs to read

\[
F_{I_2,I_3} \land \neg G_{I_2,I_1} \land G_{I_2,I_1}
\]

Slide 40, ‘LSC Semantics with Pre-Chart’

Each of the four inner table cells had one \(\land\) too much, and the second lines need to read

\[
\land w^1, \ldots, w^m \in \text{Lang}_{\text{fin}}(\mathcal{B}(PC))
\]

and

\[
\land w^{k+1}, \ldots, w^m \in \text{Lang}_{\text{fin}}(\mathcal{B}(PC)),
\]

respectively.

Meaning: the sub-word consisting of the 1st (or \(k+1\)-th) up to \(m\)-th letter of word \(w\) is in the ‘finite’ language of the pre-chart TBA, i.e., we read the pre-chart TBA as a standard Deterministic Finite Automaton (DFA) with the standard DFA-acceptance criterion of reaching an accepting state with consumption of the last letter of the word.

Lecture 12: Structural Software Modelling II

Slide 28, ‘More Interesting Example’

The studied Proto-OCL formula needs to read:

\[
\forall c \in \text{allInstances}_C \cdot x(n(c)) \neq 27
\]