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)

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'

See Lecture 10, "Slide 10, 'Example' and Slide 5, 'Language of LSC Body: Example"' below.

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

$$\psi^{\mathsf{Msg}}(q) = \neg \bigvee_{1 \leq i \leq n, \psi \in \mathsf{Msg}(q_i \setminus q)} \psi \wedge \underbrace{\left(\mathit{strict} \implies \bigwedge_{\psi \in \mathcal{E}^{\mathcal{I}}_{!?} \cap \mathsf{Msg}(\mathcal{L})} \neg\psi \right)}_{=:\psi_{\mathrm{strict}}(q)}$$

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

$$\neg(G^{I_2,I_1}_{!} \vee G^{I_2,I_1}_{?})$$

and the progress condition from q_4 to q_6 needs to read

$$F^{I_2,I_3}_{?} \wedge \neg G^{I_2,I_1}_{!} \wedge \neg G^{I_2,I_1}_{?}$$

Slide 40, 'LSC Semantics with Pre-Chart'

Each of the four inner table cells had one '$\wedge$' too much, and the second lines need to read

$$\wedge\, w^1, \dots, w^m \in \mathit{Lang}_{\mathit{fin}}(\mathcal{B}(PC))$$

and

$$\wedge\, w^{k+1}, \dots, w^m \in \mathit{Lang}_{\mathit{fin}}(\mathcal{B}(PC)),$$

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:

$$\forall\, c \in \mathit{allInstances}_C \bullet x(n(c)) \neq 27$$