Software Engineering
Errata for the Course Slides

September 7, 2015

Contents

Lecture 7: DTs, Uselessness

Slide 21 (new!)

A couple of ticks (') went astray. The second bullet point should read:

- A rule $r = \varphi \to \alpha$ is called **useless** (or: **redundant**) if and only if there is another (different) rule r' whose premise φ' subsumes φ and whose effect is the same as r's, i.e. if

$$\exists\, r' = \varphi' \to \alpha', r' \neq r \bullet \models (\varphi' \implies \varphi) \wedge \alpha = \alpha'.$$

  r is called **subsumed** by r'.

Lecture 9: Live Sequence Charts

Slide 20

The correct definition of **active** for local invariants:

> Local invariant $(l_o, \iota_0, \phi, l_1, \iota_1)$ is *active* at cut q if and only if $l_0 \preceq l \prec l_1$ for some front location l of cut q or $l = l_1 \wedge \iota_1 = \bullet$.

Slide 22 (new!)

In the handwritten annotations, a couple of places which should have F? (consistently) obtained an erroneous F! instead; namely all transitions originating at q_4 or below.

The correct annotations are shown on Slide 17.

Lecture 12: Structural Software Modeling

Slide 29

The second argument of the interpretation function for the 4th bullet point of the semantics is missing.

$$\mathcal{I}[\![\forall\, c : C \bullet F]\!](\sigma, \beta) = \ldots$$

Lecture 15: Software Quality Assurance

Slide 28

The first and second bullet points of the second list have an x where there should be an r, i.e. they should correctly read:

- $\{(q+1) \cdot y + r = x \wedge r \geq 0\}\ q := q + 1\ \{P\}$ by (A2),
- $\{(q+1) \cdot y + (r - y) = x \wedge (r - y) \geq 0\}\ r := r - y\ \{(q+1) \cdot y + r = x \wedge r \geq 0\}$ by (A2),

Slide 30

The error was (at least consistently) copied over to slide 30. There the lowest triple labelled with "A2" should read:

$\{(q+1) \cdot y + r = x \wedge r \geq 0\}$

- $q := q + 1$

$\{q \cdot y + r = x \wedge r \geq 0\}$

Tutorial 4: TBA Annotation

Slide 30 (new!)

The annotation of the self-loop of q_2 should read

$$\varphi_1 \wedge \ldots$$

i.e. without logical negation ("¬").

Fun fact: conditions of *hot* conditions and local invariants never appear negated in a TBA annotation constructed from an LSC as introduced in the lecture.