Contents & Goals

Last Lecture:
• System configuration
• Transformer
• Action language: skip, update

This Lecture:
• Educational Objectives:
  - Capabilities for following tasks/questions.
  - What does this State Machine mean? What happens if I inject this event?
  - Can you please model the following behaviour.
  - What is: Signal, Event, Ether, Transformer, Step, RTC.
• Content:
  - Action Language: send (create/destroy later)
  - Run-to-completion Step
  - Putting It All Together

Transformer Cont'd

abstract syntax concrete syntax
skip
intuitive semantics
do nothing
well-typedness
/./.semantics

\[ t \left[ u \times \right](\sigma, \varepsilon) = \{ (\sigma, \varepsilon) \} \]

observables
\[ \text{Obs} \left[ u \times \right](\sigma, \varepsilon) = \emptyset \]
(error) conditions
Not defined if \[ I\left[ expr_1 \right]\left(\sigma, \beta\right) \text{ or } I\left[ expr_2 \right]\left(\sigma, \beta\right) \text{ not defined.} \]

Transformer: Update

abstract syntax concrete syntax
update \((expr_1, v, expr_2)\)
intuitive semantics
Update attribute \(v\) in the object denoted by \(expr_1\) to the value denoted by \(expr_2\).
well-typedness
\(expr_1: \tau_{C}\) and \(v: \tau \in \text{atr}(C)\); \(expr_2: \tau\); \(expr_1, expr_2\) obey visibility and navigability
semantics
\[ t\text{update}\left(\left[ expr_1, v, expr_2 \right]\right)\left[ u \times \right](\sigma, \varepsilon) = \{ (\sigma', \varepsilon) \} \]
where \(\sigma' = \sigma\left[ u \mapsto \sigma(u)\right]\left[ v \mapsto I\left[ expr_2\right]\left(\sigma, u\times\right)\right]\)

observed
\[ \text{Obs} \left[ expr_1, v, expr_2 \right]\left[ u \times \right] = \emptyset \]

Update Transformer Example

SM_C:
\[
\begin{align*}
  s_1 &:= x \\
  s_2 &:= x + 1
\end{align*}
\]

\[ t\text{update}\left(\left[ expr_1, v, expr_2 \right]\right)\left[ u \times \right](\sigma, \varepsilon) = \{ (\sigma', \varepsilon) \} \]

\(\sigma\): \(u_1: C x = 4 y = 0\)
\(u_2: C x = 5 y = 0\)
\(\varepsilon\): \(\varepsilon'\)
No show-stopper, because loops in the action annotation can be converted for, s

\( \sigma, u \mapsto \{ \} \)

of the

\( \cdot \)

\( \preceq \)

\( \in\)

use full blown denotational semantics.

Note

\( \text{dom}(u) \)

observables

\( \text{dom}(\sigma) \)

\( \text{in} \)

\( \text{m} \)

\( \text{signal instance, fill in its attributes, and place it in the ether.} \)

\( \text{Send Transformer Example} \)

Sequential Composition of Transformers

\( \text{Send Transformer Example} \)

\( \text{Transformers and Transformational Reasoning} \)
Then we call the sequence $N_i \leq u$ be the maximal sequence of indices such that $\sigma(0) = 1$ for $u$.

Note that, if $\sigma(a) = 1$ between $\sigma(b) = 0$ and $\sigma(c) = 0$, $\sigma(a) = 1$.

Example: Let $x$.

Intuition: The Run-to-Completion Step is the transition that may be taken multiple times during an RTC-step. A step is a transition to the next state in the automaton, and all later steps are considered to be present.

Notions of Steps: The Run-to-Completion Step.