Specification of state-based Systems

Prof. Martin Wirsing

25.11.2002

States and Transition Systems

Goals

- Understanding the difference between explicitly and implicitly given states
- Learn to understand states and transition systems

Functional View:

- A state is represented by a constructor term
- States are passed as explicit parameters of functions

Example (Stack):

The signature of stacks over natural numbers:

sig STACKSIG=

	sorts	Nat, Stack
	ops	empty : Stack
		push : Nat $\times$ Stack $\rightarrow$ Stack
		top : Stack $\rightarrow$ Nat
		pop : Stack $\rightarrow$ Stack

end

A state z_1 is given by following constructor term:

$$z_1 =_{def} \text{push}(2, \text{push}(1, \text{empty}))$$

The invocation $\text{pop}(z_1)$ changes the state z_1 into the state $z_2 = \text{push}(1, \text{empty})$.

State-based view:

- A state is implicit and
- The operations have following types.

ops empty : Unit
push : Nat $\rightarrow$ Unit
top : Unit $\rightarrow$ Nat
pop : Unit $\rightarrow$ Unit

The invocation of pop$()$ changes the state z_1 into the state z_2. A following push(4) leads to the state

$$z_3 =_{def} \text{push}(4, \text{push}(1, \text{empty}))$$

Example (State-based view of Person):

The description of persons requires several attributes:

$\text{name} : \text{Name}$
$\text{address} : \text{Address}$
$\text{age} : \text{Nat}$

i.e. a state is given by several components.
Operations:

$\text{getName} : \text{Unit} \longrightarrow \text{Name}$
$\text{getAddress} : \text{Unit} \longrightarrow \text{Address}$
$\text{getAge} : \text{Unit} \longrightarrow \text{Nat}$

Example (Functional (algebraic) view of Person):

generated type Person ::= makePerson(getName: Name;
getAddress: Address; getAge: Nat)

This defines the set of states with a constructor

$\text{makePerson} : \text{Name} \times \text{Address} \times \text{Nat} \rightarrow \text{Person}$

and the selection operations

$\text{getName} : \text{Person} \rightarrow \text{Name}$
$\text{getAddress} : \text{Person} \rightarrow \text{Address}$
$\text{getAge} : \text{Person} \rightarrow \text{Nat}$

Definition (State Signature):

Given a signature $\Sigma = (S, F, P)$ and a Σ-structure A.

- A state signature is a family of (system-) variables $(X_s)_{s \in S}$ with X_s countable for all $s \in S$.
- An A-state is an assignment $\sigma : X_s \to A_s$ for all $s \in S$.

Example (Person):

A state signature of Person is:

name : Name
address : Address
age : Nat

A structure A with the sorts Name, Address, Nat is e.g.

$$\begin{aligned} A_{\mathsf{Name}} &=_{def} \mathsf{char}^* \\ A_{\mathsf{Address}} &=_{def} \mathsf{char}^* \times \mathbb{N} \times \mathsf{char}^* \\ A_{\mathsf{Nat}} &=_{def} \mathbb{N} \end{aligned}$$

A state s is e.g.

$$\begin{aligned} s(\mathsf{name}) &= \mathsf{wirsing} \\ s(\mathsf{address}) &= (\mathsf{Schlagintweitstr.}, 18, \mathsf{München}) \\ s(\mathsf{age}) &= 53 \end{aligned}$$

A transition system describes a set of transitions:

Definition:

Given a signature $\Sigma = (S, F, P)$ and a state signature V.

1. A transition system $\Gamma = (Z, \delta)$ is given by
   - a set Z of Σ-states and
   - a transition relation $\delta \subseteq Z \times Z$.
2. A labelled transition system $\Gamma = (Z, \mathcal{A}, \delta)$ is given by
   - a set Z of states,
   - a set $\mathcal{A}$ of actions and
   - a transition relation $\delta \subseteq Z \times \mathcal{A} \times Z$.

Example (Stack):

Let K be a stack with

$K_{\mathsf{Stack}} =_{def} \mathbb{N}^*$

and a state signature

stack : Stack

Then the set of states equals K_{Stack}, i.e.

$Z =_{def} K_{\mathsf{Stack}} = \mathbb{N}^*$

the set of actions equals the invocations of stack operations, i.e.

$A =_{def} \{\mathsf{push}(n) \ \mid \in \mathbb{N}\} \cup \{\mathsf{pop}(), \mathsf{top}()\}$

and the labelled transition system has the transition relation δ with

$$
\begin{aligned}
&(\langle \mathsf{stack} = s \rangle, \mathsf{push}(n), \langle \mathsf{stack} = \langle n, s \rangle \rangle) \in \delta \\
&(\langle \mathsf{stack} = \langle n, s \rangle \rangle, \mathsf{pop}(), \langle \mathsf{stack} = s \rangle) \in \delta \\
&(\langle \mathsf{stack} = \langle n, s \rangle \rangle, \mathsf{top}(), \langle \mathsf{stack} = \langle n, s \rangle \rangle) \in \delta
\end{aligned}
$$

for all $n \in \mathbb{N}$.

top has not been considered adequately in this example.
Better: Include arguments and results of the operation into the action.

$$\begin{array}{rll}\mathcal{A} =_{def} & & \mathbb{N} \times \mathsf{push} \times \mathsf{Unit} \\ & \cup & \mathsf{Unit} \times \mathsf{pop} \times \mathsf{Unit} \\ & \cup & \mathsf{Unit} \times \mathsf{top} \times \mathbb{N}\end{array}$$

The transition relation has the form:

$$\begin{array}{l}(\langle \mathsf{stack} = s\rangle, (n, \mathsf{push}, ()), \langle \mathsf{stack} = \langle n, s\rangle\rangle) \\ (\langle \mathsf{stack} = \langle n, s\rangle\rangle, ((), \mathsf{pop}, ()), \langle \mathsf{stack} = s\rangle) \\ (\langle \mathsf{stack} = \langle n, s\rangle\rangle, ((), \mathsf{top}, n), \langle \mathsf{stack} = \langle n, s\rangle\rangle)\end{array}$$

for all $n \in \mathbb{N}$.