Recursion: algebraic approach

Let $E$ be a recursive specification over signature $\Sigma$ and set of variables $V_R$.

Furthermore, let $A$ be a $\Sigma$-algebra and $\iota$ the associated interpretation.

**Solution**

A solution of $E$ in $A$ is an extension $\kappa$ of $\iota$ with interpretations of the recursion variables in $V_R$ as elements of $A$ such that $A, \kappa \models X = t_X$ for every equation $X = t_X$ in $E$. If $\kappa$ is a solution of $E$ and $X \in V_R$, then we shall call $\kappa(X)$ a solution of $X$ in $E$.

NB: a solution of $E$ in an algebra of transition systems modulo bisimilarity is an assignment of closed terms to recursion variables such that the recursion equations are true up to bisimilarity.

Solutions: examples

1. The recursive specification $E_1 = \{ X = a.1 \}$ has a solution both in $P(BSP(A))/\leftrightarrow$ and in $P(BSP(A)+E_1)/\leftrightarrow$.

2. The recursive specification $E_2 = \{ X = a.X \}$ has a solution in $P(BSP(A)+E_2)/\leftrightarrow$, but not in $P(BSP(A))/\leftrightarrow$.

3. The recursive specification $E_3 = \{ X = X \}$ has many solutions, both in $P(BSP(A))/\leftrightarrow$ and in $P(BSP(A)+E_3)/\leftrightarrow$.

Recursive Definition Principle

RDP

Let $\Sigma$ be a signature; we say that $\Sigma$-algebra $A$ satisfies the Recursive Definition Principle if every recursive specification $E$ over $\Sigma$ and some set $V_R$ of variables has at least one solution.

Does $P(BSP(A))/\leftrightarrow$ satisfy RDP? No!

Let's construct a model of $BSP(A)$ that does satisfy RDP! (see next slide)
Term model (1)

Denote by $\text{Rec}$ the collection of all recursive specifications. We denote by $\text{BSP}_{\text{rec}}(A)$ the extension of $\text{BSP}(A)$ with, for every recursive specification $E$ over $\text{BSP}(A)$ and for every recursion variable $X$ defined in $E$, a constant symbol $\mu X.E$, which will stand for the transition system assigned to $X$ in the solution of $E$ as defined by the operational semantics.

The term algebra $\mathbb{P}(\text{BSP}_{\text{rec}}(A))$ for $\text{BSP}_{\text{rec}}(A)$ is the algebra

$(\mathcal{C}(\text{BSP}_{\text{rec}}(A)), +, (a)_a A, 0, 1, (\mu X.E)_{E \in \text{Rec}, X \in V(R)}(E))$

Term model (2)

[It is convenient to generalise the notation $\mu X.E$ to arbitrary terms, writing $\mu t.E$ for the term $t$ in the occurrences of recursion variables are interpreted in $E$ (see book for an inductive definition).]

\[ a.x \xrightarrow{a} x \quad x + y \xrightarrow{a} x' \]
\[ y \xrightarrow{a} y' \]
\[ \frac{1}{x} \quad \frac{x + y}{x + y} \]
\[ \frac{\mu t.X.E}{\mu X.E} \xrightarrow{t_X} \quad \frac{\mu t.X.E}{\mu X.E} \xrightarrow{t_X} \]

Bisimilarity is a congruence on $\mathbb{P}(\text{BSP}_{\text{rec}}(A))$, and $\mathbb{P}(\text{BSP}_{\text{rec}}(A))/\equiv$ is a model for the equational theory $\text{BSP}(A)$.

Term model (3)

The algebra $\mathbb{P}(\text{BSP}_{\text{rec}}(A))/\equiv$ is generally referred to as the term model for $\text{BSP}(A)$.

Theorem

The term model for $\text{BSP}(A)$ satisfies RDP (notation: $\mathbb{P}(\text{BSP}_{\text{rec}}(A))/\equiv \models \text{RDP}$).

Proof.
Let $E$ be a recursive specification. Define $\kappa$ as the extension of $\iota$ such that, for every recursion variable $X$ in $E$,

$\kappa(X) = [\mu X.E]_{\equiv}$.

Then $\kappa(X) = \kappa(t_X)$ for every recursion variable $X$ in $E$ (verify!), so $\kappa$ is indeed a solution of $E$ in $\mathbb{P}(\text{BSP}_{\text{rec}}(A))/\equiv$. □

Example: equivalence of rec. vars.

Consider the recursive specification

\[ \{ \begin{array}{l} X = a.X, \\ Y = a.a.Y \end{array} \} \]

Note that we can argue that every solution of $X$ is a solution of $Y$ too:

$X = a.X = a.a.X$.

Hence, any solution $\kappa$ of $E$ in some algebra $A$ satisfies

$\kappa(X) = \kappa(a.a.X) = \iota(a.)\iota(a.)\kappa(X)$,

so $\kappa(X)$ is a solution of $Y$ in $E$. 
Example: equivalence of rec. vars.

Consider the recursive specification
\[
\begin{align*}
X &= a.X \\
Y &= a.a.Y
\end{align*}
\]

The reasoning on the previous slide allows us to conclude that every solution of \( X \) in whatever algebra (!) must also be a solution of \( Y \) in that algebra.

The converse, however, need not hold:

Exercise: construct a model of \( \text{BSP}(A) \) in which \( Y \) has a solution that is not also a solution of \( X \).

Perhaps if we exclude some models (e.g., the answers to the exercise on the previous slide), then we may be able to say more about the equivalence of \( X \) and \( Y \) in the above recursive specification.

Note that, for models in which \( X \) and \( Y \) both have a unique solution, the reasoning on slide 17 would suffice to conclude that \( X \) and \( Y \) indeed denote the same process!

Guardedness

Definition

An occurrence of a recursion variable \( X \) in a \( \text{BSP}_{\text{rec}}(A) \)-term \( s \) is guarded if it occurs in the scope of an action prefix.

A \( \text{BSP}_{\text{rec}}(A) \)-term \( s \) is completely guarded if all occurrences of all recursion variables in \( s \) are guarded.

A recursive specification \( E \) is completely guarded if all right-hand sides of all equations in \( E \) are completely guarded.

Exercise 5.5.1

Determine whether, in the following terms, the occurrences of the recursion variables \( X \) and \( Y \) are guarded, unguarded, or both:

\[ a.X , \quad Y + b.X , \quad b.(X + Y) , \quad a.Y + X \]
Guardedness

Definition
A recursive specification $E$ is guarded if there exists a completely guarded recursive specification $F$ with $\mathcal{V}_R(E) = \mathcal{V}_R(F)$ and $\text{BSP}(A) + E \vdash X = t$ for all $X = t \in F$.

Example
Although the recursive specification $E_2 = \{ X_1 = a.X_1, Y_1 = X_1 \}$ is not completely guarded, it is guarded.

Recursive Specification Principle

RSP
Let $\Sigma$ be a signature; we say that $\Sigma$-algebra $A$ satisfies the Recursive Specification Principle (RSP) if every guarded recursive specification $E$ over $\Sigma$ and some set $\mathcal{V}_R$ of variables has at most one solution.

Theorem
The term model $\mathcal{P}(\text{BSP}_{\text{rec}}(A))/\equiv$ satisfies RSP (notation: $\mathcal{P}(\text{BSP}_{\text{rec}}(A))/\equiv \models \text{RSP}$).

Proof.
[Postponed.]

RSP as a proof principle

Example
Consider rec. spec. $E$ consisting of the following equations:

\[
\begin{align*}
X &= a.X + b.X, \\
Y &= a.Y + b.Z, \text{ and} \\
Z &= a.Z + b.Y.
\end{align*}
\]

We can prove that $X = Y$ in the context of $E$ as follows:

Define two sequences of terms $\vec{t} = t_X, t_Y, t_Z$ and $\vec{u} = u_X, u_Y, u_Z$ by $t_X \equiv X$, $t_Y \equiv Y$, $t_Z \equiv Z$, and $u_X \equiv X$, $u_Y \equiv X$, $u_Z \equiv X$.

Then both $\vec{t}$ and $\vec{u}$ denote solutions of $E$ (verify!).

Since $E$ is guarded, by RSP, $\vec{t} = \vec{u}$, so $X \equiv u_Y = t_y \equiv Y$. 
We extend BSP($A$) with unary projection operators $\pi_n$ ($n \in \mathbb{N}$):

The process $\pi_n(p)$ executes the behaviour of $p$ up to depth $n$ (i.e., it executes the first $n$ actions of $p$).

Examples

- $\pi_0(a.0 + b.c.1) = 0$ ;
- $\pi_1(a.0 + b.c.1) = a.0 + b.0$ ;
- $\pi_n(a.0 + b.c.1) = a.0 + b.c.1$ if $n \geq 2$.

Projection: operational semantics

\[
\begin{align*}
& x \quad a \to x' \\
\pi_{n+1}(x) & \quad a \to \pi_n(x') \\
\pi_n(x) & \downarrow
\end{align*}
\]

Exercise 5.5.8

Prove that for all closed (BSP + PR)$_{rec}(A)$-terms $p$ and $q$ and every $n \in \mathbb{N}$, $\pi_{n+1}(p) \iff \pi_n(q)$ implies $\pi_n(p) \iff \pi_n(q)$.

Projection: axioms

To get the equational theory BSP+PR($A$) we extend the equational theory BSP($A$) with the following axioms:

- $\pi_n(1) = 1$ PR1
- $\pi_n(0) = 0$ PR2
- $\pi_0(a.x) = 0$ PR3
- $\pi_{n+1}(a.x) = a.\pi_n(x)$ PR4
- $\pi_n(x + y) = \pi_n(x) + \pi_n(y)$ PR5

Exercise 5.5.4

Consider the recursive specification $\{X = a.X + b.c.X\}$. Calculate $\pi_0(X)$, $\pi_1(X)$, and $\pi_2(X)$.

Exercise 5.5.6

Consider the recursive specification $\{X = a.X + b.X\}$. Determine $\pi_n(X)$ for every $n \in \mathbb{N}$.

Approximation Induction Principle

AIP

Let $\Sigma$ be a signature including projection operators $\pi_n$ ($n \in \mathbb{N}$); we say that $\Sigma$-algebra $A$ satisfies the Approximation Induction Principle (AIP) if, for arbitrary $\Sigma$-terms $s$ and $t$, $A \models \pi_n(s) = \pi_n(t)$ for all $n \in \mathbb{N}$ implies $A \models s = t$.

Example 5.5.18 + Exercise 5.5.5

Consider the recursive specifications $\{X_1 = a.X_1\}$ and $\{X_2 = a.a.X_2\}$. Prove that $\pi_n(X_1) = a^n.0 = \pi_n(X_2)$ for every $n \in \mathbb{N}$, and conclude, using AIP, that $X_1 = X_2$.

(See Notation 4.6.6 on p. 105 of the book for the definition of $a^n.p$.)
Read Sections 4.5–5.5.

Do Exercises 5.5.1, 5.5.2, 5.5.3, 5.5.7, 4.5.1, 4.5.2, 4.5.5, 4.5.6, 5.5.4, 5.5.5, 5.5.6