Expressiveness

General method for expressing transition systems:
1. label states with process names (a.k.a. recursion variables);
2. associate behaviour to every recursion variable by means of an equation specifying for every recursion variable its transition- and termination-behaviour.

\[ X = a.Y + c.Z \]
\[ Y = b.X + 1 \]
\[ Z = 0 \]

Recursive specifications

Let \( \Sigma \) be a signature and let \( V_R \) be a set of recursion variables.

A recursive equation over \( \Sigma \) and \( V_R \) is an equation of the form

\[ X = t \]

with \( X \in V_R \) and \( t \) a term over \( \Sigma \) and \( V_R \). We say that the recursive equation \( X = t \) defines \( X \).

A recursive specification over \( \Sigma \) and \( V_R \) is a set of recursive equations over \( \Sigma \) and \( V_R \) consisting of precisely one recursive equation defining \( X \) for every \( X \in V_R \).

Recursion: operational semantics

[ Warning: the notation in the rules below is slightly simplified; we will generalise later! ]

Simplifying assumption
We first present the operational semantics for recursion variables in the context of a fixed recursive specification \( E \) over \( \Sigma \) and \( V_R \). Furthermore, we assume that there is a term \( t_X \) associated with every \( X \in V_R \) such that \( X = t_X \) is the defining equation for \( X \) in \( E \).

Operational rules for recursion

\[
\begin{align*}
X & \xrightarrow{a} t' X \\
X & \downarrow
\end{align*}
\]
Term model (simplified)

Let $E$ be a recursive specification over $\text{BSP}(A)$ and $V_R$.

**Definition**
The term algebra for $\text{BSP}(A)+E$ is the algebra

$$P(\text{BSP}(A)+E) = (C(\text{BSP}(A)+E), +, (\cdot)_{a \in A}, 0, 1, (X)_{X \in V_R}).$$

**Proposition**
Bisimilarity is a congruence on $P(\text{BSP}(A)+E)$.

**Theorem**
The equational theory $\text{BSP}(A)+E$ is a sound axiomatisation of $P(\text{BSP}(A)+E)/\leftrightarrow$.

Is $\text{BSP}(A)+E$ also ground-complete for $P(\text{BSP}(A)+E)/\leftrightarrow$?

Equivalence of recursion variables

**Example**
Consider the recursive specification

$$\begin{align*}
X &= a.X, \\
Y &= a.a.Y
\end{align*}$$

Then $X \not\leftrightarrow Y$.

But this equation cannot be derived from $\text{BSP}(A)+E$.

Conclusion: we need additional methods to reason about the equivalence of recursion variables.

[Remark: we shall not discuss a full-fledged ground-complete axiomatisation of the algebra $P(\text{BSP}(A)+E)/\leftrightarrow$, but we'll come close.]

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 $P(\text{BSP}(A))/\leftrightarrow$ and in $P(\text{BSP}(A)+E_1)/\leftrightarrow$.
2. The recursive specification $E_2 = \{ X = a.X \}$ has a solution in $P(\text{BSP}(A)+E_2)/\leftrightarrow$, but not in $P(\text{BSP}(A))/\leftrightarrow$.
3. The recursive specification $E_3 = \{ X = X \}$ has many solutions, both in $P(\text{BSP}(A))/\leftrightarrow$ and in $P(\text{BSP}(A)+E_3)/\leftrightarrow$. 


Recursive Definition Principle

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 $Rec$ the collection of all recursive specifications. We denote by $BSP_{rec}(A)$ the extension of $BSP(A)$ with, for every recursive specification $E$ over $BSP(A)$ 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** $P(BSP_{rec}(A))$ for $BSP_{rec}(A)$ is the algebra $(C(BSP_{rec}(A)), +, (a.)_{a \in A}, 0, 1, (\mu X.E)_{E \in Rec, X \in V_R(E)})$.

Term model (2)

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

\[
\begin{align*}
  a.x & \xrightarrow{a} x \\
  x + a & \xrightarrow{a} x' \\
  y + a & \xrightarrow{a} y' \\
  x \downarrow & \xrightarrow{a} x' \\
  y \downarrow & \xrightarrow{a} y' \\
  \mu t.X.E & \xrightarrow{a} t'_X \\
  \mu X.E & \xrightarrow{a} \mu X.E' \\
  \mu X.E & \xrightarrow{a} \mu X.E'
\end{align*}
\]

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

Term model (3)

The algebra $P(BSP_{rec}(A))/\leftrightarrow$ is generally referred to as the **term model** for $BSP(A)$.

**Theorem**

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

**Proof.**

Let $E$ be a recursive specification. Define $\kappa$ as the extension of $i$ such that, for every recursion variable $X$ in $E$, $\kappa(X) = [\mu X.E]/\leftrightarrow$.

Then $\kappa(X) = \kappa(t_X)$ for every recursion variable $X$ in $E$ (verify!), so $\kappa$ is indeed a solution of $E$ in $P(BSP_{rec}(A))/\leftrightarrow$. □
Example: equivalence of rec. vars.

Consider the recursive specification

\[ \begin{align*}
X &= a.X \\
Y &= a.a.Y
\end{align*} \]

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 (I) 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 \(X\).

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:

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

Example
1. The recursive specification
   \[ E_1 = \{ X_1 = a.X_1, Y_1 = a.X_1 \} \]
   is completely guarded.
2. The recursive specification
   \[ E_2 = \{ X_1 = a.X_1, Y_1 = X_1 \} \]
   is not completely guarded.
But \( E_1 \) and \( E_2 \) have exactly the same solutions in every model!

Definition
A recursive specification \( E \) is guaraded if there exists a completely guarded recursive specification \( F \) with \( V_R(E) = V_R(F) \) 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.

Exercise 5.5.2
Determine whether the following recursive specifications are guaraded or unguaraded:
1. \( \{ X = Y, Y = a.X \} \); and
2. \( \{ X = a.Y + Z, Y = b.Z + X, Z = c.X + Y \} \).

Recursive Specification Principle

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

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

Proof.
[Postponed.]
RSP as a proof principle

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

\[
X = a.X + b.X, \\
Y = a.Y + b.Z, \text{ and} \\
Z = a.Z + b.Y.
\]

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 $X \equiv u_Y = t_y \equiv Y$.

Projection

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

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

Exercise 5.5.8
Prove that for all closed (BSP + PR)$_{rec}$($A$)-terms $p$ and $q$ and every $n \in \mathbb{N}$, $\pi_{n+1}(p) \equiv \pi_{n+1}(q)$ implies $\pi_n(p) \equiv \pi_n(q)$.

Projection: axioms

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

\[
\begin{align*}
\pi_n(1) &= 1 & \text{PR1} \\
\pi_n(0) &= 0 & \text{PR2} \\
\pi_n(a.x) &= 0 & \text{PR3} \\
\pi_{n+1}(a.x) &= a.\pi_n(x) & \text{PR4} \\
\pi_n(x + y) &= \pi_n(x) + \pi_n(y) & \text{PR5}
\end{align*}
\]

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 $\mathfrak{A}$ satisfies the Approximation Induction Principle (AIP) if, for arbitrary $\Sigma$-terms $s$ and $t$, $\mathfrak{A} \models \pi_n(s) = \pi_n(t)$ for all $n \in \mathbb{N}$ implies $\mathfrak{A} \models s = t$.

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 hence $\mathfrak{A} \models n(X_1) = n(X_2)$.

(See Notation 4.6.6 on p. 105 of the book for the definition of $a^n.p.$)

Restricting AIP

We say that a $(\text{BSP + PR})_{\text{rec}}(A)$-term $s$ is finitely branching if for every $s'$ reachable from $s$ the set \{ $s'' \mid \exists a \in A. \ s' \xrightarrow{a} s''$ \} is finite.

Theorem
Let $s$ and $t$ be $(\text{BSP + PR})_{\text{rec}}(A)$-terms, and suppose that $s$ is finitely branching. If $\mathbb{P}((\text{BSP + PR})_{\text{rec}}(A))/\not\approx \models \pi_n(s) = \pi_n(t)$ for all $n \in \mathbb{N}$, then $\mathbb{P}((\text{BSP + PR})_{\text{rec}}(A))/\not\approx \models s = t$.

Proof. [not discussed in lecture]
Define a relation $R$ by

$$R = \{(u, v) \mid \forall n \in \mathbb{N}. \ \pi_n(u) \not\subseteq \pi_n(v) \& u \text{ or } v \text{ is finitely branching}\}.$$  

By definition, $R$ is symmetric, so it suffices to check only two of the four conditions of the definition of bisimilarity. It is straightforward to show that $u \leftrightarrow u'$ implies $v \leftrightarrow v'$.

[ On the next slide it is argued that if $u \xrightarrow{a} u'$ for some $u'$, then there exists $v'$ such that $v \xrightarrow{a} v'$ and $(u', v') \in R$. ]

AIP: validity

Consider the recursive specification

$\{ X_n = a^n.0 + X_{n+1} \mid n \in \mathbb{N} \} \cup \{ Y = a.Y \}$.

Then $X_0 = \sum_{i=0}^{n} a^i.0 + X_{n+1}$ for all $n \in \mathbb{N}$, and hence $X_0 = X_0 + a^n.0$ for all $n \in \mathbb{N}$.

Therefore

$$\pi_n(X_0 + Y) = \pi_n(X_0) + \pi_n(Y) = \pi_n(X_0) + a^n.0 = \pi_n(X_0) + \pi_n(a^n.0) = \pi_n(X_0 + a^n.0) = \pi_n(X_0).$$

Note, however, that $X_0 + Y \not\approx X_0$.

Conclusion: AIP is not valid in $\mathbb{P}((\text{BSP + PR})_{\text{rec}}(A))/\not\approx$.

Restricting AIP

$[ R = \{(u, v) \mid \forall n \in \mathbb{N}. \ \pi_n(u) \not\subseteq \pi_n(v) \& u \text{ or } v \text{ is finitely branching}\} ].$

Proof (cnt'd). [not discussed in lecture]
Suppose that $u \xrightarrow{a} u'$ and consider the set

$$S_m = \{ v' \mid v \xrightarrow{a} v' \& \pi_m(u') \not\subseteq \pi_m(v') \}.$$  

Observe that $S_m \subseteq S_{m+1}$ (cf. Exercise 5.5.8) and $S_m \neq \emptyset$ for every $m \in \mathbb{N}$.

On the one hand, if $v$ is finitely branching, then $S_m$ is also finite for every $m \in \mathbb{N}$, so $\bigcap_{m \in \mathbb{N}} S_m \neq \emptyset$. Let $v' \in \bigcap_{m \in \mathbb{N}} S_m$; then $v \xrightarrow{a} v'$, and $\pi_m(u') \not\subseteq \pi_m(v')$ for all $n \in \mathbb{N}$, so $(u', v') \in R$.

On the other hand, if $v$ is not finitely branching, then $u$ is. For all $m \in \mathbb{N}$, pick $v_m \in S_m$; since $v \xrightarrow{a} v_m$, by the preceding argument there exists $u_m$ such that $u \xrightarrow{a} u_m$ and $(u_m, v_m) \in R$.

Since $u$ is finitely branching, some $u''$ occurs infinitely often in $(u_m)_{m \in \mathbb{N}}$.

Suppose, in particular, that $u''\equiv u_k$ for some $k \in \mathbb{N}$. Then it can be argued that $\pi_n(u'') \not\subseteq \pi_n(v_k)$ for all $n \in \mathbb{N}$, and hence $(u', v_k) \in R$. 
### Head Normal Form

Let $A$ be a set of actions and let $T(A)$ be some process theory.

The set of **head normal forms** for $T(A)$ is inductively defined by:

1. $0$ and $1$ are head normal forms;
2. if $a \in A$ and $t$ is a $T(A)$-term, then $a \cdot t$ is a head normal form;
3. if $s$, $t$ are head normal forms, then $s + t$ is a head normal form.

### The process theory $(BSP + PR)_{rec}(A)$

The equational theory $(BSP + PR)_{rec}(A)$ has the following axioms:

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>$x + y = y + x$</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>$(x + y) + z = x + (y + z)$</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>$x + x = x$</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>$x + 0 = x$</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>$\mu X . E = \mu t . E$</td>
<td></td>
</tr>
<tr>
<td>PR1</td>
<td>$\pi_n(1) = 1$</td>
<td></td>
</tr>
<tr>
<td>PR2</td>
<td>$\pi_n(0) = 0$</td>
<td></td>
</tr>
<tr>
<td>PR3</td>
<td>$\pi_0(a \cdot x) = 0$</td>
<td></td>
</tr>
<tr>
<td>PR4</td>
<td>$\pi_{n+1}(a \cdot x) = a \cdot \pi_n(x)$</td>
<td></td>
</tr>
<tr>
<td>PR5</td>
<td>$\pi_n(x + y) = \pi_n(x) + \pi_n(y)$</td>
<td></td>
</tr>
</tbody>
</table>

### Head Normal Form Property

A closed $(BSP + PR)_{rec}(A)$-term $t$ is **guarded** if for every occurrence of the symbol $\mu X . E$ in $t$ it holds that the recursive specification $E$ is guarded.

**Proposition 5.5.26**

$(BSP + PR)_{rec}(A)$ has the head-normal-form (HNF) property: for every guarded closed $(BSP + PR)_{rec}(A)$-term $t$ there exists a guarded head normal form $t'$ such that $(BSP + PR)_{rec}(A) \vdash t = t'$.

**Proof.**

Assume, without loss of generality, that every recursive specification occurring in $t$ is completely guarded. Then, the result follows by a straightforward induction on the structure of $t$.

---

NB: The proof in the book is not completely accurate! It does not properly deal with the case that $t \equiv \mu X . E$?
Finitely branching

Proposition 5.5.25
If $T(A)$ extends $\text{MT}(A)$, then for every head normal form $t$ for $T(A)$ there exist $n \in \mathbb{N}$, actions $a_i \in A$ and $T(A)$-terms $t_i$ ($0 \leq i < n$) s.t.

$$T(A) \vdash t = \sum_{0 \leq i < n} a_i.t_i[+1].$$

(The notation $\sum_{0 \leq i < n}$ is used to denote an alternative composition of $n$ summands, and $[+1]$ means that there may be an optional 1-summand.)

Exercise 5.5.9
Prove that every guarded $(\text{BSP} + \text{PR})_{\text{rec}}(A)$-term $t$ is finitely branching.

Restricted AIP

Theorem
The term model $P((\text{BSP} + \text{PR})_{\text{rec}}(A))/\leftrightarrow$ satisfies the following Restricted Approximation Induction Principle

if $s$ and $t$ are $(\text{BSP} + \text{PR})_{\text{rec}}(A)$-terms, $s$ is guarded, and $\pi_n(s) = \pi_n(t)$ for every $n \in \mathbb{N}$, then $s = t$.

Proof.
Immediate consequence of Exercise 5.5.9 and the theorem on slide 34.

Validity of RSP

Theorem (Projection Theorem)
Let $E$ be a guarded recursive specification and $X$ a recursion variable in $V_R(E)$. For all $(\text{BSP} + \text{PR})_{\text{rec}}(A)$-terms $s$ and $t$ satisfying the equation defining $X$ in $E$ it holds that $\pi_n(s) = \pi_n(t)$ for all $n \in \mathbb{N}$.

Corollary
$P((\text{BSP} + \text{PR})_{\text{rec}}(A))/\leftrightarrow$ satisfies RSP.

Proof.
Let $\kappa_1$ and $\kappa_2$ be solutions of some guarded recursive specification $E$. Then $\kappa_1$ and $\kappa_2$ assign equivalence classes of $(\text{BSP} + \text{PR})_{\text{rec}}(A)$-terms to every recursion variable.
Suppose that $\kappa_1(X) = [s]_{\leftrightarrow}$ and $\kappa_2(X) = [t]_{\leftrightarrow}$. Clearly, since $\kappa_1$ and $\kappa_2$ are solutions, both $s$ and $t$ satisfy the equation defining $X$ in $E$. So, by the Projection Theorem, $\pi_n(s) \leftrightarrow \pi_n(t)$ for all $n \in \mathbb{N}$, and hence, by AIP$, s \leftrightarrow t$. It follows that $\kappa_1(X) = [s]_{\leftrightarrow} = [t]_{\leftrightarrow} = \kappa_2(X)$. 
