Postgraduate Course: Process Algebra

Communicating Sequential Process

Chapter 1 Processes

Huibiao Zhu

Software Engineering Institute, East China Normal University

Events

1. A Simple Vending Machine

   coin——the insertion of a coin in the slot of a vending machine;

   choc——the extraction of a chocolate from the dispenser of the machine.

2. More Complex Vending Machine

   in1p——the insertion of one penny;

   in2p——the insertion of a two penny coin;

   small——the extraction of a small biscuit or cookie;

   large——the extraction of a large biscuit or cookie;

   out1p——the extraction of one penny in change.

Prefix

Let x be an event and let P be a process. Then

$$(x \rightarrow P) \text{ (pronounced "} x \text{ then } P\text{")}$$

describes an object which first engages in the event x and then behaves exactly as described by P. The process $(x \rightarrow P)$ is defined to have the same alphabet as P, so this notation must not be used unless x is in that alphabet; more formally,

$$\alpha(x \rightarrow P) = \alpha(P), \qquad \text{provided } x \in \alpha(P)$$

Example A simple vending machine that successfully serves two customers before breaking

$$(coin \rightarrow (choc \rightarrow (coin \rightarrow (choc \rightarrow STOP_{\alpha VMS}))))$$

Note: In future, we shall omit brackets in the case of linear sequences of events, on the convention that $\rightarrow$ is right associative.

Wrong: $P \rightarrow Q$

$x \rightarrow y$

Correct: $x \rightarrow (y \rightarrow STOP)$

Recursion

1. Example

$\alpha CLOCK = \{tick\}$

$CLOCK = (tick \rightarrow CLOCK)$

This can be regarded as an implicit definition of the behaviour of the clock.

Formal Definition:

$CLOCK = \mu X : \{tick\} \bullet (tick \rightarrow X)$

2. Formal Definition

If $F(X)$ is a guarded expression containing the process name X , and A is the alphabet of X , then we claim that the equation

$$X = F(X)$$

has a unique solution with alphabet A.

It is sometimes convenient to denote this solution by the expression

$$\mu X : A \bullet F(X)$$

Here X is a local name (bound variable), and can be changed at will.

3. Example

A simple vending machine which serves as many chocs as required

$$VMS = (coin \rightarrow (choc \rightarrow VMS))$$

Formal definition: $VMS = \mu X : \{coin, choc\} \bullet (coin \rightarrow (choc \rightarrow X))$

Choice

If x and y are distinct events

$$(x \to P | y \to Q)$$

describes an object which initially engages in either of the events x or y. After the first event has occurred, the subsequent behaviour of the object is described by P if the first event was x, or by Q if the first event was y. Since x and y are different events, the choice between P and Q is determined by the first event that actually occurs.

$\alpha(x \to P | y \to Q) = \alpha P$ provided $\{x, y\} \in \alpha P$ and $\alpha P = \alpha Q$

The bar | should be pronounced *choice*: *x then P choice y then Q*

Example 1 A machine that serves either chocolate or toffee on each transaction

$$VMCT = \mu X \bullet coin \rightarrow (choc \rightarrow X \mid toffee \rightarrow X)$$

Example 2 A more complicated vending machine, which offers a choice of coins and a choice of goods and change

$$\begin{aligned} VMC = (in2p \rightarrow (&large \rightarrow VMC \\ &|small \rightarrow out1p \rightarrow VMC) \\ \mid in1p \rightarrow (&small \rightarrow VMC \\ &\mid in1p \rightarrow (large \rightarrow VMC \\ &\qquad \mid in1p \rightarrow STOP))) \end{aligned}$$

Example A copying process engages in the following events

in.0——input of zero on its input channel

in.1——input of one on its input channel

out.0——output of zero on its output channel

out.1——output of one on its output channel

Its behaviour consists of a repetition of pairs of events. a bit and outputs the same bit

$$COPYBIT = \mu X \bullet (in.0 \rightarrow out.0 \rightarrow X | in.1 \rightarrow out.1 \rightarrow X)$$

Choice with More Than Two Alternatives

$$(x \rightarrow P \mid y \rightarrow Q \mid ... \mid z \rightarrow R)$$

$$(x : B \rightarrow P(x))$$

Incorrect: $(x \rightarrow P \mid x \rightarrow Q)$

Example: A process which at all times can engage in any event of its alphabet A

$$\alpha RUN_A = A$$

$$RUN_A = (x : A \rightarrow RUN_A)$$

Mutual Recursion

Example A drinks dispenser has two buttons labelled ORANGE and LEMON. The actions of pressing the two buttons are setorange and setlemon. The actions of dispensing a drink are orange and lemon. The choice of drink that will next be dispensed is made by pressing the corresponding button. Before any button is pressed, no drink will be dispensed. Here are the equations defining the alphabet and behaviour of the process DD. The definition uses two auxiliary definitions of O and L, which are mutually recursive

$$\alpha DD = \alpha O = \alpha L = \{setorange, setlemon, orange, lemon\}$$

$$DD = (setorange \rightarrow O | setlemon \rightarrow L)$$

$$O = (orange \rightarrow O \mid setlemon \rightarrow L \mid setorange \rightarrow O)$$

$$L = (lemon \rightarrow O \mid setorange \rightarrow O \mid setlemon \rightarrow L)$$

Laws

For examples, $(x \rightarrow P|y \rightarrow Q) \ = \ (y \rightarrow Q|x \rightarrow P)$

L1 $(x : A \rightarrow P(x)) = (y : B \rightarrow Q(y)) \equiv (A = B \wedge \forall x : A \bullet P(x) = Q(x))$

L1A $STOP \neq (d \rightarrow P)$

L1B $(c \rightarrow P) \neq (d \rightarrow Q)$ if $c \neq d$

L1C $(c \rightarrow P \mid d \rightarrow Q) = (d \rightarrow Q \mid c \rightarrow P)$

L1D $(c \rightarrow P) = (c \rightarrow Q) \equiv P = Q$

L2 If $F(X)$ is a guarded expression,

$$(Y = F(Y)) \equiv (Y = \mu X \bullet F(X))$$

L2A $\mu X \bullet F(X) = F(\mu X \bullet F(X))$

Traces

A trace will be denoted as a sequence of symbols, separated by commas and enclosed in angular brackets

$\langle x, y \rangle$ consists of two events, x followed by y.

$\langle x \rangle$ is a sequence containing only the event x.

$\langle\ \rangle$ is the empty sequence containing no events.

Operations on traces

(1) Catenation

$$s^\frown t$$

For example

$$\langle coin, choc\rangle^\frown\langle coin, toffee\rangle = \langle coin, choc, coin, toffee\rangle$$

$$\langle in1p\rangle^\frown\langle in1p\rangle = \langle in1p, in1p\rangle$$

$$\langle in1p, in1p\rangle^\frown\langle\ \rangle = \langle in1p, in1p\rangle$$

Properties:

The most important properties of catenation are that it is associative, and has $\langle\ \rangle$ as its unit

L1 $s^\frown\langle\rangle = \langle\ \rangle^\frown s = s$

L2 $s \frown (t \frown y) = (s \frown t) \frown u$

The following laws are both obvious and useful

L3 $s \frown t = s \frown u \equiv t = u$

L4 $s \frown t = u \frown t \equiv s = u$

L5 $s \frown t = \langle\ \rangle \equiv s = \langle\ \rangle \wedge t = \langle\ \rangle$

(2) Restriction

The expression $(t \upharpoonright A)$ denotes the trace t when restricted to symbols in the set A; it is formed from t simply by omitting all symbols outside A.

For example,

$$\langle around, up, down, around \rangle \upharpoonright \{up, down\} = \langle up, down \rangle$$

(3) Head and Tail

If s is a nonempty sequence, its first sequence is denoted s0, and the result of removing the first symbol is s'.

For example, $\langle x, y, x \rangle_0 = x,$ $\langle x, y, x \rangle' = \langle y, x \rangle$

(4) Star

The set A^* is the set of all finite traces (including $\langle \rangle$) which are formed from symbols in the set A.

(5) Ordering

If s is a copy of an initial subsequence of t , it is possible to find some extension u of s such that $s^\frown u = t$. We therefore define an ordering relation

$$s \preceq t = (\exists u \bullet s^\frown u = t)$$

and say that s is a prefix of t.

(6) Length

The length of the trace t is denoted $\#t$. For example $\#\langle x, y, x\rangle = 3$

Traces of a process

$traces(P)$——the complete set of all possible traces of a process P can be known in advance

(1) $traces(STOP) = \{\langle\rangle\}$

(2) $traces(coin \rightarrow STOP) = \{\langle\rangle, \langle coin\rangle\}$

(3) $traces(\mu X \bullet tick \rightarrow X) = \{\langle\rangle, \langle tick\rangle, \langle tick, tick\rangle, ...\} = tick^*$

(4) $traces(\mu X \bullet coin \rightarrow choc \rightarrow X) = \{s \mid \exists n \bullet s \preceq \langle coin, choc\rangle^n\}$

Calculating the Traces for a Process

L1 $traces(STOP) = \{t \mid t = \langle\rangle\} = \{\langle\rangle\}$

L2 $traces(c \rightarrow P) = \{t \mid t = \langle\rangle \ \vee \ (t_0 = c \wedge t' \in traces(P))\}$

$$= \{\langle\rangle\} \cup \{\langle c\rangle ^\frown t \mid t \in traces(P)\}$$

L3 $traces(c \rightarrow P \mid d \rightarrow Q) =$

$$\{t \mid t = \langle\rangle \ \vee \ (t_0 = c \wedge t' \in traces(P)) \ \vee \ (t_0 = d \wedge t' \in traces(Q))\}$$

L4

$$traces(x : B \rightarrow P(x)) = \{t \mid t = \langle\rangle \ \vee \ (t_0 \in B \wedge t' \in traces(P(t_0)))\}$$

Consider a recursive defined process:

$$X = F(X)$$

Let $$F^0(X) = X$$

$$F^{n+1} = F(F^n(X)) = F^n(F(X))$$

Then, provided that F is guarded, we can define

L5 $traces(\mu X : A \bullet F(X)) = \bigcup_{n \geq 0} traces(F^n(STOP_A))$

Example 1 Recall that RUN_A was defined in 1.1.3 X8 as $\mu X : A \bullet F(X)$ where $F(X) = (x : A \to X)$

We wish to prove that $traces(RUN_A) = A^*$

Proof: $A^* = \bigcup_{n \geq 0} \{s | s \in A^* \wedge \#s \leq n\}$

This is done by induction on n.

1.
$$\begin{aligned}
& trace(STOP_A) \\
= \ & \{\langle\rangle\} \\
= \ & \{s \mid s \in A^* \wedge \#s \leq 0\}
\end{aligned}$$

应该为t'

2.
$$\begin{aligned}
& traces(F^{n+1}(STOP_A)) & \\
= \ & traces(x : A \to F^n(STOP_A)) & \{\text{def. } F, F^{n+1}\} \\
= \ & \{t | t = \langle\rangle \ \vee \ (t_0 \in A \wedge \underline{t' \in traces(F^n(STOP_A))})\} & \{\text{L4}\} \\
= \ & \{t | t = \langle\rangle \ \vee \ (t_0 \in A \wedge \underline{(t' \in A^* \wedge \#t \leq n)})\} & \{\text{ind. hyp.}\} \\
= \ & \{t | (t = \langle\rangle \ \vee \ (\underline{t_0 \in A \wedge t' \in A^*})) \wedge \#t \leq n + 1\} & \{\text{property of } \#\} \\
= \ & \{t | t \in A^* \wedge \#t \leq n + 1\} & \{1.6.4 \text{ L4}\}
\end{aligned}$$

For the example in section 1.5 (X4), we want to prove:

$$traces(VMS) = \bigcup_{n \geq 0} \{s \mid s \preceq \langle coin, choc \rangle^n\}$$

L6 $\langle\rangle \in traces(P)$

L7 $s ^\frown t \in traces(P) \Rightarrow s \in traces(P)$

L8 $traces(P) \subseteq (\alpha P)^*$

After Operator

If $s \in traces(P)$ then P/s (P after s) is a process which behaves the same as P behaves from the time after P has engaged in all the actions recorded in the trace s. If s is not a trace of P, P/s is not defined.

Example

(1) $(VMS/\langle coin\rangle) = (choc \rightarrow VMS)$

(2) $(VMS/\langle coin, choc\rangle) = VMS$

(3) $(VMC/\langle in1p^3) = STOP$

Law

L1 $P/\langle\rangle = P$

L2 $P/(s^\frown t) = (P/s)/t$

L3 $(x : B \rightarrow P(x)/\langle c\rangle = P(c)),$ provided that $c \in B$

L3A $(c \rightarrow P)/\langle c\rangle = P$

L4 $\mathit{traces}(P/s) = \{t \mid s^\frown t \in \mathit{traces}(P)\},$
provided that $s \in \mathit{traces}(P)$

Question: How to express the properties?

(1) A process P never stops?

$$\forall s : \mathit{traces}(P) \bullet P/s \neq \mathit{STOP}$$

循环性、周期性

(2) A process is *cyclicity.*

$$\forall s : \mathit{traces}(P) \bullet \exists t \bullet (P/(s^\frown t) = P)$$

Remark: STOP is trivial cyclic.

Warning:

The use of / in a recursively defined process has the unfortunate consequence of invalidating its guards, thereby introducing the danger of multiple solutions to the recursive equations.

For example $X = (a \rightarrow (X/\langle a \rangle))$ is not guarded, and has as its solution any process of the form $a \rightarrow P$ for any P.

Proof: $(a \rightarrow ((a \rightarrow P)/\langle a \rangle)) = (a \rightarrow P)$ by **L3A**.

For this reason, we will never use the / operator in recursive process definitions.

Specification: Introduction

We will use the special variable tr to stand for an arbitrary trace of the process being specified.

Examples

(1) The owner of a vending machine does not wish to make a loss by installing it. He therefore specifies that the number of chocolates dispensed must never exceed the number of coins inserted.

$$NOLOSS = (\underline{\#(tr \upharpoonright \{choc\}}) \leq \underline{\#(tr \upharpoonright \{coin\}})$$

Abbreviation: $tr \downarrow c = \#(tr \upharpoonright \{c\})$

(2) The customer of a vending machine wants to ensure that it will not absorb further coins until it has dispensed the chocolate already paid for

$$FAIR1 = ((tr \downarrow coin) \preceq (tr \downarrow choc) + 1)$$

(2) The manufacturer of a simple vending machine must meet the requirements both of its owner and its customer

$$VMSPEC = NOLOSS \wedge FAIR1$$
$$= (0 \leq ((tr \downarrow coin) - (tr \downarrow choc)) \leq 1)$$

(3) The specification of a correction to the complex vending machine forbids it to accept three pennies in a row

$$VMCFIX = (\neg \langle in1p \rangle^3 \ \ \mathbf{in} \ \ tr)$$

Specification: Satisfaction

If P is a product which meets a specification S, we say that P satisfies S, abbreviated to $P\mathbf{sat}S$.

L1 $\quad P \;\mathbf{sat}\; \mathit{true}$

L2A
$$\begin{array}{rlll} \text{If} & P & \mathbf{sat} & S \\ \text{and} & P & \mathbf{sat} & T \\ \text{then} & P & \mathbf{sat} & (S \wedge T) \end{array}$$

L2
$$\begin{array}{rl} \text{If} & \forall n \cdot (P \;\mathbf{sat}\; S(n)) \\ \text{then} & P \;\mathbf{sat}\; (\forall n \cdot S(n)) \end{array}$$
provided that P does not depend on n

L3	If	P sat S
	and	$S \Rightarrow T$
	then	P sat T

Specification: Proofs

L4A $STOP$ sat $(tr =<>)$

L4B If P sat $S(tr)$

then $(c \to P)$ sat $(tr = \langle\rangle \vee (tr_0 = c \wedge S(tr')))$

L4C If P sat $S(tr)$

then $(c \to d \to P)$ sat $(tr \leq \langle c, d\rangle \vee (tr \geq \langle c, d\rangle \wedge S(tr'')))$

L4D If P sat $S(tr)$

and Q sat $T(tr)$

then $(c \to P | d \to Q)$ sat

$(tr = \langle\rangle \vee (tr_0 = c \wedge S(tr')) \vee (tr_0 = d \wedge T(tr')))$

L4 If $\forall x : B \cdot (P(x) \text{ sat } S(tr, x))$

then $(x : B \rightarrow P(x)) \text{ sat } (tr = \langle\rangle) \vee (tr_0 \in B \wedge S(tr', tr_0))$

L5 If $P \text{ sat } S(tr)$

and $s \in traces(P)$

then $P/s \text{ sat } S(s^\frown tr)$

L6 If $F(x)$ *is guarded*

and $STOP \text{ sat } S$

and $((X \text{ sat } S) \rightarrow (F(X) \text{ sat } S))$

then $(\mu X \cdot F(X)) \text{ sat } S$

Example We want to prove (1.1.2 X2, 1.10 X3) that

$$VMS \ \textbf{sat} \ VMSPEC$$

Proof:

1. $STOP$

   $\mathsf{sat}\ tr = \langle\rangle$ [L4A]

   $\Rightarrow\ 0 \leq (tr \downarrow coin - tr \downarrow choc) \leq 1$ [since$(\langle\rangle \downarrow coin) = (\langle\rangle \downarrow choc) = 0$]

2. Assume X sat $(0 \leq ((tr \downarrow coin) - (tr \downarrow choc)) \leq 1)$,then

   $(coin \to choc \to X)$

   $\mathsf{sat}\ (tr \leq \langle coin, choc\rangle)\vee$ [L4C]

   $(tr \geq \langle coin, choc\rangle \wedge$

   $0 \leq ((tr'' \downarrow coin) - (tr'' \downarrow choc)) \leq 1)$

   $\Rightarrow\ 0 \leq ((tr \downarrow coin - tr \downarrow choc)) \leq 1$

since $\langle\rangle \downarrow coin = \langle\rangle \downarrow choc = \langle coin\rangle \downarrow choc = 0$

and $\langle coin\rangle \downarrow coin = (\langle coin, choc\rangle \downarrow coin) = \langle coin, choc\rangle \downarrow choc = 1$

and $tr \geq \langle coin, choc\rangle \Rightarrow$

$$(tr \downarrow coin = tr'' \downarrow coin + 1 \wedge tr \downarrow choc = tr'' \downarrow choc + 1)$$