

University of Wollongong

Research Online

Department of Computing Science Working
Paper Series

Faculty of Engineering and Information
Sciences

1983

CSP at WSS

C. A. R. Hoare
University of Wollongong

Follow this and additional works at: <https://ro.uow.edu.au/compsciwp>

Recommended Citation

Hoare, C. A. R., CSP at WSS, Department of Computing Science, University of Wollongong, Working Paper 83-3, 1983, 25p.
<https://ro.uow.edu.au/compsciwp/71>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

THE UNIVERSITY OF WOLLONGONG
DEPARTMENT OF COMPUTING SCIENCE

CSP at WSS

C.A.R. Hoare

Programming Research Group
Oxford University
45 Banbury Road
Oxford OX2 6PE

Abstract

This preprint contains a copy of the visuals on Communicating Sequential Processes which were presented at the First Wollongong Summer School on the Science of Programming at Sponar's Chalet, January 31 - February 9, 1983.

Preprint No 83-3

March 25, 1983

P.O. Box 1144, WOLLONGONG, N.S.W. AUSTRALIA
telephone (042)-282-981
telex AA29022

o)

COMMUNICATING SEQUENTIAL PROCESSES

C.A.R. HOARE

SPONAR'S CHALET

JAN - FEB 1983

o

A symbol denotes a class of event in which a process may engage

coin insertion of coin in vending machine VMS

choc extraction of choc from VMS

The alphabet of a process is the set of events in which we are interested, e.g.,

$$\alpha VMS = \{coin, choc\}$$

(but not empty coin box)

2

RECURSION

3

1. $STOP_A$ never does anything.

$$\alpha STOP_A = A$$

2. If $x \in \alpha P$

$(x \rightarrow P)$ first does x

and then behaves like P

$$\alpha(x \rightarrow P) = \alpha P$$

$(coin \rightarrow STOP_{\alpha VMS})$ accepts a coin before stopping

$(coin \rightarrow choc \rightarrow coin \rightarrow choc \rightarrow STOP)$
serves two customers

A clock does nothing but tick forever

$$\alpha CLOCK = \{tick\}$$

Let X be the "unknown" behaviour

$$\therefore X = (tick \rightarrow X)$$

CLOCK is defined as the only solution

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

compare $ROOT2 = \mu(X: X \in TR: X^2 + X - 2)$

WARNING. RHS of equation must be guarded: i.e., begin with $\alpha \rightarrow \dots$

$X = X$ has many solutions

$\therefore \mu X: X$ is NOT ALLOWED.

1)

CLOCK = (tick → CLOCK)
 = (tick → (tick → CLOCK))
 = ...

A simple vending machine

VMS = coin → choc → VMS

A change giving machine

α CH5 = {in5p, out2p, out1p}

CH5 = in5p → out2p → out1p → out2p → CH5

4

CHOICE

5

Let $\alpha P = \alpha Q$

$\{x, y\} \subseteq \alpha P \wedge x \neq y$

Then $(x \rightarrow P \mid y \rightarrow Q)$

either does x and then behaves as P

or does y and then behaves as Q

The choice is made by some other process

A machine that offers a choice of goods

VMCT = $\mu X. \text{coin} \rightarrow (\text{choc} \rightarrow X \mid \text{toffee} \rightarrow X)$

6

GENERAL CHOICE

7.

A more complicated vending machine

α VMC = {in1p, in2p, large, small, out1p}

VMC = (in2p → (large → VMC
 | small → out1p → VMC)
 | in1p → (small → VMC
 | in1p → (large → VMC
 | in1p → STOP)
)
)

Let A be a set of events

Let P_x be a process with alphabet B

(For each x in A)

IF $A \subseteq B$

$(x: A \rightarrow P_x)$ First does some x in A
 then behaves as P_x

A process which always does anything desired

$\alpha \text{RUN}_A = A$

$\text{RUN}_A = (x: A \rightarrow \text{RUN}_A)$

WARNING: DO NOT INSERT THREE
 COINS.

2)

All processes (so far) can be expressed by general choice

$$STOP = (x: \{\} \rightarrow P x) = (y: \{\} \rightarrow Q y)$$

$$(b \rightarrow P) = (x: \{b\} \rightarrow P)$$

$$(b \rightarrow P \mid c \rightarrow Q) = (x: \{b, c\} \rightarrow \text{if } x = b \text{ then } P \text{ else } Q)$$

$$\mu X. (x: A \rightarrow P(x, X)) = (x: A \rightarrow P(x, \mu X. (x: A \rightarrow P(x, X))))$$

MUTUAL RECURSION

P and Q may be defined as the unique processes satisfying the simultaneous equations:

$$P = \text{coin} \rightarrow Q$$

$$Q = \text{choc} \rightarrow P$$

Theorem. $P = VMS$

Proof $P = \text{coin} \rightarrow (\text{choc} \rightarrow P)$ subst

VMS is the only solution of this equation

AN INFINITE SET OF EQUATIONS

$$\alpha CT_n = \{\text{up, down, around}\} \text{ for } n \geq 0$$

CT_n is the behaviour at height n .

$$CT_0 = (\text{around} \rightarrow CT_0 \mid \text{up} \rightarrow CT_1)$$

$$CT_{n+1} = (\text{down} \rightarrow CT_n \mid \text{up} \rightarrow CT_{n+2})$$

$$ROCKET = CT_0$$

PICTURES

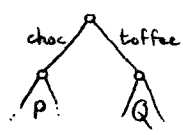
STOP



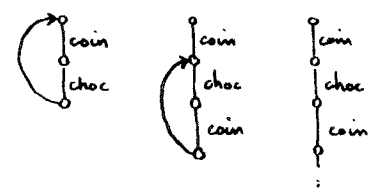
$\text{coin} \rightarrow \text{STOP}$



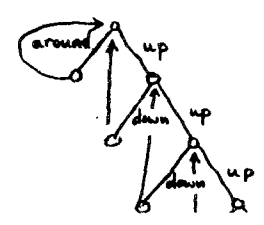
$(\text{choc} \rightarrow P \mid \text{toffee} \rightarrow Q)$



VMS



CT_0



3) LAW

12

$$(x: A \rightarrow P x) = (y: B \rightarrow Q y)$$

$$\equiv A = B \wedge \underline{A} (x: x \in A: P x = Q x)$$

Corollaries

$$\text{STOP} \neq (d \rightarrow P)$$

$$(c \rightarrow P) \neq (d \rightarrow Q) \quad \text{if } c \neq d$$

$$(c \rightarrow P | d \rightarrow Q) = (d \rightarrow Q | c \rightarrow P)$$

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

LAW

13

Let $F(Y)$ be guarded

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

Corollary

$$F(\mu X. F(X)) = \mu X. F(X)$$

recall $VMS = \mu X. \text{coin} \rightarrow \text{choc} \rightarrow X$

let $VMT = \mu X. \text{choc} \rightarrow \text{coin} \rightarrow X$

Theorem.

$$VMS = \text{coin} \rightarrow VMT$$

Proof

IMPLEMENTATION

14

(in LISP!)

$(x: A \rightarrow P x)$ is implemented by

$\lambda x. \text{if ismember}(x, A) \text{ then } P(x) \text{ else "BLEEP"}$

$\text{STOP} \approx \lambda x \text{ "BLEEP"}$

$(b \rightarrow P) \approx \text{prefix}("b, P)$

$\text{prefix}(c, P) = \lambda x. \text{if } x = c \text{ then } P \text{ else "BLEEP"}$

$VMS \approx \text{LABEL } X, \text{prefix}("coin, prefix("choc, X))$

$(b \rightarrow P | c \rightarrow Q) \approx \text{choice2}("b, P; c, Q)$

$\text{ROCKET} \equiv \text{LABEL } CT, \lambda n$

$\text{if } n = 0 \text{ then choice2}("around, CT(0),$

$\text{"up, CT(1))$

$\text{else choice2}("down, CT(n-1),$

$\text{"up, CT(n+1)) (0)$

TRACES

2.1

A trace of a process is a sequence of events recording its behaviour up to some moment.

$\langle \text{coin}, \text{choc}, \text{coin} \rangle$ is a trace of VMS

$\langle \text{coin} \rangle$ is also a trace of VMS

$\langle \rangle$ is a trace of every process

$\langle \text{coin}, \text{choc} \rangle, \langle \text{coin}, \text{toffee} \rangle$

are traces of VMT

$\langle \text{in1}, \text{in1}, \text{in1} \rangle$ is a trace of VMC

There is no x such that

$\langle \text{in1}, \text{in1}, \text{in1}, x \rangle$ is a trace of VMC

4) CATENATION

2.2

$s^{\wedge}t$ is the result of writing t after s
 $\langle \text{coin, choc} \rangle^{\wedge} \langle \text{coin, toffee} \rangle = \langle \text{coin, choc, coin, toffee} \rangle$

$$s^{\wedge} \langle \rangle = \langle \rangle^{\wedge} s = s$$

$$s^{\wedge}(t^{\wedge}u) = (s^{\wedge}t)^{\wedge}u$$

$$s^{\wedge}t = s^{\wedge}u \Rightarrow t = u, \quad t^{\wedge}s = u^{\wedge}s \Rightarrow t = u$$

$$s^{\wedge}t = \langle \rangle \Rightarrow s = t = \langle \rangle$$

POWER

$$t^0 = \langle \rangle$$

$$t^{n+1} = t^{\wedge}t^n = t^n \wedge t$$

$$(s^{\wedge}t)^{n+1} = s^{\wedge}(t^{\wedge}s)^n \wedge t$$

Exercise: prove this last law from the preceding ones.

Proofs.

2.4

To prove $A_s :: P(s)$

(1) prove $P(\langle \rangle)$

(2) prove $A_{\langle x \rangle} :: P(\langle x \rangle)$

(3) prove $A_{\langle s, t \rangle} :: P_s \wedge P_t \Rightarrow P(s^{\wedge}t)$

Example

To prove $A_s :: (s \upharpoonright A) \upharpoonright B = s \upharpoonright (A \cap B)$

$$(1) \langle \rangle \upharpoonright A \upharpoonright B = \langle \rangle \upharpoonright B = \langle \rangle = \langle \rangle \upharpoonright (A \cap B)$$

$$(2) \langle x \rangle \upharpoonright A \upharpoonright B = \{ \text{case 1: } x \in A \cap B \}$$

$$= \langle x \rangle \upharpoonright B = \langle x \rangle = \langle x \rangle \upharpoonright (A \cap B)$$

$$" = \{ \text{case 2: } x \in A - B \}$$

$$= \langle x \rangle \upharpoonright B = \langle \rangle = \langle x \rangle \upharpoonright (A \cap B)$$

$$" = \{ \text{case 3: } x \notin A \}$$

$$= \langle \rangle \upharpoonright B = \langle \rangle = \langle x \rangle \upharpoonright (A \cap B)$$

$$(3) ((s^{\wedge}t) \upharpoonright A) \upharpoonright B = ((s \upharpoonright A)^{\wedge} (t \upharpoonright A)) \upharpoonright B$$

$$= ((s \upharpoonright A) \upharpoonright B)^{\wedge} ((t \upharpoonright A) \upharpoonright B)$$

$$\{ \text{ind. hyp} \}$$

$$= s \upharpoonright (A \cap B)^{\wedge} t \upharpoonright (A \cap B)$$

$$= (s^{\wedge}t) \upharpoonright (A \cap B)$$

RESTRICTION

2.3

$t \upharpoonright A$ is formed from t by omitting symbols outside A

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

$$\langle \rangle \upharpoonright A = \langle \rangle$$

strict

$$(s^{\wedge}t) \upharpoonright A = (s \upharpoonright A)^{\wedge} (t \upharpoonright A)$$

distributive

$$\langle x \rangle \upharpoonright A = \langle x \rangle \quad \text{if } x \in A$$

$$= \langle \rangle \quad \text{if } x \notin A$$

$$\upharpoonright \{ \} = \langle \rangle$$

$$(s \upharpoonright A) \upharpoonright B = s \upharpoonright (A \cap B) = (s \upharpoonright B) \upharpoonright A$$

NOTE: a strict distributive function is defined by its effect on singleton sequences.

CHANGE OF SYMBOL

2.5

Let F be a function from symbols to symbols. - total
 $F^*(s)$ is result of applying F to each symbol of s

$$\text{double}^*(\langle 1, 5, 3, 1 \rangle) = \langle 2, 10, 6, 2 \rangle$$

LAWS

$$F^*(\langle \rangle) = \langle \rangle$$

$$F^*(\langle x \rangle) = \langle F(x) \rangle$$

$$F^*(s^{\wedge}t) = F^*(s)^{\wedge} F^*(t)$$

$$* F^*(s \upharpoonright A) = F^*(s) \upharpoonright F(A)$$

Exercise: (1) give a counter-example.
 (2) state a property of F which makes the law valid
 (3) prove it! from the previous laws.

5)

MISCELLANEOUS

2.6

A^* is the set of all finite traces
with no symbols outside A

$$A^* = \{s \mid s \upharpoonright A = s\}$$

IF $s \neq \langle \rangle$ then

s_0 is its first element

s' is result of removing first element

$$\langle \langle x \rangle \hat{s} \rangle_0 = x, \langle \langle x \rangle \hat{s} \rangle' = s$$

$$s \neq \langle \rangle \Rightarrow s = \langle s_0 \rangle \hat{s}'$$

$\#s$ is the length of s

$s.b$ is the number of occurrences
of b in s

$$s.b = \#(s \upharpoonright \{b\})$$

IMPLEMENTATION

2.8

$\langle \rangle$	NIL
$\langle x \rangle$	cons(x, NIL)
$\langle \langle x \rangle \hat{s} \rangle$	cons(x, s)
$s \hat{t}$	append(s, t)
s_0	car(s)
s'	cdr(s)

THM. $\langle \rangle \hat{t} = t \wedge \langle \langle x \rangle \hat{s} \rangle \hat{t} = \langle x \rangle \hat{(s \hat{t})}$

\therefore append(s, t) = if $s = \langle \rangle$ then t
else cons(car(s), append(cdr(s), t))

Let F be a distributive function.

$$F(s) = \text{if } s = \langle \rangle \text{ then } F(\langle \rangle) \\ \text{else } F(\langle s_0 \rangle) \hat{F}(s')$$

$$F(s) = \text{if } s = \text{NIL} \text{ then } F(\langle \rangle) \\ \text{else } \text{append}(F(\text{car}(s)), F(\text{cdr}(s)))$$

REVERSAL

2.7

\bar{s} is s written backwards.

$$\overline{\langle a, b, c \rangle} = \langle c, b, a \rangle$$

$$\text{last}(s) = \bar{s}_0$$

$$\text{trunc}(s) = \overline{\bar{s}'}$$

Exercise.

Give 10 interesting laws
governing reversal.

EXAMPLE

2.9

$$\langle \rangle \upharpoonright A = \langle \rangle$$

$$\langle s \rangle \upharpoonright A = \text{if } s_0 \in A \text{ then } \langle s \rangle \text{ else } \langle \rangle$$

$$\therefore \langle \langle s_0 \rangle \upharpoonright A \rangle \upharpoonright A = s \upharpoonright A$$

$$= \text{if } s_0 \in A \text{ then } \langle s \rangle \upharpoonright A \text{ else } \langle \rangle \upharpoonright A$$

$$\therefore \text{restrict}(s, A) = \text{if } s = \text{NIL} \text{ then } \text{NIL}$$

$$\text{else if } \text{ismember}(\text{car}(s), A)$$

$$\text{then } \text{cons}(\text{car}(s), \text{restrict}(\text{cdr}(s), A))$$

$$\text{else } \text{restrict}(\text{cdr}(s), A)$$

EXERCISE

(1) Implement $F^*(s)$ as the LISP function
star(F, s); state explicitly all theorems
used.

(2) Implement \bar{s} as reverse(s), (together
with the relevant theorems)

$s \leq t$ means t begins with a copy of s

$$s \leq t \equiv \exists (u :: s \hat{\wedge} u = t)$$

\leq is a partial order, i.e.:

$$\langle \rangle \leq s$$

$$s \leq s$$

reflexive

$$s \leq t \wedge t \leq s \Rightarrow s = t$$

antisymmetry

$$s \leq t \wedge t \leq u \Rightarrow s \leq u$$

transitive

$$s \text{ in } t \equiv \exists (u, v :: u \hat{\wedge} s \hat{\wedge} v = t)$$

in is a partial order.

$$s \leq^n t \equiv s \leq t \ \& \ \#t \leq \#s + n$$

$$s \leq^0 t \equiv s = t$$

$$s \leq^{nm} t \equiv \exists u \ s \leq^n u \wedge u \leq^m t$$

$$\text{i.e., } \leq^n; \leq^m = \leq^{nm}$$

2.12

Let f be SM, with

range(f) infinite

\bar{f} is the SM function s.t.

$$\text{range}(\bar{f}) = \overline{\text{range}(f)}$$

$$\bullet \ \bar{\bar{f}} = f$$

s interleaves $(t, u) = s \sim (t, u)$

$$EF: f \text{ is SM: } t = \text{sof} \wedge u = \text{so}\bar{f}$$

$$s \sim (t, u) \equiv s \sim (u, t)$$

$$s \sim (t, u) \wedge t \sim (v, w)$$

$$\Rightarrow \exists x \ x \sim (u, w) \wedge s \sim (v, x)$$

$$s[0] = s_0 \quad \text{if } 0 < \#s$$

$$s[i+1] = s'[i]$$

Let f, g be a total Functions from \mathbb{N} to \mathbb{N}

$$(\text{sof})[i] = s[f(i)]$$

$$(\text{sof}) \circ g = \text{so}(f \circ g)$$

f is strictly monotonic (SM) if

$$\underline{A}(i, j: 0 \leq i < j: f(i) < f(j))$$

$$s \text{ sub } t \equiv \exists (f: f \text{ is SM: } s = t \circ f)$$

sub is a partial order

4 A drinks dispenser DD

$$\alpha DD = \{\text{setgin, setwhisky, gin, whisky, coin}\}$$

DD sells gin if the most recent setting was setgin, or whisky if setwhisky

5. VMS2 behaves like VMS, except that it accepts up to two coins before dispensing up to two choes

6. Draw pictures of your answers to 3, 4 and 5

$$7. \text{VMS} \triangleq \mu X. \text{coin} \rightarrow \text{choc} \rightarrow X$$

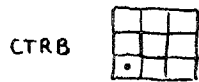
$$\text{VMS}' \triangleq \mu X. \text{choc} \rightarrow \text{coin} \rightarrow X$$

Prove that

$$\text{VMS} = \overset{\text{coin}}{\text{choc}} \rightarrow \text{VMS}'$$

7) EXERCISES

1. $\mu\text{CTR} = \{\text{up, right}\}$



2. $\mu\text{VMD} = \{\text{in1p, in2p, choc}\}$
 choc costs 3p. Any combination and order of coins μ must be accepted.

3. $\mu\text{CTRD} = \{\text{up, down, left, right}\}$
 moves on same board as CTRA

RECURSION AGAIN.

A.2

To find the solution of the equation

$$P = F(P)$$

where F is guarded, and uses only approved notations.

Define $F^0(P) = P$

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

$$= F(F(\dots F(P)\dots))$$

The required answer is

$$\underline{A}(n: n \geq 0: F^n(\text{true}))$$

μ i.e., continuous

ASSERTIONS

A.1

Let's identify a process with a predicate $P(\text{tr})$ which describes all its possible traces tr

$$\text{STOP} = \text{tr} = \langle \rangle$$

$$(\text{coin} \rightarrow \text{STOP}) = \text{tr} = \langle \rangle \vee \text{tr} = \langle \text{coin} \rangle$$

$$\text{CLOCK} = \text{tr} \uparrow \{\text{tick}\} = \text{tr}$$

$$\text{VMS} = \underline{E}n: n \geq 0: \text{tr} \leq \langle \text{coin}, \text{choc} \rangle^n$$

In general

$$(\alpha: A \rightarrow P) = \text{tr} = \langle \rangle$$

$$\vee (tr_0 \in A \wedge P(tr_0) [tr'/tr])$$

$\mu X. F(X)$ = the unique predicate P s.t.
 $P = F(P)$

$$\text{STOP} = \text{tr} = \langle \rangle$$

$$(C \rightarrow P) = \text{tr} = \langle \rangle \vee (tr_0 = C \wedge P [tr'/tr])$$

A.3

$$X \equiv \text{coin} \rightarrow \text{choc} \rightarrow X$$

$$X \equiv \text{tr} = \langle \rangle \vee (tr_0 = \text{coin} \wedge (\text{choc} \rightarrow X) [tr'/tr])$$

$$\equiv \text{tr} = \langle \rangle \vee$$

$$(tr_0 = \text{coin} \wedge (tr' = \langle \rangle \vee$$

$$\vee (tr'_0 = \text{choc} \wedge X [tr''/tr])))$$

$$\equiv \text{tr} = \langle \rangle \vee \text{tr} = \langle \text{coin} \rangle$$

$$\vee \langle \text{coin}, \text{choc} \rangle \leq \text{tr} \wedge X [tr''/tr]$$

$$P_0 \equiv \text{true}$$

$$P_1 \equiv \text{tr} \leq \langle \text{coin}, \text{choc} \rangle \vee \langle \text{coin}, \text{choc} \rangle \leq \text{tr}$$

$$P_2 \equiv \text{tr} \leq \langle \text{coin} \rangle \vee (\langle \text{coin}, \text{choc} \rangle \leq \text{tr}$$

$$\wedge (tr'' \leq \langle \text{coin}, \text{choc} \rangle \vee \langle \text{coin}, \text{choc} \rangle \leq tr''))$$

$$\equiv \text{tr} \leq \langle \text{coin}, \text{choc} \rangle^2 \vee \langle \text{coin}, \text{choc} \rangle^2 \leq \text{tr}$$

$$P_n \equiv \text{tr} \leq \langle \text{coin}, \text{choc} \rangle^n \vee \langle \text{coin}, \text{choc} \rangle^n \leq \text{tr}$$

$$\therefore \mu X. \text{coin} \rightarrow \text{choc} \rightarrow X$$

$$= \underline{A}(n: n \geq 0: \text{tr} \leq \langle \text{coin}, \text{choc} \rangle^n \vee \langle \text{coin}, \text{choc} \rangle^n \leq \text{tr})$$

=

8) CONTINUITY

A4.

A5

A chain of predicates satisfies

$$[A(n: n \geq 0: P_{n+1} \Rightarrow P_n)]$$

A predicate transformer F is continuous if

$$F(\underline{A}(n: n \geq 0: P_n)) = \underline{A}(n: n \geq 0: F(P_n))$$

for all chains

Lemma. IF F and G are continuous,

so is $F \circ G$

Lemma. IF F is continuous,

$F^n(\text{true})$ is a chain.

IF F is continuous

$$\text{Let } X = \underline{A}(n: n \geq 0: F^n(\text{true}))$$

Then $F(X) = X$

Proof. LHS =

$$\underline{A}(n: n \geq 0: F(F^n(\text{true})))$$

$$= \text{true} \wedge \underline{A}(n: n \geq 1: F^n(\text{true}))$$

$$= \text{RHS.}$$

SPECIFICATIONS.

A.6.

ALWAYS.

A.7

A process is specified by giving all desired properties of its traces.

"It mustn't accept more than 70 coins"

$$\text{VMC70} = \# \text{tr} \{i_1p, i_2p\} \leq 70$$

"It mustn't accept three consecutive pennies"

$$\text{VMCBAR3} = \neg \langle i_1p \rangle^3 \text{ in tr}$$

"An improved VMC."

$$\text{VMCOK} = \text{VMC}$$

$$\wedge \text{VMCBAR3} \wedge \text{VMC70}$$

Let P be a predicate intended to be true of all traces tr of a process. \therefore it must be satisfied by all prefixes of tr too. So we define:

$$\square P = \underline{A}(s: s \leq tr: P[s/tr])$$

Provided that $P[\epsilon/tr]$

$\square P$ uniquely defines a process, with exactly those traces which satisfy it.

9) Examples.

A.8

LAWS

A.9

"It mustn't lose money"
 NOLOSS = $\square(\text{tr. choc} \leq \text{tr. coin})$

"It mustn't cheat the client"
 FAIR = $\square(\text{tr. coin} \leq \text{tr. choc} + 1)$

"It must please owner and client"
 NOLOSS \wedge FAIR

This specifies VMS

A specification of CT_n:

$\square(\text{tr. down} \leq \text{tr. up} + n$
 $\wedge \text{last}(\text{tr}) = \text{around} \Rightarrow \text{tr. down} = \text{tr. up} + n)$

$$\square P \Rightarrow P$$

$$\square \square P \equiv \square P$$

$$\square(P \wedge Q) \equiv (\square P) \wedge (\square Q)$$

$$\square \underline{A}(n :: P_n) \equiv \underline{A}(n :: \square P_n)$$

tr \Leftarrow $\langle \rangle$ \vee

$$(\square P)[\text{tr}'/\text{tr}] \equiv \square(P[\text{tr}'/\text{tr}])$$

if F is healthy

$$(\square P)[f(\text{tr})/\text{tr}] = \square(P[f(\text{tr})/\text{tr}])$$

A.10

AFTER

A.11

Prove VMS = $\square 0 \leq \text{tr. coin} - \text{tr. choc} < 2$

Proof. coin \rightarrow choc \rightarrow VMS

$$= (\text{tr} \leq \text{coins})$$

$$\vee (\langle \text{coin}, \text{choc} \rangle \leq \text{tr}$$

$$\wedge \square 0 \leq \text{tr}'' \text{coin} - \text{tr}'' \text{choc} < 2))$$

$$= \text{VMS}$$

If s is a trace of P

P/s describes the behaviour of
 P after it has engaged in s

$$\text{VMC} / \langle \text{in} 2p, \text{small} \rangle = \text{out} 1p \rightarrow \text{VMC}$$

$$\text{VMC} / \langle \text{in} 1p \rangle = \text{STOP}$$

LAWS

$$P / \langle \rangle = P$$

$$P / (s^{\wedge} t) = (P/s) / t$$

$$(x:A \rightarrow P_x) / \langle y \rangle = P_y \quad \text{if } y \bullet A$$

ASSERTION

Provided $P[s/\text{tr}]$

$$P/s = P[s^{\wedge} \text{tr} / \text{tr}]$$

$$(\square P)/s \stackrel{!}{=} \square(P/s)$$

Let P be specification of a mechanism driven by "menu-select" & keyboard.

αP contains a symbol for each key except backspace

We wish to transform P to a spec $\text{backable}(P)$

with alphabet $\alpha P \cup \{\text{backspace}\}$

which specifies a mechanism which behaves as P except that

backspace cancels the effect of the most recent uncanceled symbol in αP

$$\begin{aligned} \text{backable}(P) &= \square(P[\text{clean}(tr)/tr]) \\ \text{clean}(s) &= s \text{ if } s \notin \{\text{backspace}\}^* \\ \text{clean}(s^{\langle x \rangle}) &= (\text{clean } s)^{\langle x \rangle} \quad x \in \alpha P \\ \text{clean}(s^{\langle x \rangle} \langle \text{backspace} \rangle^{\langle t \rangle}) &= \text{clean}(s^{\langle t \rangle}) \quad \text{for } x \in \alpha P \end{aligned}$$

EXERCISES

Specify the following predicate transformers:

- (1) $\text{restart} \in \alpha P$
restartable(P) behaves like P ; but when restart is pressed, it starts from the beginning again.
- (2) $\alpha P \cap \{\text{save, restore}\} = \{\}$
saveable(P) behaves like P ; but when restore is pressed it returns to the state just before the most recent save

$\text{back}(Q, P)$ behaves like $\text{backable}(P)$ except that an extra backspace will cause it to behave like Q instead.

$$\text{back}(Q, P) = \lambda x.$$

if $x = \text{backspace}$ then Q

else if $P(x) = \text{"BLEEP"}$ then "BLEEP

else $\text{back}(\text{back}(Q, P), P(x))$

$$\text{backable}(P) = \lambda x.$$

if $P(x) = \text{"BLEEP"}$ then "BLEEP

else $\text{back}(\text{backable}(P), P(x))$

INTERSECTION

3.1

Let $\alpha P = \alpha Q = \alpha(P \wedge Q)$

$P \wedge Q$ behaves like both P and Q

Each event that occurs requires simultaneous participation of both.

$$\begin{aligned} \text{GRCUST} &= (\text{choc} \rightarrow \text{GRCUST} \\ &\quad | \text{coin} \rightarrow \text{choc} \rightarrow \text{GRCUST}) \end{aligned}$$

$$\text{GRCUST} \wedge \text{VMS} = \mu X. \text{coin} \rightarrow \text{choc} \rightarrow X$$

$$\begin{aligned} \text{FOOLC} &= (\text{in2p} \rightarrow \text{large} \rightarrow \text{FOOLC} \\ &\quad | \text{in1p} \rightarrow \text{large} \rightarrow \text{FOOLC}) \end{aligned}$$

$$\begin{aligned} \text{FOOLC} \wedge \text{VMC} &= \mu X. (\text{in2p} \rightarrow \text{large} \rightarrow X \\ &\quad | \text{in1p} \rightarrow \text{STOP}) \end{aligned}$$

11)

P || Q

$\alpha(P || Q) = \alpha P \cup \alpha Q$

If $x \in \alpha P - \alpha Q$, (P || Q) does x whenever P does

If $y \in \alpha Q - \alpha P$, (P || Q) does y whenever Q does

If $z \in \alpha P \cap \alpha Q$, (P || Q) does z

whenever both P and Q do z

$\alpha \text{NOISYVM} = \{\text{coin, clink, choc, toffee}\}$

$\text{NOISYVM} = (\text{coin} \rightarrow \text{clink} \rightarrow \text{choc} \rightarrow \text{NOISYVM})$

$\alpha \text{CUST} = \{\text{coin, startcourse, endcourse, choc, toffee}\}$

$\text{CUST} = \text{coin} \rightarrow (\text{toffee} \rightarrow \text{CUST} \mid \text{startcourse} \rightarrow \text{endcourse} \rightarrow \text{choc} \rightarrow \text{CUST})$

NOISYVM || CUST =

$\mu X. \text{coin} \rightarrow (\text{clink} \rightarrow \text{startcourse} \rightarrow \text{endcourse} \rightarrow \text{choc} \rightarrow X \mid \text{startcourse} \rightarrow \text{clink} \rightarrow \text{endcourse} \rightarrow \text{choc} \rightarrow X \mid \text{startcourse} \rightarrow \text{endcourse} \rightarrow \text{clink} \rightarrow \text{choc} \rightarrow X)$



$\alpha \text{CTR} = \{\text{up, down, left, right}\}$

Let $\alpha P = \{\text{up, down}\}$

$P = (\text{up} \rightarrow \text{down} \rightarrow P)$

$\alpha Q = \{\text{left, right}\}$

$Q = (\text{right} \rightarrow \text{left} \rightarrow Q \mid \text{left} \rightarrow \text{right} \rightarrow Q)$

$\text{CTR} = P || Q$

LAWS.

$(P || Q) / s = (P / s) \cup (Q / s)$

|| is symmetric and associative

Let $P = x : A \rightarrow P x$

$Q = y : B \rightarrow Q y$

Then $P || Q =$

$x : (A \wedge \alpha P \wedge \alpha Q \wedge B) \rightarrow$

(if $x \in A$ then $P x$ else P)

|| if $x \in B$ then $Q x$ else Q)

where $A \wedge X \wedge B = (X \wedge A \wedge B) \cup (\bar{X} \wedge A) \cup (\bar{X} \wedge B)$

ASSERTION

$P || Q = \text{tr } \alpha (\alpha P \cup \alpha Q)^*$

$\wedge P [(\text{tr } \alpha P) / \text{tr}]$

$\wedge Q [(\text{tr } \alpha Q) / \text{tr}]$

EXAMPLE

$\alpha P = \{a, c\}$ $P = a \rightarrow c \rightarrow P$

$\alpha Q = \{b, c\}$ $Q = c \rightarrow b \rightarrow Q$

1. $P || Q = a \rightarrow (c \rightarrow P || c \rightarrow b \rightarrow Q)$

$= a \rightarrow c \rightarrow (P || b \rightarrow Q)$

2. $P || b \rightarrow Q = (a \rightarrow (c \rightarrow P || b \rightarrow Q) \mid b \rightarrow (P || Q))$

$\mid b \rightarrow (P || Q)$

$= (a \rightarrow b \rightarrow c \rightarrow (P || b \rightarrow Q) \mid b \rightarrow a \rightarrow c \rightarrow (P || b \rightarrow Q))$

$\mid b \rightarrow a \rightarrow c \rightarrow (P || b \rightarrow Q)$

$= \mu X. (a \rightarrow b \rightarrow c \rightarrow X \mid b \rightarrow a \rightarrow c \rightarrow X)$

$\mid b \rightarrow a \rightarrow c \rightarrow X)$

3. $= (P || Q) / \langle a, c \rangle$

12)

3.6

Let $n \leq m \wedge n \leq m \wedge m \leq n+k$

$$P = \square \text{tr}.c \leq \text{tr}.a \quad \square 0 \leq \text{tr}.a - \text{tr}.c < 2$$

$$Q = \square \text{tr}.b \leq \text{tr}.c$$

$$P \parallel Q = \text{tr} \in \{a, b, c\}^*$$

$$\wedge \square \text{tr}\{a, c\}.c \leq \text{tr}\{a, c\}.a$$

$$\wedge \square \text{tr}\{b, c\}.b \leq \text{tr}\{b, c\}.c$$

$$= \square \text{tr}.b \leq \text{tr}.c \wedge \text{tr}.c \leq \text{tr}.a$$

$$\therefore (P \parallel Q) / \langle a, c \rangle =$$

$$= \square \langle a, c \rangle^* \text{tr}.b \leq \langle a, c \rangle^* \text{tr}.c \leq \langle a, c \rangle^* \text{tr}.a$$

$$= \square \text{tr}.b \leq \text{tr}.c + 1 \leq \text{tr}.a + 1$$

$$\alpha(P \sim Q) = \alpha P \cup \alpha Q \cup \{\text{switch}\}$$

$(P \sim Q)$ first behaves like P .

After switch it behaves like Q .

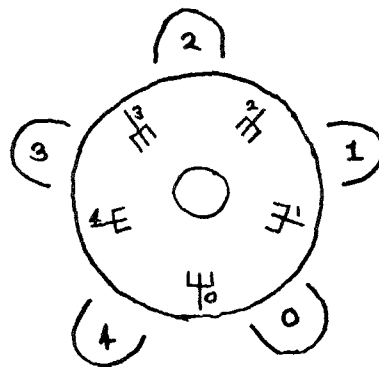
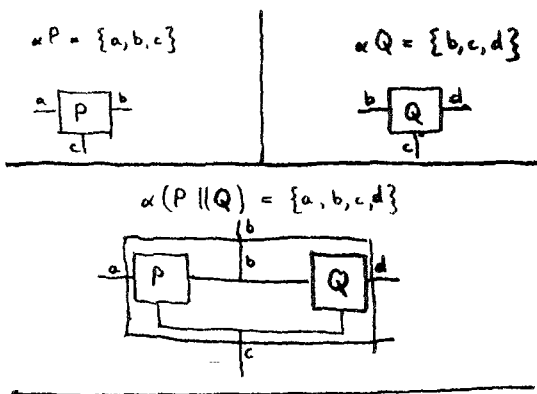
On each subsequent switch it suspends execution of the current one of $\{P, Q\}$, and resumes execution of the other at the point it was most recently suspended.

PICTURES.

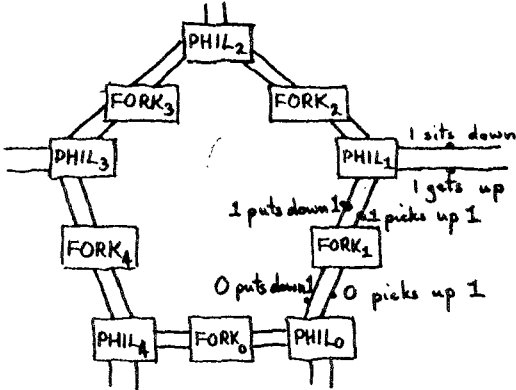
3.7

THE DINING PHILOSOPHERS.

3.8



\oplus mean arithmetic modulo 5.



$\alpha PHIL_i = \{ i \text{ sits down, } i \text{ gets up, } i \text{ picks up } i, i \text{ picks up } i \oplus 1, i \text{ puts down } i, i \text{ puts down } i \oplus 1 \}$

$\alpha FORK_i = \{ i \text{ picks up } i, i \oplus 1 \text{ picks up } i, i \text{ puts down } i, i \oplus 1 \text{ puts down } i \}$
DEADLOCK 3.11

Let $sad = \langle 0 \text{ sits down, } 1 \text{ sits down, } \dots, 4 \text{ sits down, } 0 \text{ picks up } 0, 1 \text{ picks up } 1, \dots, 4 \text{ picks up } 4 \rangle$

$COLLEGE / sad = STOP.$

SOLUTION

introduce a FOOTMAN

$\alpha FOOTMAN = \bigcup_{i=0}^4 \{ i \text{ sits down, } i \text{ gets up} \}$

$FOOTMAN = \square \text{ ups } \stackrel{\leq}{=} \text{ downs}$

where $ups = \sum_{i=0}^4 tr. i \text{ gets up}$

$downs = \sum_{i=0}^4 tr. i \text{ sits down}$

$NEW COLLEGE = COLLEGE \parallel FOOTMAN$

$PHIL_i = \mu X. i \text{ sits down} \rightarrow i \text{ picks up } i \rightarrow i \text{ picks up } i \oplus 1 \rightarrow i \text{ puts down } i \rightarrow i \text{ puts down } i \oplus 1 \rightarrow i \text{ gets up} \rightarrow X.$

$FORK_i = \mu X. (i \text{ picks up } i \rightarrow i \text{ puts down } i \rightarrow X \mid i \oplus 1 \text{ picks up } i \rightarrow i \oplus 1 \text{ puts down } i \rightarrow X)$

$PHILOS = PHIL_0 \parallel PHIL_1 \parallel \dots \parallel PHIL_4$

$FORKS = FORK_0 \parallel FORK_1 \parallel \dots \parallel FORK_4$

$COLLEGE = PHILOS \parallel FORKS$

CHANGE OF SYMBOL

Let f map αP onto αQ

$P = f^{-1}(Q)$

means P performs α exactly on those occasions that Q performs $f(\alpha)$.

$\alpha f^{-1}(Q) = f^{-1}(\alpha Q)$

where $f^{-1}(B) = \{ \alpha \mid f(\alpha) \in B \}$

To deal with inflation

$f(in10p) = in2p$

$f(\text{small}) = \text{large}$

$f(in5p) = in1p$

$f(\text{very small}) = \text{small}$

$f(out5p) = out1p$

$NEWVMC = f^{-1}(VMC)$

$g(i \text{ sits down}) = \text{sits down for } 0 \leq i < 5$

$g(i \text{ gets up}) = \text{gets up}$

$LKY = g^{-1}(\mu X. \text{ sits down} \rightarrow \text{gets up} \rightarrow X)$

14) LAWS

3.13

NONDETERMINISM

4.1

$$F^{-1}(STOP_R) = STOP_{F^{-1}(R)}$$

$$F^{-1}(x \rightarrow P) = (y: F^{-1}(\{x\}) \rightarrow F^{-1}(P))$$

$$F^{-1}(x: A \rightarrow P_x) = (x: F^{-1}(A) \rightarrow (F^{-1} \circ P \circ F) x)$$

$$\times F^{-1}(P/s) = F^{-1}(P)/F^{-1}(s)$$

$$F^{-1}(P/F^{-1}(s)) = F^{-1}(P)/s$$

$$F^{-1}(P \parallel Q) = F^{-1}(P) \parallel F^{-1}(Q)$$

$$F^{-1}(\mu X. F) = \mu X. F^{-1}(F)$$

$$g^{-1}(F^{-1}(P)) = (f \circ g)^{-1}(P)$$

Let $\alpha P = \alpha Q$

$P \sqcap Q$ behaves either like P or like Q
the choice is not made by another process.

$$VMX = \text{coin} \rightarrow (\text{choc} \rightarrow VMX$$

$$\sqcap \text{toffee} \rightarrow VMX)$$

each transaction may be different

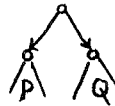
$$VMY = (\mu X. \text{coin} \rightarrow \text{choc} \rightarrow X)$$

$$\sqcap (\mu X. \text{coin} \rightarrow \text{toffee} \rightarrow X)$$

it's always the same.

Picture

$P \sqcap Q$



LAWS

4.2

IMPLEMENTATIONS

4.3

$$P \sqcap P = P$$

$$P \sqcap Q = Q \sqcap P$$

$$P \sqcap (Q \sqcap R) = (P \sqcap Q) \sqcap R$$

An operator F is distributive if it distributes through \sqcap . i.e.,

$$F(P \sqcap Q) = F(P) \sqcap F(Q)$$

$$G(P, Q \sqcap R) = G(P, Q) \sqcap G(P, R)$$

$$G(P \sqcap Q, R) = G(P, R) \sqcap G(Q, R)$$

All operators so far are distributive, e.g.

$$x \rightarrow (P \sqcap Q) = (x \rightarrow P) \sqcap (x \rightarrow Q)$$

$$(x: A \rightarrow (P_x \sqcap Q_x)) = (x: A \rightarrow P_x) \sqcap (x: A \rightarrow Q_x)$$

WARNING. $\mu X.$ is not distributive.

compare VMX with VMY .

A non deterministic process has more than one implementation, e.g.

$$\text{or } 1(P, Q) = P$$

$$\text{or } 2(P, Q) = Q$$

Or even allow environment to choose:

$$\text{or } 3(P, Q) = \lambda x. \text{if } P(x) = \text{"BLEEP then } Q(x)$$

$$\text{else if } Q(x) = \text{"BLEEP then } P(x)$$

$$\text{else } \text{or } 3(P(x), Q(x))$$

or 1 and or 2 are efficient but asymmetric.

The set of implementations is symmetric

$$\{\text{or } 1(P, Q), \text{or } 2(P, Q), \text{or } 3(P, Q)\}$$

$$\{\text{or } 1(Q, P), \text{or } 2(Q, P), \text{or } 3(Q, P)\}$$

$P \square Q$ behaves like P or like Q .
The choice is made by the environment on the first step only.

$$(c \rightarrow P) \square (d \rightarrow Q) = (c \rightarrow P \mid d \rightarrow Q) \text{ if } c \neq d$$

$$(c \rightarrow P) \square (c \rightarrow Q) = c \rightarrow (P \sqcap Q)$$

LAWS

\square is idempotent, symmetric, associative and distributive, and has unit STOP i.e.,
 $P \square \text{STOP} = P$

$$\begin{aligned} (x: A \rightarrow P x) \square (y: B \rightarrow Q y) = \\ x: A \cup B \rightarrow \begin{cases} \text{if } z \in A - B \text{ then } P z \\ \text{else if } z \in B - A \text{ then } Q z \\ \text{else } (P z) \sqcap (Q z) \end{cases} \end{aligned}$$

INTERLEAVING

4.5

Let s be a sequence of sequences

$$\hat{\wedge} s = s_0 \hat{\wedge} s_1 \hat{\wedge} s_2 \hat{\wedge} \dots \hat{\wedge} s_{n-1}$$

zip(s, t) takes alternate elements from s & t :

$$\text{zip}(s, t) = \begin{cases} \text{if } s = \langle \rangle \text{ then } t \\ \text{else } \langle s_0 \rangle \hat{\wedge} \text{zip}(t, s') \end{cases}$$

u interleaves $(v, w) \equiv$

$$\exists s, t. u = \hat{\wedge} \text{zip}(s, t) \wedge v = \hat{\wedge} s \wedge w = \hat{\wedge} t$$

ASSERTION

$$\begin{aligned} P \parallel Q = \underline{E} (s, t: \text{tr interleaves } (s, t) \\ \wedge P[s/\text{tr}] \\ \wedge Q[t/\text{tr}]) \end{aligned}$$

$(P \parallel Q)$ does α whenever either P or Q does α . If both can do α , the choice is non-determinate

$$\alpha(P \parallel Q) = \alpha P \cup \alpha Q.$$

A footman is constructed from four lackey

$$\text{FOOTMAN} = \text{LKY} \parallel \text{LKY} \parallel \text{LKY} \parallel \text{LKY}$$

LAWS

\parallel is associative, commutative, distributive, with unit STOP and zero RUN i.e.,
 $P \parallel \text{RUN} = \text{RUN}$.

$$\text{Let } P = (x: A \rightarrow P x), Q = (y: B \rightarrow Q y)$$

$$\begin{aligned} \text{Then } P \parallel Q = (x: A \rightarrow (P x \parallel Q)) \\ \sqcap (y: B \rightarrow (P \parallel Q y)) \end{aligned}$$

CONCEALMENT

4.6

$P \setminus C$ behaves like P except that events in set C occur autonomously and invisibly.

$$\alpha(P \setminus C) = \alpha P - C.$$

Soundproofing

$$\text{VMS} = \text{NOISYVM} \setminus \{\text{dink, toffee}\}$$

$$((\mu X. a \rightarrow c \rightarrow X) \parallel (\mu X. c \rightarrow b \rightarrow X)) \setminus \{c\}$$

$$= (a \rightarrow c \rightarrow (\mu X. (a \rightarrow b \rightarrow c \rightarrow X \mid b \rightarrow a \rightarrow c \rightarrow X))) \setminus \{c\}$$

$$= (a \rightarrow \mu X. (a \rightarrow b \rightarrow X \mid b \rightarrow a \rightarrow X))$$

16) LAWS

4.7

$$\begin{aligned}
 P \setminus \{\} &= P \\
 (P \setminus B) \setminus C &= P \setminus (B \cup C) \\
 (P \cap Q) \setminus C &= (P \setminus C) \cap (Q \setminus C) \\
 STOP_A \setminus C &= STOP_{A-C} \\
 (x \rightarrow P) \setminus C &= x \rightarrow (P \setminus C) \quad \text{if } x \notin C \\
 &= P \setminus C \quad \text{if } x \in C
 \end{aligned}$$

if $A \cap C = \{\}$
 $(x:A \rightarrow P_x) \setminus C = (x:A \rightarrow (P_x \setminus C))$

if $A \neq \{\} \wedge A \subseteq C$ *A finite.*
 $(x:A \rightarrow P_x) \setminus C = \prod_{x \in A} P_x \setminus C$

if $A \cap C \neq \{\}$ *A finite.*
 $(x:A \rightarrow P_x) \setminus C = Q \cap (Q \sqcup x:A-C \rightarrow (P_x \setminus C))$
 where $Q = \prod_{x \in A \cap C} (P_x \setminus C)$

if $X = \{x, y, \dots, z\}$

$$\prod_{w \in X} P_w = P_x \cap P_y \cap \dots \cap P_z$$

IMPLEMENTATION

4.9

hide(P, c) implements $P \setminus \{c\}$

hide(P, x) = if P(x) = "BLEEP" then
 $\lambda y. \text{if } P(y) = \text{"BLEEP" then "BLEEP"}$
 else hide(P(y), x)
 else *hide(P(x), x)

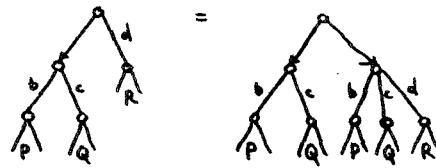
*NOTE danger of infinite recursion:

such hiding is not allowed.

PICTURES

4.8

Just remove hidden symbols from arcs.
 Single unlabelled arcs may be suppressed:



WARNING: is NOT ALLOWED



$$(\mu X. (c \rightarrow X \mid d \rightarrow STOP)) \setminus \{c\}$$

ALPHABET EXTENSION

4.10

Let $\alpha P \cap B = \{\}$

$P \underline{\alpha} B$ behaves like P, except that

$$\alpha(P \underline{\alpha} B) = \alpha P \cup B$$

It never performs any action from B

$$(P \underline{\alpha} B) \setminus B = P$$

Alphabet extension usually implicit:

$$\text{If } \alpha P \cap \alpha Q = \{\}$$

$$P \parallel Q = P \parallel Q$$

$$(\text{means: } (P \underline{\alpha} Q) \parallel (Q \underline{\alpha} (P))) = P \parallel Q$$

17) PROCESS NAMING

4.11

MASTER AND SLAVE

4.12

A compound event

$m.e$

records performance of e by a process named m

$m:P$ does $m.e$ whenever P would do e .

$$m:P = \text{strip}_m^{-1}(P)$$

where $\text{strip}_m(m.e) = e$

A pair of vending machines.

left: VMS || right: VMS

NOTE: VMS || VMS = VMS

RECURSION

4.13

Another definition of ROCKET

$$Z = (\text{around} \rightarrow Z \\ | \text{up} \rightarrow [m:Z \parallel \text{LOOP}])$$

$$\text{LOOP} = (\text{up} \rightarrow m.\text{up} \rightarrow \text{LOOP} \\ | \text{down} \rightarrow (m.\text{down} \rightarrow \text{LOOP} \\ | m.\text{around} \rightarrow Z \underline{x} \alpha(m:Z)))$$

$$\alpha Z = \{\text{up}, \text{down}, \text{around}\} \\ \alpha \text{LOOP} = \alpha Z \cup \alpha(m:Z) \\ = \{\text{up}, \text{down}, \text{around}, \\ m.\text{up}, m.\text{down}, m.\text{around}\}$$

Let $\alpha(m:P) \subseteq U$

$[m:P \parallel U]$ is a process in which U controls $m:P$ by actions in $\alpha(m:P)$, and these are concealed from environment

$$[m:P \parallel U] = ((m:P) \parallel U) \setminus \alpha(m:P)$$

A PILOT flies a rocket named m

$$[m:\text{ROCKET} \parallel \text{PILOT}]$$

To land the rocket she does:

$$\mu X. (m.\text{down} \rightarrow X \\ | m.\text{around} \rightarrow \dots)$$

A pilot with two rockets:

$$[n:\text{ROCKET} \parallel [m:\text{ROCKET} \parallel \text{PILOT}]]$$

STACK

4.14

$$\alpha \text{STACK} = \{\text{in1p}, \text{out1p}, \text{in2p}, \text{out2p}, \text{isempty}\}$$

$$\text{STACK} = \text{EMPTY}$$

$$\text{EMPTY} = (\text{isempty} \rightarrow \text{EMPTY} \\ | \text{in1p} \rightarrow [m:\text{EMPTY} \parallel \text{LOOP1}] \\ | \text{in2p} \rightarrow [m:\text{EMPTY} \parallel \text{LOOP2}])$$

$$\text{LOOP1}_m = (\text{in1p} \rightarrow m.\text{in1p} \rightarrow \text{LOOP1} \\ | \text{in2p} \rightarrow m.\text{in1p} \rightarrow \text{LOOP2})$$

$$\text{LOOP2}_m = (\text{out1p} \rightarrow \text{CONTRACT})$$

$$\text{CONTRACT}_m = (m.\text{out1p} \rightarrow \text{LOOP1} \\ | m.\text{out2p} \rightarrow \text{LOOP2} \\ | m.\text{isempty} \rightarrow \text{EMPTY})$$

$$\text{LOOP1}_n = \underline{x} \alpha(m:\text{EMPTY})$$



9)

WARNING

4.15

4.16

RECURSION UNDER HIDING
is not guarded.

$$\begin{aligned} \text{Let } F(X) &= a \rightarrow (X \setminus \{a\}) \underline{x} \{a\} \\ \alpha X &= \{a, b, c\} \\ F^2(X) &= a \rightarrow ((a \rightarrow (X \setminus \{a\}) \underline{x} \{a\}) \setminus \{a\}) \underline{x} \{a\} \\ &= a \rightarrow (X \setminus \{a\} \underline{x} \{a\} \setminus \{a\}) \underline{x} \{a\} \\ &= a \rightarrow X \setminus \{a\} \underline{x} \{a\} \\ &= F(X) \end{aligned}$$

\therefore For all X , $F(X)$ is a solution of
 $Y = F(Y)$

i.e., there are many solutions.

$$\begin{aligned} F(X) / \langle a \rangle &= X \setminus \{a\} \underline{x} \{a\} \\ \underline{x} &= \mu b \rightarrow X \\ \underline{x} &= \mu X c \rightarrow X \\ &\text{REFUSALS.} \end{aligned}$$

.1

.2

$$\begin{aligned} \text{Let } x \neq y, \alpha P = \alpha Q = \{x, y\}, \\ P = (x \rightarrow P), \quad Q = (y \rightarrow Q) \end{aligned}$$

$$\begin{aligned} (P \parallel Q) \parallel P &= (x \rightarrow P \mid y \rightarrow Q) \parallel (x \rightarrow P) \\ &= (x \rightarrow (P \parallel P)) = P \end{aligned}$$

$$\begin{aligned} (P \cap Q) \parallel P &= (P \parallel P) \cap (Q \parallel P) \\ &= P \cap (y \rightarrow Q \parallel x \rightarrow P) \\ &= P \cap \text{STOP} \end{aligned}$$

$$\therefore P \parallel Q \neq P \cap Q$$

because $P \cap Q$ can refuse $\{x\}$
on its first step

$P \parallel Q$ can't.

Let r be a new event

$$\mu X. F(X) = (\mu X. r \rightarrow F(X)) \setminus \{r\}$$

The recursion fails just when
the hiding fails.

$B \in \text{refusals}(P)$ means

$$P \parallel (\alpha : B \rightarrow Q \alpha)$$

can deadlock immediately.

$$\text{refusals}(\alpha : A \rightarrow P \alpha) = \{B \mid \overset{B \in \alpha P}{A \cap B} = \{\}\}$$

$$\text{refusals}(P \cap Q) = \text{refusals}(P) \cup \text{refusals}(Q)$$

$$\text{refusals}(P \parallel Q) = \text{refusals}(P) \cap \text{refusals}(Q)$$

$$\text{refusals}(P \parallel\parallel Q) = \{X \cup Y \mid X \in \text{refusals}(P) \wedge Y \in \text{refusals}(Q)\}$$

$$\text{refusals}(P \parallel\parallel\parallel Q) = \text{refusals}(P \parallel Q)$$

$$\text{refusals}(F^{-1}(P)) = \{X \mid F(X) \in \text{refusals}(P)\}$$

$$\begin{aligned} \text{refusals}(P \setminus C) &= \bigcup_{\substack{S \in C^* \\ S \in \text{trans}(P) \wedge X \cap C = \{\}}} \{X \mid X \in \text{refusals}(P/S)\} \\ &\text{provided this is finite union} \end{aligned}$$

19)

LAWS

.3

$$X \in \text{refusals}(P) \Rightarrow X \subseteq \alpha P$$

$$\{\} \in \text{refusals}(P)$$

$$(X \cup Y) \in \text{refusals}(P) \Rightarrow X \in \text{refusals}(P)$$

$$X \in \text{refusals}(P) \wedge \alpha \in \alpha P \Rightarrow \\ X \cup \{\alpha\} \in \text{refusals}(P) \\ \forall \langle \alpha \rangle \in \text{traces}(P)$$

OBSERVATIONS.

.4

(s, B) is an observation of P if

$$B \in \text{refusals}(P/s) \wedge s \in \text{traces}(P)$$

$(s, B) \leq (t, C)$ means.

$$s = t \wedge B \subseteq C$$

$$\vee s < t \wedge B = \{\}$$

ASSERTIONS

.5

Let's identify a process with a predicate $P(\text{tr}, \text{ref})$ which describes all its possible observations. (tr, ref)

$$(\alpha: A \rightarrow P\alpha) = (\text{tr} = \langle \alpha \rangle \wedge \bar{A} \subseteq \text{ref} \subseteq \alpha P) \\ \vee (\text{tr}_0 \in A \wedge P(\text{tr}_0)[\text{tr}'/\text{tr}])$$

$$\text{STOP}_A = \text{tr} = \langle \rangle \wedge \bar{A} \subseteq \text{ref} \subseteq A$$

$$(c \rightarrow P) = (\text{tr} = \langle c \rangle \wedge c \notin \text{ref} \subseteq \alpha P)$$

$$\text{tr} \neq \langle c \rangle \wedge \forall \text{tr}_0 = c \wedge P[\text{tr}'/\text{tr}]$$

$$P/s = P[s^{\wedge} \text{tr}/\text{tr}]$$

$$(P \parallel Q) = \exists X, Y: \text{ref} = X \cup Y:$$

$$P[X/\text{ref}, \text{tr}'/\alpha P/\text{tr}] \\ \wedge Q[Y/\text{ref}, \text{tr}'/\alpha Q/\text{tr}]$$

$$F^*(P) = P[F(\text{ref})/\text{ref}, F^*(\text{tr})/\text{tr}]$$

.6

$$P \cap Q = P \vee Q$$

$$P \square Q = \text{if } \text{tr} \neq \langle \rangle \text{ then } (P \cap Q) \text{ else } (P \vee Q)$$

$$P \parallel\parallel Q = \underline{\exists} s, t: \text{tr} \text{ interleaves } (s, t): \\ P[s/\text{tr}] \wedge Q[t/\text{tr}]$$

$$P \setminus C = \underline{\exists} s: s \uparrow C = \text{tr}: \\ P[s/\text{tr}, (\text{ref} \cup C)/\text{ref}]$$

provided that

$$\underline{A}t: [\text{tr} \uparrow \text{tr}' \uparrow \bar{C} = t \wedge P[\{\}/\text{ref}]]$$

is FINITE

$$P \sqsupset \text{tr} \neq \langle \rangle \sqsupset Q$$

Bill Roscoe
Steve Brookes.

20) EXAMPLES

.7

VM with buffering of coins perhaps.

Let $BAL = tr.coin - tr.choc$

$$VM = \square((BAL = 0 \wedge \{coin\} \leq \overline{ref}) \vee (BAL > 0 \wedge \{choc\} \leq \overline{ref}))$$

$$= VM_0 = coin \rightarrow VM_1$$

$$VM_{n+1} = choc \rightarrow VM_n$$

$$\Pi (coin \rightarrow VM_{n+1} \square choc \rightarrow VM_n)$$

$$VMS = VM \wedge \square BAL \leq 1$$

$$VMS \Rightarrow VM$$

i.e VMS is a valid implementation of VM

8.

CD is a cash dispenser

$$\alpha CD = \{withdraw, deposit\}$$

$$BAL = tr.deposit - tr.withdraw$$

$$CD = \square(|deposit \in \overline{ref} \wedge (BAL \leq 0 \vee withdraw \in \overline{ref}))$$

EXERCISE

A. Write process CD

B. Write a process CDB such that

$$CDB \Rightarrow CD$$

COMMUNICATING PROCESSES. 5.1

A compound event

c.v

denotes communication of a message with value v on channel c

$$\alpha c(P) = \{v \mid c.v \in \alpha P\}$$

gives the set of messages which P can communicate on channel c

IF E is an expression with value v

$$(c!E \rightarrow P) = (c.v \rightarrow P)$$

outputs v on channel c

$$(c?x \rightarrow P_x) = (y: strip_c^{-1}(\alpha c(P)) \rightarrow P(strip_c(y)))$$

inputs a message x on channel c

Loosely: $c \in \alpha P$ means $\alpha c(P) \neq \{\}$

EXAMPLES

5.2

$$COPY = \mu X. left?y \rightarrow right!y \rightarrow X$$

$$DOUBLE = \mu X. left?y \rightarrow right!(y+y) \rightarrow X$$

$$UNPACK = P_{<} \quad \alpha left(P) = (\alpha right(P))^*$$

$$P_{<} = left?s \rightarrow P_s$$

$$P_{<x>s} = right!x \rightarrow P_s$$

$$PACK = P_{<} \quad (\alpha left(P))^* = \alpha right(P)$$

$$P_l = left?x \rightarrow P_{l<x>} \quad \text{if } \#l < 125$$

$$P_l = right!l \rightarrow P_{<} \quad \text{if } \#l = 125$$

$$SQUASH = \mu X. left?x \rightarrow$$

$$\underline{\text{if}} \ x = "a" \ \underline{\text{then}} \ (right!x \rightarrow X)$$

$$\underline{\text{else}} \ left?y \rightarrow$$

$$\underline{\text{if}} \ y = "a" \ \underline{\text{then}} \ (right!"a" \rightarrow X)$$

$$\underline{\text{else}} \ (right!"a" \rightarrow right!y \rightarrow X)$$

21)

VAR = left?x → VAR_x
 VAR_x = (left?y → VAR_y
 | right!x → VAR_x)

MERGE =
 (left1?x → right!x → MERGE
 | left2?x → right!x → MERGE)

BUFFER = P_<
 P_< = left?x → P<sub><x>
 P<sub><x>s = (left?y → P<sub><x>s<y>
 | right!x → P_s)</sub></sub></sub>

STACK = P_<
 P_< = (isempty → P_< | left?x → P_x)
 P_{x>s =}

PIPES

A pipe is a process with only two channels left and right

(P>>Q) is formed by joining right channel of P to left channel of Q



Communications on connecting channel hidden

Provided that $\alpha\text{right}(P) = \alpha\text{left}(Q)$

$(P \gg Q) = (P[\text{mid}/\text{right}] \parallel Q[\text{mid}/\text{left}]) \setminus \alpha\text{mid}$

where $P[d/c] = F^{-1}(P)$

$F(d.v) = c.v$

$F(b.v) = b.v$ for $b \neq d$

$\alpha(P \gg Q) = \alpha\text{left}(P) \cup \alpha\text{right}(Q)$

$(P \gg Q) \gg R = P \gg (Q \gg R)$

5.3

TRACES

5.4

$\text{tr}.c = \text{strip}_c^*(\text{tr}[\text{tr}.c])$
 IF $\text{tr} = \langle \text{left}.3, \text{right}.3, \text{left}.37 \rangle$
 then $\text{tr}.left = \langle 3, 37 \rangle$
 $\text{tr}.right = \langle 3 \rangle$

Define $t \geq^s = s \leq t \ \&\#t \leq \#s + n$

COPY = $\square \text{tr}.left \geq^1 \text{tr}.right$

DOUBLE = $\square \text{double}^*(\text{tr}.left) \geq^1 \text{tr}.right$

UNPACK = $\square \text{tr}.left \geq \text{tr}.right$
 $\& \wedge (\text{trunc}(\text{tr}.left)) \leq \text{tr}.right$

where $\text{trunc}(s \langle x \rangle) = s$

PACK = $\square \text{tr}.left \geq^{125} \wedge \text{tr}.right$
 $\& \forall s: s \in \text{range tr}.right: \#s = 125$

5.5

EXAMPLES

5.6

QUADRUPLE = DOUBLE >> DOUBLE

LISTING = UNPACK >> PACK

CONWAY = UNPACK >> SQUASH >> PACK

BUFFERED = UNPACK >> BUFFER >> PACK

BUFFER3 = COPY >> COPY >> COPY

\gg is associative

$$(\text{right!}v \rightarrow P) \gg (\text{left?}x \rightarrow Q) = P \gg (Q \vee)$$

$$\begin{aligned} (\text{right!}v \rightarrow P) \gg (\text{right!}w \rightarrow Q) \\ = (\text{right!}w \rightarrow ((\text{right!}v \rightarrow P) \gg Q)) \end{aligned}$$

$$\begin{aligned} (\text{left?}x \rightarrow P) \gg (\text{left?}y \rightarrow Q) \\ = (\text{left?}x \rightarrow (P \wedge \gg (\text{left?}y \rightarrow Q))) \end{aligned}$$

$$\begin{aligned} (\text{left?}x \rightarrow P) \gg (\text{right!}w \rightarrow Q) \\ = (\text{left?}x \rightarrow (P \wedge \gg (\text{right!}w \rightarrow Q))) \\ \text{right!}w \rightarrow ((\text{left?}x \rightarrow P) \gg Q) \end{aligned}$$

As above, with \gg replaced by $\gg R \gg$

$$\mathcal{L}(S;T)_r = \{u :: \mathcal{L}Su \& uTr\}$$

\mathcal{L} is associative, with unit ϵ .

$$(\mathcal{L}; \mathcal{L}) = \mathcal{L}$$

$$\mathcal{L}^m; \mathcal{L}^n = \mathcal{L}^{m+n}$$

$$f^*; \mathcal{L}^n = \mathcal{L}^n; f^*$$

$$\text{where } \mathcal{L}(f^*)_r = r = f^*(\mathcal{L})$$

$$f^*; g^* = (g \circ f)^*$$

S is grounded if for all ℓ, r :

$$\begin{aligned} \mathcal{L}S_r \Rightarrow \mathcal{L}r = \epsilon \Leftrightarrow \\ \vee \text{trunc}(\mathcal{L})S_r \\ \vee \mathcal{L}S \text{trunc}(r) \end{aligned}$$

If S and T are grounded
so is $S;T$

Let R and S be grounded: 5.9

$$\begin{aligned} \text{If } P &= \square \text{tr. left } R \text{ tr. right} \\ Q &= \square \text{tr. left } S \text{ tr. right} \end{aligned}$$

then $P \gg Q = \square \text{tr. left } (R;S) \text{ tr. right}$
provided that for all s, t
 $\{u | sRu \& uSt\}$ is finite.

$$\begin{aligned} \text{BUFFER} \gg \text{BUFFER} \\ = \square \text{tr. left } (\mathcal{L}; \mathcal{L}) \text{ tr. right} \\ = \text{BUFFER} \end{aligned}$$

$$(\text{COPY} \gg \text{COPY}) = \square \text{tr. left } \mathcal{L}^2 \text{ tr. right}$$

$$\begin{aligned} (\text{DOUBLE} \gg \text{DOUBLE}) &= \\ \square \text{tr. left } (\text{double}^*; \mathcal{L}; \text{double}^*; \mathcal{L}) \text{ tr. right} \\ &= (\text{double}^*; \text{double}^*; \mathcal{L}; \mathcal{L}) \\ &= (\text{quadruple}^*; \mathcal{L}^2) \end{aligned}$$

SEQUENTIAL PROCESSES. 6.1

$P;Q$ behaves as P until P terminates
successfully; then behaves as Q

SKIP_A does nothing but terminate
successfully.
 $\mu \text{SKIP}_A = A$

$$\text{SKIP}_A \neq \text{STOP}_A$$

If P is guarded.

$$\mu X. P;X$$

2.3) EXAMPLES

6.2

6.3

$$VM1 = (\text{coin} \rightarrow \text{choc} \rightarrow \text{SKIP})$$

$$VM3 = VM1; VM1; VM1$$

$$VMS = *VM1$$

$$AnBCn = \mu X. (b \rightarrow \text{SKIP} \\ | a \rightarrow (X; (c \rightarrow \text{SKIP})))$$

accepts n a's followed by a b

followed by n c's, for any n

$$\exists n:: \text{tr} \leq \langle a \rangle^n \langle b \rangle \langle c \rangle^n \checkmark$$

where \checkmark denotes successful termination

$$POS = (\text{down} \rightarrow \text{SKIP} \mid \text{up} \rightarrow (POS; POS))$$

POS terminates when number of downs
first exceeds the number of ups

$$C_0 = (\text{around} \rightarrow C_0 \mid \text{up} \rightarrow C_1)$$

$$C_{n+1} = POS; C_n \quad \text{for all } n \geq 0$$

$$C_0 = CT_0 \quad !$$

LAWS

6.4

$;$ is associative with unit SKIP
distributive

$$(x: A \rightarrow Px); Q = (x: A \rightarrow (Px; Q))$$

Theorem $C_0 = CT_0$

Proof: C_0 satisfies equations of CT_0

Lemma (1) $C_0 = (\text{around} \rightarrow C_0 \mid \text{up} \rightarrow C_1)$ by def. C_0

Lemma (2) $C_{n+1} = (\text{up} \rightarrow C_{n+2} \mid \text{down} \rightarrow C_n)$

$$\begin{aligned} \text{Proof. LHS} &= POS; C_n && \text{def } C_{n+1} \\ &= (\text{down} \rightarrow \text{SKIP} \mid \text{up} \rightarrow POS; POS); C_n && \text{def } POS \\ &= (\text{down} \rightarrow \text{SKIP}; C_n \mid \text{up} \rightarrow POS; POS; C_n) && \text{dist} \\ &= (\text{down} \rightarrow C_n \mid \text{up} \rightarrow POS; (POS; C_n)) && \text{assoc} \\ &= (\text{down} \rightarrow C_n \mid \text{up} \rightarrow POS; C_{n+1}) && \text{def } C_{n+1} \\ &= (\text{down} \rightarrow C_n \mid \text{up} \rightarrow C_{n+2}) && \text{def } C_{n+2} \\ &= \text{RHS} \end{aligned}$$