ETH Zürich

Chair of Software Engineering

Concurrent Object-Oriented Programming

Prof. Dr. Bertrand Meyer

Lecture 11: An introduction to CSP

Origin

Communicating Sequential Processes: C.A.R. Hoare

1978 paper, based in part on ideas of E.W. Dijkstra (guarded commands, 1978 paper and "A Discipline of Programming" book)

Revised with help of S. D. Brooks and A.W. Roscoe

1985 book, revised 2004

CSP purpose

Concurrency formalism

- Expresses many concurrent situations elegantly
- Influenced design of several concurrent programming languages, in particular Occam (Transputer)

Calculus

- Formally specified: laws
- Makes it possible to prove properties of systems

Basic notions

Processes engage in events

Example:

$$BDVM = (coin \rightarrow coffee \rightarrow coin \rightarrow coffee \rightarrow STOP)$$

$\alpha(BDVM) = \{coin, coffee\}$

Basic CSP syntax

P ::=

Stop	|	-- Does not engage in any events
$a \rightarrow P$	|	-- Accepts a, then engages in P
$P \sqcap P$	|	-- Internal choice
$P \;\Box\; P$	|	-- External choice
$P \;\|\|\; P$	|	-- Concurrency
$P \;\|\|\|\; P$	|	-- Interleaving
$P \setminus H$	|	-- Hiding (H: alphabet symbols)
$\mu P \bullet f(P)$		-- Recursion

Some examples

CLOCK = (tick → CLOCK)

This is an abbreviation for

$CLOCK = \mu P \bullet (tick \rightarrow P)$

CVM = (in1f → (coffee → CVM))

= (in1f → coffee → CVM) -- Right-associativity

CHM1 = (in1f → out50rp → out20rp → out20rp → out10rp)

CHM2 = (in1f → out50rp → out50rp)

CHM = CHM1 □ CHM2

More examples

$$COPYBIT = (in.0 \rightarrow out.0 \rightarrow COPYBIT \;\square\; in.1 \rightarrow out.1 \rightarrow COPYBIT)$$

More examples

VMC =

$$
\begin{aligned}
&(in2f \rightarrow \\
&\qquad ((large \rightarrow VMC) \;\Box \\
&\qquad (small \rightarrow out1f \rightarrow VMC)) \\
&\Box \\
&(in1f \rightarrow \\
&\qquad ((small \rightarrow VMC) \;\Box \\
&\qquad (in1f \rightarrow large \rightarrow VMC))
\end{aligned}
$$

$$
\begin{aligned}
FOOLCUST = (&in2f \rightarrow large \rightarrow FOOLCUST \;\Box \\
&in1f \rightarrow large \rightarrow FOOLCUST)
\end{aligned}
$$

FOOLCUST || VMC =

$$\mu P \bullet (in2f \rightarrow large \rightarrow P \;\Box\; in2f \rightarrow STOP)$$

Formal semantics: through traces

Internal non-deterministic choice

CH1F = (in1f →
((out20rp → out20rp →
out20rp →out20rp →out20rp → CH1F)
⊓
(out50rp → out50rp → CH1F)))

Laws of concurrency

$P \;||\; Q \;=\; Q \;||\; P$

$P \;||\; (Q \;||\; R)) \;=\; ((P \;||\; Q) \;||\; R)$

$P \;||\; STOP_{\alpha P} = STOP_{\alpha P}$

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

$(c \rightarrow P) \;||\; (d \rightarrow Q) \;=\; STOP$ -- If $c \neq d$

$(x\!: A \rightarrow P\,(x)) \;||\; (y\!: B \rightarrow Q\,(y)) = (z\!: (A \cap B) \rightarrow (P\,(z) \;||\; Q\,(z))$

Laws of non-deterministic internal choice

$P \sqcap Q = Q \sqcap P$

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

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

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

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

The recursion operator is not distributive; consider:

$$P = \mu X \bullet ((a \rightarrow X) \sqcap (b \rightarrow X))$$

$$Q = (\mu X \bullet (a \rightarrow X)) \sqcap (\mu X \bullet (b \rightarrow X))$$