COMMUNICATING SEQUENTIAL PROCESSES

C.A.R Hoare
The behaviour pattern of an object described in terms of the events in its alphabet.

Example:

- A counter starts on the bottom left square of a board and can only move up or right to a white square
- $\alpha_{CTR} = \{\text{up, right}\}$
- $CTR = \{\text{right} \rightarrow \text{up} \rightarrow \text{right} \rightarrow \text{right} \rightarrow \text{STOP}_{\alpha_{CTR}}\}$
RECURSION

- Notation: $X = F(X)$ becomes $\mu X: A. F(X)$
  - Where $X$ is the bound variable and $A$ is the alphabet

- Example: A perpetual clock
  - $CLOCK = \mu X: \{tick\}. (tick \rightarrow X)$
CHOICE

- If x and y are distinct choices and P and Q processes then
  - \((x \rightarrow P | y \rightarrow Q)\)

- Example:
  - A vending machine which offers a choice of input coins and a choice of either a small or large biscuit and change
  - \(VMC = (\text{in}2p \rightarrow (\text{large} \rightarrow VMC | \text{small} \rightarrow \text{out}1p \rightarrow VMC) | \text{in}1p \rightarrow (\text{small} \rightarrow VMC | \text{in}1p \rightarrow (\text{large} \rightarrow VMC | \text{in}1p \rightarrow \text{STOP}))))\)
IMPLEMENTATION OF PROCESSES

- Every process can be written in the form \((x: B \rightarrow F(x))\)
  - The process may be regarded as a function \(F\).
  - With a domain \(B\) defining the set of events in which the process is initially prepared to engage.
  - For each \(x\) in \(B\), \(F(x)\) defines the future behaviour of the process if the first event was \(x\).
LISP

- Each event is an atom
- A process is a function which can be applied to a symbol
- If the symbol is not a possible first event for the process then the function returns “BLEEP

Example: Binary choice \((c \rightarrow P | d \rightarrow Q)\)

\[
\text{choice2}(c, P, d, Q) = \lambda x. \begin{cases} 
    P & \text{if } x = c \\
    Q & \text{if } x = d \\
    \text{"BLEEP"} & \text{else}
\end{cases}
\]
TRACES

- A trace of the behaviour of a process is a finite sequence of symbols recording the events in which the process has engaged up to some moment in time.

- Example: A trace of a vending machine at the moment it is finished serving its first two customers
  - <coin, choc, coin, choc>
OPERATIONS ON TRACES

1. Catenation
   \[ <\text{coin, choc}> \land <\text{coin, toffee}> = <\text{coin, choc, coin, toffee}> \]

2. Restriction
   \[ <\text{around, up, down, around}> \uparrow \{\text{up, down}\} = <\text{up, down}> \]

3. Head and Tail
   \[ <x, y, x>_{0} = x \quad <x, y, x>^{' } = <y, x> \]

4. Star
   \[ A^* = \{s | s \uparrow A = s\} \]

5. Ordering
   \[ s \leq t = (\exists u. s^u u = t) \]

6. Length
   \[ \# <x, y, x> = 3 \]
IMPLEMENTATION OF TRACES

- Traces are implemented by lists of atoms representing their events
- Operations on traces can be readily implemented as functions on lists
  - \( s^t = \text{append}(s, t) \)
- Example: Restriction
  - \( \text{isMember}(x, B) = \text{if } B = \text{NIL then false} \)
    - \( \text{else if } x = \text{car}(B) \text{ then true} \)
    - \( \text{else isMember}(x, \text{cdr}(B)) \)
  - \( \text{restrict}(s, B) = \text{if } s = \text{NIL then NIL} \)
    - \( \text{else if isMember(car(s), B)} \)
      - \( \text{then cons(car(s), restrict(cdr(s), B))} \)
    - \( \text{else restrict(cdr(s), B)} \)
TRACES OF A PROCESS

- Before a process starts it is not known which trace will occur the choice depends on environmental factors beyond the control of the process
- However we can know the complete set of possible traces and this is denoted as the function traces(P)
- Example: A perpetual clock
  - $\text{traces}(\mu X. \text{tick} \rightarrow X) = \{\langle\rangle, \langle\text{tick}\rangle, \langle\text{tick}, \text{tick}\rangle, \ldots\}$
  - $= \{\text{tick}\}^*$

Marie Farrell IRCHSS Scholar Principles of Programming Research Group
IMPLEMENTATION

- \( \text{isTrace}(s, P) = \text{if } s = \text{NIL then true} \)
  
  \( \text{else if } P(s_0) = \text{"BLEEP then false} \)
  
  \( \text{else isTrace } (s', P(s_0)) \)
AFTER

- If $s \in \text{traces}(P)$
  then $P/s$

Is a process that behaves the same as $P$ behaves from the time after $P$ has engaged in all the actions recorded in the trace $s$
SPECIFICATIONS

- In the case of a process the most relevant observation of behaviour is the trace of events that occur up to a given moment in time.
- Use the variable $tr$ to stand for an arbitrary trace of the process being specified.
- Use $tr \downarrow c = \#(tr \uparrow \{c\})$
  - i.e. the number of occurrences of the symbol $c$ in $tr$
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.

- $FAIR_1 = ((tr \downarrow coin) \leq (tr \downarrow choc) + 1)$
SATISFACTION

- If P is a product which meets a specification S we say that P satisfies S.
  - P sat S
  - ∀tr. tr ∈ traces(P) ⇒ S
QUESTIONS?