Theory of concurrency

Lecture 9

Nondeterminism

$P = (x \to y \to P)$, $Q = (y \to x \to Q)$, $\alpha P = \alpha Q = \{x, y\}$
$(P \sqcap Q) \parallel P = (x \to y \to P) = P$
$(P \sqcap Q) \parallel P = (P \parallel P) \sqcap (Q \parallel P) = P \sqcap STOP$

Nondeterminism: **Concealment**

- Concealment is hiding the structure of process components from its environment:
  - to prevent observation or control by the environment
  - some events have to be hidden.

- $P \setminus C$ is a process which behaves like P, except that
  - each occurrence of any event in finite set of events C is concealed.
  - $\alpha(P \setminus C) = (\alpha P) - C$

X1. A noisy vending machine can be placed in a soundproof box:
  - $NOISYVM \setminus \{clink, clunk\}$

- The resulting process is equal to the simple vending machine:
  - $VMS = NOISYVM \setminus \{clink, clunk\}$

$NOISYVM = (coin \to clink \to choc \to clunk \to NOISYVM)$
$VMS = (coin \to (choc \to VMS))$

Nondeterminism: Concealment

- Mutual interactions of concurrent processes are usually
  - regarded as *internal workings* of the resulting systems.
  - intended to occur
    - autonomously and as quickly as possible,
    - without the knowledge or intervention of the environment.
- Then the symbols in the intersection of the alphabets need to be concealed.

X2. Let

- $aP = \{a, c\}$ and $aQ = \{b, c\}$
- $P = (a \to c \to P)$ and $Q = (c \to b \to Q)$

- The action c in the alphabet of both P and Q is an internal action, to be concealed:

$$
\begin{aligned}
(P \parallel Q) \setminus \{c\} &= (a \to c \to \mu X \bullet (a \to b \to c \to X \\
&\qquad \mid b \to a \to c \to X)) \setminus \{c\} \\
&= a \to \mu X \bullet (a \to b \to X \\
&\qquad \mid b \to a \to X)
\end{aligned}
$$

Nondeterminism: Concealment

- Concealment of nothing leaves everything revealed:

L1. $P \setminus \{\} = P$

- Sequential concealment:

L2. $(P \setminus B) \setminus C = P \setminus (B \cup C)$

- Concealment distributes through nondeterministic choice:

L3. $(P \sqcap Q) \setminus C = (P \setminus C) \sqcap (Q \setminus C)$

- Concealment affect only the alphabet of a stopped process:

L4. $STOP_A \setminus C = STOP_{A-C}$

Nondeterminism: Concealment: **Laws**

- Concealment of nothing leaves everything revealed:

L1. $P \setminus \{\} = P$

- Sequential concealment:

L2. $(P \setminus B) \setminus C = P \setminus (B \cup C)$

- Concealment distributes through nondeterministic choice:

L3. $(P \sqcap Q) \setminus C = (P \setminus C) \sqcap (Q \setminus C)$

- Concealment affect only the alphabet of a stopped process:

L4. $STOP_A \setminus C = STOP_{A-C}$

Nondeterminism: Concealment: **Laws**

- Unconcealed events remain unchanged:

L5. $(x \to P) \setminus C = x \to (P \setminus C)$ **if** $x \notin C$

$= P \setminus C$ **if** $x \in C$

- Concealment distributes through concurrency if it hides independent events:

L6. $(P \parallel Q) \setminus C = (P \setminus C) \parallel (Q \setminus C)$ **when** $\alpha P \cap \alpha Q \cap C = \{\}$

  - Usually events of interest are in $\alpha P \cap \alpha Q$.

- Concealment distributes through symbol change by a one-one function:

L7. $f(P \setminus C) = f(P) \setminus f(C)$

Nondeterminism: Concealment: **Laws**

- The initial choice remains the same if the menu does not include concealed events:

L8. $(x : B \rightarrow P(x)) \setminus C = (x : B \rightarrow (P(x) \setminus C))$ **if** $B \cap C = \{\}$

- The concealment of events can introduce nondeterminism:

L9. $(x : B \rightarrow P(x)) \setminus C = \sqcap_{x \in B} (P(x) \setminus C)$ **if** $B \subseteq C$, and B is finite and not empty.

  - when several different concealed events can happen,
    - it is not determined which of them will occur.
  - The choice is disappeared.

Nondeterminism: Concealment: Laws

- Consider the case, when some of the initial events are concealed and some are not.
- In the process $(c \rightarrow P \mid d \rightarrow Q) \setminus C$, where $c \in C$, $d \notin C$
  - The concealed event c may happen immediately.
    - The total behaviour is defined by $(P \setminus C)$, and the event d does not happen.
  - If d occurs, it might have been performed by $(P \setminus C)$ after the hidden occurrence of c.
    - The total behaviour is defined by $(P \setminus C) \,\square\, (d \rightarrow (Q \setminus C))$
      - The choice between this and $(P \setminus C)$ is nondeterministic.
- This reasoning is summarized in the law:
  - $(c \rightarrow P \mid d \rightarrow Q) \setminus C = (P \setminus C) \sqcap ((P \setminus C) \,\square\, (d \rightarrow (Q \setminus C)))$
- The general law:

L10. If $C \cap B$ is finite and non-empty, **then**

$$(x : B \rightarrow P(x)) \setminus C = Q \sqcap (Q \,\square\, (x : (B - C) \rightarrow P(x) \setminus C)),\ \textbf{where}\ Q = \sqcap_{x \in B \cap C} P(x) \setminus C$$

Nondeterminism: Concealment: **Laws**

- Concealment does not distribute backwards through general choice $\square$:
- A counterexample:

$$
\begin{aligned}
&(c \rightarrow STOP \,\square\, d \rightarrow STOP) \setminus \{c\} \\
&\qquad = STOP \sqcap (STOP \,\square\, (d \rightarrow STOP)) && \text{[L10]} \\
&\qquad = STOP \sqcap (d \rightarrow STOP) && \text{[L4]} \\
&\qquad \neq d \rightarrow STOP \\
&\qquad = STOP \,\square\, (d \rightarrow STOP) \\
&\qquad = ((c \rightarrow STOP) \setminus \{c\}) \,\square\, ((d \rightarrow STOP) \setminus \{c\})
\end{aligned}
$$

L4. $P \,\square\, STOP = P$

L5. $(x \to P) \setminus C = x \to (P \setminus C)$ **if** $x \notin C$
$= P \setminus C$ **if** $x \in C$

Nondeterminism: Concealment: **Laws**

- The extension of the alphabet of a process P by inclusion of symbols of a set B:
  - $\alpha(P_{+B}) = \alpha P \cup B$ and $P_{+B} = (P \parallel STOP_B)$ if $B \cap \alpha P = \{\}$
- None of the new events of B will ever actually occur:

L11. $traces(P_{+B}) = traces(P)$

- Concealment of B reverses the extension of the alphabet by B:

L12. $(P_{+B}) \setminus B = P$

- In simple cases, concealment distributes through recursion:

$$\begin{aligned}&(\mu X : A \bullet (c \to X)) \setminus \{c\}\\ &\quad = \mu X : (A - \{c\}) \bullet ((c \to X_{+\{c\}}) \setminus \{c\})\\ &\quad = \mu X : (A - \{c\}) \bullet X \qquad \text{[by L12, L5]}\end{aligned}$$

- The attempt to conceal an *infinite* sequence of consecutive events leads to the same unfortunate result as an infinite loop or unguarded recursion.
  - The *divergence*.

Nondeterminism: Concealment: **Laws**

- If the divergent process is infinitely often capable of some unconcealed event
  - the recursion is unguarded and leads to divergence:

$$
\begin{aligned}
&(\mu X \bullet (c \to X \mathbin{\square} d \to P)) \setminus \{c\} \\
&\qquad = \mu X \bullet ((c \to X \mathbin{\square} d \to P) \setminus \{c\}) \\
&\qquad = \mu X \bullet (X \setminus \{c\}) \sqcap ((X \setminus \{c\}) \mathbin{\square} d \to (P \setminus \{c\})) \qquad \text{[by L10]}
\end{aligned}
$$

- Even though the environment is infinitely often offered the choice of selecting d
  - the process may infinitely often choose to perform the hidden event instead.
- In this case, we prefer not to insist on fairness of nondeterminism.

Nondeterminism: Concealment: **Laws**

- In some sense, hiding is in fact fair.
- Let $d \in \alpha R$, and consider the process

$$
\begin{aligned}
&((c \to a \to P \mid d \to STOP) \setminus \{c\}) \parallel (a \to R) \\
&\quad = ((a \to P \setminus \{c\}) \sqcap (a \to P \setminus \{c\} \,\square\, d \to STOP)) \parallel (a \to R) && \text{[L10]} \\
&\quad = (a \to P \setminus \{c\}) \parallel (a \to R) \sqcap (a \to P \setminus \{c\} \,\square\, d \to STOP) \parallel (a \to R) \\
&\quad = a \to ((P \setminus \{c\}) \parallel R)
\end{aligned}
$$

- A process which offers the choice between a hidden action c and a nonhidden one d
  - cannot insist that the nonhidden action shall occur.
- If the environment (in this example, $a \to R$) is not prepared for d, then
  - the hidden event must occur, so that
    - the environment has the chance to interact with the resulting process
      - e.g. $(a \to P \setminus \{c\})$.

Nondeterminism: Concealment: **Traces**

- The trace of $P \setminus C$ is obtained from trace t of P by
  - removing all occurrences of any of the symbols in C.

L1. $traces(P \setminus C) = \{t \upharpoonright (\alpha P - C) \mid t \in traces(P)\}$ **if** $\forall t : traces(P) \bullet \neg diverges(P / t, C)$

- *Divergence*
  - $diverges(P,C) = \forall n \bullet \exists s : traces(P) \cap C^* \bullet \#s > n$
    - P diverges immediately on concealment of C
      - it can engage in an unbounded sequence of hidden events.

Nondeterminism: Concealment: **Traces**

- There can be several traces t of P which cannot be distinguished after the concealment
  - $t \upharpoonright (\alpha P - C) = s$, s in $P \setminus C$.
- After s it is not determined which of the possible subsequent behaviours of P defines the subsequent behaviour of $(P \setminus C)$.

L2. $(P \setminus C) \;/\; s = (\sqcap_{t \in T} P \;/\; t) \setminus C$, **where** $T = traces(P) \cap \{t \mid t \upharpoonright (\alpha P - C) = s\}$

if T is finite and $s \in traces(P \setminus C)$

- L1 and L2 are restricted to the case when the process does not diverge.
  - Divergences is never the intended result of the definition of a process.

Nondeterminism: Concealment: Pictures

- *Nondeterministic choice* is represented in a picture by
  - a node from which emerge two or more unlabelled arrows
    - on reaching this node, a process passes along one of the emergent arrows
      - the choice being nondeterministic.
- $P \sqcap Q$:

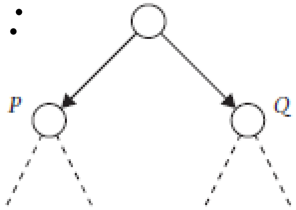

Figure 3.1

- Associativity of $\sqcap$

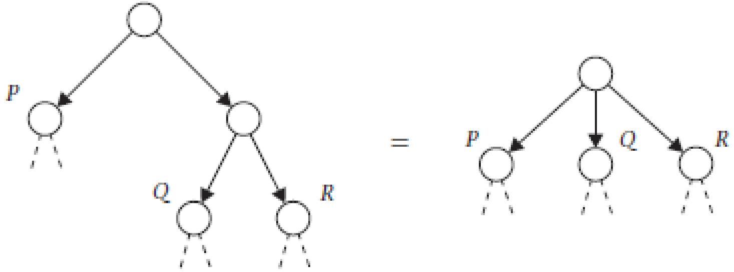

Figure 3.2

L10. If $C \cap B$ is finite and non-empty, **then**
$(x : B \to P(x)) \setminus C = Q \sqcap (Q \,\Box\, (x : (B - C) \to P(x)))$, **where** $Q = \sqcap_{x \in B \cap C} P(x) \setminus C$

Nondeterminism: Concealment: **Pictures**

- *Concealment* removes concealed symbols from all arrows:

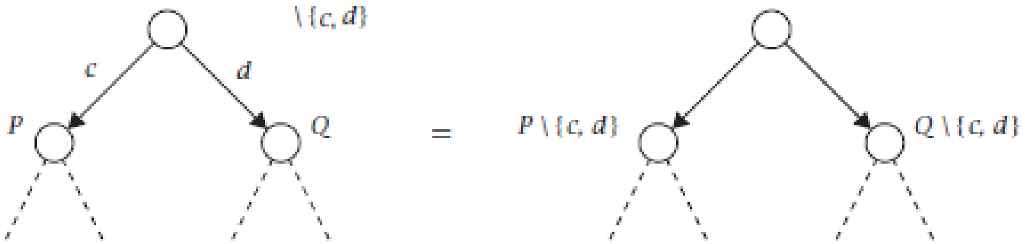

Figure 3.3

- When some arcs of a node are labelled and some are not:
  - By the law L10 such a node can be eliminated:

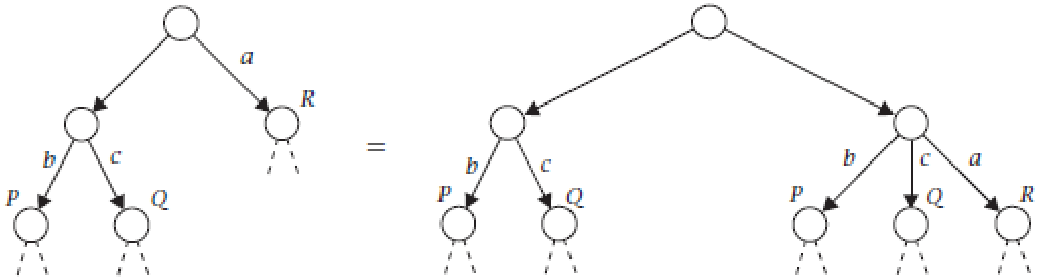

Figure 3.4

L10. If $C \cap B$ is finite and non-empty, **then**
$(x : B \rightarrow P(x)) \setminus C = Q \sqcap (Q \,\Box\, (x : (B - C) \rightarrow P(x)))$, **where** $Q = \sqcap_{x \in B \cap C} P(x) \setminus C$

Nondeterminism: Concealment: Pictures

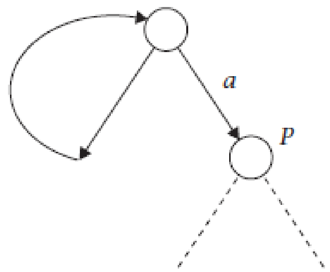

Figure 3.5

- These eliminations are always possible for
  - finite trees.
  - infinite graphs
    - if the graph contains no infinite path of consecutive unlabelled arrows:
      - Such a picture can arise only in the case of divergence.

- It is possible that the node may acquire two emergent lines with the same label
  - by the transformation L10
- Such nodes can be eliminated by the law LX

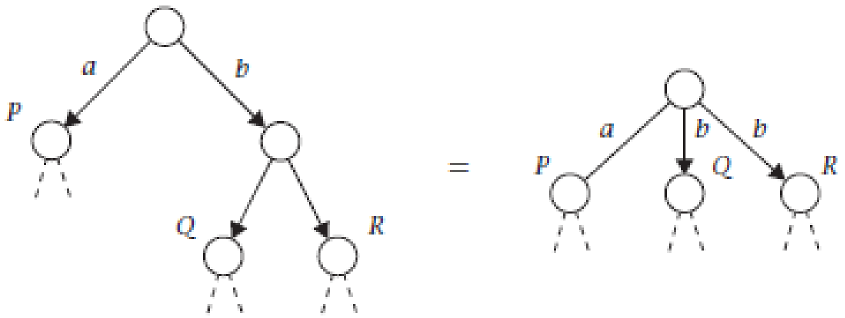

Figure 3.6

LX. $(c \rightarrow P \,\Box\, c \rightarrow Q) = (c \rightarrow P \sqcap c \rightarrow Q)$

Nondeterminism: **Interleaving**

- The concurrency operator $\|$
  - actions in the alphabet of both operands require their simultaneous participation,
  - actions not in the alphabet of both operands occur in an arbitrary interleaving,
  - combines interacting processes with differing alphabets into
    - systems exhibiting concurrent activity, but without nondeterminism.
- *The interleaving operator*
- $P \,|||\, Q$ $\quad\quad$ P interleave Q
  - $\alpha(P \,|||\, Q) = \alpha P = \alpha Q$
  - joins processes with the same alphabet to operate concurrently without
    - directly interacting or synchronising with each other.
  - Each action of the system is an action of exactly one of the processes.
    - If one of the processes cannot engage in the action, the other one must act;
    - if both processes have engaged in the same action,
      - the choice between them is nondeterministic.

Nondeterminism: **Interleaving**

X1. A vending machine accepts up to two coins before dispensing up to two chocolates

- $(VMS \mathbin{|||} VMS) = VMS2$

$$VMS = (coin \rightarrow (choc \rightarrow VMS))$$
$$VMS2 = (coin \rightarrow VMCRED)$$
$$VMCRED = \mu X \bullet (coin \rightarrow choc \rightarrow X \mid choc \rightarrow coin \rightarrow X)$$

X2. A footman made from four lackeys, each serving only one philosopher at a time.

- $FOOT = (LACKEY \mathbin{|||} LACKEY \mathbin{|||} LACKEY \mathbin{|||} LACKEY)$

$$LACKEY = (sits\ down \rightarrow gets\ up \rightarrow LACKEY)$$
$$FOOT_0 = (x : D \rightarrow FOOT_1)$$
$$FOOT_j = (x : D \rightarrow FOOT_{j+1} \mid y : U \rightarrow FOOT_{j-1})$$
$$FOOT_4 = (y : U \rightarrow FOOT_3)$$
$$U = \bigcup_{i=0}^{4}\{i.gets\ up\} \quad D = \bigcup_{i=0}^{4}\{i.sits\ down\}$$

L6. $(a \to P) \parallel (b \to Q) = a \to (P \parallel (b \to Q)) \mid b \to ((a \to P) \parallel Q))$

Nondeterminism: Interleaving: **Laws**

L1–L2. $|||$ is associative and symmetric.

- $|||\ s$ distributes through $\sqcap$:

L3. $(P \sqcap Q) \ ||| \ R = (P \ ||| \ R) \sqcap (Q \ ||| \ R)$

L4. $P \ ||| \ STOP = P$

L5. $P \ ||| \ RUN = RUN$ **if** P does not diverge

L6. $(x \to P) \ ||| \ (y \to Q) = (x \to (P \ ||| \ (y \to Q)) \ \square \ y \to ((x \to P) \ ||| \ Q))$

L7. If $P = (x : A \to P(x))$ **and** $Q = (y : B \to P(y))$

then $P \ ||| \ Q = (x : A \to (P(x) \ ||| \ Q) \ \square \ y : B \to (P \ ||| \ Q(y)))$

- $|||$ does not distribute through $\square$ (let $b \neq c$):

$$
\begin{aligned}
&((a \to STOP) \ ||| \ (b \to Q \ \square \ c \to R)) \ / \ \langle a \rangle \\
&= (b \to Q \ \square \ c \to R) \\
&\neq ((b \to Q) \sqcap (c \to R)) \\
&= ((a \to STOP \ \square \ b \to Q) \ ||| \ (a \to STOP \ \square \ c \to R)) \ / \ \langle a \rangle
\end{aligned}
$$

Nondeterminism: Interleaving: Laws

- On the left-hand side of this chain $(a \to STOP) \mid\mid\mid (b \to Q \,\square\, c \to R)$
  - the occurrence of a can involve progress only of the left operand of $\mid\mid\mid$,
    - no nondeterminism is introduced.
  - The left operand stops, and the choice between b and c is left to the environment.
- On the right-hand side of the chain $(a \to STOP \,\square\, b \to Q) \mid\mid\mid (a \to STOP \,\square\, c \to R)$
  - the event a may be an event of either operand of $\mid\mid\mid$, the choice is nondeterministic.
  - the environment can no longer choose whether the next event will be b or c.

$$
\begin{aligned}
&((a \to STOP) \mid\mid\mid (b \to Q \,\square\, c \to R)) \,/\, \langle a \rangle \\
&\quad = (b \to Q \,\square\, c \to R) \\
&\quad \neq ((b \to Q) \sqcap (c \to R)) \\
&\quad = ((a \to STOP \,\square\, b \to Q) \mid\mid\mid (a \to STOP \,\square\, c \to R)) \,/\, \langle a \rangle
\end{aligned}
$$

- L6 and L7 state that
  - the environment chooses between the initial events offered by the operands of $\mid\mid\mid$.
- Nondeterminism arises only when the chosen event is possible for both operands.

X2. $P = (a \to c \to P)$ and $Q = (c \to b \to Q)$

$(P \parallel Q) \setminus \{c\} = (a \to c \to \mu X \bullet (a \to b \to c \to X \mid b \to a \to c \to X)) \setminus \{c\}$

$= a \to \mu X \bullet (a \to b \to X \mid b \to a \to X)$

Nondeterminism: Interleaving: Laws

X1. Let $R = (a \to b \to R)$, then

$$
\begin{aligned}
&(R \mid\mid\mid R) \\
&\quad = (a \to ((b \to R) \mid\mid\mid R) \;\square\; a \to (R \mid\mid\mid (b \to R))) && \text{[L6]} \\
&\quad = a \to ((b \to R) \mid\mid\mid R) \sqcap (R \mid\mid\mid (b \to R)) \\
&\quad = a \to ((b \to R) \mid\mid\mid R) && \text{[L2]}
\end{aligned}
$$

- Also

$$
\begin{aligned}
&(b \to R) \mid\mid\mid R \\
&\quad = (a \to ((b \to R) \mid\mid\mid (b \to R)) \;\square\; b \to (R \mid\mid\mid R)) && \text{[L6]} \\
&\quad = (a \to (b \to ((b \to R) \mid\mid\mid R)) \square\, b \to (a \to ((b \to R) \mid\mid\mid R))) && \text{[as shown above]} \\
&\quad = \mu X \bullet (a \to b \to X \;\square\; b \to a \to X) && \text{[since the recursion is guarded]}
\end{aligned}
$$

- Thus $(R \mid\mid\mid R)$ is identical to the example X2.
- A similar proof shows that $(VMS \mid\mid\mid VMS) = VMS2$.

L3. $(\langle x\rangle ^\frown s)\ interleaves\ (t, u) \equiv$
$(t \neq \langle\rangle \land t_0 = x \land s\ interleaves\ (t, u)) \lor (u \neq \langle\rangle \land u_0 = x \land s\ interleaves\ (t, u))$

Nondeterminism: Interleaving: Traces and refusals

- A trace of $(P \parallel\mid Q)$ is an arbitrary interleaving of a trace from P with a trace from Q.

L1. $traces(P \parallel\mid Q) = \{\ s \mid \exists t : traces(P) \bullet \exists u : traces(Q) \bullet s\ interleaves\ (t, u)\ \}$

- $(P \parallel\mid Q)$ can engage in any initial action possible for either P or Q
  - it can refuse only those sets which are refused by both P and Q:

L2. $refusals(P \parallel\mid Q) = refusals(P \Box Q)$

- The behaviour of $(P \parallel\mid Q)$ after the trace s:

L3. $(P \parallel\mid Q)\ /\ s = \ \sqcap_{t,u \in T}\ (P\ /\ t) \parallel\mid (Q\ /\ u),$

where $T = \{(t, u) \mid t \in traces(P) \land u \in traces(Q) \land s\ interleaves\ (t, u)\}$

- There is no way of knowing the structure of a trace s of $(P \parallel\mid Q)$ as an interleaving of a trace from P and a trace from Q;
  - after s, the future behaviour of $(P \parallel\mid Q)$ may reflect any of the possible interleavings.
    - The choice between them is not known and not determined.

Nondeterminism: **Specifications**

- Specification of indirectly observable aspects of the behaviour.
  - describe the desired properties of process refusal sets as well as its traces.
- The variable *ref* denotes an arbitrary refusal set of a process.
- Specification of nondeterministic process P:
  - P sat $S(tr, ref)$ iff
    - $\forall tr, ref \bullet tr \in traces(P) \land ref \in refusals(P / tr) \Rightarrow S(tr, ref)$

$$VMS = (coin \rightarrow (choc \rightarrow VMS))$$
$$VMCRED = \mu X \bullet (coin \rightarrow choc \rightarrow X \mid choc \rightarrow coin \rightarrow X)$$
$$VMS2 = (coin \rightarrow VMCRED)$$

Nondeterminism: Specifications

X1. When a vending machine has ingested more coins than it has dispensed chocolates,

- it must not refuse to dispense a chocolate

$$FAIR = (tr \downarrow choc < tr \downarrow coin \Rightarrow choc \notin ref)$$

- *Every* trace *tr* and *every* refusal *ref* of the specified process *at all times*
  - should satisfy the specification.

X2. When a vending machine has given out as many chocolates as have been paid for

- it must not refuse a further coin

$$PROFIT1 = (tr \downarrow choc = tr \downarrow coin \Rightarrow coin \notin ref)$$

X3. A simple vending machine should satisfy the combined specification

$$NEWVMSPEC = FAIR \wedge PROFIT \wedge (tr \downarrow choc \leq tr \downarrow coin)$$

- This specification is satisfied by *VMS* and *VMS2*
  - may accept several coins in a row, and then give out several chocolates.

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

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

$VMS2 = (coin \rightarrow VMCRED)$

Nondeterminism: **Specifications**

$NEWVMSPEC = FAIR \wedge PROFIT \wedge (tr \downarrow choc \leq tr \downarrow coin)$

X4. A limit on the balance of coins which may be accepted in a row

$$ATMOST2 = (tr \downarrow coin - tr \downarrow choc \leq 2)$$

X5. The machine accept at least two coins in a row:

$$ATLEAST2 = (tr \downarrow coin - tr \downarrow choc < 2 \Rightarrow coin \notin ref)$$

X6. The process *STOP* refuses every event in its alphabet.

- The predicate specifies that a process with alphabet A never stops
  - $NONSTOP = (ref \neq A)$
- If P sat $NONSTOP$ and an environment allows all events in A,
  - P must perform one of them.
- Since any process which satisfies $NEWVMSPEC$ will never stop.

$$NEWVMSPEC \Rightarrow ref \neq \{coin, choc\}$$

$NEWVMSPEC = FAIR \wedge PROFIT \wedge (tr \downarrow choc \leq tr \downarrow coin)$

Nondeterminism: **Specifications**

- A *divergent process* can do anything and refuse anything.
  - if there is a set which *cannot* be refused, then the process is not divergent.
    - A sufficient condition for non-divergence
      - $NONDIV = (ref \neq A)$
- Proof of absence of divergence is not more complex than proof of absence of deadlock.
  - $NONSTOP \equiv NONDIV$

Nondeterminism: Specifications: **Proofs**

- The notation:
  - a specification – S, $S(tr)$, $S(tr, ref)$,
    - tr and ref are its free variables.

- $(P \sqcap Q)$ behaves either like P or like Q.
  - Every observation of its behaviour is an observation possible for P or for Q or for both.
    - described by the specification of P or by the specification of Q or by both.
- The proof rule for nondeterminism:

L1. If P sat S **and** Q sat T **then** $(P \sqcap Q)$ sat $(S \vee T)$

L4B. If P sat $S(tr)$ **then** $(c \rightarrow P)$ sat $(tr = \langle\rangle \lor (tr_0 = c \land S(tr')))$

Nondeterminism: Specifications: **Proofs**

- The proof rule for $STOP$ states that it does nothing and refuses anything:

L2A. $STOP_A$ sat $(tr = \langle\rangle \land ref \subseteq A)$

  - The clause $ref \subseteq A$ can be omitted.
  - if we omit the alphabet, this law is identical to that for deterministic processes:
    - $STOP$ sat $tr = \langle\rangle$

- The law for prefixing extends the one for deterministic processes:

L2B. If P sat $S(tr)$ **then** $(c \rightarrow P)$ sat $((tr = \langle\rangle \land c \notin ref) \lor (tr_0 = c \land S(tr')))$

  - In the initial state, when $tr = \langle\rangle$, the initial action cannot be refused

- The law for general choice is similarly strengthened:

L2. If $\forall x : B \bullet P(x)$ sat $S(tr, x)$

then $(x : B \rightarrow P(x))$ sat $((tr = \langle\rangle \land (B \cap ref = \{\}) \lor (tr_0 \in B \land S(tr', tr_0))))$

Nondeterminism: Specifications: **Proofs**

- The law for parallel composition deals correctly with refusals:

L3. If P sat $S(tr, ref)$ **and** Q sat $T(tr, ref)$ **and** neither P **nor** Q diverges

then $(P \parallel Q)$ sat $(\exists\, X, Y, ref \bullet ref = (X \cup Y) \wedge S(tr \upharpoonright \alpha P, X) \wedge T(tr \upharpoonright \alpha Q, Y))$

- The law for change of symbol needs a similar adaptation:

L4. If P sat $S(tr, ref)$ **then** $f(P)$ sat $S(f^{-1*}(tr), f^{-1}(ref))$ **if** f is one-one.

- The law for $\square$:

L5. If P sat $S(tr, ref)$ **and** Q sat $T(tr, ref)$ **and** neither P **nor** Q diverges

then $(P \square Q)$ sat (if $tr = \langle\rangle$ then $(S \wedge T)$ else $(S \vee T)$)

- If $tr = \langle\rangle$, a set is refused by $(P \square Q)$ only if it is refused by both P and Q.
  - This set must be described by *both* their specifications.
- If $tr \neq \langle\rangle$, each observation of $(P \square Q)$ must be an observation either of P or of Q.
  - It must be described by one of their specifications (or both).

$diverges(P,C) = \forall n \bullet \exists s : traces(P) \cap C^* \bullet \#s > n$

Nondeterminism: Specifications: **Proofs**

- The law for interleaving:

L6. If P sat $S(s)$ **and** Q sat $T(t)$ **and** neither P **nor** Q diverges

then $(P \mid\mid\mid Q)$ sat $(\exists\, s, t \bullet (tr \textit{ interleaves } (s, t) \wedge S(s) \wedge T(t)))$

- The law for concealment:

L7. If P sat $(NODIV \wedge S(tr, ref))$

then $(P \setminus C)$ sat $\exists\, s \bullet tr = s \restriction (\alpha P - C) \wedge S(tr, ref \cup C)$

  - $NODIV$ states that the number of hidden symbols that can occur is bounded by some function of the non-hidden symbols that have occurred
    - $NODIV = \#(tr \restriction C) \leq f(tr \restriction (\alpha P - C))$
      - f is a total function from traces to natural numbers.
  - $P \setminus C$ can refuse a set X only when P can refuse the whole set $X \cup C$.
  - $P \setminus C$ cannot refuse to interact with its external environment until it has reached a state in which it cannot engage in any further concealed internal activities.
- This kind of fairness is an important feature of a reasonable definition of concealment.

L6. If $F(x)$ is guarded **and** $STOP$ sat S **and** $((X \text{ sat } S) \Rightarrow (F(X) \text{ sat } S))$ **then** $(\mu X \bullet F(X))$ sat S

Nondeterminism: Specifications: **Proofs**

- The proof method for deterministic recursion is strengthened:

L8. If $S(0)$ **and** $(X \text{ sat } S(n)) \Rightarrow f(X) \text{ sat } S(n+1))$ **then** $(\mu X \bullet f(X))$ sat $(\forall n \bullet S(n))$

  - $S(n)$ is a predicate with the variable $n \in \mathbb{N}$.
  - This law is valid even for an unguarded recursion
    - The strongest specification which can be proved for an unguarded recursion process is the vacuous specification *true*.

Nondeterminism: **Divergence**

- Consider the infinite recursion $\mu X.X$
  - Every process is a solution of the recursive equation X=X.
  - $\mu X.X$ may behave like any process
    - the most nondeterministic, the least predictable, the least controllable, the worst.
  - $CHAOS_A = \mu X{:}A.X$.
    - A slightly better case is $\mu X.(c \rightarrow (X \setminus \{c\})) = c \rightarrow CHAOS$.
- $CHAOS$ is also result of
  - engaging a process in an infinite sequence of consecutive hidden events:

$$
\begin{aligned}
&(\mu X : A \bullet (c \rightarrow X)) \setminus \{c\} \\
&\quad = \mu X : (A - \{c\}) \bullet ((c \rightarrow X) \setminus \{c\}) \\
&\quad = \mu X : (A - \{c\}) \bullet (X \setminus \{c\}) && \text{[by L12, L5]} \\
&\quad = \mu X : (A - \{c\}) \bullet X \\
&\quad = CHAOS_{A - \{c\}} && \text{def. } CHAOS.
\end{aligned}
$$

Nondeterminism: Divergence: **Laws**

- *CHAOS* is the most nondeterministic process
  - it cannot be changed by adding yet nondeterministic choices.
- *CHAOS* is a zero of $\sqcap$

L1. $P \sqcap CHAOS = CHAOS$

- A function of processes is *strict* iff
  - It gives *CHAOS* if any of its arguments is *CHAOS*.

L2. The following operations are strict $/s$, $\|$, f, $\square$, $\backslash C$, $|||$, and $\mu\, X$ **I** ?

- Prefixing is not strict:

L3. $CHAOS \neq (a \rightarrow CHAOS)$

  - The right-hand side relies upon to do *a* before becoming completely unreliable.
- There is nothing that *CHAOS* might not do:

L4. $traces(CHAOS_A) = A^*$

- There is nothing that *CHAOS* might not refuse to do:

L5. $refusals(CHAOS_A)$ = all subsets of A.

Nondeterminism: Divergence: **Divergences**

- A *divergence* of a process is
  - any trace of the process after which the process behaves chaotically.
- The set of all divergences is
  - $divergences(P) = \{s \mid s \in traces(P) \land (P \,/\, s) = CHAOS_{aP}\}$

L1. $divergences(P) \subseteq traces(P)$

- The divergences of a process are extension-closed:

L2. $s \in divergences(P) \land t \in (aP)^* \Rightarrow (s^\frown t) \in divergences(P)$

- Because $/\, t$ is strict and $CHAOS \,/\, t = CHAOS$

- $CHAOS_A$ may refuse any subset of its alphabet A

L3. $s \in divergences(P) \land X \subseteq aP \Rightarrow X \in refusal(P \,/\, s)$

Nondeterminism: Divergence: **Divergences**

- The laws show for the divergences of compound processes.
- Firstly, the process *STOP* never diverges:

L4. $divergences(STOP) = \{\}$

- Every trace of *CHAOS* leads to *CHAOS*:

L5. $divergences(CHAOS_A) = A^*$

- For a process defined by choice, the divergences are determined by what happens after the first step:

L6. $divergences(x : B \rightarrow P(x)) = \{<x> \wedge s \mid x \in B \wedge s \in divergences(P(x))\}$

- Any divergence of P is also a divergence of $(P \sqcap Q)$ and of $(P \square Q)$:

L7. $divergences(P \sqcap Q) = divergences(P \square Q) = divergences(P) \cup divergences(Q)$

Nondeterminism: Divergence: **Divergences**

- Since $\|$ is strict, a divergence of $(P \| Q)$ starts with a trace of the nondivergent activity of both P and Q, which leads to divergence of either P or of Q (or of both):

L8. $divergences(P \| Q) = \{ s \wedge t \mid t \in (aP \cup aQ)^* \wedge$

$((s \upharpoonright aP \in divergences(P) \wedge s \upharpoonright aQ \in traces(Q)) \vee (s \upharpoonright aP \in traces(P) \wedge s \upharpoonright aQ \in divergences(Q)))\}$

- A similar explanation for $|||$:

L9. $divergences(P \mid\mid\mid Q) = \{ u \mid \exists s, t \bullet u \text{ interleaves } (s, t) \wedge$

$((s \in divergences(P) \wedge t \in traces(Q)) \vee (s \in traces(P) \wedge t \in divergences(Q)))\}$

- Divergences of a process with concealment include traces
  - derived from the original divergences,
  - plus those resulting from the attempt to conceal an infinite sequence of symbols:

L10. $divergences(P \setminus C) = \{ (s \upharpoonright (aP - C)) \wedge t \mid t \in (aP - C)^* \wedge$

$(s \in divergences(P) \vee (\forall n \bullet \exists u \in C^* \bullet \#u > n \wedge (s \wedge u) \in traces(P)))\}$

Nondeterminism: Divergence: **Divergences**

- A process defined by symbol change diverges only when its argument diverges

L11. $divergences(f(P)) = \{f^*(s) \mid s \in divergences(P)\}$ **if** f is one-one.

- Why divergence, if divergence is always something we do *not* want?
  - A consequence of any efficient or even computable method of implementation.
  - It can arise from either *concealment* or *unguarded recursion*.
  - A system designer *must prove* that for his particular design *the problem will not occur*.
    - In order *to prove* that something can't happen
      - we need to use *a mathematical theory* in which it can.