◇ Traces and Choice ◇

Which traces can be produced by $P \mathrel{\Box} Q$ and $P \sqcap Q$? We know that $P \mathrel{\Box} Q$ can do the first event of either P or Q, and then behave like the remainder of P or Q. Therefore any trace of either P or Q can be produced by $P \mathrel{\Box} Q$, and we have

$$traces(P \mathrel{\Box} Q) = traces(P) \cup traces(Q).$$

$P \sqcap Q$ always does τ first, and then behaves like either P or Q. Because τ does not appear in traces, we also have

$$traces(P \sqcap Q) = traces(P) \cup traces(Q).$$

We have previously considered *trace equivalence*, written $P =_T Q$, as a definition of when two processes should be considered equal or interchangeable. However, we can now see that $P \mathrel{\Box} Q =_T P \sqcap Q$, even though internal and external choice have been designed to behave in different ways.

In general, trace equivalence is not suitable as a definition of process equivalence.

Before we introduced $\sqcap$ and $\Box$ all processes were deterministic — the internal state was always determined by the observable events. For deterministic processes, *traces* are all we need to know, and trace equivalence is adequate. But the whole point of introducing the $\sqcap$ operator was so that a process could make an internal state change without doing anything observable. Similarly, if P and Q have a common event a available at the first step, then observation of the event a from $P \mathrel{\Box} Q$ does not tell us what the internal state has become.

We will now try to say exactly what the difference between $P \sqcap Q$ and $P \mathrel{\Box} Q$ is, and develop a new notion of process equivalence accordingly.

◇ Refusals ◇

Suppose we have the following definitions.

$$\begin{aligned} P &= a \to P \\ Q &= b \to Q \end{aligned}$$

What happens if we put each of $P \mathrel{\Box} Q$ and $P \sqcap Q$ in an environment consisting of P? i.e. if we look at $(P \mathrel{\Box} Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P$ and $(P \sqcap Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P$.

First, we have $P \mathrel{\Box} Q \xrightarrow{a} P$
and $P \xrightarrow{a} P$
so

$$(P \mathrel{\Box} Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P \xrightarrow{a} P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P.$$

Also,

$$P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P \xrightarrow{a} P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P$$

so

$$P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P = P$$

(they both satisfy the same recursive definition).

So

$$(P \mathrel{\Box} Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P = a \to P$$

i.e.

$$(P \mathrel{\Box} Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P = P.$$

On the other hand,

$$(P \sqcap Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P \xrightarrow{\tau} P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P$$

and

$$(P \sqcap Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P \xrightarrow{\tau} Q\ {}_{\{a,b\}}\|_{\{a,b\}}\ P$$

so

$$(P \sqcap Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P =$$

$$(P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P) \sqcap (Q\ {}_{\{a,b\}}\|_{\{a,b\}}\ P).$$

(This is a loose statement as we haven't decided what "$=$" means yet.)

We know that $P\ {}_{\{a,b\}}\|_{\{a,b\}}\ P = P$
and $Q\ {}_{\{a,b\}}\|_{\{a,b\}}\ P = STOP$
So

$$(P \sqcap Q)\ {}_{\{a,b\}}\|_{\{a,b\}}\ P = P \sqcap STOP.$$

This shows that $P \mathrel{\Box} Q$ and $P \sqcap Q$ behave differently when put in parallel with P. One is just P, the other can internally choose to deadlock (become $STOP$).

We can use this observation to develop a general approach to distinguishing between nondeterministic processes. We will consider putting a process P in an environment Q, where the alphabets of P and Q are the same, i.e. constructing $P\ {}_{\alpha(P)}\|_{\alpha(P)}\ Q$.

Let X be a set of events which are offered initially by Q. If it is possible for $P\ {}_{\alpha(P)}\|_{\alpha(P)}\ Q$ to deadlock at the first step, then we say that X is a *refusal* of P. The set of all refusals of P is obtained by considering all possible sets X which could be initial event sets of Q.

Examples:

1. The empty set is a refusal of every process, because if $Q = STOP$ then $P\ {}_{\alpha(P)}\|_{\alpha(P)}\ Q = STOP$.

2. Any set of events X is a refusal of $STOP$.

3. If $a \notin X$ then X is a refusal of $a \to P$. So if $\alpha(P) = \{a, b, c\}$ then the refusals of $a \to P$ are $\{\}$, $\{b\}$, $\{c\}$ and $\{b, c\}$. Processes Q causing

$$(a \to P)\ {}_{\{a,b,c\}}\|_{\{a,b,c\}}\ Q$$

to deadlock include $STOP$, $b \to STOP$, $c \to a \to STOP$, $(b \to STOP) \Box (c \to c \to STOP)$, etc.

4. The refusals of $(a \to c \to STOP) \Box (b \to STOP)$ are $\{\}$ and $\{c\}$.

5. The refusals of $(a \to c \to STOP) \sqcap (b \to STOP)$ are $\{\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, c\}$ and $\{b, c\}$.

We can define

$$refusals(P) = \{X \mid X \subseteq \alpha(P) \text{ and } X \text{ is a refusal of } P\}.$$

Note that $refusals(P)$ is a set of sets of events. For example,
$refusals((a \to STOP) \sqcap (b \to STOP)) = \{\{\}, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}$.

In the examples we saw that
$refusals((a \to STOP) \Box (b \to STOP)) \neq refusals((a \to STOP) \sqcap (b \to STOP))$.

In general, $refusals(P \Box Q) \neq refusals(P \sqcap Q)$, and this will be the basis for a new definition of process equality which allows us to distinguish between internal and external choice.

We can now define $refusals$ for processes defined in terms of the operators we have seen so far.

$$refusals(STOP) = \{X \mid X \subseteq \Sigma\}$$

where Σ is the set of all events being considered — the universal set of events.

$$refusals(a \to P) = \{X \mid X \subseteq (\alpha(P) - \{a\})\}$$

Both of these definitions are subsumed by the definition for menu choice: if $P = x : A \to P(x)$ then

$$refusals(P) = \{X \mid X \subseteq (\alpha(P) - A)\}$$

If P can refuse X then so will $P \sqcap Q$ if P is selected. Similarly every refusal of Q is a possible refusal of $P \sqcap Q$.

$$refusals(P \sqcap Q) = refusals(P) \cup refusals(Q)$$

$P \Box Q$ can only refuse X if both P and Q can refuse X.

$$refusals(P \Box Q) = refusals(P) \cap refusals(Q)$$

$P\ {}_A\|_B\ Q$ can refuse all events refused by P and all events refused by Q.

$$refusals(P\ {}_A\|_B\ Q) = \{X \cup Y \mid X \in refusals(P) \text{ and } Y \in refusals(Q)\}$$

Refusals allow us to distinguish formally between deterministic and nondeterministic processes. If a process is deterministic then it can never refuse any event which it could possibly do. In other words, if P is deterministic and a is a possible initial event for P, then a does not appear in any refusal set of P.

Writing $initials(P)$ for the set of possible initial events of P (so $initials(P) = \{x \mid \langle x \rangle \in traces(P)\}$), we can say that if P is deterministic then

$$refusals(P) = \{X \mid X \subseteq \alpha(P) \text{ and } X \cap initials(P) = \{\}\}.$$

Determinism means that any event which is possible cannot be taken away by an internal state transition.

Examples: If

$$P = a \to c \to STOP \mid b \to STOP$$

then $initials(P) = \{a, b\}$ and $refusals(P) = \{\{\}, \{c\}\}$.

If

$$P = (a \to c \to STOP) \sqcap (b \to STOP)$$

then $initials(P) = \{a, b\}$ and (as before)

$$refusals(P) = \{\{\}, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}.$$

Although a is a possible initial event for P, P could also internally choose to be $b \to STOP$ which refuses a.

To define nondeterminism properly, we need to consider events refused not just at the first step, but after any sequence of events. For example,

$$(a \to b \to STOP) \Box (a \to c \to STOP)$$

is nondeterministic, but this does not become apparent until after the first event.

So: P is deterministic if and only if
$$\forall\, tr \in traces(P) \bullet (refusals(P/tr) = \{X \subseteq \alpha(P) \mid X \cap initials(P/tr) = \{\}\}).$$

P/tr is the process whose behaviour is whatever P could do after the trace tr.