

◇ Traces and Choice ◇

Which traces can be produced by $P \sqcap Q$ and $P \sqcup Q$? We know that $P \sqcap Q$ 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 $P \sqcap Q$, and we have

$$\text{traces}(P \sqcap Q) = \text{traces}(P) \cup \text{traces}(Q).$$

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

$$\text{traces}(P \sqcup Q) = \text{traces}(P) \cup \text{traces}(Q).$$

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 $P \sqcap Q =_t P \sqcup Q$, 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 \sqcup 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 $P \sqcap Q$ does not tell us what the internal state has become.

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

◇ Refusals ◇

Suppose we have the following definitions.

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

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

First, we have $P \sqcap Q \xrightarrow{a} P$
and $P \xrightarrow{a} P$

so

$$(P \sqcap Q)_{\{a,b\}} \parallel_{\{a,b\}} P \xrightarrow{a} P_{\{a,b\}} \parallel_{\{a,b\}} P.$$

Also,

$$P_{\{a,b\}} \parallel_{\{a,b\}} P \xrightarrow{a} P_{\{a,b\}} \parallel_{\{a,b\}} P$$

so

$$P_{\{a,b\}} \parallel_{\{a,b\}} P = P$$

(they both satisfy the same recursive definition).

So

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

i.e.

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

On the other hand,

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

and

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

so

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

$$(P_{\{a,b\}} \parallel_{\{a,b\}} P) \sqcap (Q_{\{a,b\}} \parallel_{\{a,b\}} P).$$

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

We know that $P_{\{a,b\}} \parallel_{\{a,b\}} P = P$

and $Q_{\{a,b\}} \parallel_{\{a,b\}} P = \text{Stop}$

So

$$(P \sqcap Q)_{\{a,b\}} \parallel_{\{a,b\}} P = P \sqcap \text{Stop}.$$

This shows that $P \sqcap Q$ 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 $P_{\alpha P} \parallel_{\alpha P} Q$.

Let X be a set of events which are offered initially by Q . If it is possible for $P \alpha P \parallel_{\alpha P} Q$ 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 $Q = \text{Stop}$ then $P \alpha P \parallel_{\alpha P} Q = \text{Stop}$.

2. Any set of events X is a refusal of Stop .

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

$$(a \rightarrow P) \{a, b, c\} \parallel_{\{a, b, c\}} Q$$

to deadlock include Stop , $b \rightarrow \text{Stop}$, $c \rightarrow a \rightarrow \text{Stop}$, $(b \rightarrow \text{Stop}) \square (c \rightarrow c \rightarrow \text{Stop})$, etc.

4. The refusals of $(a \rightarrow c \rightarrow \text{Stop}) \square (b \rightarrow \text{Stop})$ are $\{\}$ and $\{c\}$.

5. The refusals of $(a \rightarrow c \rightarrow \text{Stop}) \sqcap (b \rightarrow \text{Stop})$ are $\{\}$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, c\}$ and $\{b, c\}$.

We can define

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

Note that $\text{refusals}(P)$ is a set of sets of events. For example,

$$\text{refusals}((a \rightarrow \text{Stop}) \sqcap (b \rightarrow \text{Stop})) = \{\{\}, \{a\}, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}.$$

In the examples we saw that

$$\text{refusals}((a \rightarrow \text{Stop}) \square (b \rightarrow \text{Stop})) \neq \text{refusals}((a \rightarrow \text{Stop}) \sqcap (b \rightarrow \text{Stop})).$$

In general, $\text{refusals}(P \square Q) \neq \text{refusals}(P \sqcap Q)$, 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.

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

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

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

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

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

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 $P \sqcap Q$.

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

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

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

$P _A \parallel_A Q$ can refuse all events refused by P and all events refused by Q .

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

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

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

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

Examples: If

$$P = a \rightarrow c \rightarrow \text{Stop} \mid b \rightarrow \text{Stop}$$

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

If

$$P = (a \rightarrow c \rightarrow \text{Stop}) \sqcap (b \rightarrow \text{Stop})$$

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

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

Although a is a possible initial event for P , P could also internally choose to be $b \rightarrow \text{Stop}$ 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,

$$(a \rightarrow b \rightarrow \text{Stop}) \sqcap (a \rightarrow c \rightarrow \text{Stop})$$

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

So: P is deterministic if and only if

$\forall s \in \text{traces}(P)$.

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

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