Department of Computer Science
University of Bristol

COMS20001 - Concurrent Computing

www.cs.bris.ac.uk/Teaching/Resources/COMS20001

Lecture 08

CSP: Choice, Refusals, Failures

Sion Hannuna | hannuna@cs.bris.ac.uk
Tilo Burghardt| tilo@cs.bris.ac.uk
Dan Page | daniel.page@bristol.ac.uk

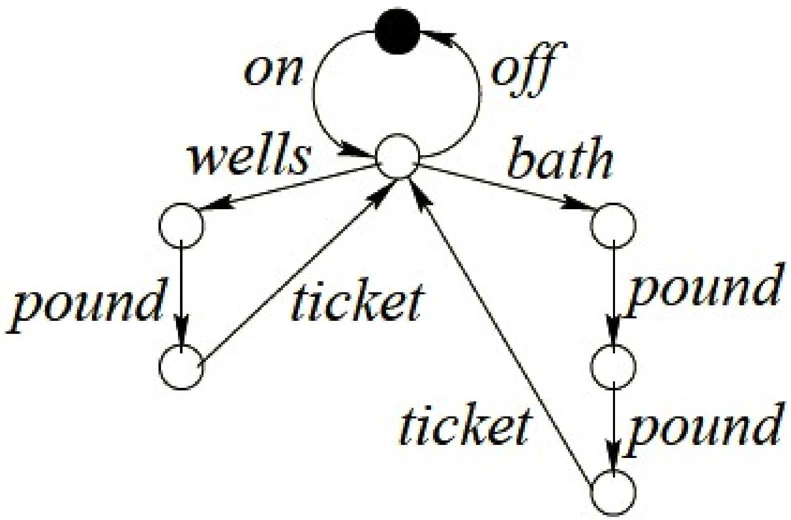
23 Slides

Recap: Processes and Traces

Connection between transition diagram of a process, and its traces.

- $$\begin{aligned} MACHINE &= on \rightarrow TICKETS \\ TICKETS &= wells \rightarrow pound \rightarrow ticket \rightarrow TICKETS \\ &\mid\ bath \rightarrow pound \rightarrow pound \rightarrow ticket \rightarrow TICKETS \\ &\mid\ off \rightarrow MACHINE \end{aligned}$$

Transition diagram:



- $traces(MACHINE)$ is the set of traces corresponding to the paths in the diagram starting from the filled-in (black or white) state.

Recap: Students & Colleges…

$$\begin{aligned}
STUDENT &= yr1 \rightarrow (pass \rightarrow YEAR2 \mid fail \rightarrow STUDENT) \\
YEAR2 &= yr2 \rightarrow (pass \rightarrow YEAR3 \mid fail \rightarrow YEAR2) \\
YEAR3 &= yr3 \rightarrow (pass \rightarrow graduate \rightarrow STOP \mid fail \rightarrow YEAR3)
\end{aligned}$$

$$\begin{aligned}
COLLEGE &= fail \rightarrow STOP \mid pass \rightarrow C1 \\
C1 &= fail \rightarrow STOP \mid pass \rightarrow C2 \\
C2 &= fail \rightarrow STOP \mid pass \rightarrow prize \rightarrow STOP
\end{aligned}$$

Combine student and college: $SYSTEM = STUDENT\ {}_{S}\|_{C}\ COLLEGE$ where

$$S = \{yr1, yr2, yr3, pass, graduate, fail\}$$

$$C = \{pass, fail, prize\}$$

- Which events do student and college synchronise on?
- What happens if the student fails?
- NOTE: $COLLEGE$ stops after $fail$!

Recap: Specification via Trace Refinement

$$
\begin{aligned}
STUDENT &= yr1 \rightarrow (pass \rightarrow YEAR2 \mid fail \rightarrow STUDENT) \\
YEAR2 &= yr2 \rightarrow (pass \rightarrow YEAR3 \mid fail \rightarrow YEAR2) \\
YEAR3 &= yr3 \rightarrow (pass \rightarrow graduate \rightarrow STOP \mid fail \rightarrow YEAR3)
\end{aligned}
$$

$$
\begin{aligned}
COLLEGE &= fail \rightarrow STOP \mid pass \rightarrow C1 \\
C1 &= fail \rightarrow STOP \mid pass \rightarrow C2 \\
C2 &= fail \rightarrow STOP \mid pass \rightarrow prize \rightarrow STOP
\end{aligned}
$$

Combine student and college: $SYSTEM = STUDENT\ {}_{S}\|_{C}\ COLLEGE$

$$
\begin{aligned}
SPECP &= pass \rightarrow S1 \mid fail \rightarrow SPECF \\
S1 &= pass \rightarrow S2 \mid fail \rightarrow SPECF \\
S2 &= pass \rightarrow prize \rightarrow STOP \mid fail \rightarrow SPECF \\
SPECF &= pass \rightarrow SPECF \mid fail \rightarrow SPECF
\end{aligned}
$$

- Are all traces of $SYSTEM$ covered by $SPECP$?

Process $EXTRA = x : E \rightarrow EXTRA$ with $E = \{yr1, yr2, yr3, graduate\}$

Extended specification: $SPEC = SPECP\ {}_{SP}\|_{E}\ EXTRA$

Is $SPEC \sqsubseteq_T SYSTEM$ satisfied?

Students and Parents

Process $STUDENT$ has alphabet: $S = \{yr1, yr2, yr3, pass, graduate, fail\}$

$$\begin{aligned} STUDENT &= yr1 \rightarrow (pass \rightarrow YEAR2 \mid fail \rightarrow STUDENT) \\ YEAR2 &= yr2 \rightarrow (pass \rightarrow YEAR3 \mid fail \rightarrow YEAR2) \\ YEAR3 &= yr3 \rightarrow (pass \rightarrow graduate \rightarrow STOP \mid fail \rightarrow YEAR3) \end{aligned}$$

Some students have generous parents, who buy a present every time a student passes the exams.

$$PARENT = pass \rightarrow present \rightarrow PARENT$$

with $\alpha(PARENT) = \{pass, present\} = P$

How many "states" has a student? How many "states" has a parent?

In parallel combination $STUDENT \;{}_S\|_P\; PARENT$ only event $pass$ needs synchronisation!

Cardinality of State Spaces

- Make a transition diagram for $STUDENT \; {}_S\|_P \; PARENT$.

- After the student has passed an exam, events $present$ and next year ($yr?$) can happen in either order!

How many states?

- Process P and Q completely **independent** (i.e. $A \cap B = \{\}$), number of states of combined process $P \; {}_A\|_B \; Q$ is product of number of states in P and number of states in Q.
- No longer true if processes must synchronise on some events!

- $STUDENT$ has 8 states, $PARENT$ has 2 states, parallel combination has 14 states!

Traces and Prefix Closure

$\alpha(VM) = \{coin, choc, beep\} = A \qquad \alpha(CUST) = \{coin, choc, shout\} = B$

$$\begin{aligned} VM &= coin \rightarrow beep \rightarrow choc \rightarrow STOP \\ CUST &= coin \rightarrow shout \rightarrow choc \rightarrow STOP \end{aligned}$$

What are the traces of $VM_A \|_B CUST$?

$$\begin{aligned} &traces(VM_A \|_B CUST) = \\ &\quad \{\langle\rangle, \langle coin\rangle, \langle coin, beep\rangle, \langle coin, shout\rangle, \\ &\quad \langle coin, beep, shout\rangle, \langle coin, shout, beep\rangle, \\ &\quad \langle coin, beep, shout, choc\rangle, \langle coin, shout, beep, choc\rangle\} \end{aligned}$$

NOTE: If a process can be observed to perform a sequence of events, it can also be observed to *perform any prefix of that sequence*.

MORE FORMALLY: If $tr_1 \frown tr_2 \in traces(P)$ then $tr_1 \in traces(P)$.

- Sets of traces are **prefix closed**.

traces(): Potentially **Infinite** Sets of Finite Traces

We only consider *finite* traces.

Processes which are defined *without recursion* have a *bound* on the length of their trace.

- $PHONE = ring \rightarrow answer \rightarrow STOP$

  Traces of $PHONE$ are: $\{\langle\rangle, \langle ring\rangle, \langle ring, answer\rangle\}$

Recursive processes can perform events forever.

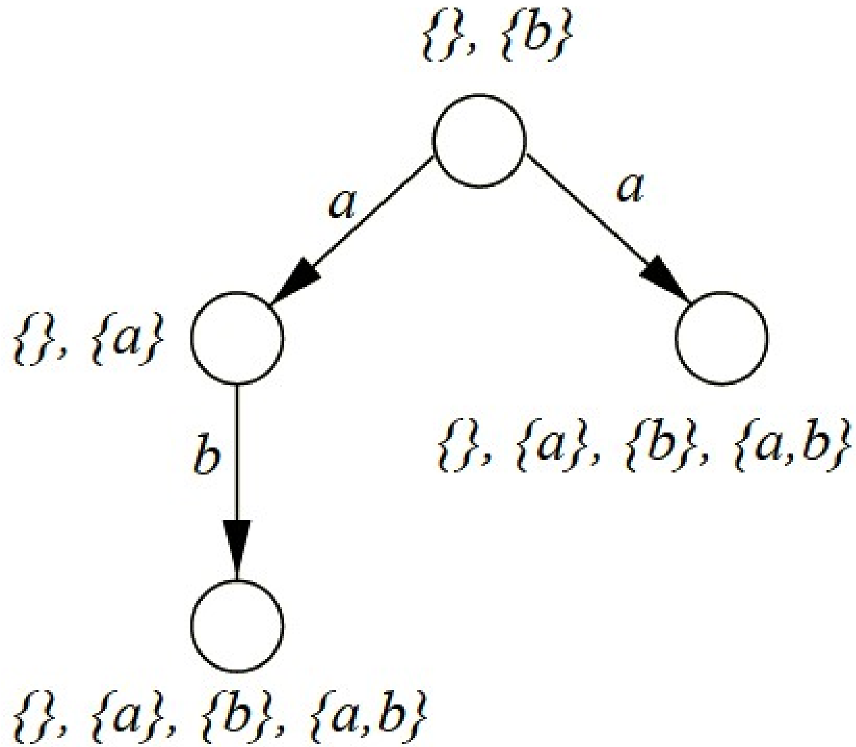
- $CLOCK = tick \rightarrow CLOCK$

  Traces of $CLOCK$ are: $\{\langle\rangle, \langle tick\rangle, \langle tick, tick\rangle, \langle tick, tick, tick\rangle, \ldots\}$

- *Recursive processes can have an* infinite *set of traces.*

It is important to understand that we are interested in potentially *infinite sets* of *finite* traces.

What could be the meaning of the annotated sets?



Refusals: Event sets leading to immediate deadlock

Put a process P into an environment ENV, where the alphabets of P and ENV are the same, e.g. $P \;_{\alpha(P)}\|_{\alpha(P)}\; ENV$.

- Let X be the set of events which are offered initially by ENV.
- If it is possible for $P \;_{\alpha(P)}\|_{\alpha(P)}\; ENV$ to deadlock at the first step, then we say that the set X is a *refusal* of P.

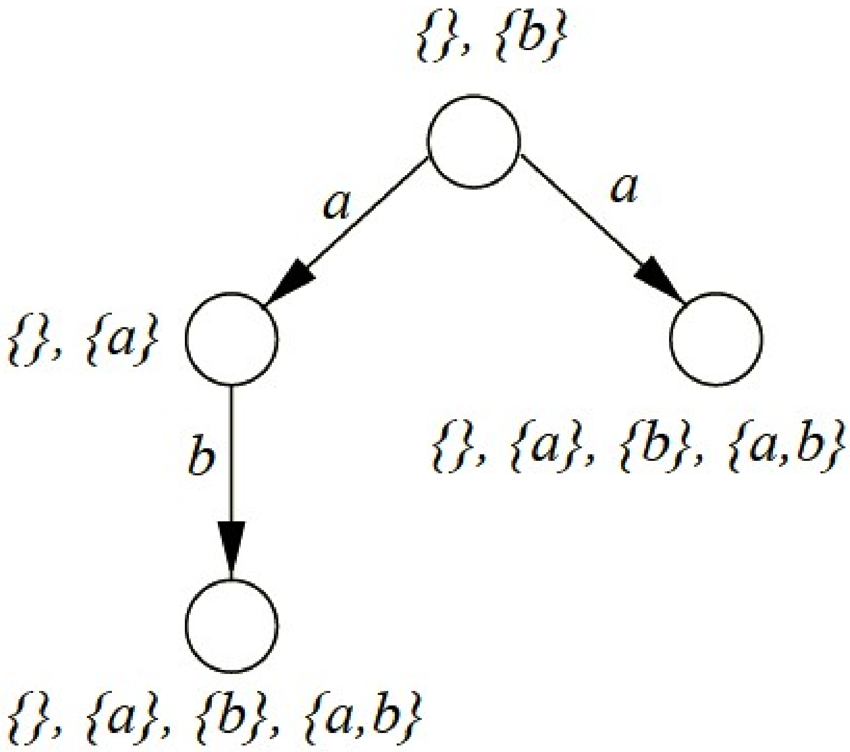
$$\boxed{refusals(P) = \{X \mid X \subseteq \alpha(P) \text{ and } X \text{ is a refusal of } P\}}$$

- BUT: Looking at refusals can only detect differences *at the first step.*

We need to look at events refused after *arbitrary traces* have been observed.

- Write P/tr for the process whose behaviour is whatever P could do after the trace tr has been observed.

Example: Understanding Refusal Sets



$$traces(P) = \{\langle\rangle, \langle a\rangle, \langle a,b\rangle\}$$

$$refusals(P/\langle\rangle) = \{\{\}, \{b\}\}$$

$$refusals(P/\langle a\rangle) = \{\{\}, \{a\}, \{b\}, \{a,b\}\}$$

$$refusals(P/\langle a,b\rangle) = \{\{\}, \{a\}, \{b\}, \{a,b\}\}$$

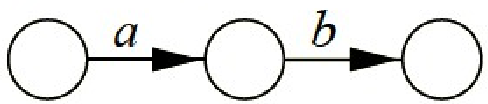
Failures are Trace-Refusal Pairs

We write P/tr for the process whose behaviour is whatever P could do after the trace tr has been observed.

Failures of a process:

$failures(P) = \{(tr, X) \mid tr \in traces(P)$ and $X \in refusals(P/tr)\}$

- $P = a \to b \to STOP$ with $\alpha(P) = \{a, b\}$

Transition Diagram of P: 

$$\begin{aligned}
traces(P) &= \{\langle\rangle, \langle a\rangle, \langle a, b\rangle\}
\end{aligned}$$

$$\begin{aligned}
refusals(P/\langle\rangle) &= \{\{\}, \{b\}\} \\
refusals(P/\langle a\rangle) &= \{\{\}, \{a\}\} \\
refusals(P/\langle a, b\rangle) &= \{\{\}, \{a\}, \{b\}, \{a, b\}\}
\end{aligned}$$

$$\begin{aligned}
failures(P) = \{&(\langle\rangle, \{\}), (\langle\rangle, \{b\}), \\
&(\langle a\rangle, \{\}), (\langle a\rangle, \{a\}), \\
&(\langle a, b\rangle, \{\}), (\langle a, b\rangle, \{a\}), (\langle a, b\rangle, \{b\}), (\langle a, b\rangle, \{a, b\})\}
\end{aligned}$$

Relationship between Failures and Traces

Recall that $\{\} \in refusals(P)$ for every process P.

$\Rightarrow$ Means that for every process P and every trace $tr \in traces(P)$, $(tr, \{\}) \in failures(P)$.

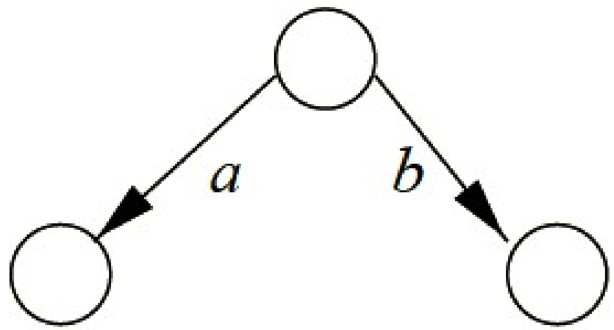
Traces can be recovered from failures!

$$traces(P) = \{tr \mid (tr, \{\}) \in failures(P)\}$$

Hence, if $failures(P) = failures(Q)$ then $traces(P) = traces(Q)$.

Example 2: Failure Representation

Transition Diagram of P:

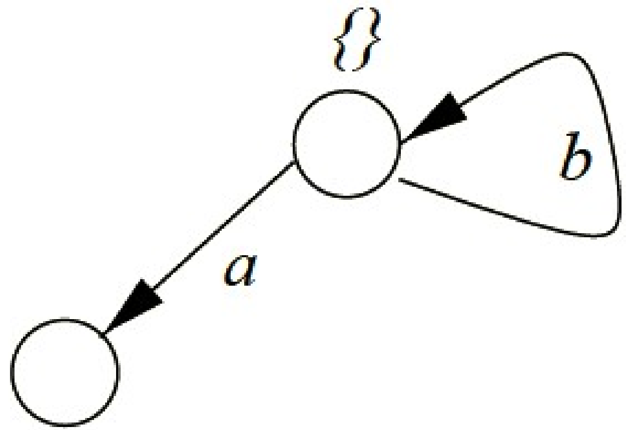


$$traces(P) = \{\langle\rangle, \langle a\rangle, \langle b\rangle\}$$

$$\begin{aligned} refusals(P/\langle\rangle) &= \{\{\}\} \\ refusals(P/\langle a\rangle) &= X \subseteq \{a,b\} \\ refusals(P/\langle b\rangle) &= X \subseteq \{a,b\} \end{aligned}$$

$$\begin{aligned} failures(P) = \{(\langle\rangle, \{\})\} &\cup \{(\langle a\rangle, X) \mid X \subseteq \{a,b\}\} \\ &\cup \{(\langle b\rangle, X) \mid X \subseteq \{a,b\}\} \end{aligned}$$

Example 3: Failure Representation



Transition Diagram of P: $\{\}, \{a\}, \{b\}, \{a,b\}$

$$failures(P) = \{(\langle b\rangle^n, \{\}) \mid n \geq 0\} \cup \{(\langle b\rangle^n \frown \langle a\rangle, X) \mid n \geq 0 \wedge X \subseteq \{a,b\}\}$$

Explicit External Choice

Process $P \square Q$ (Pronounce "P external choice Q".)

- Initially prepared to do any event P or Q could do.
- After first event, behaviour is either that of P or that of Q, depending on which process did the event.

External choice because environment (another process in parallel) can choose the first event.

In general:

- $a \to P \square b \to Q$ is equivalent to $a \to P \mid b \to Q$
- Possible to use $\square$ instead of $|$.
- However, external choice permits $(a \to P)\square(a \to Q)$.
- ($a \to P \mid a \to Q$ is illegal!)

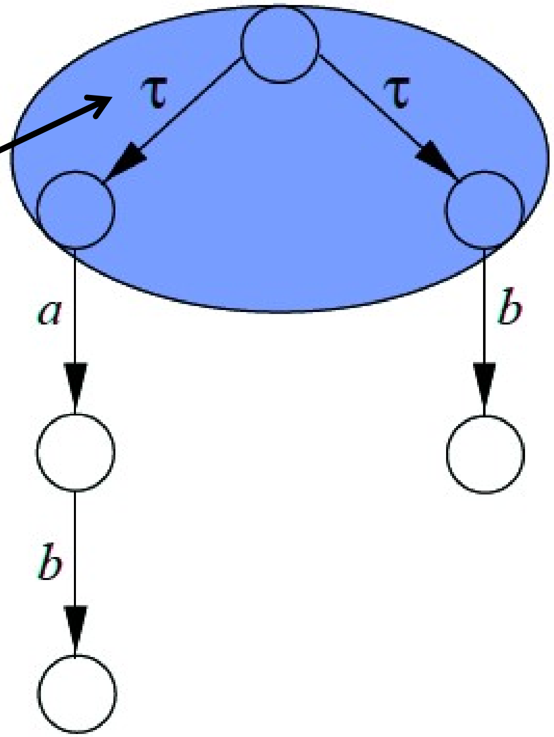
Internal Choice

Process $P \sqcap Q$ (Pronounce "P internal choice Q".)

- Choice between P and Q outside of environment's control.
- Resolve choice internally.

Nondeterministic choice!

- $P = a \to b \to STOP \sqcap b \to STOP$ with $\alpha(P) = \{a, b\}$



Internal Choice in Transition Diagrams using ***tau*** – (invisible/silent event) controlled internally

Internal Choice and Traces

Consider $P = a \to P$ and $Q = b \to Q$.

Traces of $P \Box Q$:

- Any trace of either P or Q can be produced by $P \Box Q$.
- $traces(P \Box Q) = traces(P) \cup traces(Q)$

Traces of $P \sqcap Q$:

- Always does *invisible* event τ first!
- $traces(P \sqcap Q) = traces(P) \cup traces(Q)$

BUT what happens if we put each of $P \Box Q$ and $P \sqcap Q$ **into an environment** consisting of P?

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

They behave differently when put in parallel with P. (One behaves just as P, the other one can internally choose to deadlock.)

Choice and Refusals I

- $P = a \rightarrow c \rightarrow STOP \ \Box \ b \rightarrow STOP$

  $initials(P) = \{a, b\}$

  $refusals(P) = \{\{\}, \{c\}\}$

- $P = (a \rightarrow c \rightarrow STOP) \sqcap (b \rightarrow STOP)$

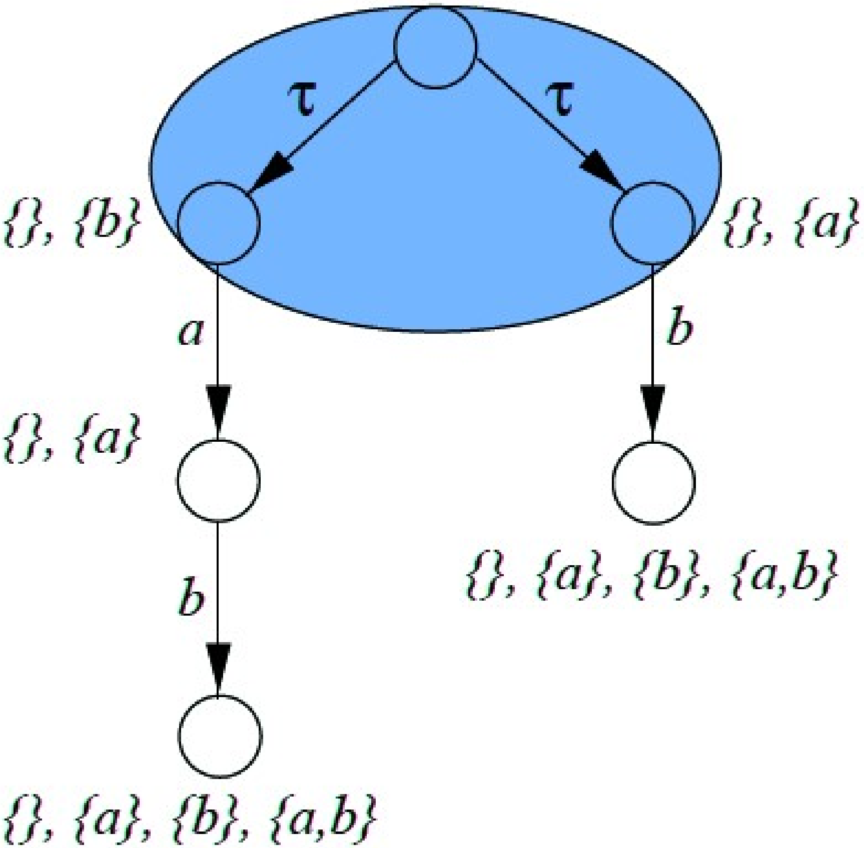
  $initials(P) = \{a, b\}$

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

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

Choice and Refusals II

- $P = a \rightarrow b \rightarrow STOP \sqcap b \rightarrow STOP$ with $\alpha(P) = \{a, b\}$



$$traces(P) = \{\langle\rangle, \langle a\rangle, \langle b\rangle, \langle a, b\rangle\}$$

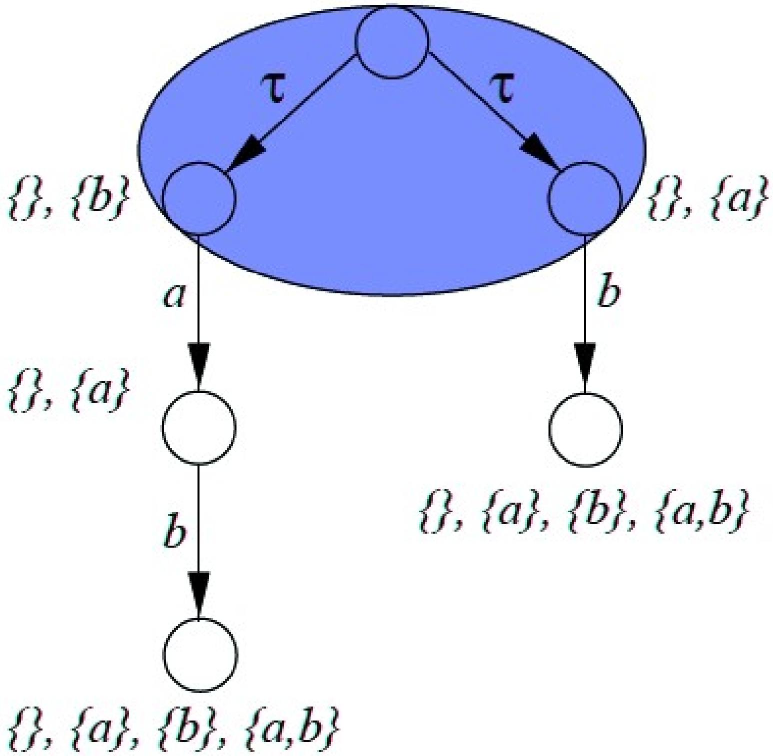
$$refusals(P/\langle\rangle) = \{\{\}, \{a\}, \{b\}\}$$
$$refusals(P/\langle a\rangle) = \{\{\}, \{a\}\}$$
$$refusals(P/\langle b\rangle) = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$$
$$refusals(P/\langle a, b\rangle) = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$$

Choice and Non-Determinism of Processes

P is ***deterministic*** if and only if

$\forall tr \in traces(P) \;.\; (refusals(P/tr) = \{X \subseteq \alpha(P) \mid X \cap initials(P/tr) = \{\}\})$

- $P = a \to b \to STOP \sqcap b \to STOP$ with $\alpha(P) = \{a, b\}$



$traces(P) = \{\langle\rangle, \langle a\rangle, \langle b\rangle, \langle a,b\rangle\}$

$refusals(P/\langle\rangle) = \{\{\}, \{a\}, \{b\}\}$

$refusals(P/\langle a\rangle) = \{\{\}, \{a\}\}$

$refusals(P/\langle b\rangle) = \{\{\}, \{a\}, \{b\}, \{a,b\}\}$

$refusals(P/\langle a,b\rangle) = \{\{\}, \{a\}, \{b\}, \{a,b\}\}$

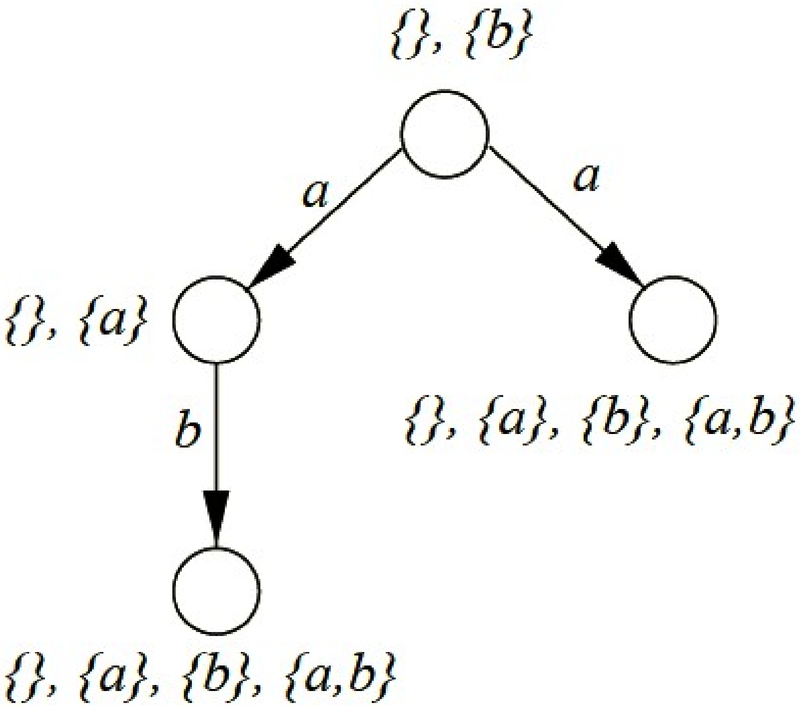
NOTICE: Nondeterminism caused by $\sqcap$.

Choice and Non-Determinism of Processes

P is *deterministic* if and only if

$$\forall tr \in traces(P)\ .\ (refusals(P/tr) = \{X \subseteq \alpha(P) \mid X \cap initials(P/tr) = \{\}\})$$

- $P = a \rightarrow b \rightarrow STOP \;\square\; a \rightarrow STOP$ with $\alpha(P) = \{a,b\}$



Is P deterministic?

$traces(P) = \{\langle\rangle, \langle a\rangle, \langle a,b\rangle\}$

$refusals(P/\langle\rangle) = \{\{\},\{b\}\}$

$refusals(P/\langle a\rangle) = \{\{\},\{a\},\{b\},\{a,b\}\}$

$refusals(P/\langle a,b\rangle) = \{\{\},\{a\},\{b\},\{a,b\}\}$

NOTICE: Nondeterminism caused by same initial action for $\square$.

Looking ahead...

Paradigms of Parallelism