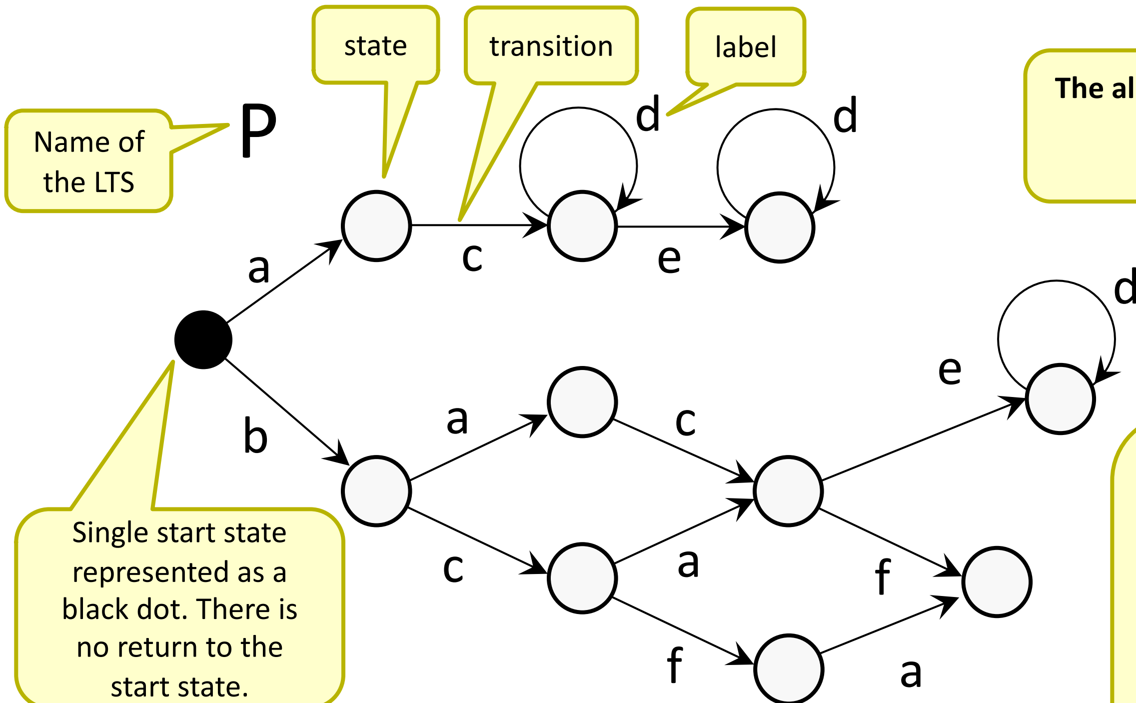
Extracting Concurrency from a Sequential System

a bit like:

Extracting Sunbeams from a Cucumber

Dr Ashley McNeile
Birkbeck College – March 2023

Labelled Transition Systems (LTS)



The alphabet of an LTS is the set of labels used:

alphabet(P) = {a, b, c, d, e}

Each trace of an a LTS represents a possible ordering of the labels in an *enactment*, e.g.:

a, c, d, e, d, d
b, a, c, e
b, c

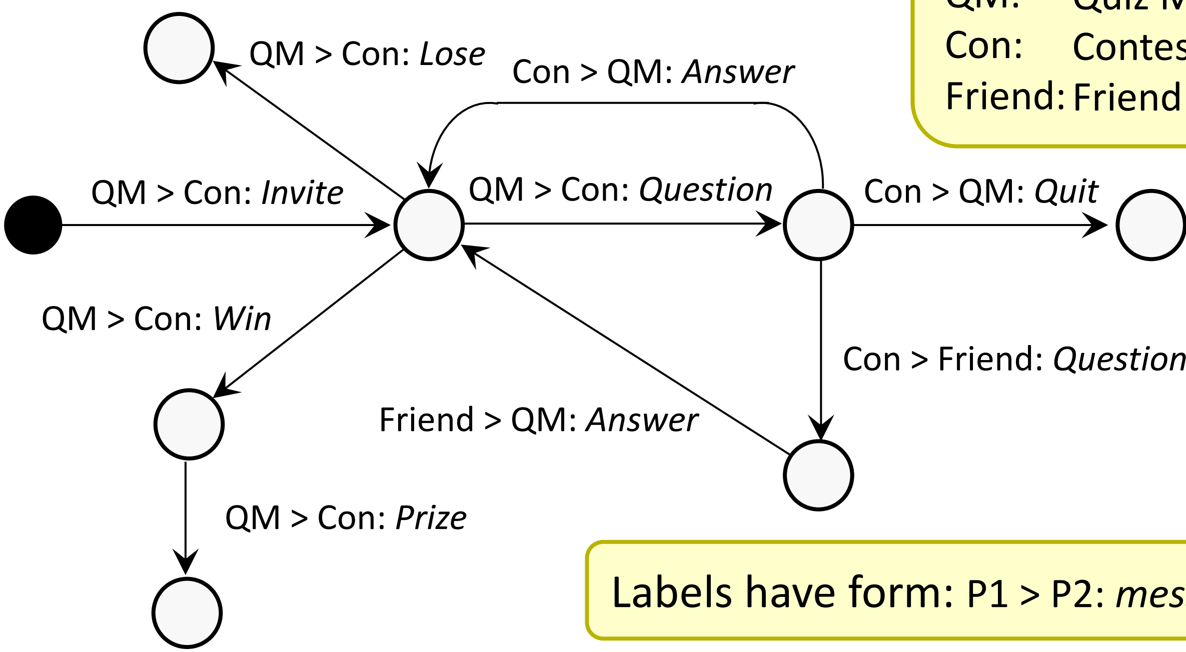
If two <u>deterministic</u> LTSs have the same set of traces they are deemed *equal*.

Deterministic: Distinct transitions from a given state have different labels.

We will only use deterministic LTSs.

A Simple Choreography

Quiz Game



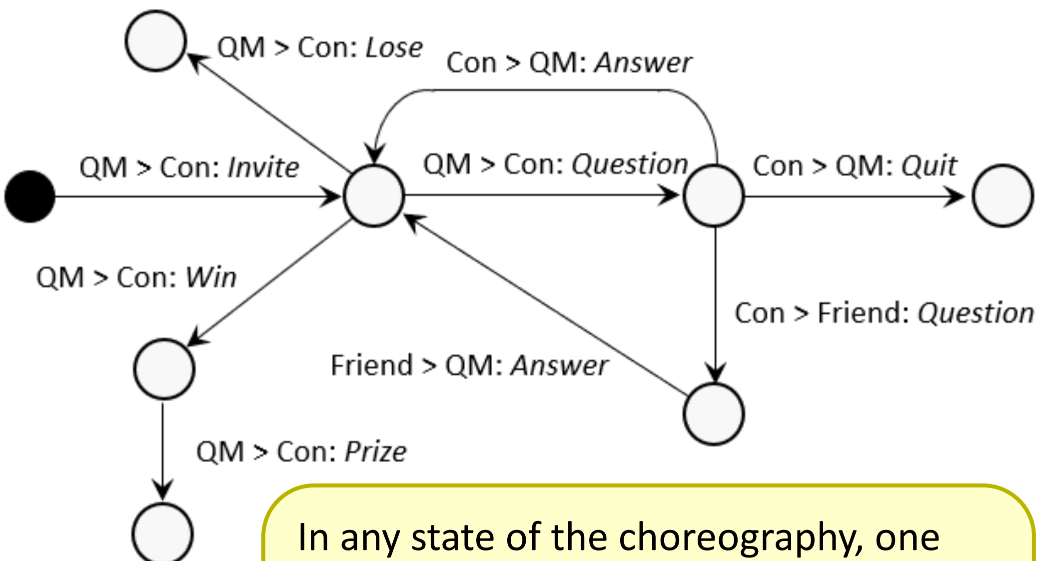
Participants:
QM: Quiz Master
Con: Contestant
Friend: Friend of Contestant

Labels have form: P1 > P2: *message*

Send and receive of a given message are separate events
The choreography defines the possible sequence of sends

Relay Form

Quiz Game



Relay Form:

1. Every state has a single sender
2. The sender of a state is the sender or receiver of any transition ending at the state.

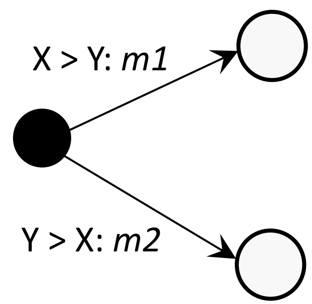
In any state of the choreography, one participant holds the baton:

- Only the holder may send
- A transition can transfer the baton to the receiver

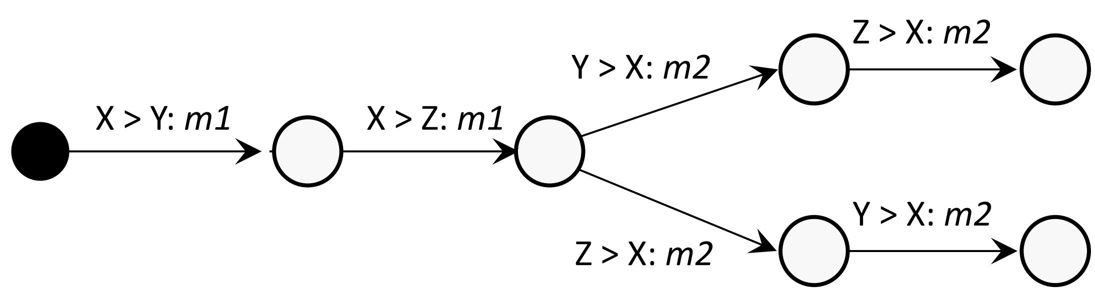
Relay Form is <u>sufficient</u> for Realisability.

Two Non-Relay Examples

Not Realisable

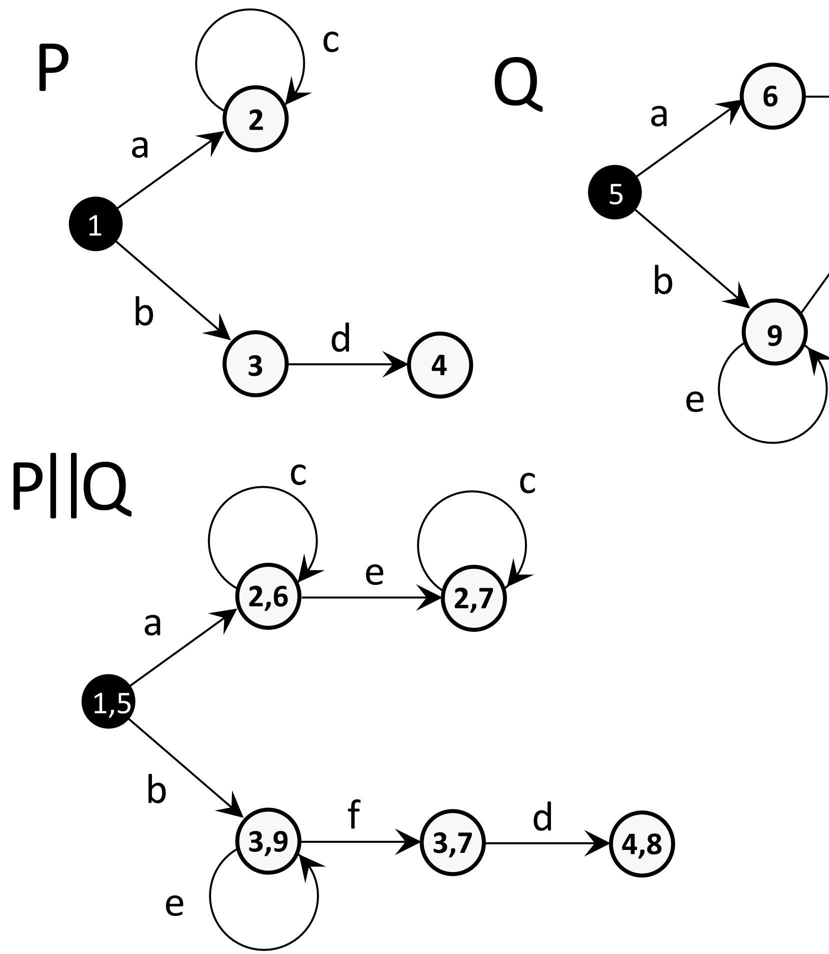


Realisable



Relay Form is not <u>necessary</u> for Realisability.

Composition (||)



The set of states of P||Q is the Cartesian product of the states of P and Q

alphabet(P) = {a,b,c,d}

alphabet(Q) = {a,b,d,e,f}

alphabet(P||Q) = *alphabet*(P) ∪ *alphabet*(Q)

A transition is allowed in the composition if and only if it is allowed by every component that has the transition’s label in its alphabet.

Composition Preserves Realisability

- If C1 and C2 are realisable choreographies then so is C1||C2

- This means that a <u>sufficient</u> condition for realisability of a choreography is that it can be expressed as a composition of relay components (called a “relay composition”)

- This is like a relay race with *multiple batons*. In particular, it is possible for two participants to send at the same state of the choreography

- However, is this a <u>necessary</u> condition for realisability?

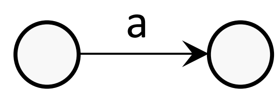
Factorisation

- To show being a “relay composition” is a necessary condition for realisability we need to show that any realisable choreography can be *factorised* into a set of relay components
- It turns out that this requires that we *relabel* the choreography, so that the concurrency is encoded in the labels
- However it also turns out that this relabelling is *benign* – more on this later

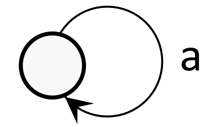
The three concurrency topologies

Two transition topologies

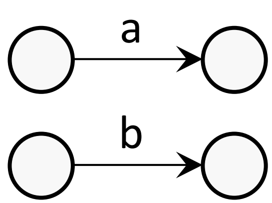
Line
(start and end states different)

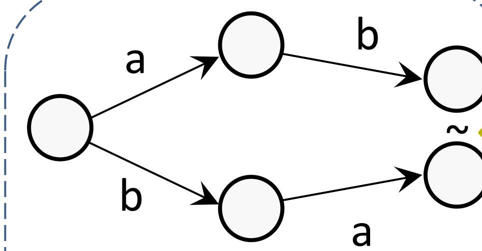


Loop
(start and end states the same)



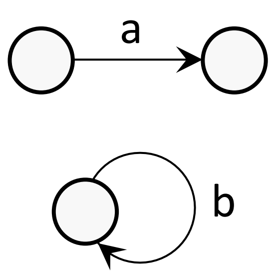
Line || Line

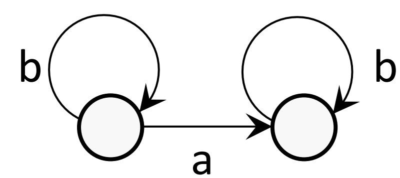




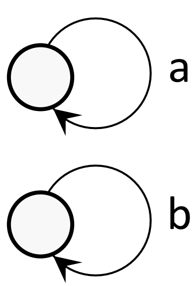
Final states are equivalent

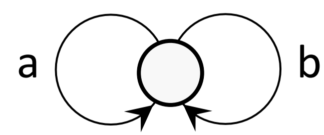
Line || Loop





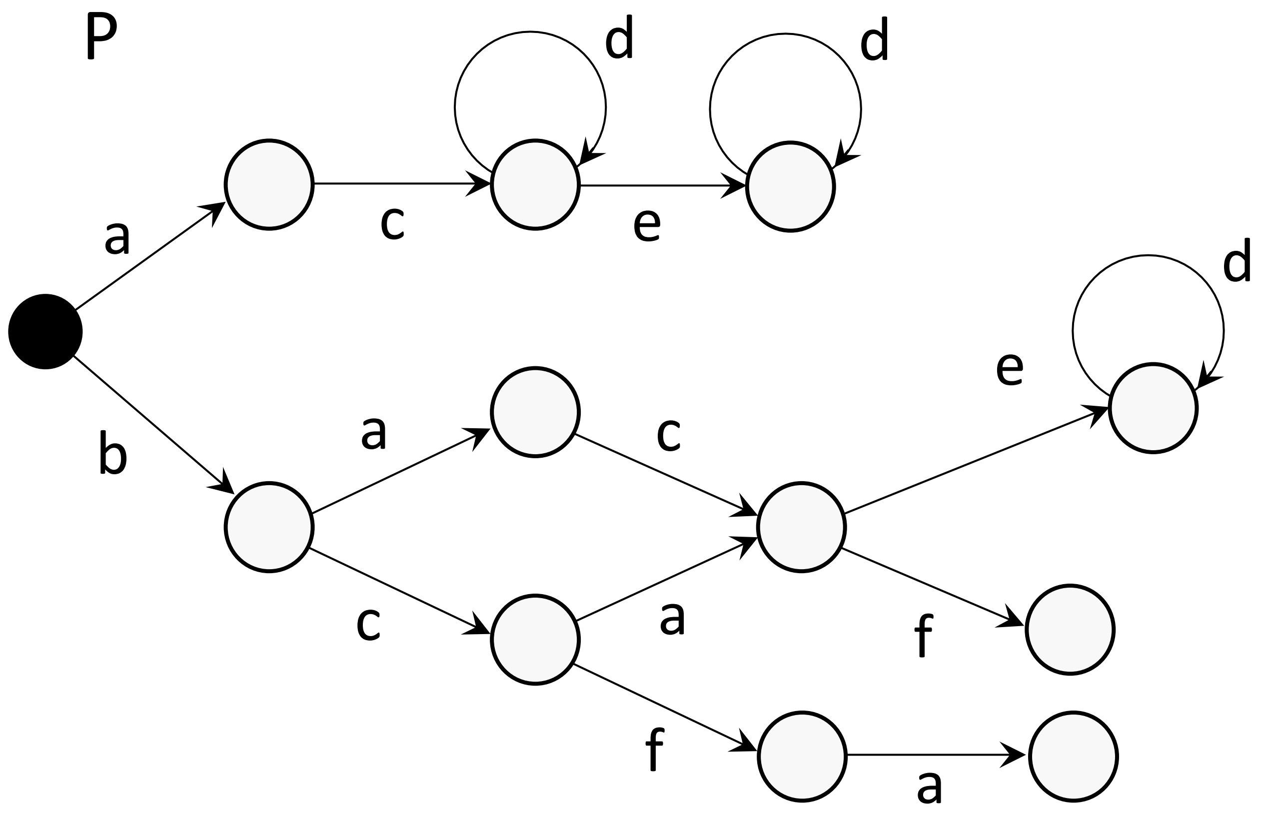
Loop || Loop



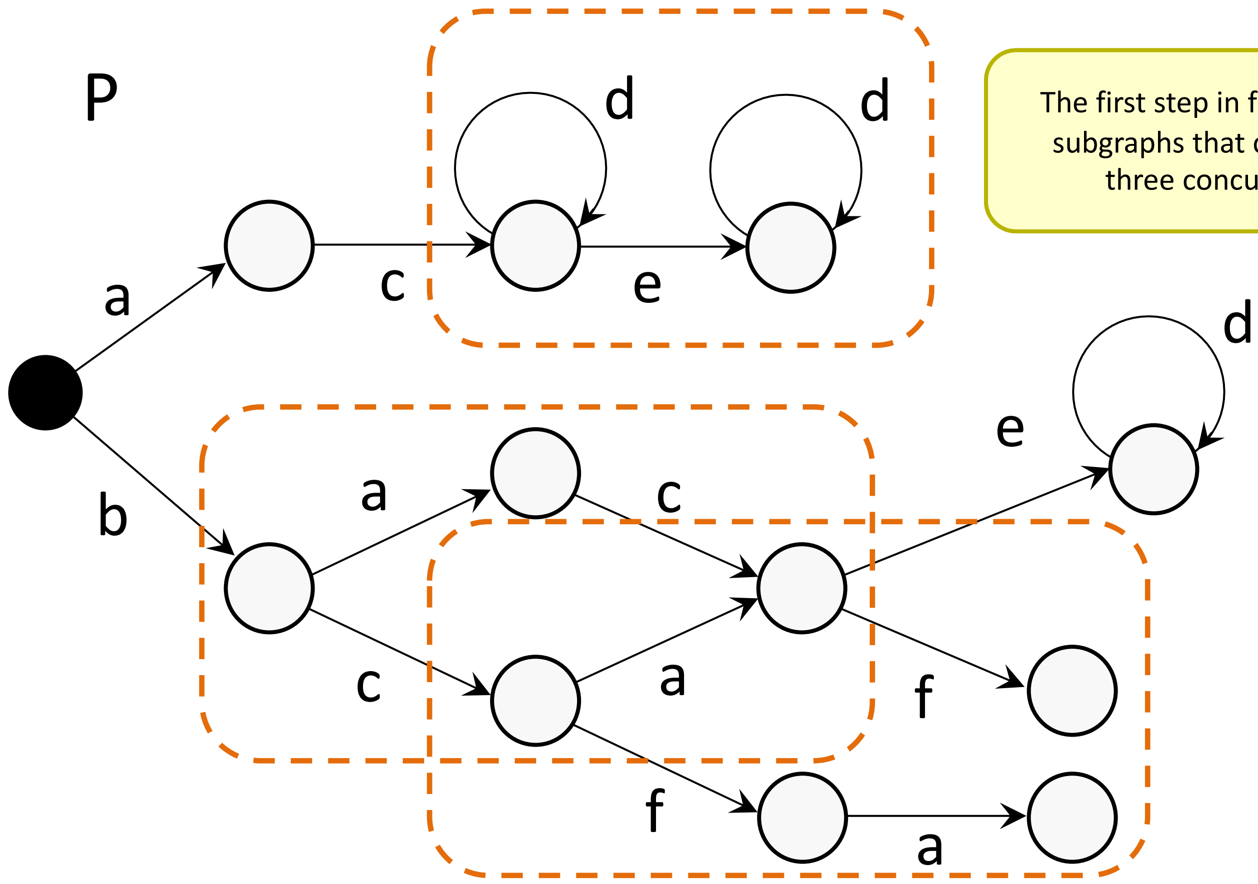


The three concurrency topologies

Factorising

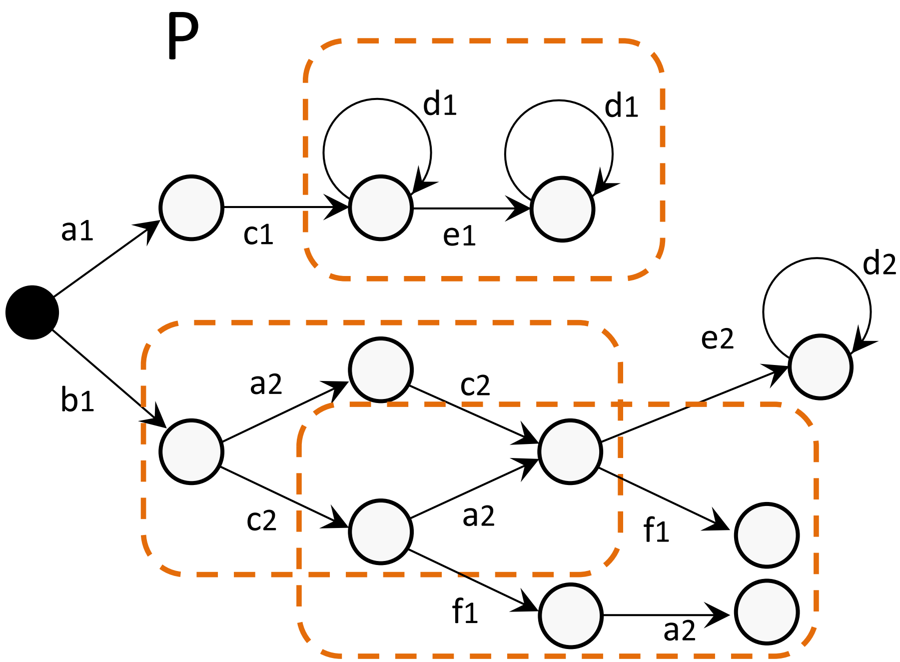


Factorising Step 1: Identify Concurrency Sub-graphs



The first step in factorising is to identify subgraphs that conform to one of the three concurrency topologies

Factorising Step 2: Relabel to preserve Concurrency



Relabelling adds a subscript to the label on each transition so that:

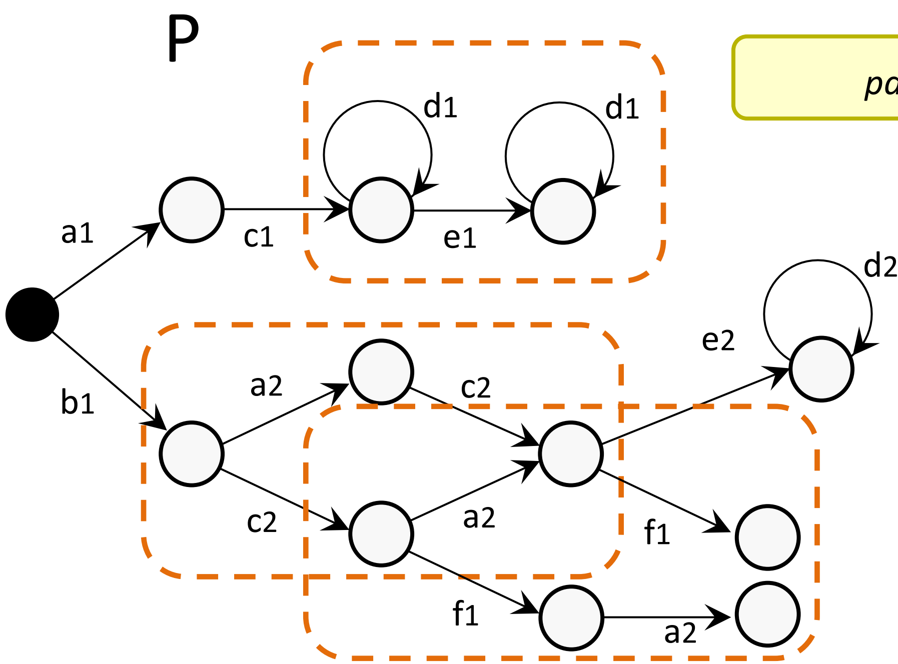
- *Concurrency is preserved*
- *The alphabet is maximised*

The purpose of relabelling is to <u>encode the concurrency in the labels</u>

After relabelling the original label can still be recovered by removing the subscript.

After relabelling: *alphabet*(P) = {a1, a2, b1, c1, c2, d1, d2, e1, e2, f1}

Factorising Step 3: Determine *pass*(P)



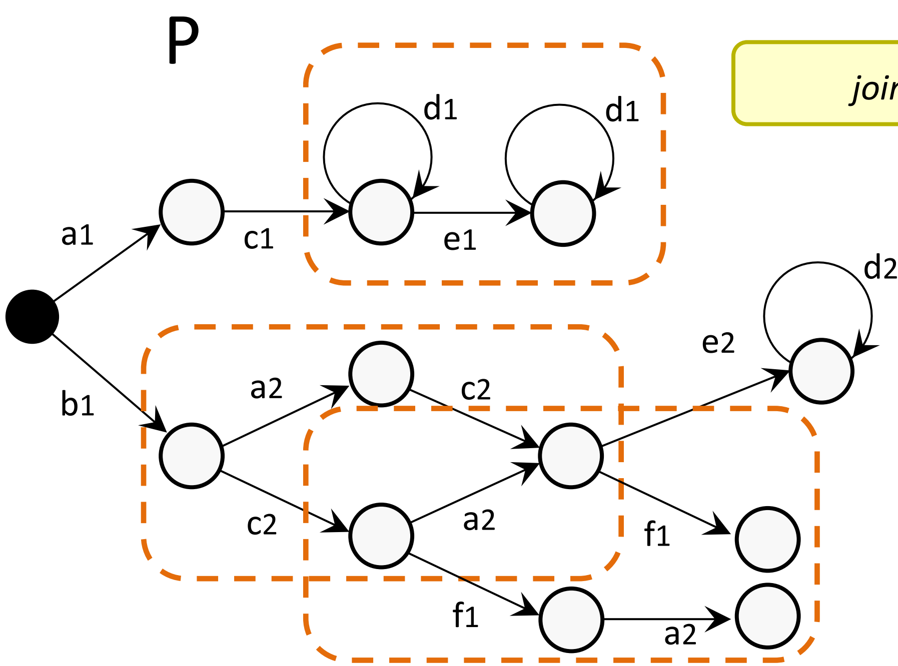
pass(P) $\subseteq$ { {λ1, λ2} | λ1, λ2 $\in$ *alphabet*(P)}

After relabelling the concurrency sub-graphs define the set of pairs called *pass*(P):

{d1,e1} $\in$ *pass*(P)
{a2, c2} $\in$ *pass*(P)
{a2, f1} $\in$ *pass*(P)

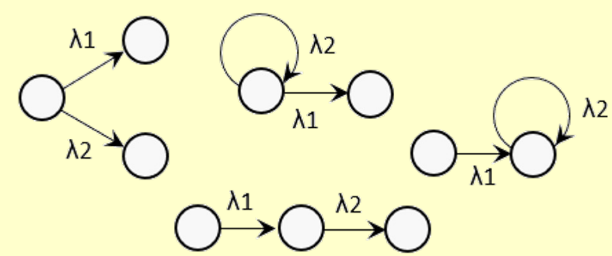
There is a member of *pass*(P) for each concurrency sub-graph.

Factorising Step 4: Determine *joined*(P)



joined(P) $\subseteq$ { {λ1, λ2} | λ1, λ2 $\in$ *alphabet*(P)}

joined(P) identifies labels which are not concurrent and occur in any one of these patterns:

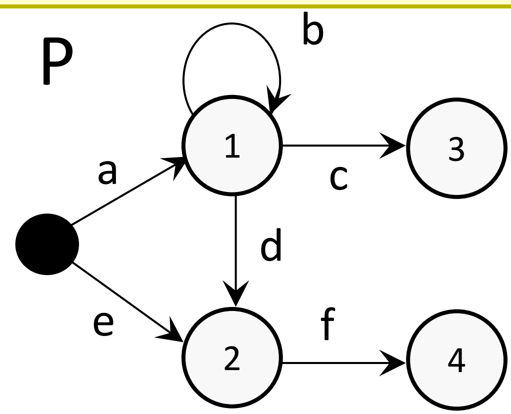


joined(P) = {{a1,c1}, {c1, d1}, {c1, e1}, {a1, b1}, {b1, a2}, {b1, c2}, {c2, f1}, {a2, e2}, {c2, e2}, {e2, f1}, {e2, d2}}

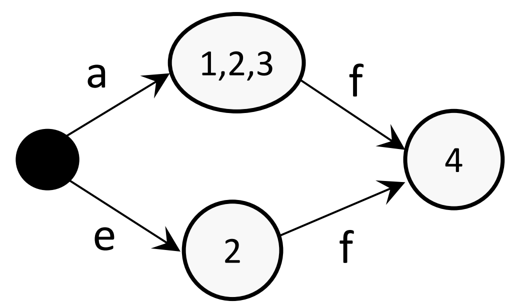
Reduction

Reduction is a transformation that limits the elements of the alphabet that are visible.

Given an LTS P and a set R $\subseteq$ *alphabet*(P) we form the new LTS: *reduction*(P,R) with *alphabet*(*reduction*(P,R)) = R



reduction(P, {a,e,f})



Σ is the original set the states.

The set of states in the reduction is $\Sigma' \subseteq$ *power*(Σ) where *power*(S) is the power set of S.

Each state of Σ' represents the uncertainty about the current state of the original.

For instance: After event **a** in **P** the reduction on the right is in the state {1,2,3} as the reduction does not know whether **c** or **d** have happened or not, as these are not in its alphabet.

Factorising Step 5: Determine *factors*

We define the *full reduction set*, $\mathbb{R}(\mathrm{P})$, of all possible reductions of an LTS P as follows:

$$\mathbb{R}(\mathrm{P}) = \{\mathit{reduction}(\mathrm{P},\mathrm{R}) \mid \mathrm{R} \in \mathit{power}(\mathit{alphabet}(\mathrm{P}))\}$$

The set of factors of P is the smallest subset $\mathbb{F}(\mathrm{P}) \subseteq \mathbb{R}(\mathrm{P})$ satisfying all of the following:

$$\bigcup_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} \mathit{alphabet}(\mathcal{F}) = \mathit{alphabet}(\mathrm{P})$$

The factors account for the whole alphabet

$$\bigcup_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} \mathit{joined}(\mathcal{F}) = \mathit{joined}(\mathrm{P})$$

The factors include all the constraints represented by *joined*

$$\bigcup_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} \mathit{pass}(\mathcal{F}) = \emptyset$$

The factors themselves have no concurrency

It is possible to show that: $\displaystyle \big\|_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} \mathcal{F} = \mathrm{P}$

Existence of *factors*

The set of factors of P is the smallest subset $\mathbb{F}(\mathrm{P}) \subseteq \mathbb{R}(\mathrm{P})$ satisfying all of the following:

$$\bigcup_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} alphabet(\mathcal{F}) = alphabet(\mathrm{P})$$

$$\bigcup_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} joined(\mathcal{F}) = joined(\mathrm{P})$$

$$\bigcup_{\mathcal{F} \in \mathbb{F}(\mathrm{P})} pass(\mathcal{F}) = \emptyset$$

Does $\mathbb{F}(\mathrm{P})$ exist?

The set of reductions generated by:

$$\left\{ \{\lambda1\} \mid \lambda1 \in alphabet(\mathrm{P})\} \right\} \cup \left\{ \{\lambda2, \lambda3\} \mid \{\lambda2, \lambda3\} \in joined(\mathrm{P})\} \right\}$$

gives a set of reductions that satisfies the above (but not necessarily the smallest such set).

This guarantees the existence of $\mathbb{F}(\mathrm{P})$.

The Proposition

- Any realisable choreography can be factorised using the method described, by subscripting the message types.

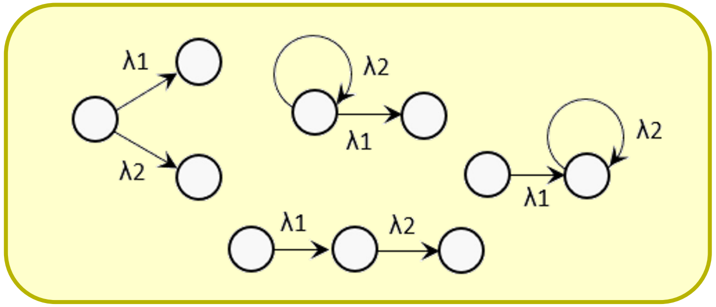
- The resultant factors will have relay form.

- The relabelled choreography is still realisable.

If this proposition is true then we can say that being expressible as a relay composition is necessary for realisability (up to relabelling).

Factors have Relay Form

Relay Form of factors follows directly from the form of the *joined* patterns. If labels in these patterns are not to be concurrent, they must have relay form.



The relabelled Choreography is Realisable

It can be shown that the participant processes of a realisable choreography can also participate successfully in the relabelled choreography without material change (just the addition of the subscripts to message types).

In particular, the relabelling does not introduce non-determinism into the participant processes.

Other Articulations

Realisability of Control-State Choreographies
Klaus-Dieter Schewe · Yamine Aït-Ameur · Sarah Benyagoub

If two transitions do not conform to either Sequence Condition or Choice Condition (= Relay Form) they must be "swappable".

Deciding Choreography Realizability
Samik Basu · Tevfik Bultan · Meriem Ouederni

If a choreography is realisable when the message queue between two participants is limited to a single message, then it is realisable with unlimited queues. Realisability with limited queues has a finite execution space so can be analysed exhaustively.

Gulliver’s Travels
Jonathan Swift (1726)

Gulliver visits the academy in Lagado where he encounters men engaged in many bizarre projects.

“He had been eight years upon a project for extracting sunbeams out of cucumbers, which were to be put in vials hermetically sealed, and let out to warm the air in raw inclement summers. He told me, he did not doubt in eight years more, that he should be able to supply the Governor's gardens with sunshine at a reasonable rate; but he complained that his stock was low since this had been a very dear season for cucumbers.”