* What a state contains?
* How to analyse a state in Bigrapher!?
* What do we need to know? Do we know what we need to know? How to know what we need to know!? (paper title!)

# How to analyse states and transitions?

The aim is to identify assets, relationships, and actions that are relevant to the incident being modelled in the space. This involves analysing possible state transitions that satisfy the preconditions and postconditions of each activity in an incident.

From previous phases, I did the following:

1. identified potential assets that match entities in the incident model.
2. Generated predicates (preconditions and postconditions): preconditions and postconditions of each activity are replaced with matched assets from the space. Then inserted into a Bigrapher file, which is a representation of the space.
3. Extracted states that satisfy these predicates through executing a bigraph representation of the space using Bigrapher that included the generated predicates.
4. Identified state transitions between states that satisfy the predicates.

Currently, I have identified all possible state transitions that satisfy the pre/postconditions of an activity and also transitions that link one activity to the next.

What is left is to analyse these transitions to reach our aim that I stated at the beginning of the section. To do so, we need to know what a state in Bigrapher contains. This is discussed in the next subsection.

## What a state contains?

A state generated by Bigrapher has the following information:

* Nodes: A Bigrapher state has a definition of the nodes, where each node is defined by a node id, and a control and its arity. For example:

*{"node\_id": 4,*

*"control": {"control\_id": "M", "control\_arity": 2}}*

*{"node\_id": 3,*

*"control": {"control\_id": "Snd", "control\_arity": 0}}*

* Directed graph (place graph) which represents the containment. A containment is represented by the source -> target using nodes ids defined in the nodes above. The place graph also holds information about the number of regions, nodes, and sites. For example:

*{"source": 4, "target": 3}*, numbers refer to nodes ids

* Link graph representing the connectivity graph between nodes in that state. A connectivity is represented by inner and out interfaces. An interface such as outer interface is defined by a name and ports. Each port has a node id and a port arity (i.e. how many connections to that node are there).

*{"inner": [], "outer": [{"name": "v\_b"}], "ports": [{"node\_id": 4, "port\_arity": 1}]}*

## Identifying Actions for State Transitions

For actions, we can identify actions through state transitions, which I implemented using the *LabelExtractor* class since currently Bigrapher does not label transactions. However, the assumption made is that each reaction rule should have a unique keyword that is placed in the redex and dropped in the reactum. A file containing all reaction rules keywords separated by semicolon should also be provided. For example:

*react snd = A{a}.****Snd****.(M{a1, v} | id) | Mail -> A{a} | Mail.(M{a1, v} | id);*

*react ready = A{a}.****Ready*** *| Mail.(M{a, v} | id) -> A{a} | Mail | {v};*

keywords used = {Snd; Ready}

I have implemented another approach that does not require having unique keywords but it is computationally more expensive and it can return more than one possible action for a transition. The approach is to use the redex and reactum of a reaction rule as predicates in the Bigrapher file then extract states that satisfy them and identify states that are immediately a transition to each other depending on the transition system or the digraph object that represent the transition system.

## Identifying Relevant Parts of a State

We need to identify part(s) of a state that are relevant to the incident being investigated. We can extract assets and their relationships that concerns us from a given state depending on earlier generated predicates (phase-2), which can tell us what assets and relationships we are interested in.

Identifying which action took place helps in identifying relevant part(s) of stats. It can help identify nodes and their relationships at a state and the change that happened to that part in the next state.

There could be different parts that evolve/change over a state transition.