* 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!)
* The goal of state analysis is to identify **nodes**, **relationships**,and **actions** in the state that are “relevant” to some incident(s) being analysed.

# What a state contains?

* 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}*

* 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}]}*

# 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:

* identified potential assets that match entities in the incident model.
* Generated predicates (preconditions and postconditions).
* Extracted states that satisfy these predicates through executing a bigraph representation of the space using Bigrapher that included the generated predicates.
* Identified state transitions between states that satisfy the predicates.

Through state transitions we can identify actions, which I implemented using the label extractor class since currently Bigrapher does not label transactions.