

# CT5132/CT5148 Lab Week 8

James McDermott

## Finite State Machines

The Pacman ghost FSM was very simplified. It had no actions, just transitions from one state to another.

We're now going to use an FSM to model the behaviour of a vending machine. The rules are:

1. Each drink costs EUR3.
2. The only coins allowed are EUR1 and EUR2.
3. If the customer enters too much money, we just return it.
4. If the customer enters the right amount, they can then select a drink, and we vend it.

How should we model this as an FSM? Let's draw it on paper. Issues to think about:

- What are the states?
- What are the possible inputs?
- Are there actions?

Here is some code which implements a general FSM (also in Bb `fsm.py`):

```
def fsm(SISAs, start_state, end_states, input_symbols):
    state = start_state
    # (current_state, input_symbol) -> (next_state, action)
    for input_symbol in input_symbols:
        yield state
        if state in end_states:
            break
        try:
            state, action = SISAs[state, input_symbol]
            if action is not None and callable(action):
                action()
        except KeyError:
            pass # stay in same state
```

Implement our vending machine FSM as a data structure `SISAs`, and confirm that it works by passing in various possible sequences of inputs. E.g. with input `[1, 1, 1]`, we should see that the FSM says `vend drink` at the end and comes back to state 0.

## What was the most important family in Renaissance Florence?

This is a short demo of **betweenness centrality** in `networkx`. Betweenness centrality is one way to measure how important each node in a graph is, due to being on the paths between other nodes. We don't need to understand the maths, but let's see how to use `networkx` for this.

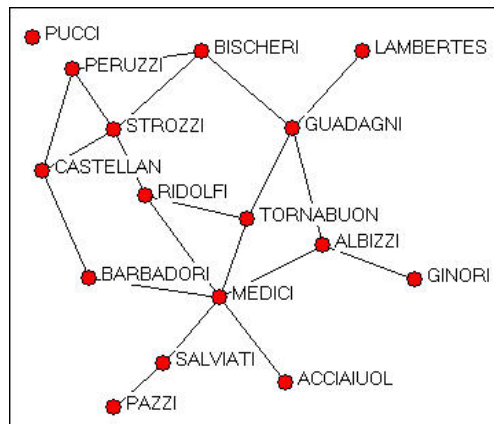


Figure 1: This graph represents the **marriages** between important families in Renaissance Florence (Padgett, figure from Hanneman and Riddle):

Suppose we are given the data in this format (`florentine_families.csv`):

```

Acciaiuoli,Medici
Medici,Barbadori
Medici,Ridolfi
Medici,Tornabuoni
Medici,Albizzi
Medici,Salviati
Castellani,Peruzzi
Castellani,Strozzi
Castellani,Barbadori
Peruzzi,Strozzi
Peruzzi,Bischeri
Strozzi,Ridolfi
Strozzi,Bischeri
Ridolfi,Tornabuoni
Tornabuoni,Guadagni
Albizzi,Gionori
Albizzi,Guadagni
Salviati,Pazzi
Bischeri,Guadagni
Guadagni,Lamberteschi

```

1. Import `networkx`.
2. Create an empty graph. Should it be directed or undirected for this data?
3. Add the edges.

4. Look up how to calculate betweenness centrality in **networkx**, and call it.
5. Which family seems the most important? Can you interpret this from the graph?
6. Optional homework: there are other measures of centrality, e.g. **eigenvector centrality**, related to Google PageRank. Try them out. Do they all agree?
7. Optional homework: use **networkx** to make an image of the graph.