Untitled26

December 7, 2023

[1]:
```
import matplotlib.pyplot as plt
import networkx as nx
```

[3]:
```
G = nx.DiGraph()
in_out_labels = []
fsm = dict()
ins = []

with open("automaton.dat", "r") as f:
    start_node = f.readline()
    start_node = start_node.split(": ")[-1].strip()
    end_node = f.readline().split(": ")[-1].strip()
    default_output = f.readline().split(": ")[-1].strip()
    for line in f.readlines():
        tmp = line.split(": ")[1].strip()
        current_node, in_out, next_node = tmp.split(" -> ")
        in_out_labels.append(in_out)
        in_sym, out_sym = [i.strip() for i in in_out.split("/")]
        ins.append(in_sym)
        if current_node not in fsm:
            fsm[current_node] = dict()
            fsm[current_node][in_sym] = (out_sym, next_node)
        else:
            fsm[current_node][in_sym] = (out_sym, next_node)
        G.add_edge(current_node, next_node)

fsm["s27"]["e83"] = (default_output, "s27")
```

[8]:
```
states = fsm.keys()
inputs = fsm["s27"].keys()
inputs
```

[8]:
```
dict_keys(['e88', 'e18', 'e83'])
```

[16]:
```
# inputs, (), ()

fronta = [("", states)]
```

```
undistinguished = set(states)
while len(undistinguished) > 0:
    a = fronta.pop(0)
    for i in inputs:
        p = dict()
        for j in a[1]:
            b, c = fsm[j][i]
            if b not in p:
                p[b] = [c]
            else:
                p[b].append(c)
        for k in p:
            if len(p[k]) == 1:
                if p[k][0] in undistinguished:
                    undistinguished.remove(p[k][0])
                    # add to final sequences
            else:
                fronta.append((a[0]+"-"+i, set(p[k])))
```

[17]:
```
fronta
```

[17]:
```
[('-e88-e18-e83', {'s27', 's9'}),
 ('-e88-e83-e88', {'s51', 's76', 's9'}),
 ('-e88-e83-e18', {'s50', 's89'}),
 ('-e88-e83-e83', {'s51', 's92'}),
 ('-e88-e83-e88', {'s55', 's70'}),
 ('-e88-e88-e88', {'s55'}),
 ('-e88-e88-e83', {'s44', 's9'}),
 ('-e88-e88-e88', {'s34', 's92'}),
 ('-e88-e88-e18', {'s10', 's12'}),
 ('-e88-e88-e83', {'s51', 's55'}),
 ('-e88-e18-e88', {'s55', 's70', 's9'}),
 ('-e88-e18-e18', {'s12', 's51', 's7'}),
 ('-e88-e18-e83', {'s51', 's70'}),
 ('-e88-e18-e18', {'s50', 's70'}),
 ('-e88-e18-e83', {'s27', 's55'}),
 ('-e88-e83-e88', {'s55', 's70'}),
 ('-e88-e83-e18', {'s0', 's76'}),
 ('-e88-e83-e83', {'s9'}),
 ('-e88-e83-e88', {'s51', 's76'}),
 ('-e88-e83-e18', {'s50', 's89'}),
 ('-e18-e88-e88', {'s34', 's55'}),
 ('-e18-e88-e18', {'s12', 's70'}),
 ('-e18-e88-e18', {'s27', 's76'}),
 ('-e18-e88-e83', {'s51', 's55'}),
 ('-e18-e88-e83', {'s44', 's9'}),
 ('-e18-e88-e88', {'s55', 's70'}),
```

```
('-e18-e88-e18', {'s12', 's51'}),
('-e18-e18-e83', {'s34', 's70'}),
('-e18-e18-e88', {'s34', 's70'}),
('-e18-e18-e88', {'s34', 's55'}),
('-e18-e18-e18', {'s0', 's27'}),
('-e18-e18-e18', {'s50', 's70'}),
('-e18-e18-e83', {'s27', 's51', 's55'}),
('-e18-e83-e88', {'s55', 's70'}),
('-e18-e83-e18', {'s0', 's76'}),
('-e18-e83-e18', {'s12', 's50'}),
('-e18-e83-e83', {'s44', 's9'}),
('-e18-e88-e88', {'s55', 's70', 's76'}),
('-e18-e88-e18', {'s76', 's89'}),
('-e18-e88-e83', {'s10', 's9'}),
('-e18-e88-e88', {'s34', 's55'}),
('-e18-e88-e18', {'s12', 's70'}),
('-e18-e88-e83', {'s27', 's55'}),
('-e18-e18-e88', {'s55', 's70'}),
('-e18-e18-e88', {'s34', 's89'}),
('-e18-e18-e18', {'s0', 's7'}),
('-e18-e18-e18', {'s0', 's27'}),
('-e18-e18-e83', {'s0', 's27', 's9'}),
('-e18-e18-e88', {'s34', 's55'}),
('-e18-e18-e18', {'s0', 's27'}),
('-e18-e18-e83', {'s27', 's51'}),
('-e18-e83-e88', {'s55', 's70'}),
('-e18-e83-e18', {'s12', 's51'}),
('-e18-e83-e83', {'s44', 's9'}),
('-e18-e83-e88', {'s34', 's55'}),
('-e18-e83-e18', {'s27', 's89'}),
('-e18-e83-e83', {'s27', 's51'}),
('-e83-e88-e88', {'s34', 's70', 's76'}),
('-e83-e88-e18', {'s0', 's89'}),
('-e83-e88-e83', {'s10', 's9'}),
('-e83-e88-e88', {'s55'}),
('-e83-e88-e83', {'s44', 's9'}),
('-e83-e18-e88', {'s0', 's55', 's76'}),
('-e83-e18-e18', {'s51', 's76'}),
('-e83-e18-e83', {'s27', 's34'}),
('-e83-e83-e88', {'s55', 's92'}),
('-e83-e83-e18', {'s10', 's50'}),
('-e83-e83-e83', {'s27', 's55'}),
('-e83-e88-e88', {'s55', 's70'}),
('-e83-e18-e88', {'s55', 's70', 's9'}),
('-e83-e18-e18', {'s0', 's51'}),
('-e83-e18-e18', {'s12', 's7'}),
('-e83-e18-e83', {'s34', 's51'}),
```

```
('-e83-e18-e83', {'s27', 's9'}),
('-e83-e18-e83', {'s51', 's55'}),
('-e83-e83-e88', {'s55', 's70'}),
('-e83-e83-e18', {'s50', 's51'}),
('-e83-e83-e83', {'s27', 's70'}),
('-e83-e83-e88', {'s51', 's76'}),
('-e83-e83-e18', {'s50', 's89'}),
('-e88-e88-e88-e88', {'s55', 's70'}),
('-e88-e88-e88-e18', {'s12', 's51'}),
('-e88-e88-e88-e83', {'s44', 's9'}),
('-e88-e88-e18-e88', {'s55', 's70'}),
('-e88-e88-e18-e18', {'s51', 's7'}),
('-e88-e88-e18-e83', {'s27', 's9'}),
('-e88-e88-e83-e88', {'s55', 's70'}),
('-e88-e88-e83-e88', {'s51', 's76'}),
('-e88-e88-e83-e18', {'s50', 's89'}),
('-e88-e18-e18-e18', {'s50', 's70'}),
('-e88-e18-e18-e83', {'s27', 's55'}),
('-e88-e18-e83-e88', {'s55', 's70'}),
('-e88-e18-e83-e18', {'s0', 's76'}),
('-e88-e18-e83-e83', {'s9'}),
('-e88-e18-e88-e88', {'s55', 's70'}),
('-e88-e18-e18-e83', {'s0', 's9'})]
```

[ ]: