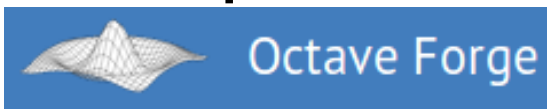
3-1-14-visualizationFSM-2019-4-15

CTI One Corporation

Version: x0.4
Date: Sep 7, 2018
Project Lead: Harry Li, Ph.D.

Team members: Zhixuan Zhou

April-21-2019 Octave Neural Fuzzy Tool



https://octave.sourceforge.io/fuzzy-logic-toolkit/overview.html

fuzzy-logic-toolkit

https://www.youtube.com/watch?v=HWHO3ApA9RA

File Input/Output of Fuzzy Inference Systems

Fuzzy Inference Systems

Membership functions

Fuzzy Demo Systems

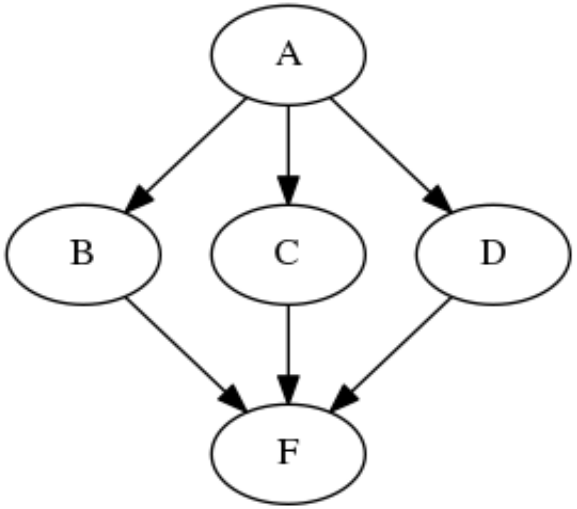
Fuzzy Clustering

April-21-2019 Graphviz Eamples

https://renenyffenegger.ch/notes/tools/Graphviz/examples/index

Example 1:

```
digraph D {

  A -> {B, C, D} -> {F}

}
```

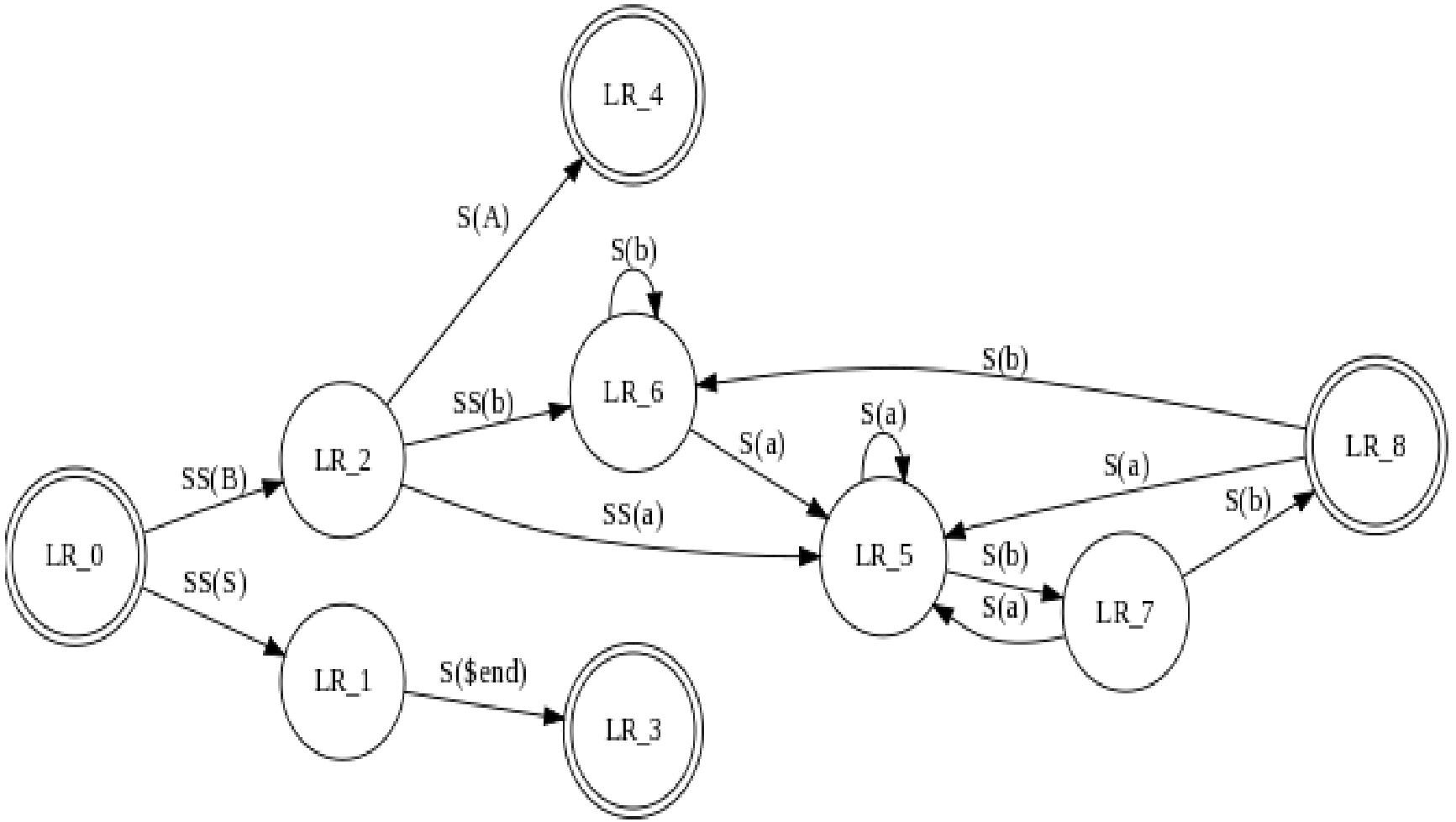



```
digraph finite_state_machine {
        rankdir=LR;
        size="8,5"
        node [shape = doublecircle]; LR_0 LR_3 LR_4 LR_8;
        node [shape = circle];
        LR_0 -> LR_2 [ label = "SS(B)" ];
        LR_0 -> LR_1 [ label = "SS(S)" ];
        LR_1 -> LR_3 [ label = "S($end)" ];
        LR_2 -> LR_6 [ label = "SS(b)" ];
        LR_2 -> LR_5 [ label = "SS(a)" ];
        LR_2 -> LR_4 [ label = "S(A)" ];
        LR_5 -> LR_7 [ label = "S(b)" ];
        LR_5 -> LR_5 [ label = "S(a)" ];
        LR_6 -> LR_6 [ label = "S(b)" ];
        LR_6 -> LR_5 [ label = "S(a)" ];
        LR_7 -> LR_8 [ label = "S(b)" ];
        LR_7 -> LR_5 [ label = "S(a)" ];
        LR_8 -> LR_6 [ label = "S(b)" ];
        LR_8 -> LR_5 [ label = "S(a)" ];
}
```

https://graphviz.gitlab.io/_pages/Gallery/directed/fsm.gv.txt

Flow diagram on the next page

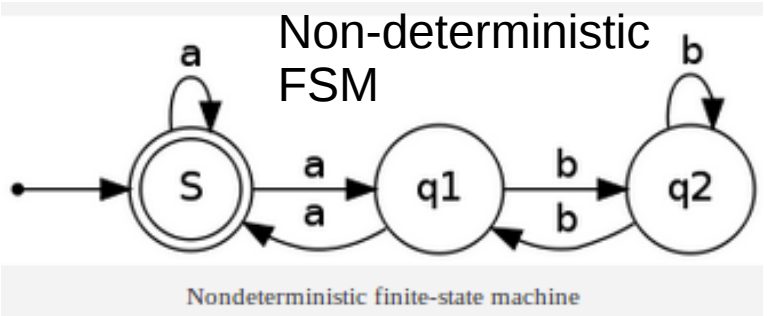
April-21-2019 Graphviz FSM Examples



April-21-2019 How to Draw FSM

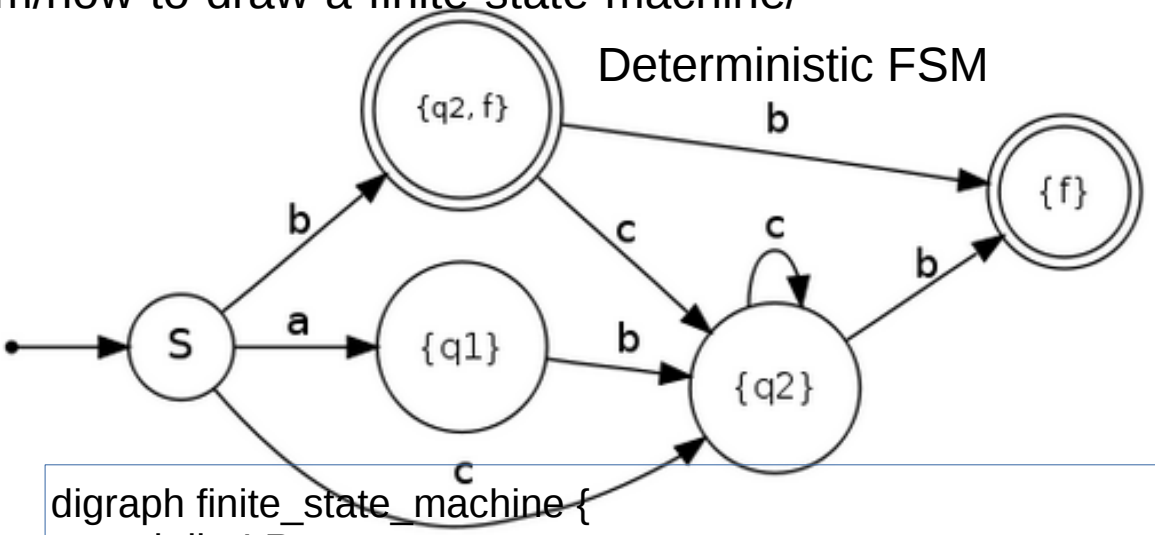
https://martin-thoma.com/how-to-draw-a-finite-state-machine/

Non-deterministic FSM



Nondeterministic finite-state machine

Deterministic FSM



Example: FSM.gv

```
digraph finite_state_machine {
    rankdir=LR;
    size="8,5"

    node [shape = doublecircle]; S;
    node [shape = point ]; qi

    node [shape = circle];
    qi -> S;
    S  -> q1 [ label = "a" ];
    S  -> S  [ label = "a" ];
    q1 -> S  [ label = "a" ];
    q1 -> q2 [ label = "b" ];
    q2 -> q1 [ label = "b" ];
    q2 -> q2 [ label = "b" ];
}
```

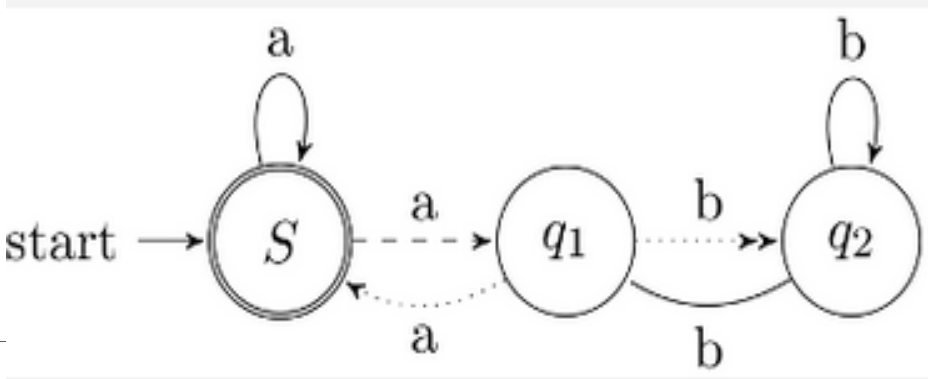
```
digraph finite_state_machine {
    rankdir=LR;
    size="8,5"
    node [shape = doublecircle, label="{f}", fontsize=12] f;
    node [shape = doublecircle, label="{q2, f}", fontsize=10] q2f;
    node [shape = circle, label="S", fontsize=14] S;
    node [shape = circle, label="{q1}", fontsize=12] q1;
    node [shape = circle, label="{q2}", fontsize=12] q2;
    node [shape = point ]; qi
    qi -> S;
    S   -> q1  [ label = "a" ];
    S   -> q2f [ label = "b" ];
    S   -> q2  [ label = "c" ];
    q1  -> q2  [ label = "b" ];
    q2f -> f   [ label = "b" ];
    q2f -> q2  [ label = "c" ];
    q2  -> f   [ label = "b" ];
    q2  -> q2  [ label = "c" ];
}
```

April-21-2019 How to Draw FSM Latex

https://martin-thoma.com/how-to-draw-a-finite-state-machine/

```
\documentclass{scrartcl}
\usepackage{tikz}
\usetikzlibrary{arrows,automata}

\begin{document}
\begin{tikzpicture}[>=stealth',shorten >=1pt,auto,node distance=2cm]
  \node[initial,state,accepting] (S)      {$S$};
  \node[state]         (q1) [right of=S]  {$q_1$};
  \node[state]         (q2) [right of=q1] {$q_2$};

  \path[->]          (S)  edge [loop above] node {a} (S);
  \path[->, dashed]  (S)  edge              node {a} (q1);
  \path[->, dotted]  (q1) edge [bend left]  node {a} (S);
  \path[->>, dotted] (q1) edge              node {b} (q2);
  \path               (q2) edge [loop above] node {b} (q2)
            edge [bend left]  node {b} (q1);
\end{tikzpicture}
\end{document}digraph finite_state_machine {
    rankdir=LR;
    size="8,5"

    node [shape = doublecircle, label="{f}", fontsize=12] f;
    node [shape = doublecircle, label="{q2, f}", fontsize=10] q2f;

    node [shape = circle, label="S", fontsize=14] S;
    node [shape = circle, label="{q1}", fontsize=12] q1;
    node [shape = circle, label="{q2}", fontsize=12] q2;
```

```
    node [shape = point ]; qi
        qi -> S;

        S   -> q1  [ label = "a" ];
        S   -> q2f [ label = "b" ];
        S   -> q2  [ label = "c" ];

        q1  -> q2  [ label = "b" ];

        q2f -> f   [ label = "b" ];
        q2f -> q2  [ label = "c" ];

        q2  -> f   [ label = "b" ];
        q2  -> q2  [ label = "c" ];
    }
```



April-21-2019 Graphviz and Python

https://pypi.org/project/graphviz/ Simple Python interface for Graphviz

Create a graph object:

```
>>> from graphviz import Digraph

>>> dot = Digraph(comment='The Round Table')

>>> dot  #doctest: +ELLIPSIS
<graphviz.dot.Digraph object at 0x...>
```

Add nodes and edges:

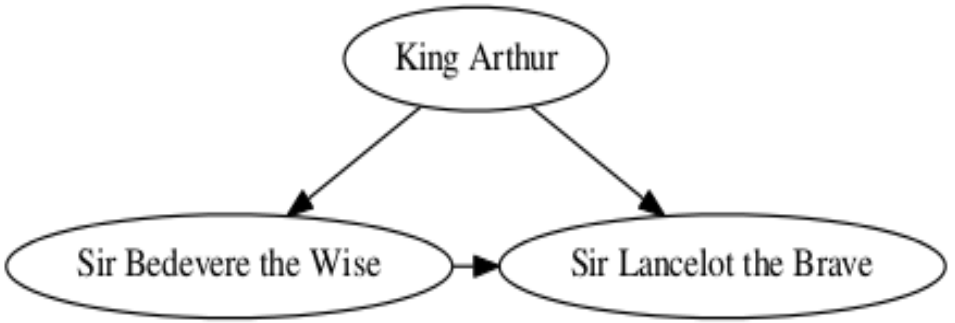
```
>>> dot.node('A', 'King Arthur')
>>> dot.node('B', 'Sir Bedevere the Wise')
>>> dot.node('L', 'Sir Lancelot the Brave')

>>> dot.edges(['AB', 'AL'])
>>> dot.edge('B', 'L', constraint='false')
```

Check the generated source code:

```
>>> print(dot.source)
// The Round Table
digraph {
    A [label="King Arthur"]
    B [label="Sir Bedevere the Wise"]
    L [label="Sir Lancelot the Brave"]
    A -> B
    A -> L
    B -> L [constraint=false]
}
```

Save and render the source code, view the result:

```
>>> dot.render('test-output/round-table.gv', view=True)
```



April-21-2019 ns2 Simulator Installation

https://www.linuxquestions.org/questions/ubuntu-63/ns2-34-installation-on-ubuntu-14-04-lts-done-4175508517/

Step by step installation of ns-2.34 on ubuntu 14.04

http://surajpatilworld.blogspot.com/2015/02/step-by-step-installation-of-ns-234-on.html