OpenFST Library

SOME of the available FST operations

https://www.openfst.org

OpenFst Library

- *Definition of the symbols* (syms.txt)

```
red       1
green     2
blue      3
yellow    4
```

- *Definition of a transducer* (t.txt)

```
0   0   red     yellow     .5
0   1   green   blue       .3
1   2   blue    green
1   2   yellow  red        .6
2   .8
```

- *Generation of the binary version* (t.fst)
  - **fstcompile** --isymbols=syms.txt --osymbols=syms.txt **t.txt | fstarcsort > t.fst**
- *Generation of the graphical version* (t.pdf)
  - **fstdraw** --portrait --isymbols=syms.txt --osymbols=syms.txt **t.fst | dot -Tpdf > t.pdf**

OpenFst Library

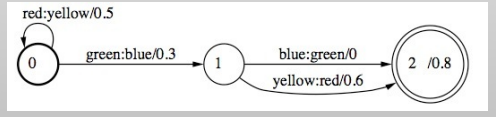
- *Definition of the symbols* (syms.txt)

```
red       1
green     2
blue      3
yellow    4
```

- *Definition of a transducer* (t.txt)

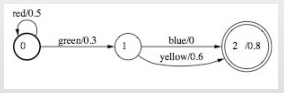
```
0   0   red     yellow     .5
0   1   green   blue       .3
1   2   blue    green
1   2   yellow  red        .6
2   .8
```

- *Graphical version* (t.pdf)

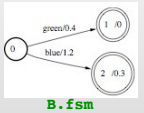


UNION of TRANSDUCES

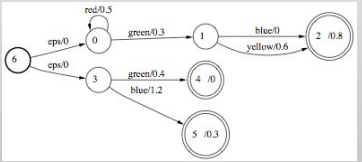
```
fstunion A.fsm B.fsm > C.fsm
```



A.fsm



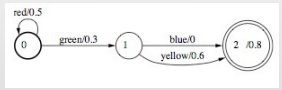
B.fsm



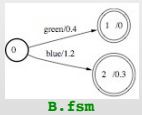
C.fsm

CONCATENATION of TRANSDUCES

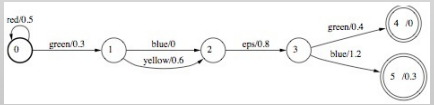
```
fstconcat A.fsm B.fsm > C.fsm
```



A.fsm



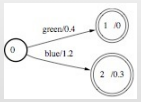
B.fsm



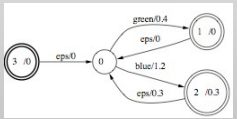
C.fsm

CLOSURE of TRANSDUCES

```
fstclosure B.fsm > C.fsm
```



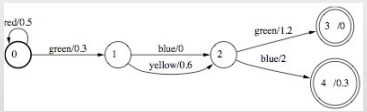
B.fsm



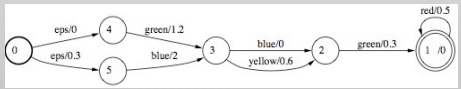
C.fsm

"REVERSAL" of TRANSDUCES

```
fstreverse A.fsm > C.fsm
```



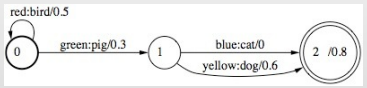
A.fsm



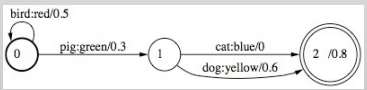
C.fsm

INVERTION of TRANSDUCES

```
fstinvert A.fsm > C.fsm
```



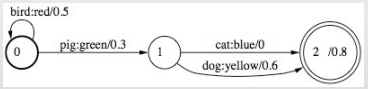
A.fsm



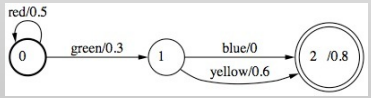
C.fsm

PROJECTION (output) of TRANSDUCES

```
fstproject --project_type=output A.fsm > C.fsm
fstproject --project_output=true A.fsm > C.fsm (in previous versions)
```



A.fsm



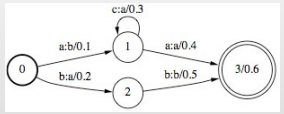
C.fsm

COMPOSITION of TRANSDUCES

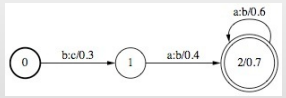
- *Para obter o transdutor composto:*
  - Cria um novo estado (x,y) para todos os pares de estados $x \epsilon Q_1$ e $y \epsilon Q_2$
  - A função de transição da composição é definida por
    $\delta((x,y),i:o)=(v,z)$ se $\delta_1(x,i:c) = v$ e $\delta_2(y,c:o) = z$

COMPOSITION of TRANSDUCES

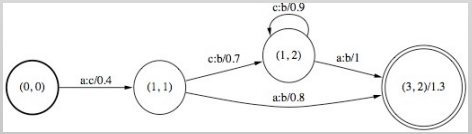
```
fstcompose A.fsm B.fsm > C.fsm
```



A.fsm



B.fsm



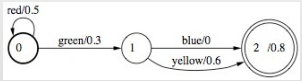
C.fsm

INTERSECTION of TRANSDUCES

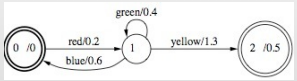
- *O algoritmo de intersecção apenas considera o produto cartesiano dos estados*
  - Para cada estado q_i do primeiro transdutor, e q_j do segundo transdutor, cria-se um novo estado q_{ij}
  - Para o símbolo de entrada a, se o primeiro transdutor transitava para o estado q_n e o segundo transdutor transitava para o estado q_m o novo transdutor transita para o estado q_{nm}

INTERSECTION of TRANSDUCES

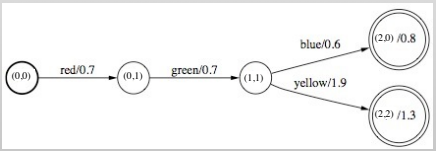
```
fstintersect A.fsm B.fsm > C.fsm
```



A.fsm



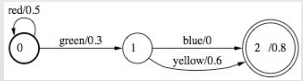
B.fsm



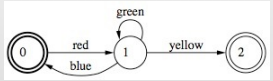
C.fsm

DIFERENÇA DE TRANSDUTORES

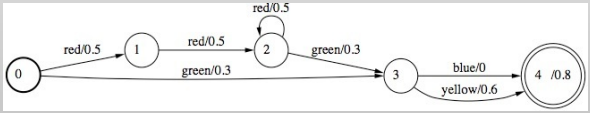
```
fsmdifference A.fsm B.fsm > C.fsm
```



A.fsm



B.fsm

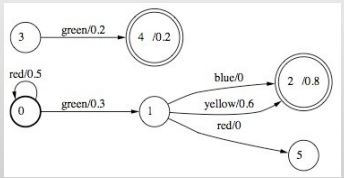


C.fsm

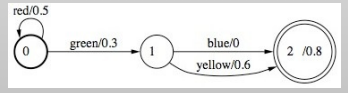
REMOVAL OF INACCESSIBLE STATES

- With the option -t, returns (exit status) 1 if the output has no states, useful to test the empty output ...

```
fstconnect A.fsm > C.fsm
```



A.fsm



C.fsm