

Carmel

The format of FSA / FST

final_state

(from_state1 (to_state1 "input_symbol" "output_symbol"? weight?)*)

(from_state2 (to_state2 "input_symbol" "output_symbol"? weight?)*)

...

A state can be a number or string.

The from_state in the first edge-line is the start state.

ϵ is represented as *e*

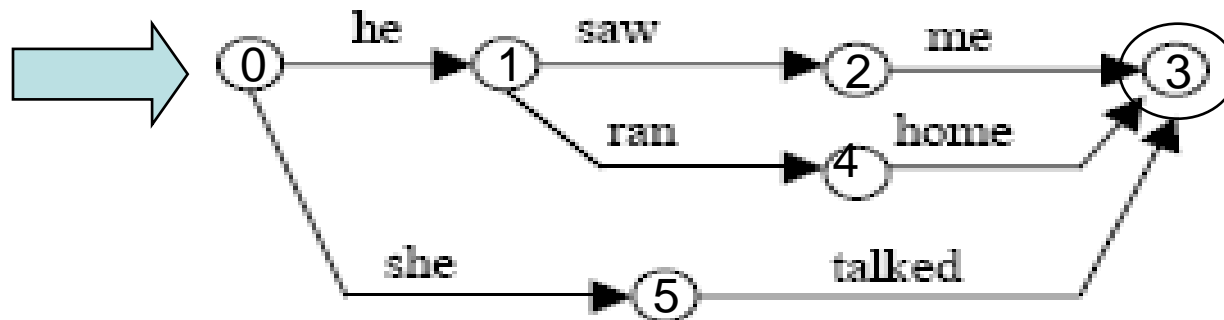
output_symbol and prob are optional.

An FSA example: fsa1

Filename: fsa1

3

```
(0 (1 "he"))  
(1 (2 "saw"))  
(2 (3 "me"))  
(1 (4 "ran"))  
(4 (3 "home"))  
(0 (5 "she"))  
(5 (3 "talked"))
```

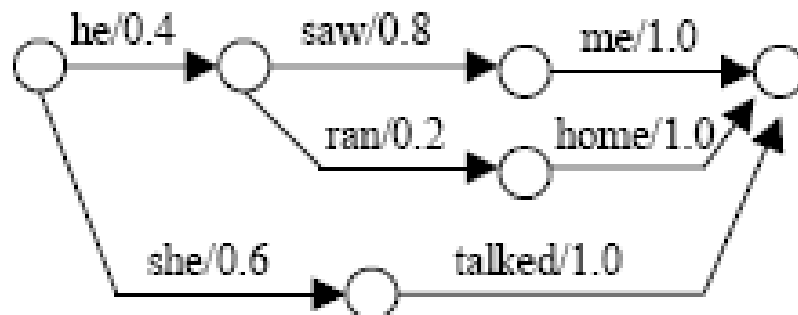


An WFSA example: wfsa1

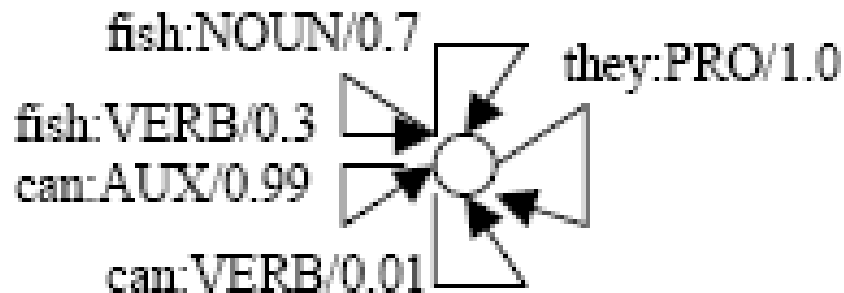
```
##### Filename: wfsa1 #####
```

```
3
```

```
(0 (1 "he" 0.4))  
(1 (2 "saw" 0.8))  
(2 (3 "me" 1.0))  
(1 (4 "ran" 0.2))  
(4 (3 "home" 1.0))  
(0 (5 "she" 0.6))  
(5 (3 "talked" 1.0))
```



An WFST example: wfst1



~~~~~ Filename: wfst1 ~~~~~

S

```
(S (S "they" "PRO" 1.0))  
(S (S "can" "AUX" 0.99))  
(S (S "can" "VERB" 0.01))  
(S (S "fish" "NOUN" 0.7))  
(S (S "fish" "VERB" 0.3))
```

# To use Carmel

- `carmel fst1 fst2`  
=> return a new fst, which composes `fst1` and `fst2`.
- `carmel -k N wfst1`  
=> return the `N` most probable paths
- `carmel -Ok N wfst1`  
=> return the `N` most probable output strings

# To use Carmel (cont)

- `cat input_file | carmel -sli fst1`
  - create a `foo_fst` that corresponds to the first line in `input_file`
  - `carmel foo_fst fst1`
  - Ex: `input_file` is  
“they” “can” “fish”
- `cat input_file | carmel -sri fst1`
  - create a `foo_fst` that corresponds to the first line in `input_file`
  - `carmel fst1 foo_fst`
  - Ex: `input_file` is  
“PRO” “AUX” “VERB”
- `cat input_file | carmel -b -sli fst1`