Language Understanding Systems

WFST Solutions

Evgeny A. Stepanov

SISL, DISI, UniTN & VUI, Inc.
evgeny.stepanov@unitn.it

FSA Exercises

- Mohri et al. (1996) FSM Toolkit Exercises

Exercise 1

Given the alphabet $L = \{a, b, ..., z, A, B, ..., Z, < space >\}$, create an automaton that:

a) accepts a letter in L (including space).

b) accepts a single space.

c) accepts a capitalized word (where a word is a string of letters in L excluding space and a capitalized word has its initial letter uppercase and remaining letters lowercase).

d) accepts a word containing the letter a.

Exercise 1: *Solution* - Step 0

Create a lexicon file:

```
#!/bin/bash
# epsilon
echo "<eps> 0"

cnt=1

# lowercase letters
for c in {a..z} ; do echo "$c $cnt" ; ((cnt++)) ; done

# capital letters
for c in {A..Z} ; do echo "$c $cnt" ; ((cnt++)) ; done

# space
echo "<space> $cnt"
```

Exercise 1: *Solution* - (a)

```
#!/bin/bash

# since only 1 letter should be accepted
# state transitions are all from 0 to 1

# lowercase letters
for c in {a..z} ; do echo "0 1 $c" ; done

# uppercase letters
for c in {A..Z} ; do echo "0 1 $c" ; done

# space
echo "0 1 <space>"

# final state is 1
echo "1"
```

Exercise 1: *Solution* - (b)

```
#!/bin/bash

# since only <space> should be accepted
# single transition from 0 to 1

# space
echo "0 1 <space>"

# final state is 1
echo "1"
```

Exercise 1: *Solution* - (c)

```
#!/bin/bash

# capitalized word:
# capital letter transitions from 0 to 1
# lowercase letter transition from 1 to 1

# uppercase letters
for c in {A..Z} ; do echo "0 1 $c" ; done

# lowercase letters
for c in {a..z} ; do echo "1 1 $c" ; done

# final state is 1
echo "1"
```

Exercise 1

Given the alphabet $L = \{a, b, ..., z, A, B, ..., Z, < space >\}$, create an automaton that:

a) accepts a letter in L (including space).

b) accepts a single space.

c) accepts a capitalized word (where a word is a string of letters in L excluding space and a capitalized word has its initial letter uppercase and remaining letters lowercase).

d) accepts a **word** containing the letter a.

Exercise 1: *Solution* - (d)

```
#!/bin/bash
# word containing ’a’:
# depends on whether we consider a ’word’ to mean:
# (a) capitalized + lowercase
# (b) lowercase only (since not specified)

# capital letters transition from 0 to 1 (once)
for c in {A..Z} ; do echo "0 1 $c" ; done
# lowercase letters except ’a’
for c in {b..z} ; do echo "0 1 $c" ; done
# lowercase letters except ’a’ again, from 1 to 1
for c in {b..z} ; do echo "1 1 $c" ; done
# ’a’ both from 0 and 1 to 2
echo "0 2 a"
echo "1 2 a"
# lowercase letters loop on 2
for c in {a..z} ; do echo "2 2 $c" ; done
# final state
echo "2"
```

Exercise 1: *Solution*

Compile the defined automata as:

```
fstcompile --acceptor --isymbols=sym.lex fsm.txt > fsm.fsa
```

Exercise 2

Using the automata in Exercise 1 as the building blocks, use appropriate FSM operations on them to create an automaton that:

- (a) accepts zero or more capitalized words followed by spaces.
- (c) accepts a word that is capitalized and contains the letter a.
- (d) accepts a word that is capitalized or does not contain an a.

Exercise 2: *Solution*

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

```
fstconcat 1c.fsa 1b.fsa | fstclosure > 2a.fsa
```

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

```
fstconcat 1c.fsa 1b.fsa | fstclosure > 2a.fsa
```

(c) accepts a word that is capitalized and contains the letter *a*

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

```
fstconcat 1c.fsa 1b.fsa | fstclosure > 2a.fsa
```

(c) accepts a word that is capitalized and contains the letter *a*

NOTE: indicates that 1d is not capitalized only

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

```
fstconcat 1c.fsa 1b.fsa | fstclosure > 2a.fsa
```

(c) accepts a word that is capitalized and contains the letter *a*

NOTE: indicates that 1d is not capitalized only

```
fstintersect 1c.fsa 1d.fsa > 2c.fsa
```

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

```
fstconcat 1c.fsa 1b.fsa | fstclosure > 2a.fsa
```

(c) accepts a word that is capitalized and contains the letter *a*

NOTE: indicates that 1d is not capitalized only

```
fstintersect 1c.fsa 1d.fsa > 2c.fsa
```

(d) accepts a word that is capitalized or does not contain an *a*

Exercise 2: *Solution*

(a) accepts zero or more capitalized words followed by spaces

```
fstconcat 1c.fsa 1b.fsa | fstclosure > 2a.fsa
```

(c) accepts a word that is capitalized and contains the letter *a*

NOTE: indicates that 1d is not capitalized only

```
fstintersect 1c.fsa 1d.fsa > 2c.fsa
```

(d) accepts a word that is capitalized or does not contain an *a*

```
fstdifference 1c.fsa 1d.fsa | fstunion 1c.fsa - > 2d.fsa
```

Exercise 3

Epsilon-remove, determinize, and minimize each of the automata in Exercise 2. Give the number of states and arcs before and after these operations.