Speech and Language Processing

Lecture 6

Weighted finite state transducer (WFST) and speech decoding

Information and Communications Engineering Course

Takahiro Shinozaki

Lecture Plan (Shinozaki's part)

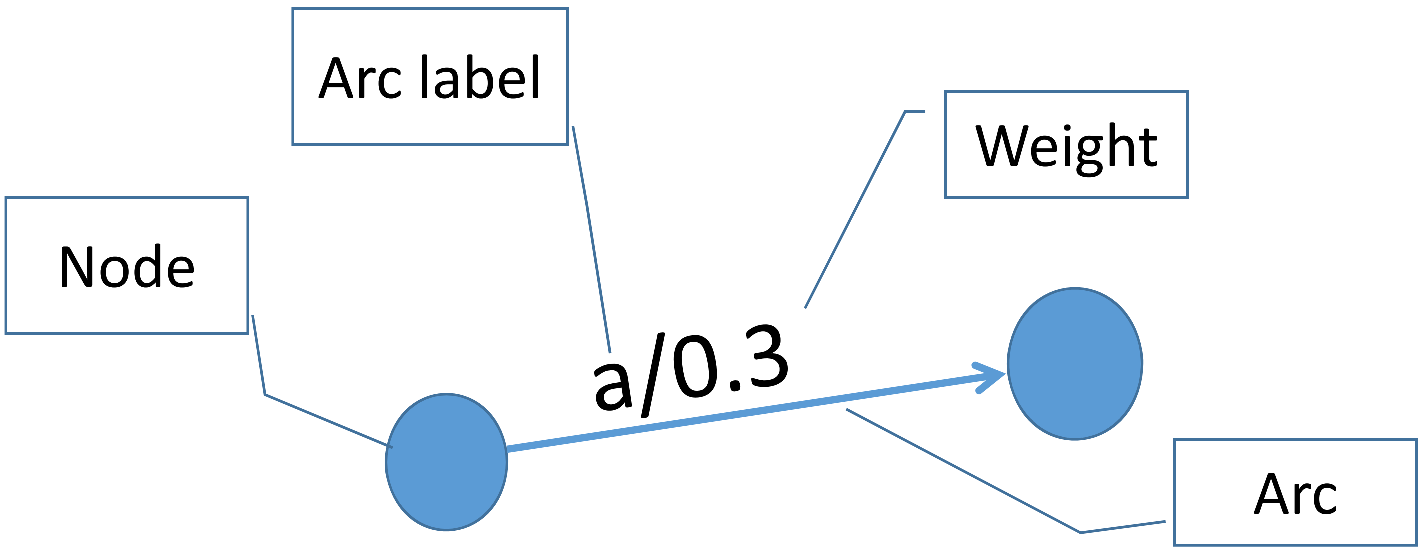
I gives the first 6 lectures about speech recognition. Through these lectures, the backbone of the latest speech recognition techniques is explained.

1. 10/19 (remote)
   Speech recognition based on GMM, HMM, and N-gram
2. 10/19 (remote)
   Maximum likelihood estimation and EM algorithm
3. 10/20 (remote)
   Bayesian network and Bayesian inference
4. 10/20 (remote)
   Variational inference and sampling
5. 10/22 (remote)
   Neural network based acoustic and language models
6. 10/22 (remote)
   Weighted finite state transducer (WFST) and speech decoding

Weighted finite state transducer (WFST)

Weighted Finite State Acceptor (WFSA)

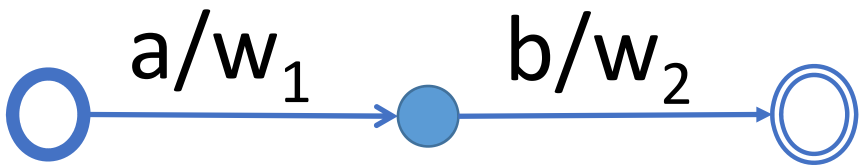
- Defined by a set of nodes, arcs, arc labels, and arc weights
- An extension of finite state automaton
- Accepts a sequence of symbols (string) assigning a weight to the sequence



Variations of Weight Types

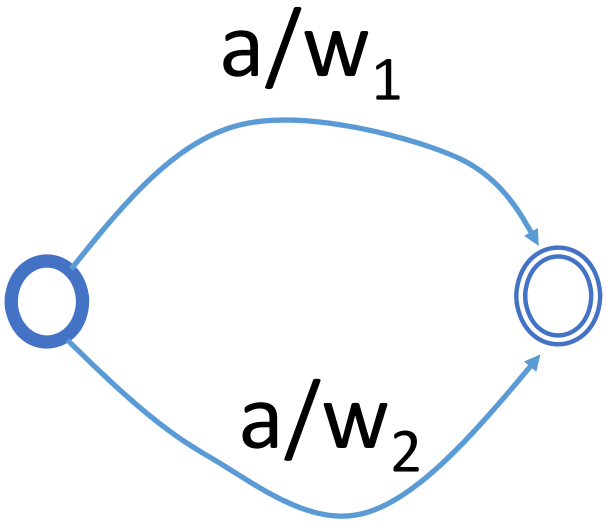
Name	Set	Plus $\oplus$	Times $\otimes$	Zero	One
Boolean	{0,1}	$\vee$	$\wedge$	0	1
Real	{0, ∞}	+	*	0	1
Log	{-∞, ∞}	$-\log(e^{-x}+e^{-y})$	+	∞	0
Tropical	{-∞, ∞}	min	+	∞	0

Weight of Acceptance (Tropical Weight)



Weight of accepting <a,b> is :

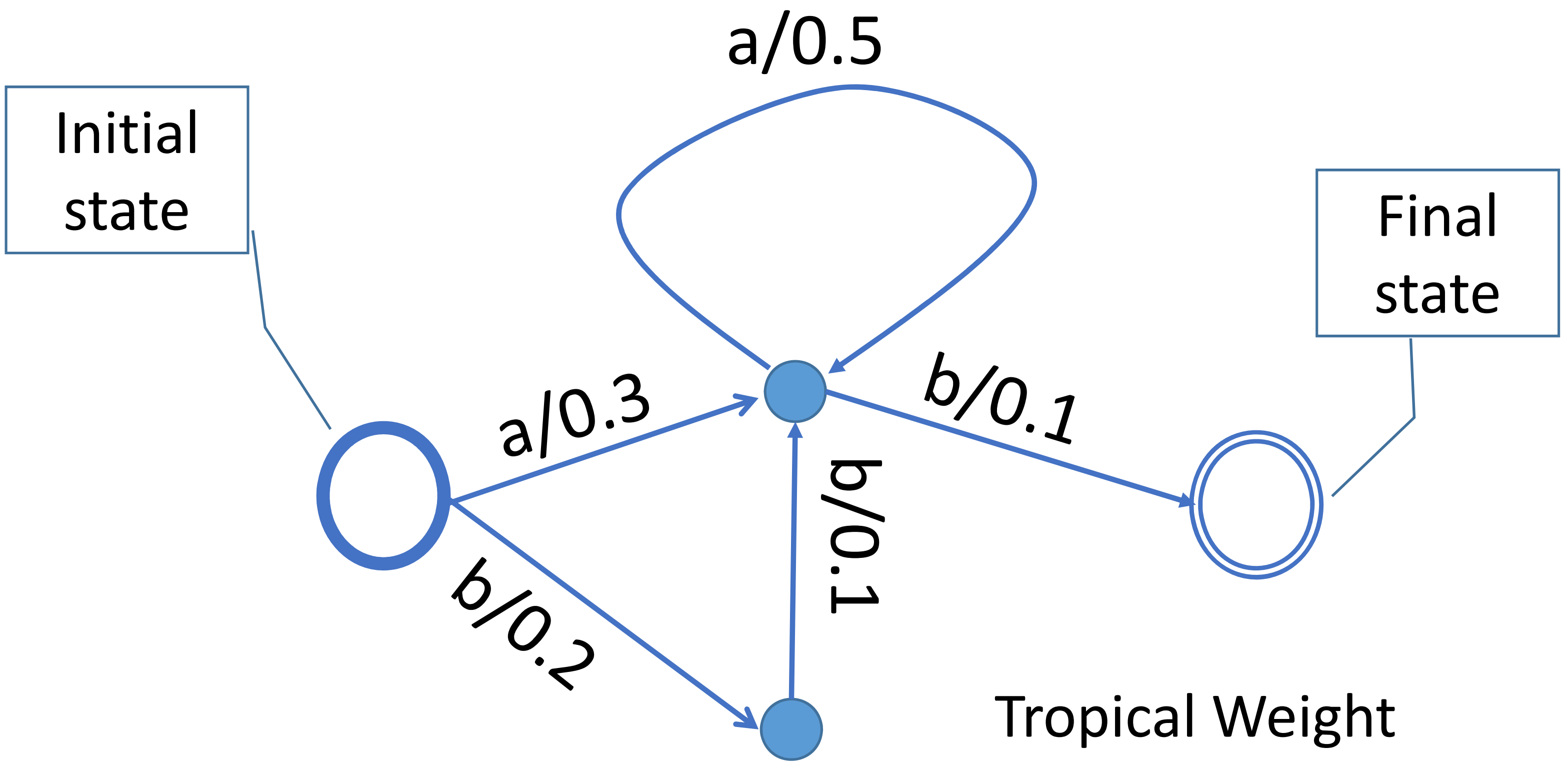
$w_1 \otimes w_2 = w_1 + w_2$



Weight of accepting <a> is :

$w_1 \oplus w_2 = \min\{w_1, w_2\}$

Example

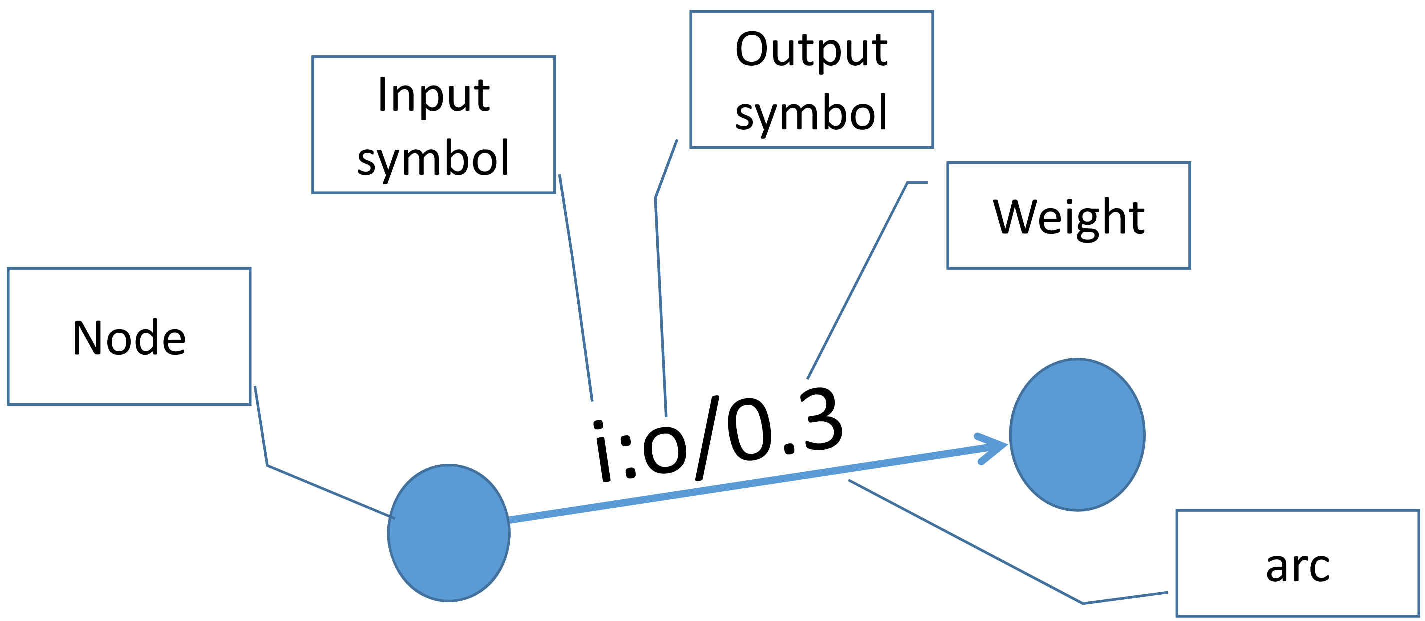


<a, a, a, b> (1.4)

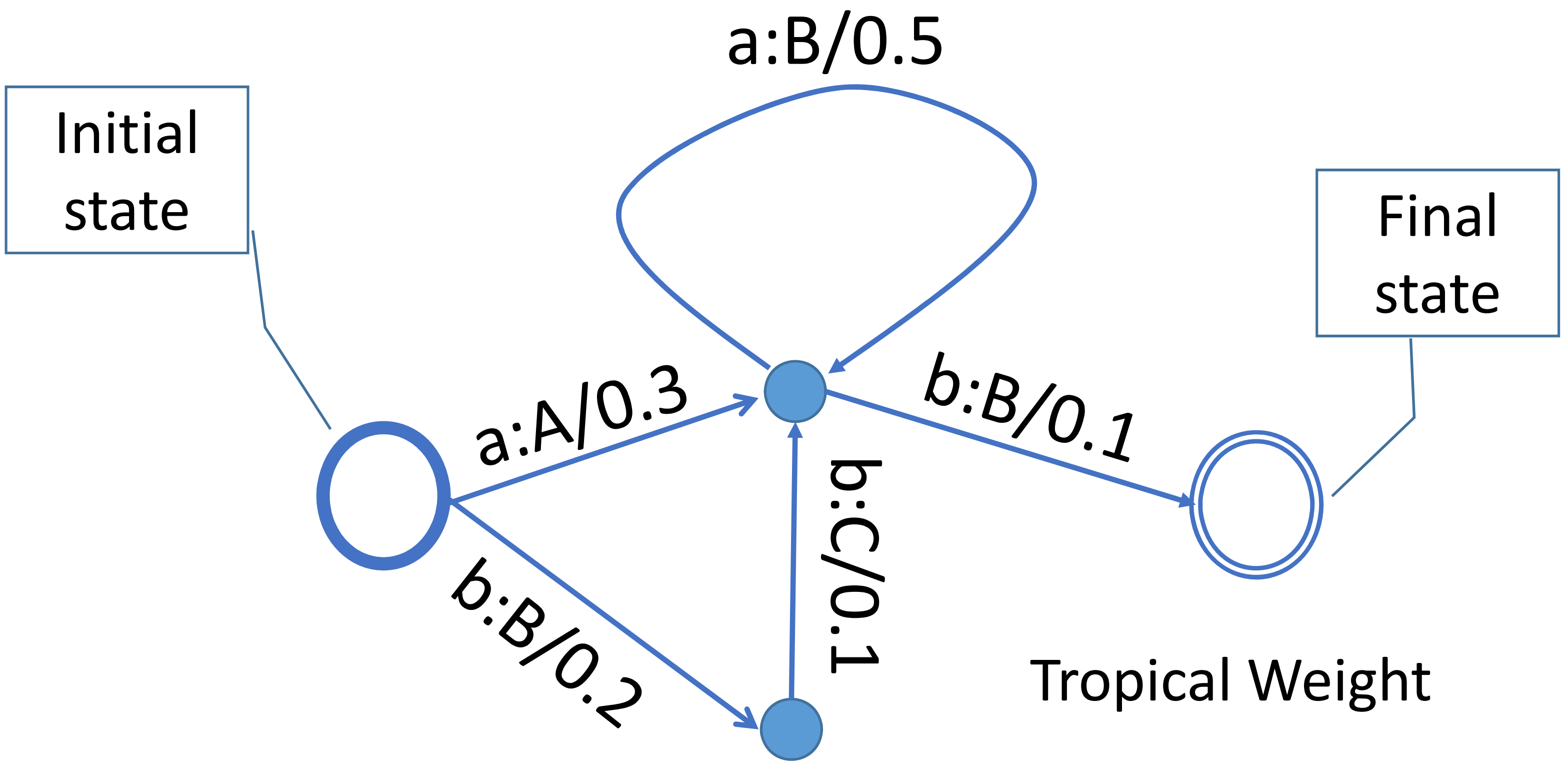
<b, b, b> (0.4)

Weighted Finite State Transducer

- Defined by a set of nodes, arcs, input/output arc labels, and arc weights
- An extension of finite state acceptor
- Transduces an input string to an output string, assigning a weight



Example

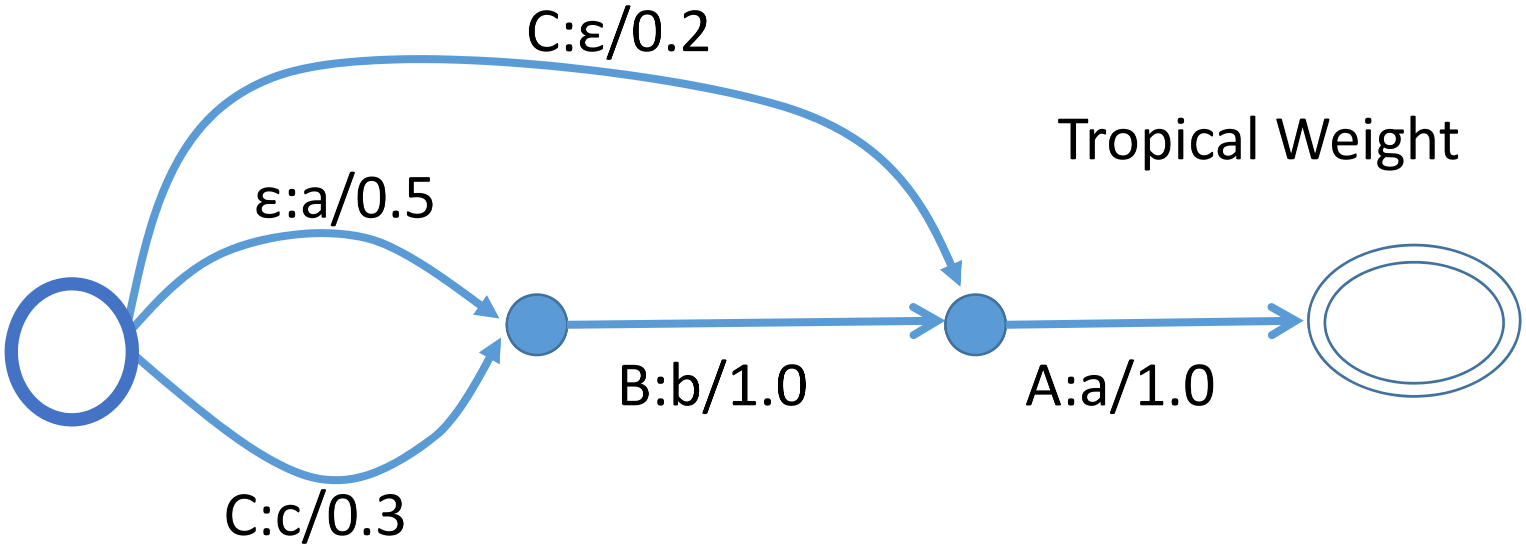


<a, a, a, b> → <A, B, B, B> (1.4)

<b, b, b> → <B, C, B> (0.4)

Null (ε) Symbol

- Input symbol is ε : No input
- Output symbol is ε: No output

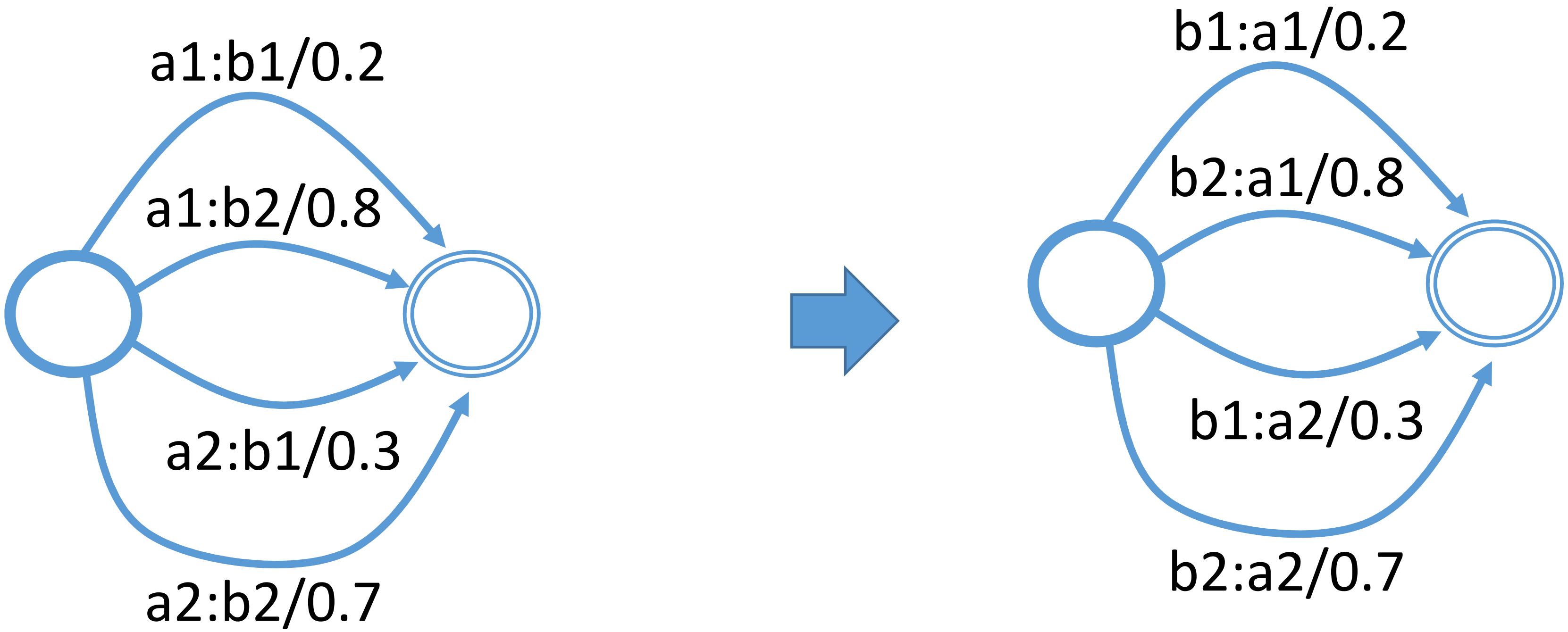


<B, A> → <a, b, a> (2.5)

<C, A> → <a> (1.2)

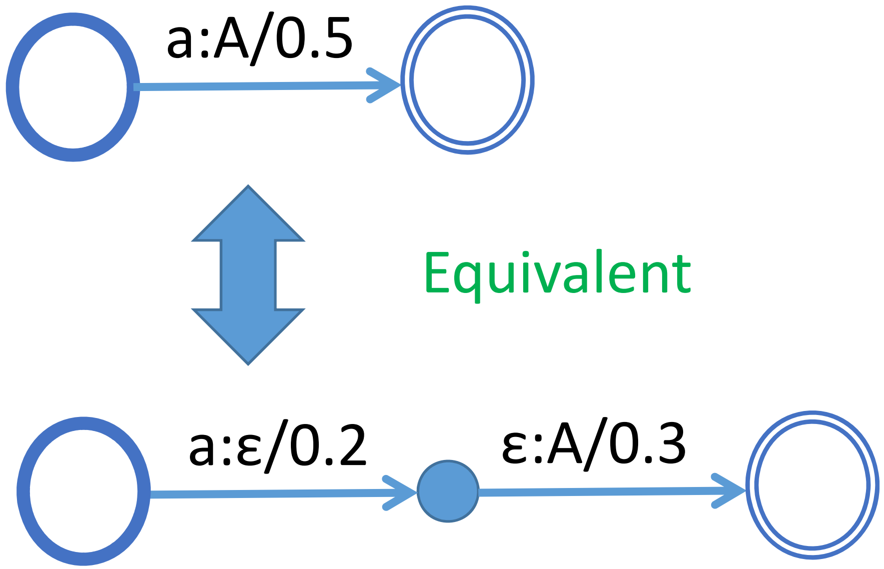
Invert of WFST

- Swap input symbol and output symbol



Equivalence of WFST

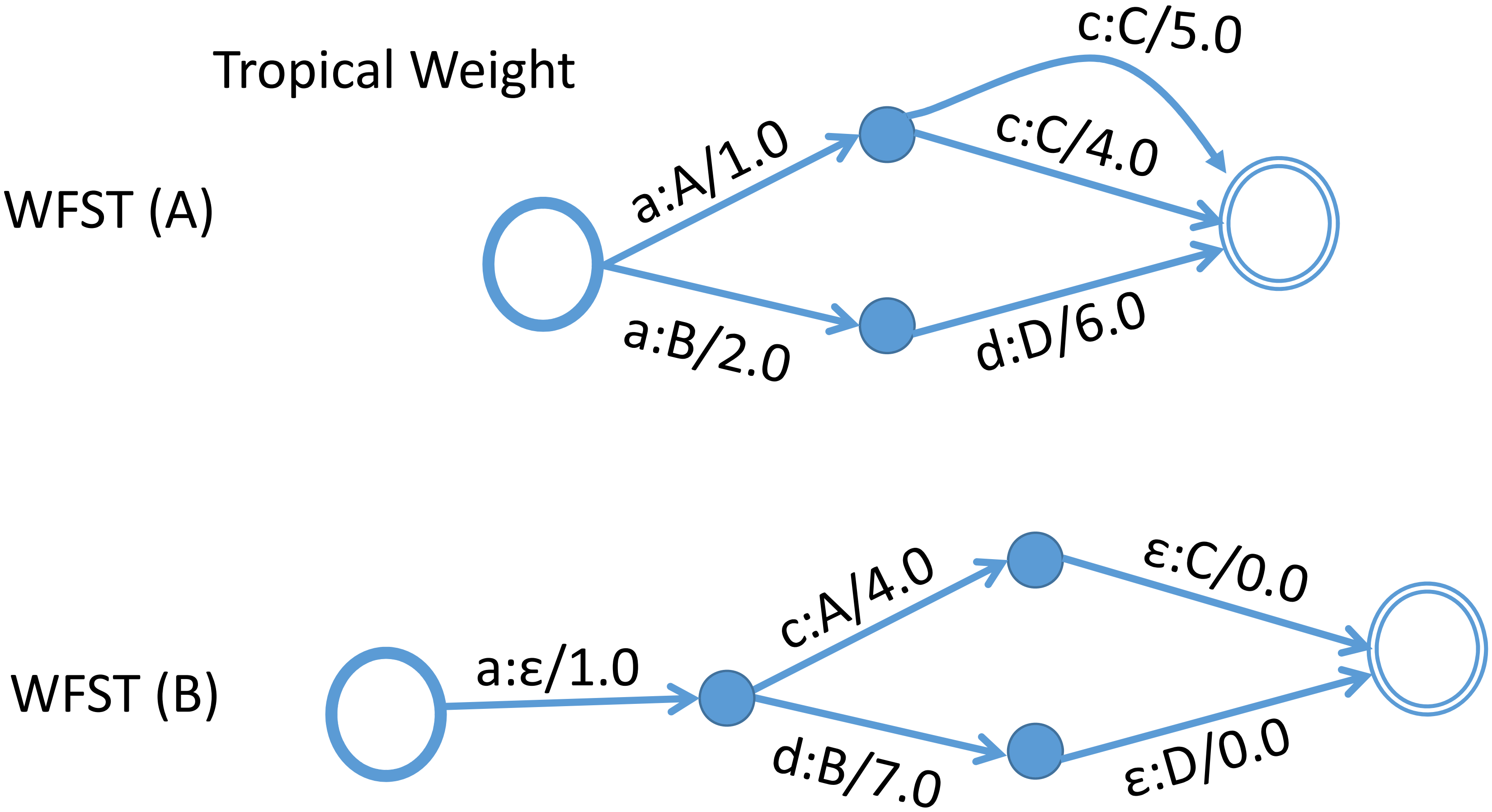
- Two transducers are equivalent if for each input string, they produce the same output strings with the same weight



Tropical Weight

Determinization of WFST

- Makes equivalent WFST so that no state has two transitions with the same input label



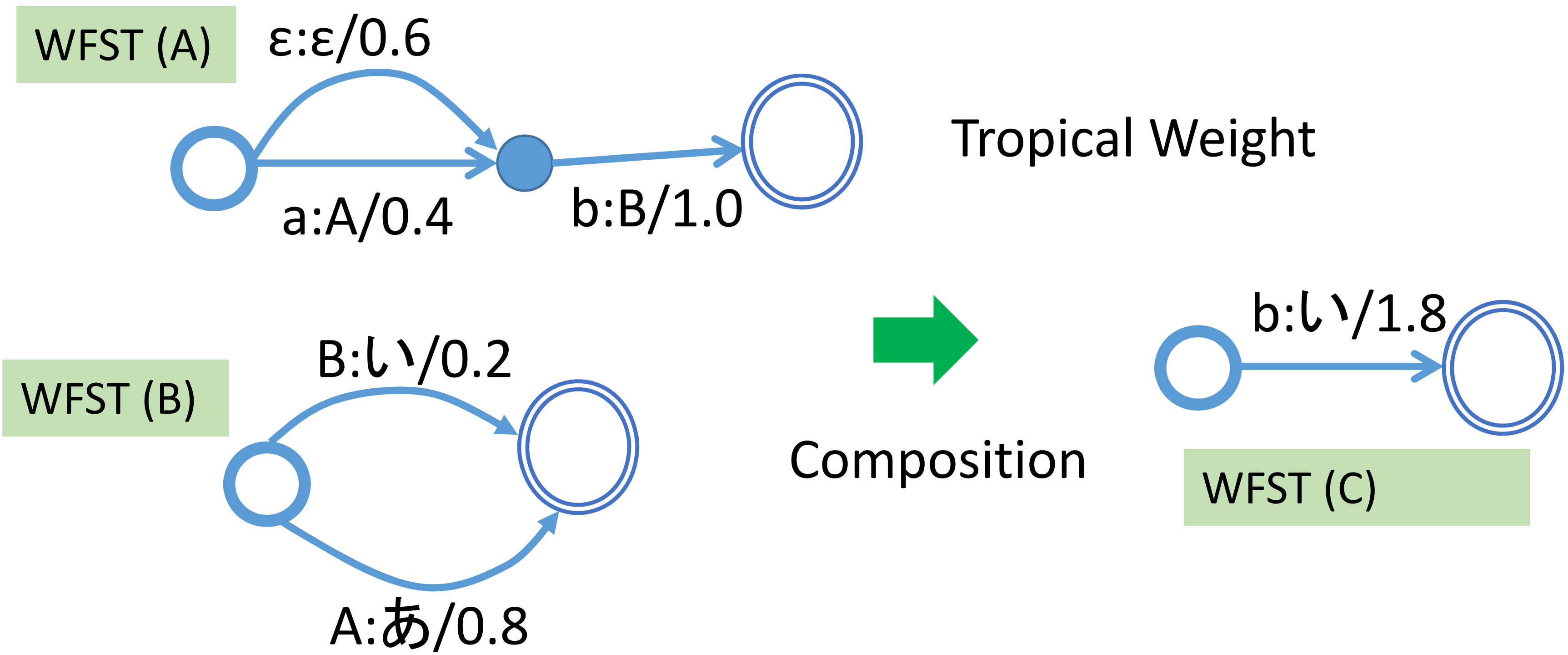
Exercise 6.1

Consider WFST (A) and (B) in previous slide

- Obtain path weight when <a, c> is input to WFST(A)
- Obtain path weight when <a, c> is input to WFST(B)
- Are WFST (A) and (B) equivalent?

Composition of WFSTs

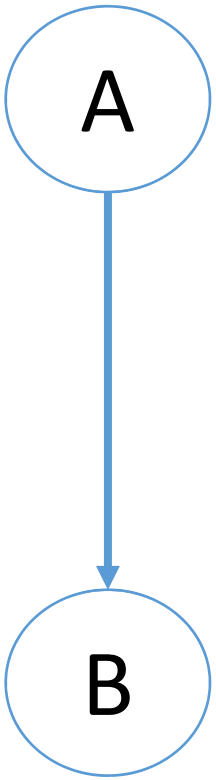
- WFST1 transduces string x to string y with weight w_1
- WFST2 transduces string y to string z with weight w_2
- Composition of WFST1 and WFST2 (WFST1・WFST2) makes WFST that transduces x to z with weight $w_1 \otimes w_2$



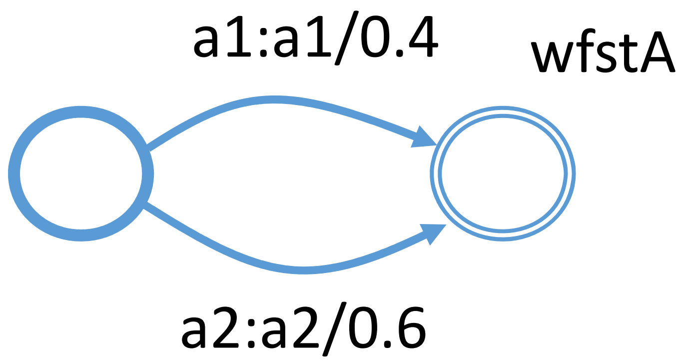
Exercise 6.2

- Check that WFST (C) in previous slide is equivalent to WFST (A) ・WFST(B) by enumerating all possible input strings. Are they equivalent?

Probability Distribution and WFST

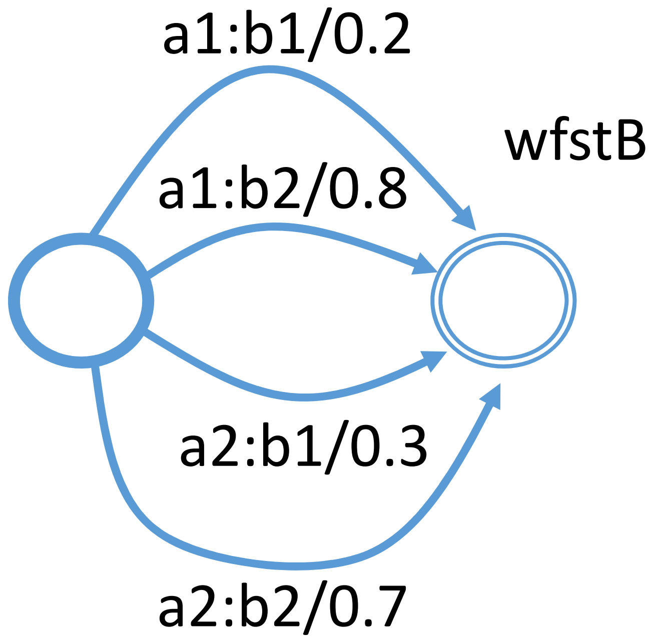


	A=a1	A=a2
P(A)	0.4	0.6



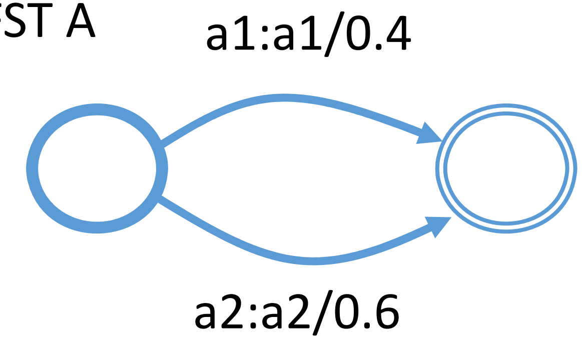
Real Weight

P(B|A)	B=b1	B=b2
A=a1	0.2	0.8
A=a2	0.3	0.7

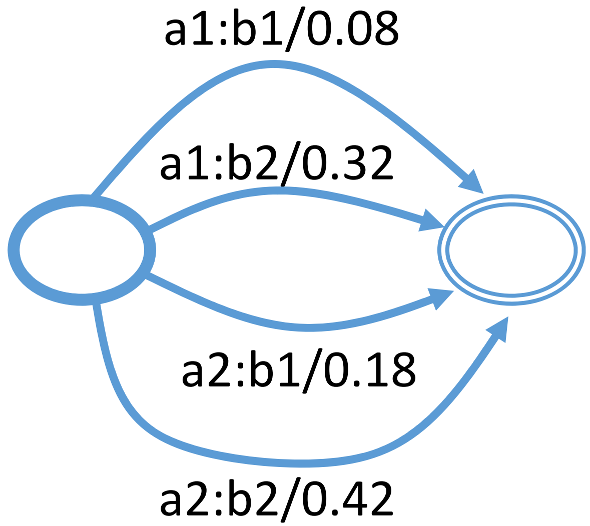


Composition and Joint Probability

WFST A

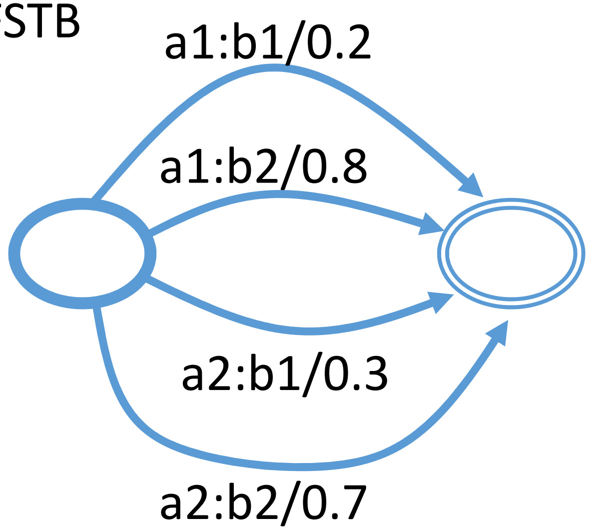


A▪B

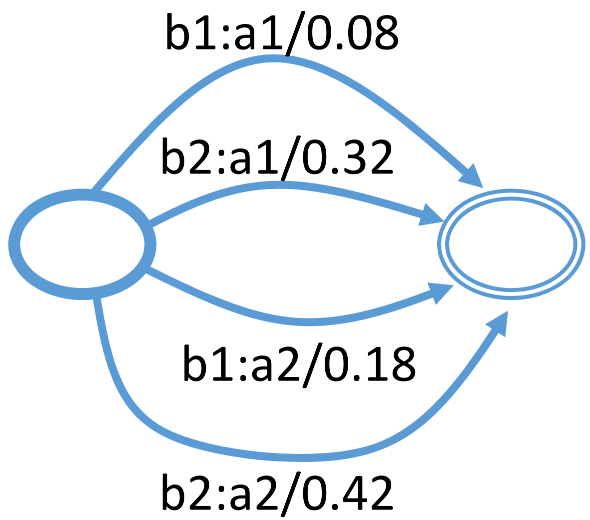


P(A,B)
∝P(B|A=a)

WFSTB



Inv(B)▪Inv(A)



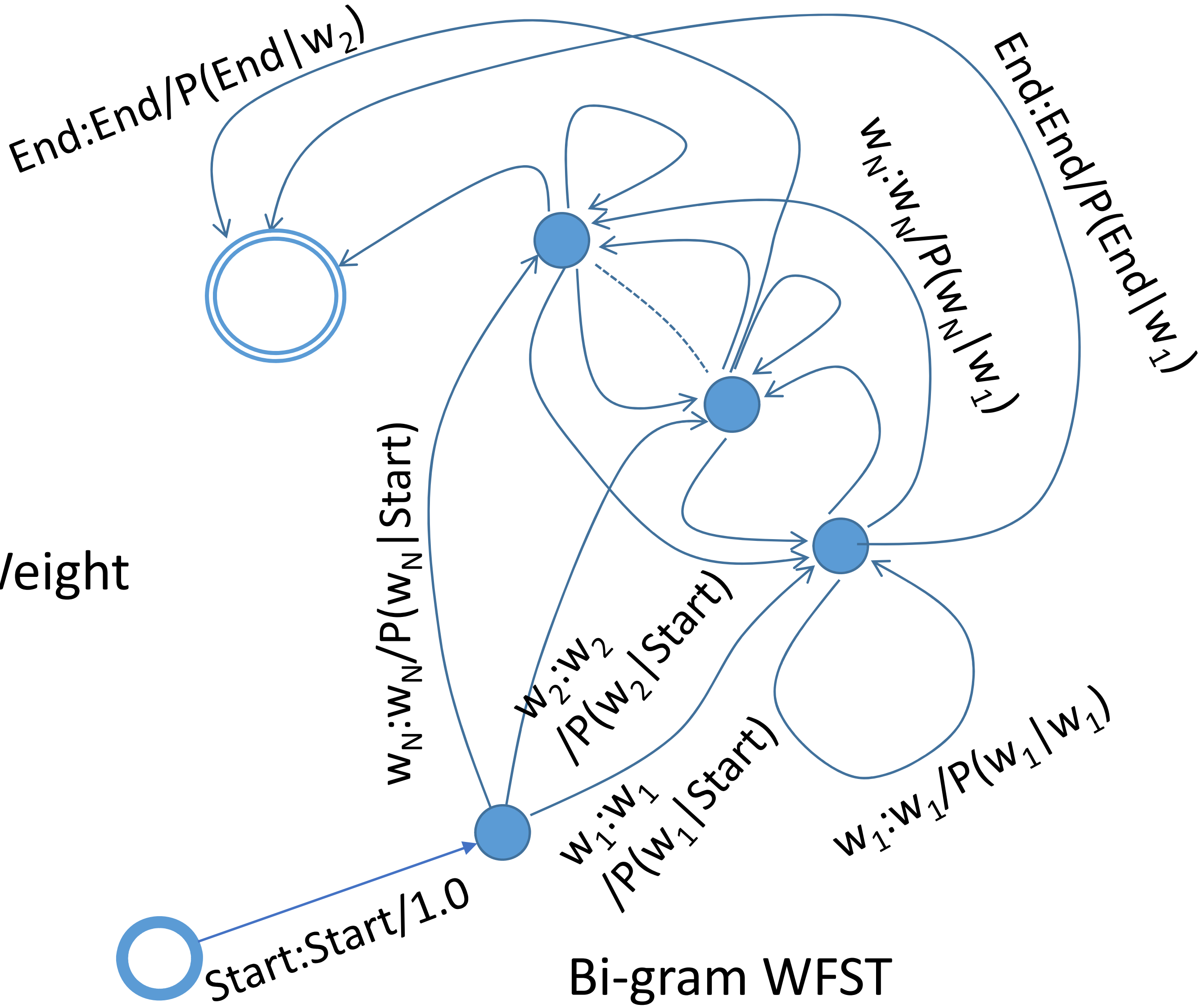
P(A,B)
∝P(A|B=b)

Real Weight

Speech decoding

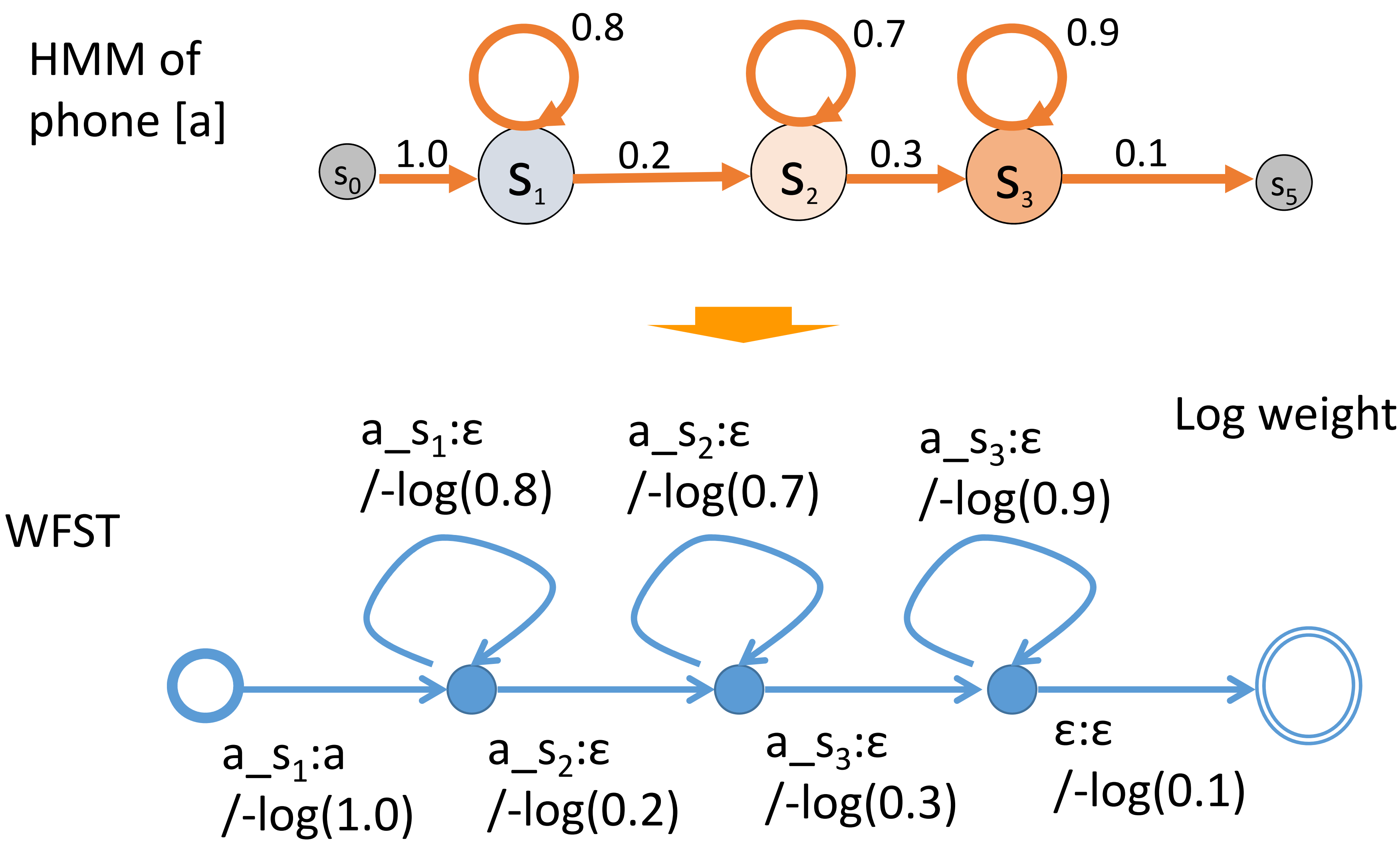
WFST Representation of N-gram

Real Weight



Bi-gram WFST

WFST Representation of HMM



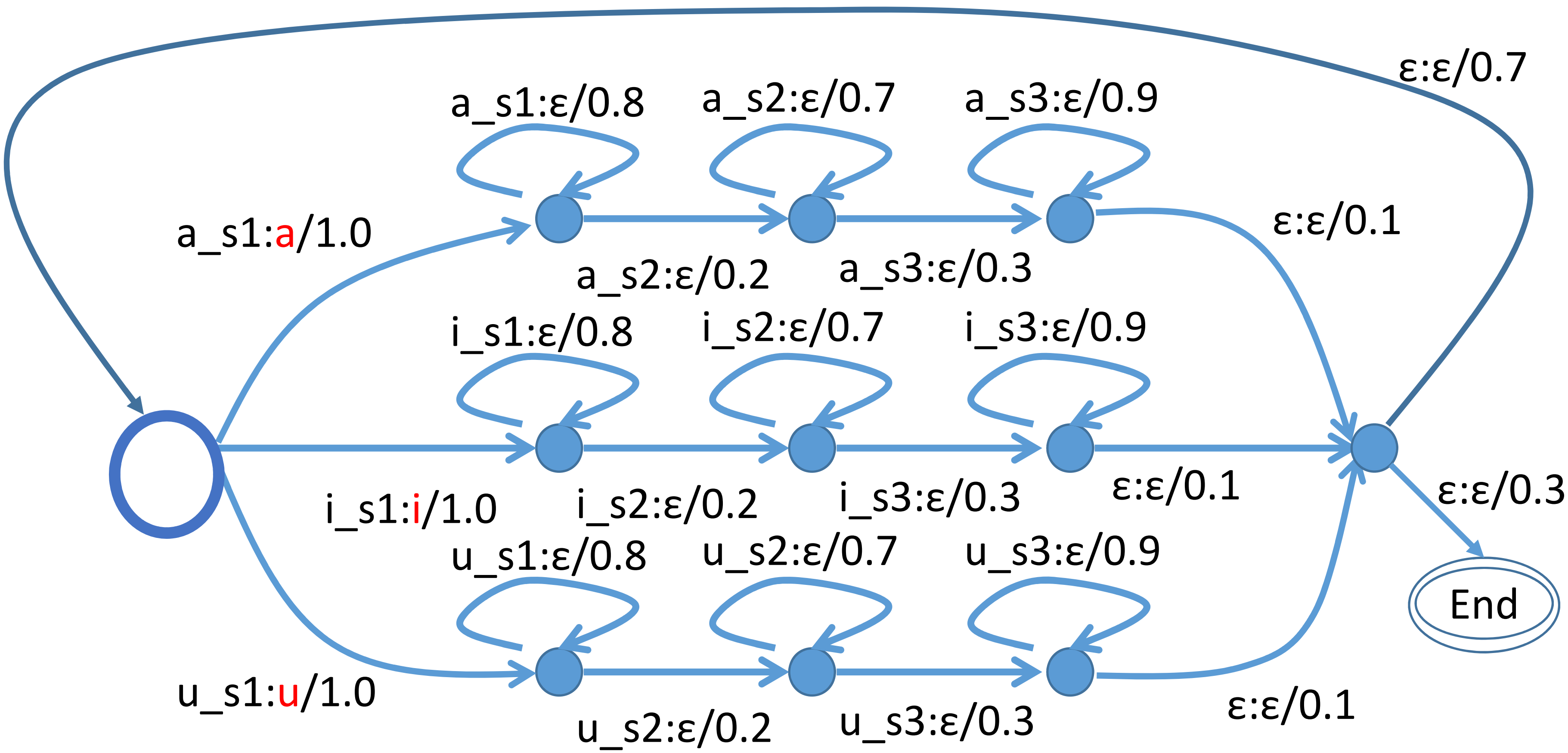
Exercise 6.3

- Make a WFST that represents the following bi-gram

C＼W	today	is	End
Start	0.6	0.3	0.1
today	0.2	0.5	0.3
is	0.4	0.1	0.5

*P(Start)=1.0

WFST To Recognize Phone Sequence

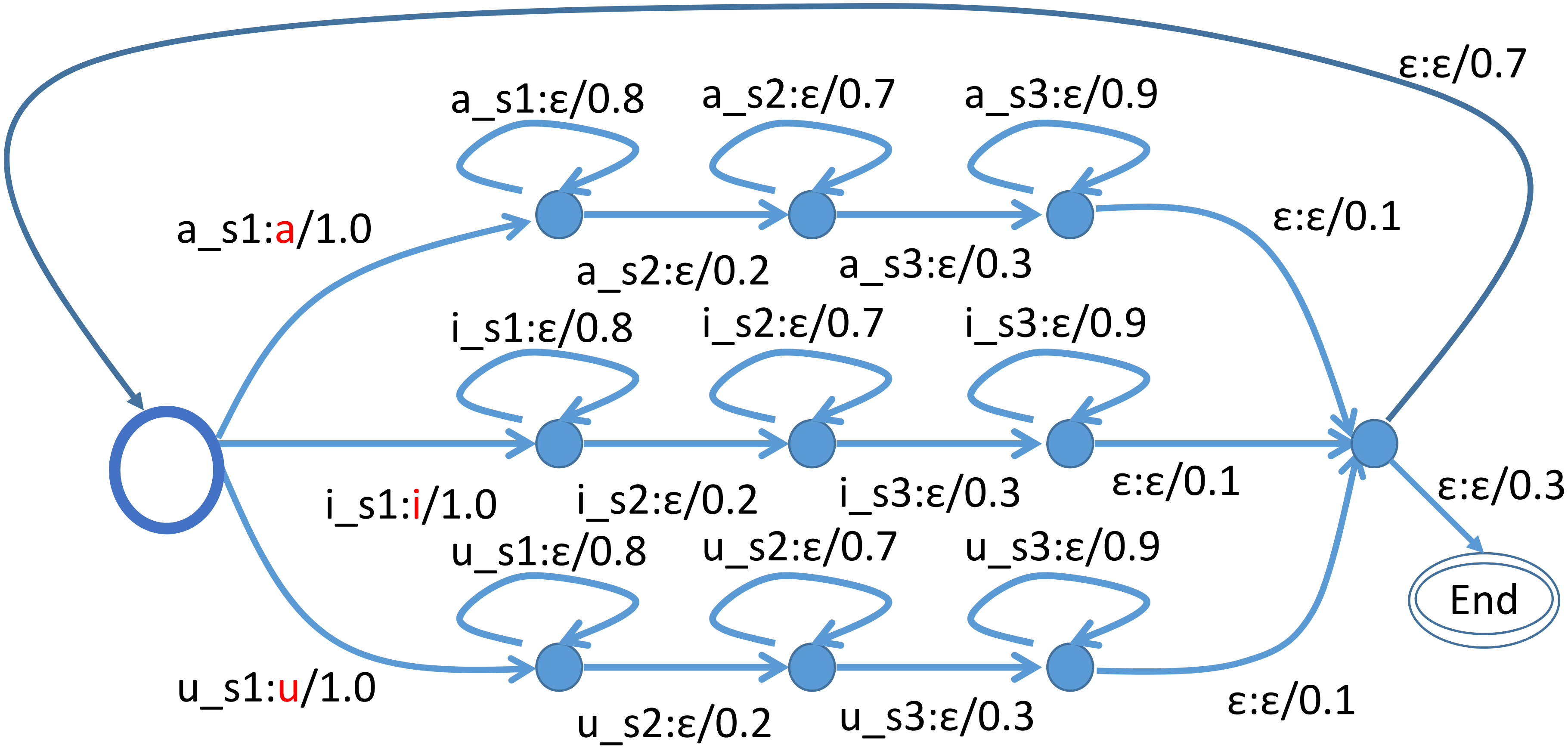


Input: Sequence of HMM state names

Output: Sequence of phonemes

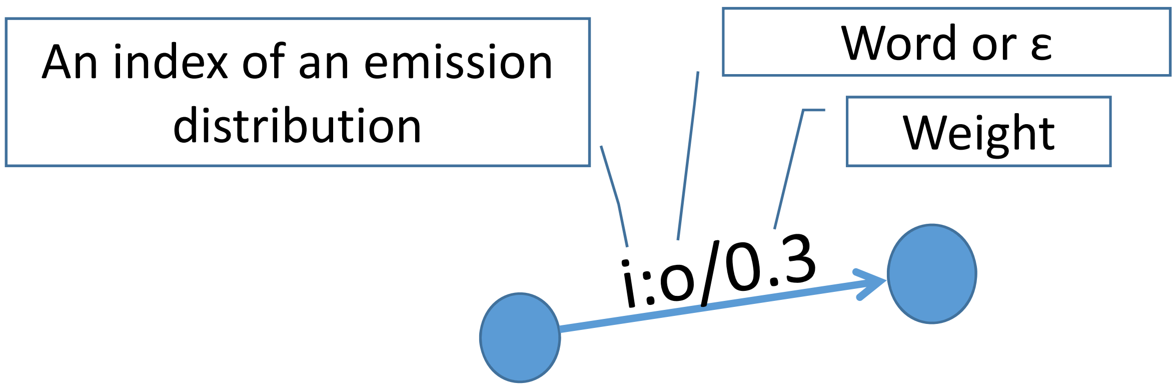
Exercise 6.4

Obtain an output symbol sequence and its weight for an input sequence <i_s1, i_s1, i_s1, i_s2, i_s3, i_s3, i_s1, i_s2, i_s3>. Assume Real weight. When you answer, remove NULL (ε) symbol.



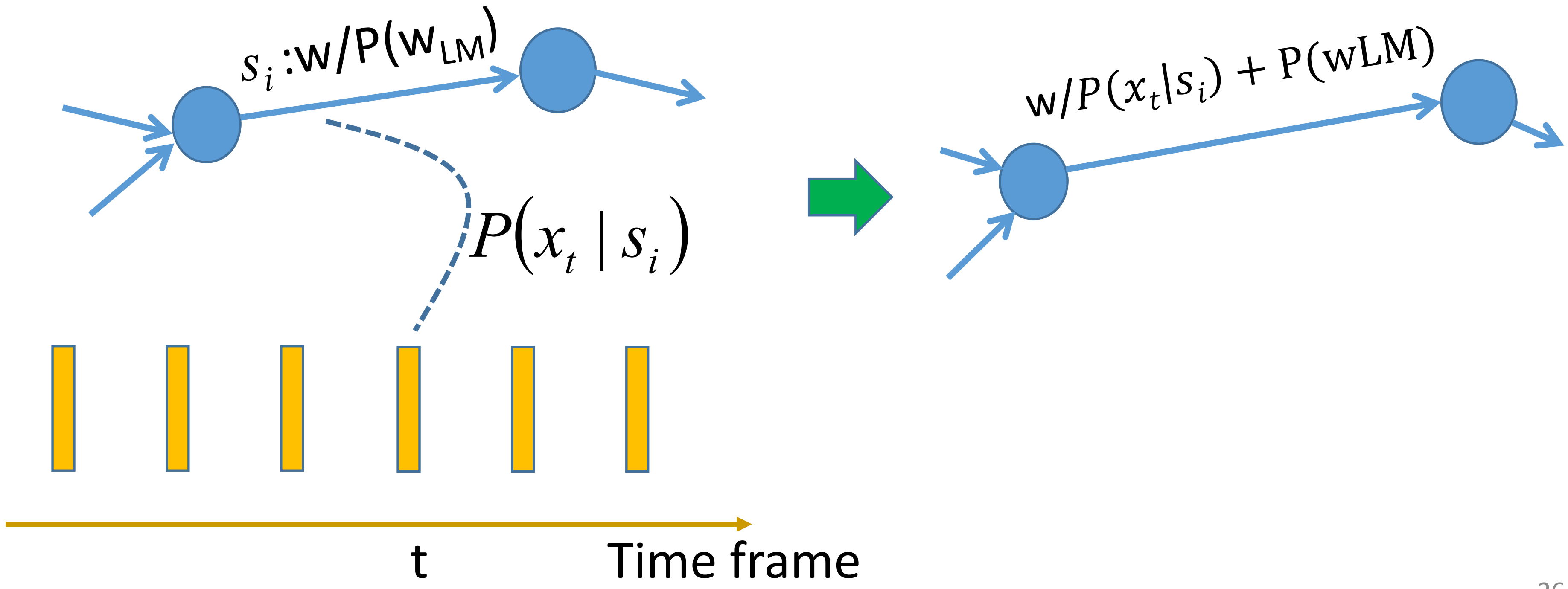
Construction of a Speech Recognition System

- Prepare:
  - WFST representing phone HMMs
  - WFST representing pronunciation dictionary: L
  - WFST representing word network or N-gram: G
- Compose:
  - WFST(Recognition system)
    =H・L・G
  - The WFST H・L・G has input labels indicating a HMM state name (=emission distribution) and output labels of a word



Speech Decoding based on WFST

- Starting from the initial state, t-th transition with non-epsilon input label s_i corresponds to t-th time frame x_t in input acoustic feature sequence
- By computing acoustic probability and adding it to language probability, an WFSA is obtained



Search the Best Recognition Hypothesis

- By performing minimum cost path search on the obtained WFSA, a word sequence that best matches to the input sound is obtained as the sequence of the output labels
- The popular search algorithm is Viterbi beam search

Appendix

Tools for WFST

- OpenFST
  - http://www.openfst.org
- Graphviz
  - http://www.graphviz.org

Example

wfst1.txt

```
0 1 <eps> a 0.5
0 1 C c 0.3
0 2 C <eps> 0.2
1 2 B b 1.0
2 3 A a
3
```

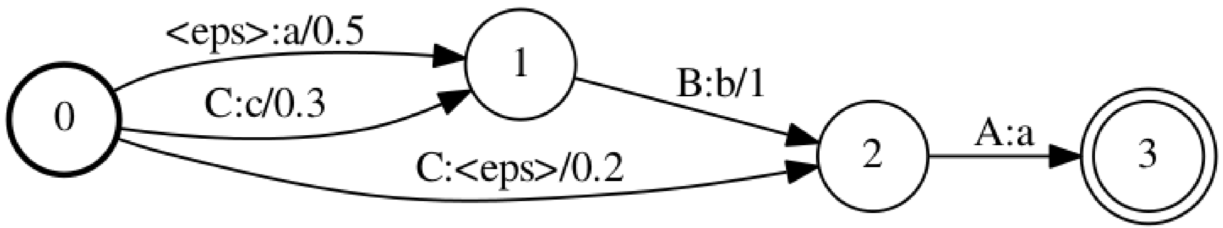
isym.txt

```
<eps> 0
A 1
B 2
C 3
```

osym.txt

```
<eps> 0
a 1
b 2
c 3
```

```
$ fstcompile --isymbols=isym.txt --osymbols=osym.txt --keep_isymbols --keep_osymbols
wfst1.txt wfst1.fst
$ fstdraw wfst1.fst | dot  -Tpdf > wfst1.pdf
$ evince wfst1.pdf &
```



Textbooks of Machine Learning

- Christopher Bishop,
  Pattern Recognition and Machine Learning,
  Springer-Verlag New York, 2006
- Kevin P. Murphy,
  Machine Learning A Probabilistic Perspective,
  The MIT Press, 2012