SPEECH RECOGNITION WITH WEIGHTED FINITE-STATE TRANSDUCERS

Mehryar Mohri, Fernando Pereira, Michael Riley

Professor：陳嘉平

Reporter：陳柏含

Introduction

- Weighted finite-state automata
  - Acceptor: accept each string that can be read along a path from the start state to a final state
  - Each accepted string is assigned a weight (or accumulated weights along accepting path)
- Weighted finite-state transducer
  - It is quite similar to a weighted acceptor except that it has an input label, an output label and a weight on each of its transitions

Introduction (cont.)

- Key transducer operations for speech application:
  - Composition: combine different level of transducer
  - Determinization: redundant path removal
  - Minimization: size reduction

Weighted finite-state acceptor

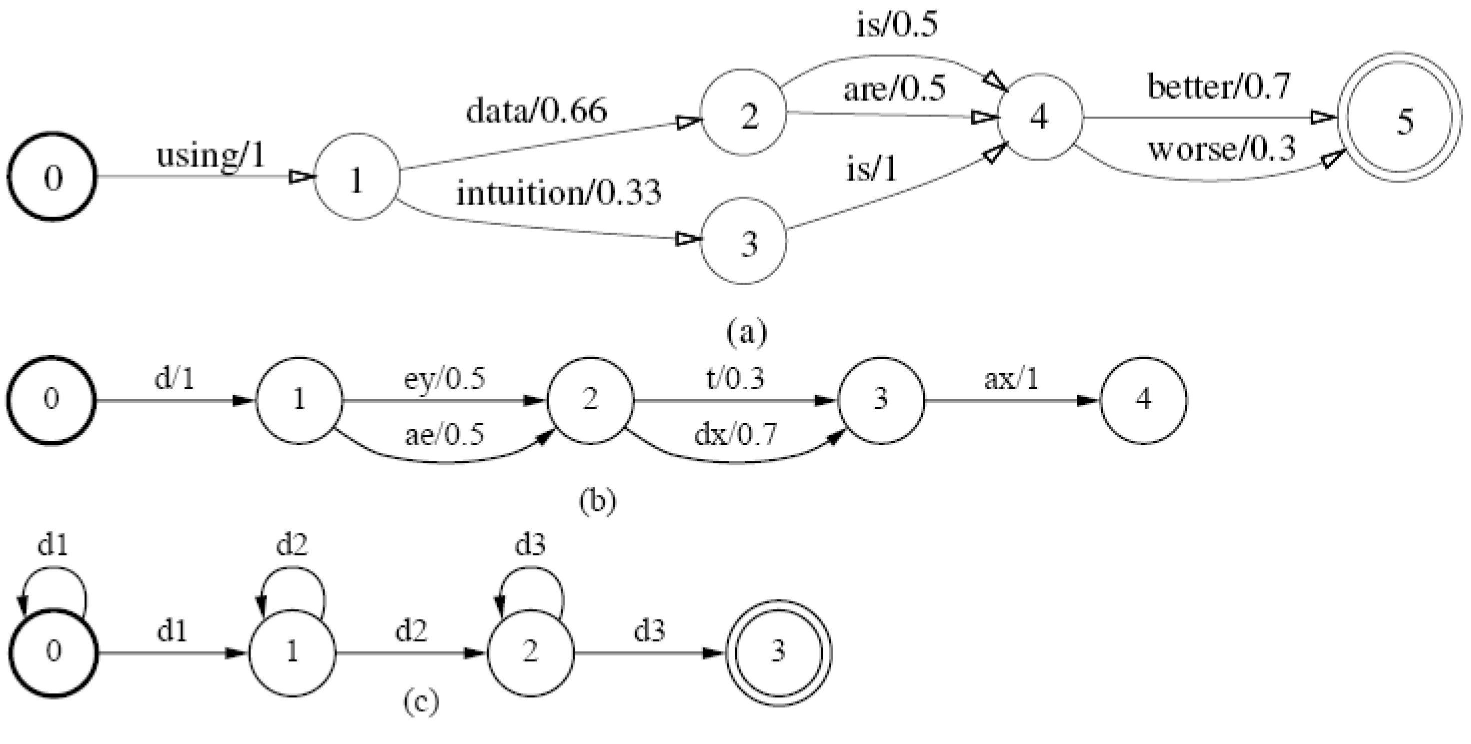
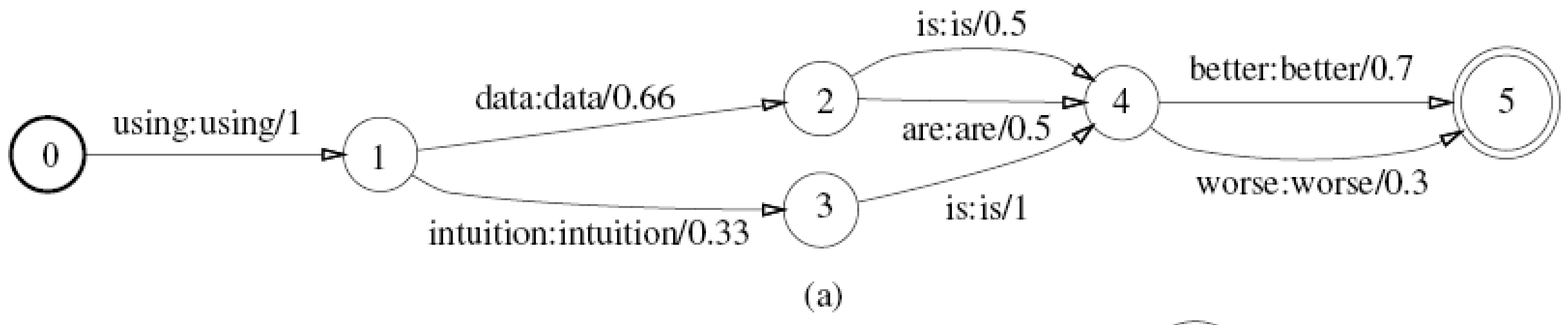


Figure 1: Weighted finite-state acceptor examples. By convention, the states are represented by circles and marked with their unique number. The initial *state* is represented by a bold circle, final states by double circles. The label l and weight w of a transition are marked on the corresponding directed arc by l/w. When explicitly shown, the final weight w of a final state f is marked by f/w.

Weighted finite-state transducer



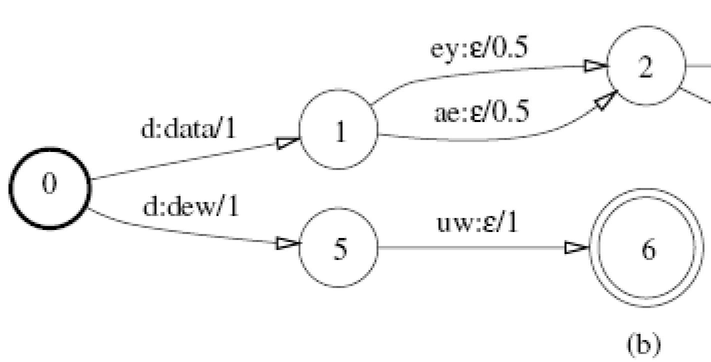


Figure 2: Weighted finite-state transducer examples. These are similar to the weighted acceptors in Figure 1 except output labels are introduced on each transition. The input label i, the output label o, and weight w of a transition are marked on the corresponding directed arc by $i : o/w$.

Definition and operation of WFST

- Weighted finite-state transducer $T = (A, B, Q, I, F, E, \lambda, \rho)$ over a semring K
  - A : input alphabet
  - B : output alphabet
  - Q : a finite set of states
  - I : a set of initial state, $I \subseteq Q$
  - F : a set of final state, $F \subseteq Q$
  - E : a finite set of transitions, $E \subseteq Q \times (A \cup \{\varepsilon\}) \times (B \cup \{\varepsilon\}) \times k \times Q$
  - λ : initial state weight assignment $\lambda : I \to K$
  - ρ : initial state weight assignment $\rho : I \to K$
- $|T|$ denotes the sum of the number of states and transitions of T
- $E[q]$ denotes the set of transitions leaving state $q \in Q$

Definition and operation of WFST (cont.)

- Weighted automata is defined in a similar way by simply omitting the input or output labes.
- Given a transition e belong to E
  - $p[e]$: e's origin or previous state
  - $n[e]$: e's destination state
  - $i[e]$: input label of e
  - $o[e]$: output label of e
  - $w[e]$: weight of e
- A path $\pi = e_1 \cdots e_k$ is a sequence of consecutive transitions $w[\pi] = w[e_1] \otimes \cdots \otimes w[e_k]$
  - Path weight:
  - $P(q, q')$: denotes the set of paths from q to q'

Definition of semiring

Definition 1 *A* semiring *is a system* $(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ *such that*

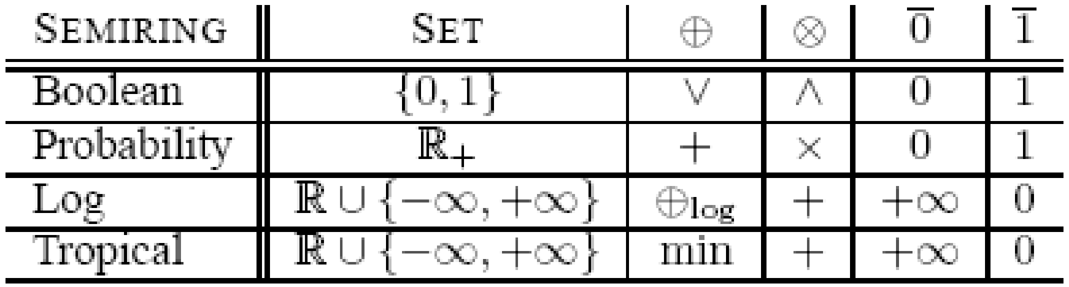
1. $(\mathbb{K}, \oplus, \overline{0})$ *is a commutative monoid with* $\overline{0}$ *as the identity element for* $\oplus$,
2. $(\mathbb{K}, \otimes, \overline{1})$ *is a monoid with* $\overline{1}$ *as the identity element for* $\otimes$,
3. $\otimes$ *distributes over* $\oplus$*: for all* a, b, c *in* $\mathbb{K}$,

$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c),$$
$$c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b),$$

4. $\overline{0}$ *is an annihilator for* $\otimes$*:* $\forall a \in \mathbb{K},\ a \otimes \overline{0} = \overline{0} \otimes a = \overline{0}$.

Semiring example

Table 1: *Semiring examples.* $\oplus_{\log}$ *is defined by:* $x \oplus_{\log} y = -\log(e^{-x} + e^{-y})$.



SEMIRING	SET	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$
Boolean	$\{0,1\}$	$\vee$	$\wedge$	0	1
Probability	$\mathbb{R}_+$	$+$	$\times$	0	1
Log	$\mathbb{R} \cup \{-\infty, +\infty\}$	$\oplus_{\log}$	$+$	$+\infty$	0
Tropical	$\mathbb{R} \cup \{-\infty, +\infty\}$	min	$+$	$+\infty$	0

Composition

- T_1 and T_2 are two transducers
  - Composition $T = T_1 \circ T_2$, T maps string u to string w while T_1 maps u to some string v and T_2 maps v to w
  - Compute the weight of a path in T from corresponding the weights of paths in T_1 and T_2 by weight operation $\otimes$

    e.g. If the transition weight represents probabilities or log probability, that $\otimes$ operation is product or sum, respectively.
  - Weight operations for a weighted transducer can be specified by a *semiring*

Composition (cont.)

- States in the composition T are identified with pairs of a state of T_1 and a state of T_2
- A transition of $T_1 \circ T_2$ from appropriate transitions of T_1 and T_2

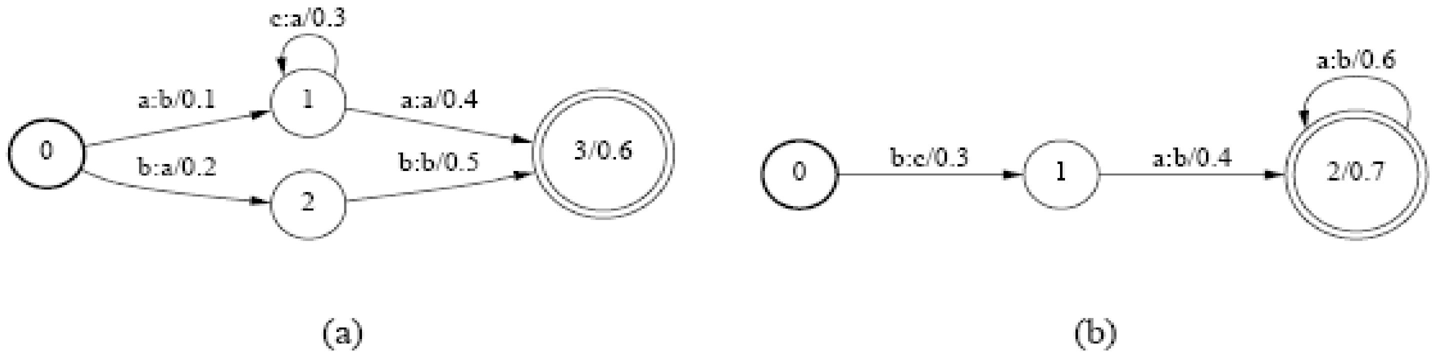
$$(q_1, a, b, w_1, r_1) \text{ and } (q_2, b, c, w_2, r_2) \Rightarrow ((q_1, q_2), a, c, w_1 \otimes w_2, (r_1, r_2))$$

Composition Algorithm

```
WEIGHTED-COMPOSITION$(T_1, T_2)$
$Q \leftarrow I_1 \times I_2$
$S \leftarrow I_1 \times I_2$
while $S \neq \emptyset$ do
    $(q_1, q_2) \leftarrow$ HEAD$(S)$
    DEQUEUE$(S)$
    if $(q_1, q_2) \in I_1 \times I_2$ then
        $I \leftarrow I \cup \{(q_1, q_2)\}$
        $\lambda(q_1, q_2) \leftarrow \lambda_1(q_1) \otimes \lambda_2(q_2)$
    if $(q_1, q_2) \in F_1 \times F_2$ then
        $F \leftarrow F \cup \{(q_1, q_2)\}$
        $\rho(q_1, q_2) \leftarrow \rho_1(q_1) \otimes \rho_2(q_2)$
    for each $(e_1, e_2) \in E[q_1] \times E[q_2]$ such that $o[e_1] = i[e_2]$ do
        if $(n[e_1], n[e_2]) \notin Q$ then
            $Q \leftarrow Q \cup \{(n[e_1], n[e_2])\}$
            ENQUEUE$(S, (n[e_1], n[e_2]))$
        $E \leftarrow E \cup \{((q_1, q_2), i[e_1], o[e_2], w[e_1] \otimes w[e_2], (n[e_1], n[e_2]))\}$
return $T$
```

Figure 7: Pseudocode of the composition algorithm.

Example of transducer composition



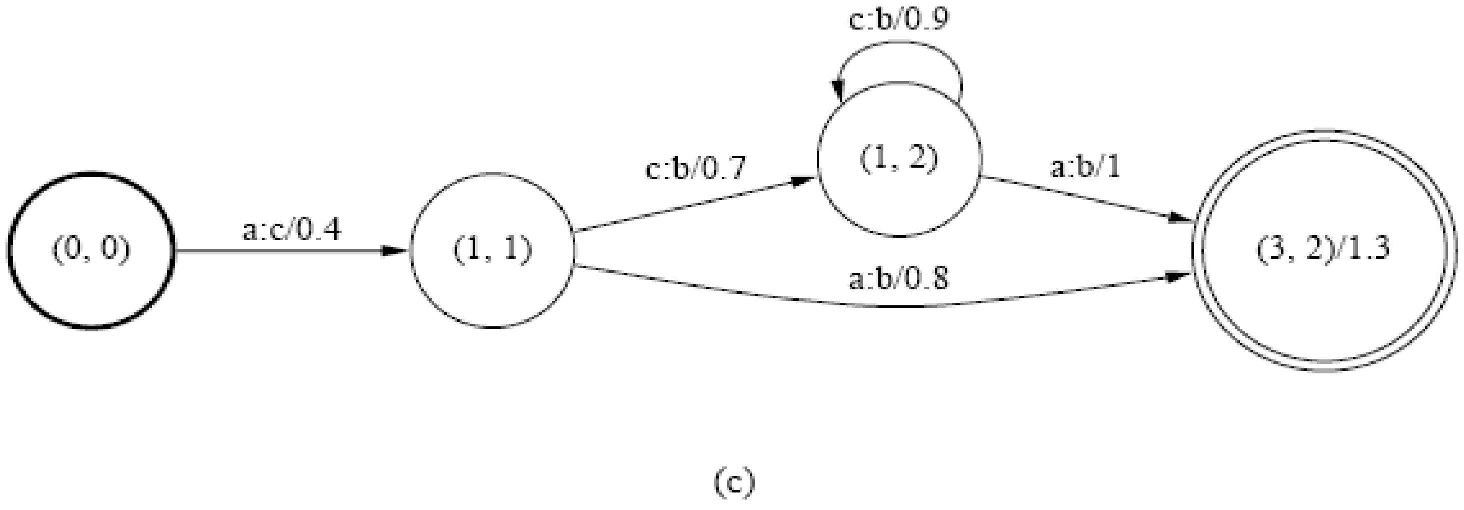
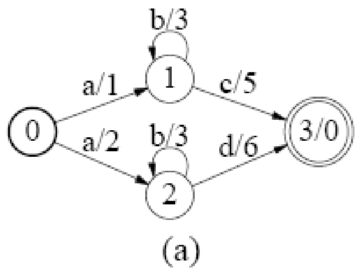


Figure 3: Example of transducer composition.

Determinization

- In a *deterministic automaton*, each state has at most one transition with any given input label and there are no input epsilon label.
- The key advantage of a deterministic automaton
  - at most one path matching any given input string
  - reducing the time and space needed to process the string
  - Tree lexicon in ASR is deterministic pronunciation representation



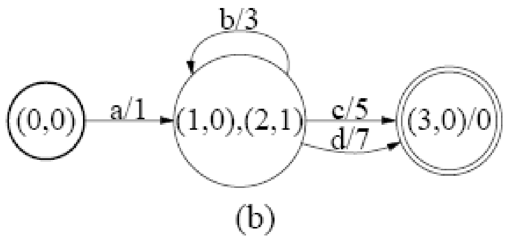
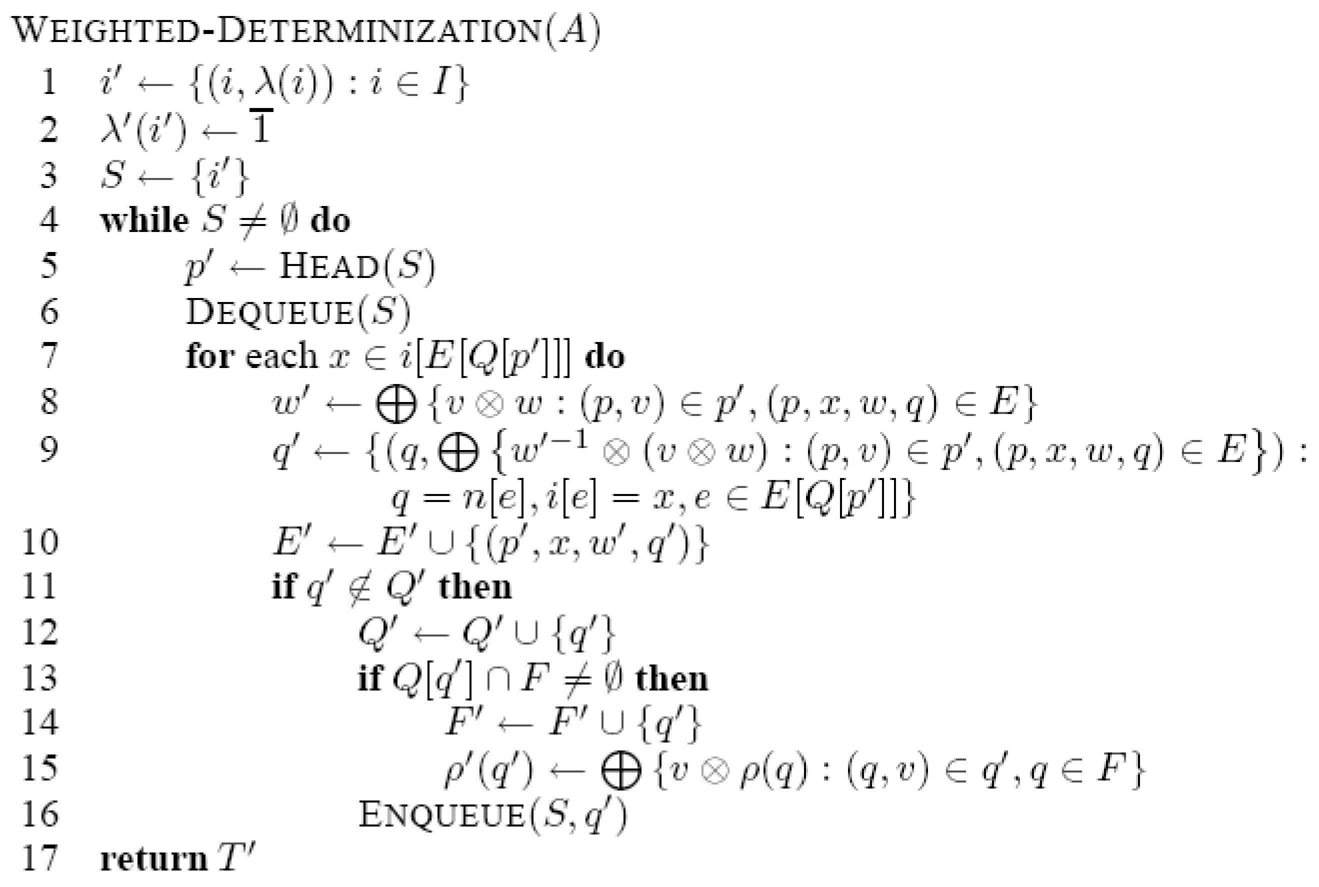


Figure 4: Determinization of weighted automata. (a) Weighted automaton over the tropical semiring A. (b) Equivalent weighted automaton B obtained by determinization of A.

Determinization algorithm



```
WEIGHTED-DETERMINIZATION($A$)
$i' \leftarrow \{(i, \lambda(i)) : i \in I\}$
$\lambda'(i') \leftarrow \overline{1}$
$S \leftarrow \{i'\}$
while $S \neq \emptyset$ do
    $p' \leftarrow$ HEAD($S$)
    DEQUEUE($S$)
    for each $x \in i[E[Q[p']]]$ do
        $w' \leftarrow \bigoplus \{v \otimes w : (p, v) \in p', (p, x, w, q) \in E\}$
        $q' \leftarrow \{(q, \bigoplus \{w'^{-1} \otimes (v \otimes w) : (p, v) \in p', (p, x, w, q) \in E\}) :$
                $q = n[e], i[e] = x, e \in E[Q[p']]\}$
        $E' \leftarrow E' \cup \{(p', x, w', q')\}$
        if $q' \notin Q'$ then
            $Q' \leftarrow Q' \cup \{q'\}$
            if $Q[q'] \cap F \neq \emptyset$ then
                $F' \leftarrow F' \cup \{q'\}$
                $\rho'(q') \leftarrow \bigoplus \{v \otimes \rho(q) : (q, v) \in q', q \in F\}$
            ENQUEUE($S, q'$)
return $T'$
```

Figure 10: Pseudocode of the weighted determinization algorithm [Mohri, 1997].

- In the classical subset construction for determinizing unweighted automaton, all the states reachable by a given input label strings from the initial state are placed in the same subset.

- In the weighted case, transitions with the same input label can have different weights, but only the minimum of those weights is needed and the leftover weights must be kept track of.

- Thus, the subsets in weighted determinization contain pairs (q,w) of a state q of the original

Test for Determinizable

- Two state *q* and *q*' of A are said to be siblings if
  - there is input label string *x* in *A* such that *q* and *q*' can be reached from *I* by paths labeled with *x*
  - there is input label string *y* in *A* while there is a cycle at *q* and a cycle at *q*' both labeled with *y*
- Two sibling states are said to be *twins* when for every string y

  $w[P(q,y,q)]=w[P(q',y,q')]$

- *A* has the twins property if any two sibling states of A are twins

Test for Determinizability (cont.)

- The following theorem relates the twins property and weighted determinization [Morhi, 1997]
  - **Theorem 1** *Let A be a weighted automaton over the tropical semiring. If A has the twins property, then A is determinizable.*

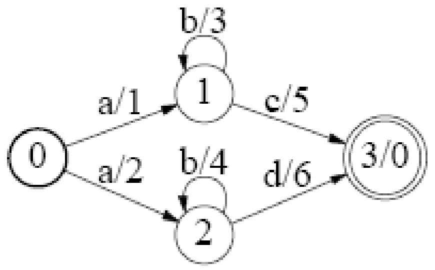


Figure 11: Non-determinizable weighted automaton over the tropical semiring. States 1 and 2 are non-twin siblings.

Minimization

- Automaton A has the least number of states and the least number of transitions by minimization

- The size of A can be reduced by using weighted minimization algorithm which works in two steps
  - 1. Pushes weight (reweighting) among transition
  - 2. applies the classical minimization algorithm to each distinct label-weight pair viewed as a distinct symbol

Weight Pushing

- Let $V: Q \rightarrow \mathbb{R}$ be an arbitrary potential function on states (assign each state lowest path weight from that state to final state)
- Update each weight $w[t]$ of the transition t from state $p[t]$ to $n[t]$ as follows

$$w[t] \leftarrow w[t] + (V(n[t]) - V(p[t])),$$

- and the final weight $\rho[f]$ of final state f as follows:

$$\rho[f] \leftarrow \rho[f] + (V(i) - V(f)).$$

- The weight of successful path is not changed

$$(V(f) - V(i)) + (V(i) - V(f)) = 0$$

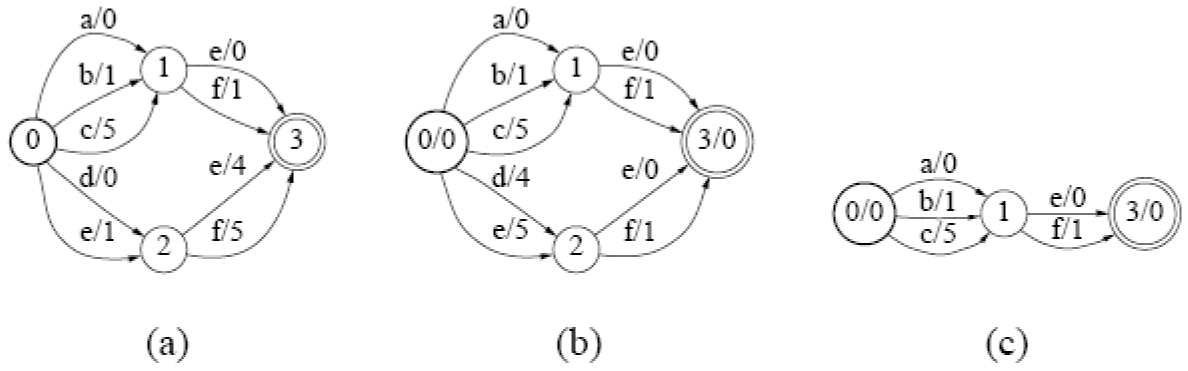


Figure 5: Weight pushing and minimization. (a) Deterministic weighted automaton A. (b) Equivalent weighted automaton B obtained by weight pushing in the tropical semiring. (c) Minimal weighted automaton equivalent to A.

Weight Pushing over semiring

- For any state q belong to Q, assume that the following sum is well defined and in K:

$$d[q] = \bigoplus_{\pi \in P(q,F)} (w[\pi] \otimes \rho[n[\pi]])$$

—$d[q]$ is the shortest-distance from q to F

- Reweighting the transition weights

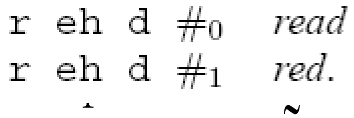
$$w[e] \leftarrow d[p[e]]^{-1} \otimes w[e] \otimes d[n[e]], if\, d[p[e]] \neq \bar{0}$$
$$\lambda[i] \leftarrow \lambda[i] \otimes d[i],$$
$$\rho[f] \leftarrow d[f]^{-1} \otimes \rho[f] \quad if\, d[f] \neq \bar{0}$$

Application For Speech Recognition

- Represent the various models in speech recognition as WFST
  - Word-level grammar G
  - Pronunciation lexicon L
  - Context-dependency transducer C
  - HMM transducer H

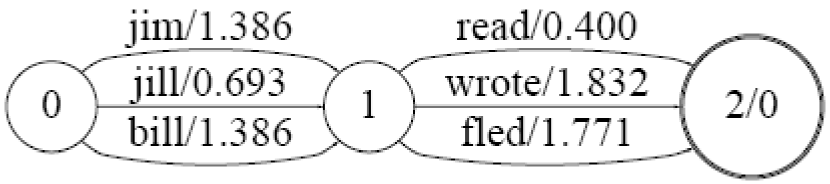
Pronunciation lexicon

- Pronunciation lexicon L transducer
  - Union all pronunciation lexicon transducers from each word in Grammar transducer G
    - Connecting the epsilon-transition from super-initial state to the start states of each pronunciation transducer
  - Kleene closure by connecting an epsilon-transition from each final state to the initial state
- In presence of homophones, L is *not determinizable.*
  - Solution: Add auxiliary phone symbol $\#_0$ to the end of each uniqu and other symbol $\#_1 \cdots \#_{k-1}$ to distinguis

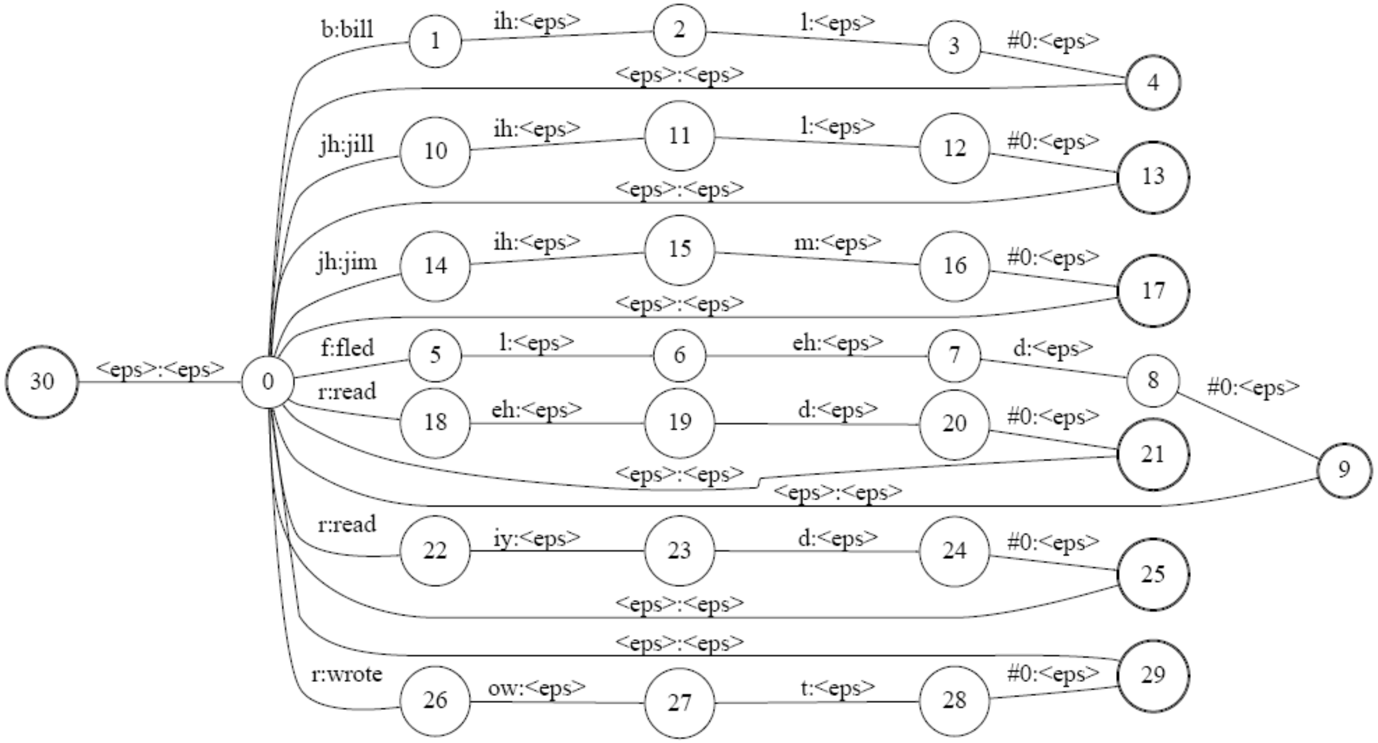


r eh d $\#_0$ *read*
r eh d $\#_1$ *red.*

$\tilde{L}$



(a) grammar G



(b) lexicon L

Context-dependency transducer

- Maps context-independent phones to context-dependent triphones
  - State encode the knowledge of the previous and next phone
  - Output of transition is labeled as *left context_right context*
  - make the transducer deterministic by choosing the right phone as input label instead of center phone as input

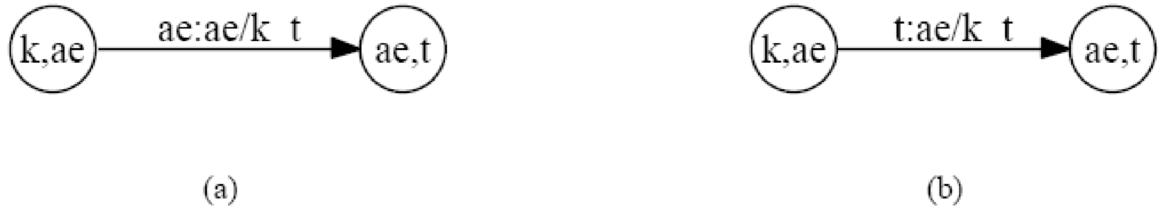
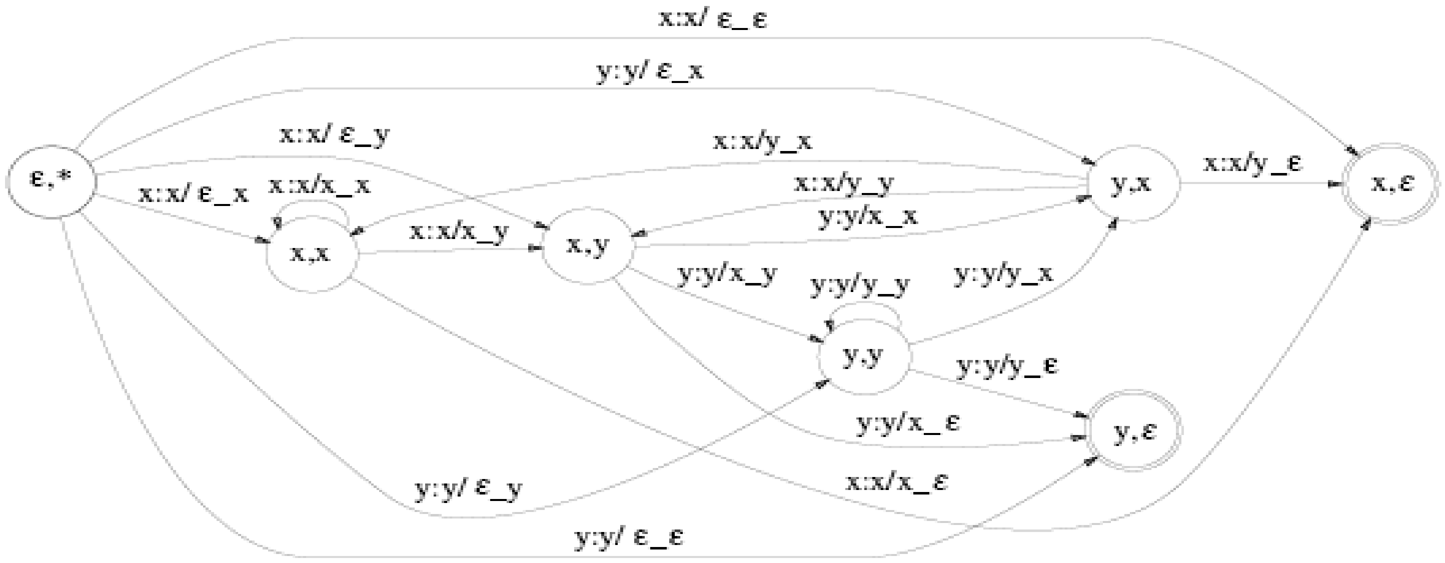
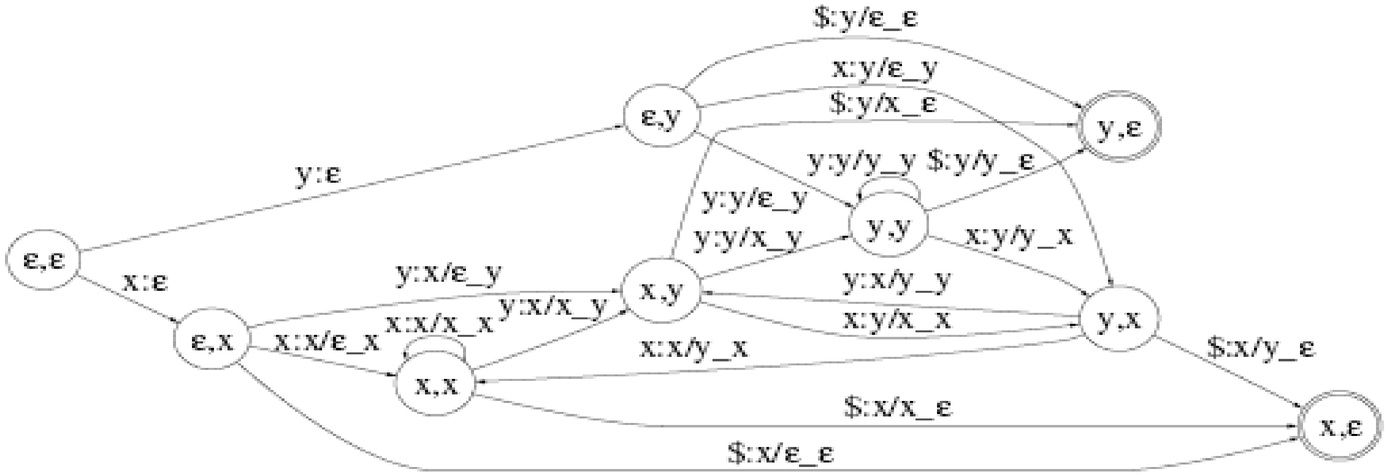


Figure 6: Context-dependent triphone transducer transition: (a) non-deterministic, (b) deterministic.



(a)



(b)

Input: xyx$ output context-depend model: ε _x, x_y, y_ ε

Figure 15: Context-dependent triphone transducers: (a) non-deterministic, (b) deterministic.

Composition

- Maps from distribution to word strings restricted by G

$$\tilde{H} \circ \tilde{C} \circ \tilde{L} \circ G$$

$\tilde{C}$:Self loops are added at each state of C mapping each auxiliary phone to new auxiliary context-dependent phone

$\tilde{H}$: Self loops are added at the initial state of H with auxiliary distribution name input labels and auxiliary context-dependent phone output labels

Determinization & Minimization

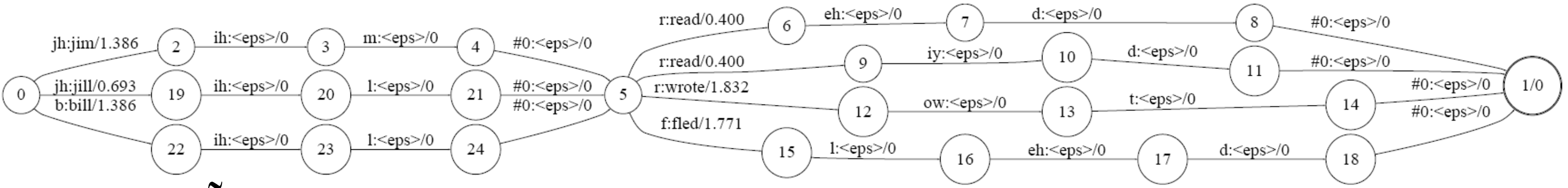
- Determinization

  $N = \pi_{\varepsilon}\left(\det\left(\tilde{H} \circ \det\left(\tilde{C} \circ \det\left(\tilde{L} \circ G\right)\right)\right)\right)$

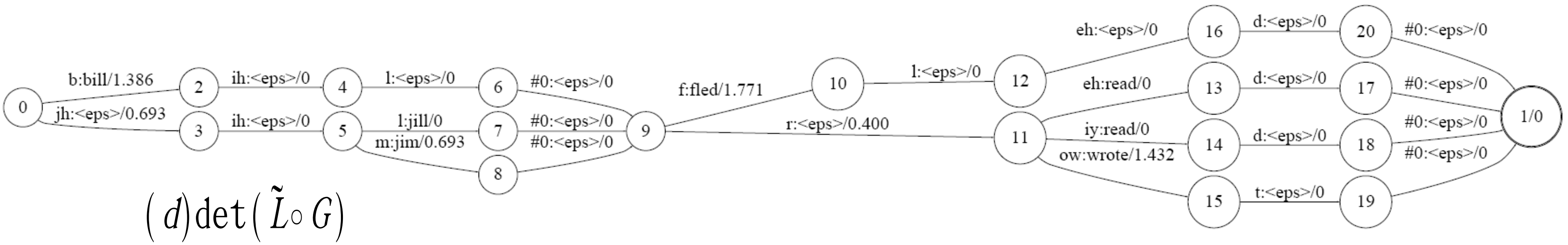
  – π_{ε} : replace auxiliary distribution symbols with epsilon

- Minimization

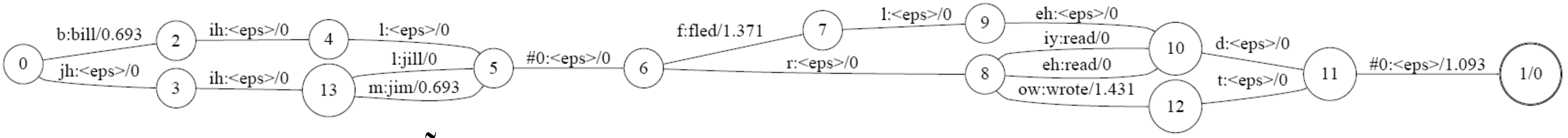
  $N = \pi_{\varepsilon}\left(\min\left(\det\left(\tilde{H} \circ \det\left(\tilde{C} \circ \det\left(\tilde{L} \circ G\right)\right)\right)\right)\right)$



$(c)\tilde{L}\circ G$



$(d)\mathrm{det}(\tilde{L}\circ G)$



$(e)\mathrm{min}_{tropical}(\mathrm{det}(\tilde{L}\circ G))$