On the Disambiguation of Weighted Automata

Mehryar Mohri
Courant Institute of Mathematical Sciences
mohri@cs.nyu.edu

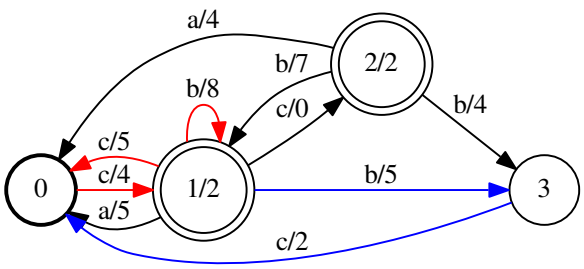
Michael Riley
Google NY Research
riley@google.com

August 18, 2015

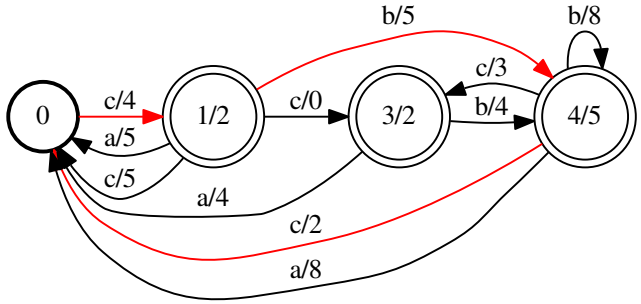
CIAA 2015, Umea, Sweden

Problem

- Weighted automata (WFA): automata where each path is labeled with a string and a weight
- Unambiguous weighted automaton: no two accepting paths labeled with the same string
- Disambiguation algorithm: produce an equivalent weighted automaton that is unambiguous



ambiguous A



equivalent, unambiguous A'

In the depictions:

- Initial state: bold circle; initial weight: 0
- Final states: double circles; final weight: state/weight
- Transitions: label/weight
- Weights: (tropical semiring) sum along paths; minimum when combining ambiguously-labeled paths (e.g., the two accepting paths of `cbcc` and `cbcc`).

Motivation

Example: Weighted automata are often used to represent hypothesis sets, so-called *lattices*, in speech and language processing problems. Ambiguities can be introduced in lattice generation that present problems with their use, e.g.:

- n-best generation:
  - classical n-shortest path algorithms do not ensure uniqueness as often needed in applications
  - *At Google:* almost a billion speech lattices a day searched for unique n-best
- counting:
  - enumerating n-grams present for language modeling applications
  - *At Google:* applications where billions of lattices counted at various n-gram orders in search of improved language models
- discrimintative training:
  - Trains acoustic or language models using both positive and negative examples found in lattices.

Alternative Approaches

- Weighted determinization can be used for disambiguation but:
  - not all weighted automata admit weighted determinization or halt with weighted determization algorithm
  - weighted determinization result can be *exponentially larger* than weighted disambiguation
- Previous disambiguation algorithms either:
  - do not apply to the weighted case
  - have more restrictions on the ambiguity (finitely or polynomially ambiguous only)
  - have more restrictions on the weights (fewer applicable semirings)
  - are less efficient (super-exponential)

Outline

- Preliminaries: Basic definitions and notation
- Pre-disambiguation: Initial step that constructs a weighted automaton whose paths leaving the initial state and labeled with the same string have the same weight
- Disambiguation: Second step that removes some transitions to make the result unambiguous. Disambiguation algorithm can be applied whenever pre-disambiguation terminates
- Sufficient Conditions: Introduce *weak twins property* for the applicability of weighted disambiguation
- Experiments: Comparison of weighted disambiguation to determinization in speech recognition and machine translation applications.

Preliminaries I

- A *semiring* $(\mathbb{S}, \oplus, \otimes, \overline{0}, \overline{1})$ is a ring over $\mathbb{S}$ that may lack negation. A semiring is:
  - *cancellative:* when for any x, x' and z in $\mathbb{S}$ with $z \neq \overline{0}$, $x \otimes z = x' \otimes z$ implies $x = x'$
  - *left divisible:* if any element $x \in \mathbb{S} - \{\overline{0}\}$ admits a left inverse $x' \in \mathbb{S}$, that is $x' \otimes x = \overline{1}$
  - *weakly left divisible:* if for any x and x' in $\mathbb{S}$ such that $x \oplus x' \neq \overline{0}$, $\exists z\colon\ x = (x \oplus x') \otimes z$. When also cancellative, z is unique and we write: $z = (x \oplus x')^{-1} \otimes x$
- A *weighted finite automaton* A is a 7-tuple $(\Sigma, Q, I, F, E, \lambda, \rho)$ with
  - Σ: finite alphabet
  - Q, $I \subseteq Q$, $F \subseteq Q$: states, initial states and final states resp.
  - $E \subseteq Q \times \Sigma \times \mathbb{S} \times Q$: transitions $e = (\mathrm{orig}[e], \mathrm{lab}[e], w[e], \mathrm{dest}[e]) \in E$
  - $\lambda : I \to \mathbb{S}$: *initial weight function*
  - $\rho : F \to \mathbb{S}$: *final weight function*

Preliminaries II

- A path $\pi = e_1 \cdots e_n$ is an element of E^* with consecutive transitions. A path is *accepting* or *successful* if $\text{orig}[\pi] = \text{orig}[e_1] \in I$ and $\text{dest}[\pi] = \text{dest}[e_n] \in F$.
- $P(U, x, V)$: denotes all paths from a state in $U \subseteq Q$ to $V \subseteq Q$ labeled with $x \in \Sigma^*$
- $\delta(U, x)$: denotes the set of states reached by paths starting in $U \subseteq Q$ and labeled with $x \in \Sigma^*$
- The weight of a path:
  - path: $w[\pi] = w[e_1] \otimes \cdots \otimes w[e_n]$,
  - initial path: $w_{\mathcal{I}}[\pi] = \lambda(orig[\pi]) \otimes w[\pi]$ when $\text{orig}[\pi] \in I$
- The weight for a string x of:
  - paths from U to V: $W(U, x, V) = \bigoplus_{\pi \in P(U,x,V)} w[\pi]$
  - initial paths to V: $W_{\mathcal{I}}(x, V) = \bigoplus_{\pi \in P(I,x,V)} w_{\mathcal{I}}[\pi]$.
  - accepting paths: $A(x) = \bigoplus_{\pi \in P(I,x,F)} w_{\mathcal{I}}[\pi] \otimes \rho(\text{dest}[\pi])$

Pre-disambiguation - Relations over $Q \times Q$

- Two states $q, q' \in Q$ *share a common future* if there exists a string $x \in \Sigma^*$ such that $P(q, x, F)$ and $P(q', x, F)$ are not empty
- Let R^* be the relation: $q \,\mathsf{R}^*\, q'$ iff $q = q'$ or q and q' share a common future
  - R^* is reflexive and symmetric, but in general not transitive
  - R^* is *compatible with the inverse transition function*: if $q \,\mathsf{R}^*\, q'$, $q \in \delta(p, x)$ and $q' \in \delta(p', x)$ for some $x \in \Sigma^*$ with $(p, p') \in Q \times Q$, then $p \,\mathsf{R}^*\, p'$
- Denote by R_0 the *complete relation*: $q \,\mathsf{R}_0\, q'$ for all $(q, q') \in Q \times Q$
  - R_0 is also compatible with the inverse transition function
- The construction we will define holds for any relation R of the *set of admissible relations* $\mathcal{R}$ defined as the reflexive relations over $Q \times Q$ that are compatible with the inverse transition function and coarser than R^*.

Pre-disambiguation - Definitions

Fix R in $\mathcal{R}$:

- $\delta_q(U, x)$: For any $x \in \Sigma^*$ and $q \in \delta(U, x)$ define:

$$\delta_q(U, x) = \delta(U, x) \cap \{p \colon p \,\mathsf{R}\, q\}$$

  - the set of states in $\delta(U, x)$ that are in relation with q
  - $q \in \delta_q(I, x)$ since R is reflexive

- weighted subset $s(x, q)$: For any $x \in \Sigma^*$ and $q \in \delta(I, x)$ define:

$$s(x, q) = \Big\{ (p_1, w_1), \ldots, (p_t, w_t) \colon \big(\{p_1, \ldots, p_t\} = \delta_q(I, x)\big) \\ \wedge \big(\forall i \in [1, t], w_i = W_{\mathcal{I}}(x, \{p_1, \ldots, p_t\})^{-1} \otimes W_{\mathcal{I}}(x, p_i)\big) \Big\}$$

- set(s): For a weighted subset s, define $\mathrm{set}(s) = \{p_1, \ldots, p_t\}$

Pre-disambiguation - Construction

For any automaton A define $A' = (\Sigma, Q', I', F', E', \lambda', \rho')$ as follows:

$$Q' = \{(q, s(x,q)) \colon x \in \Sigma^*, q \in \delta(I, x)\}$$

$$I' = \{(q, s(\epsilon, q)) \colon q \in I\}$$

$$F' = \{(q, s(x,q)) \colon x \in \Sigma^*, q \in \delta(I, x) \cap F\}$$

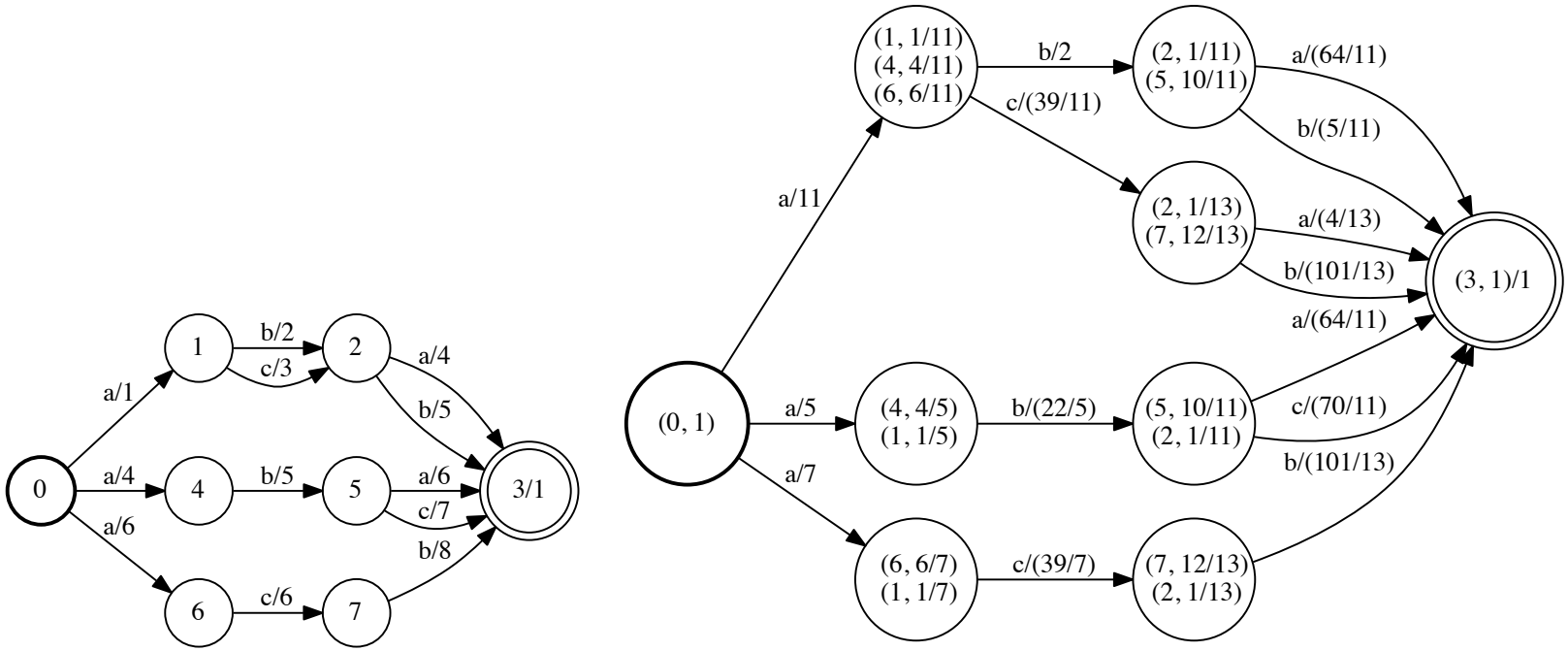
$$E' = \Big\{((q,s), a, w, (q',s')) \colon (q,s), (q',s') \in Q', a \in \Sigma,$$
$$\exists x \in \Sigma^* \mid \; s = s(x,q) = \{(p_1, w_1), \ldots, (p_t, w_t)\},$$
$$s' = s(xa, q') = \{(p'_1, w'_1), \ldots, (p'_{t'}, w'_{t'})\},$$
$$q' \in \delta(q, a),$$
$$w = \bigoplus_{i=1}^{t} \Big(w_i \otimes W(p_i, a, \mathrm{set}(s'))\Big)\Big\}$$

$$\lambda'((q,s)) = \bigoplus_{i \in [1,t]} \lambda(p_i), \quad \forall (q,s) \in I', s = \{(p_1, w_1), \ldots, (p_t, w_t)\},$$

$$\rho'((q,s)) = \bigoplus_{\substack{p_i \in F \\ i \in [1,t]}} (w_i \otimes \rho(p_i)), \quad \forall (q,s) \in F', s = \{(p_1, w_1), \ldots, (p_t, w_t)\},$$

Pre-disambiguation - Example

Illustration of pre-disambiguation construction in semiring $(\mathbb{R}_+, +, \times, 0, 1)$:



- For each result state (q, s) , the subset s is explicitly shown
- q is the state of the first pair in s shown
- The weights are rational numbers, e.g., $\frac{1}{11} \approx .091$

Pre-disambiguation - Properties

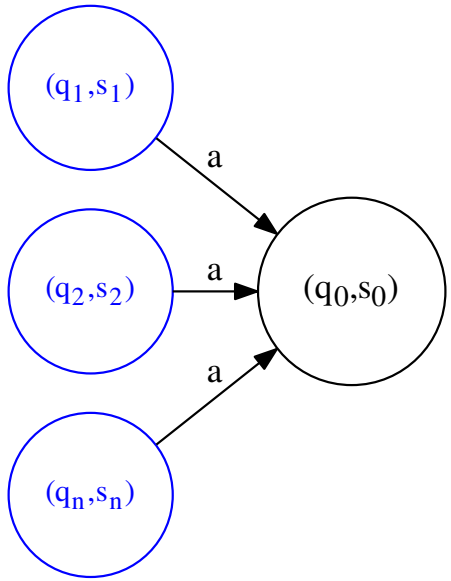
Let $A' = (\Sigma, Q', I', F', E', \lambda', \rho')$ be the finite automaton returned by the pre-disambiguation of the WFA $A = (\Sigma, Q, I, F, E, \lambda, \rho)$. Let π be a path in $P(I', x, (q, s))$ from A' with string $x \in \Sigma^*$ and subset $s = \{(p_1, w_1), \ldots, (p_t, w_t)\}$:

- Propositions:
  1. $w_{\mathcal{I}}[\pi] = W_{\mathcal{I}}(x, \mathrm{set}(s)) \quad \textit{and} \quad \forall i \in [1, t],\ w_{\mathcal{I}}[\pi] \otimes w_i = W_{\mathcal{I}}(x, p_i)$
  2. $w_{\mathcal{I}}[\pi] \otimes \rho'((q, s)) = A(x)$, *$\pi$ is an accepting path with* $(q, s) \in F'$
  3. *Any string* $y \in \Sigma^*$ *accepted by* A *is accepted by* A'
- Summary:
  A and A' accept the same strings. The weight of each path from an initial state of A' equals the $\oplus$-sum of the weights of all paths with the same label in the input automaton starting at an initial state.

Disambiguation - Definitions I

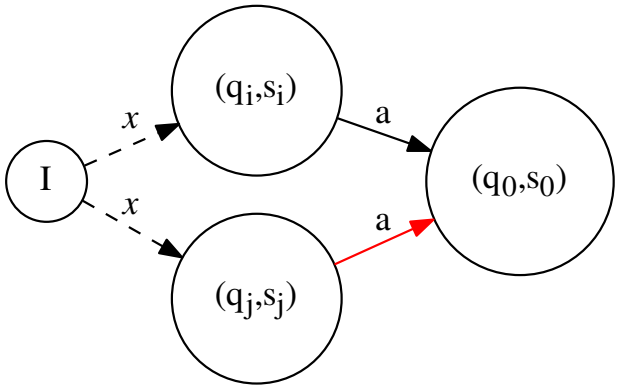
Let A' be the WFA returned by the pre-disambiguation of WFA $A = (\Sigma, Q, I, F, E, \lambda, \rho)$:

- *co-reachable:* Two states q and q' are co-reachable if they each can be reached by a path from I labeled with a common string $x \in \Sigma^*$
- *list:* $\mathcal{L}(q_0, s_0, a) = ((q_1, s_1), \ldots, (q_n, s_n))$, $n \geq 1$: all distinct states of A' admitting a transition labeled with $a \in \Sigma$ to (q_0, s_0) of A' with $q_1 \leq \cdots \leq q_n$



Disambiguation - Definitions II

- *processing of a list:* the states in $\mathcal{L}(q_0, s_0, a)$ are processed in order: for each state (q_j, s_j), $j \geq 2$, remove its a-transition to (q_0, s_0) iff there exists a co-reachable state (q_i, s_i) with $1 \leq i < j$ whose a-transition to (q_0, s_0) has not been removed



- *final list:* $\mathcal{F} = ((q_1, s_1), \ldots, (q_n, s_n))$, $n \geq 1$: all distinct final states of A' with $q_1 \leq \cdots \leq q_n$
- *processing of the final list:* the states in $\mathcal{F}$ are processed in order: for each state (q_j, s_j), $j \geq 1$, make it non-final if and only if there exists a co-reachable state (q_i, s_i) with $i < j$ whose finality was maintained

Disambiguation - Algorithm

Assume that A is pre-disambiguable. Define the disambiguation algorithm DISAMBIGUATION for A as follows:

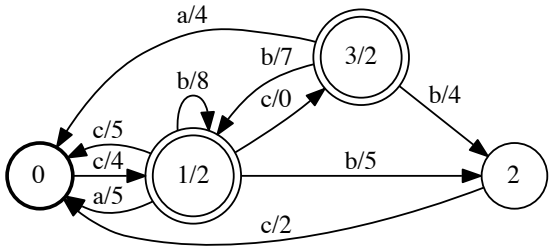
1. construct A', the result of the pre-disambiguation of A
2. process $\mathcal{L}(q_0, s_0, a)$ for any state (q_0, s_0) of A' and label $a \in \Sigma$
3. process the list of final states $\mathcal{F}$

Theorem: *Let A be a pre-disambiguable weighted automaton. Then algorithm* DISAMBIGUATION *run on input A generates an unambiguous WFA B equivalent to A.*

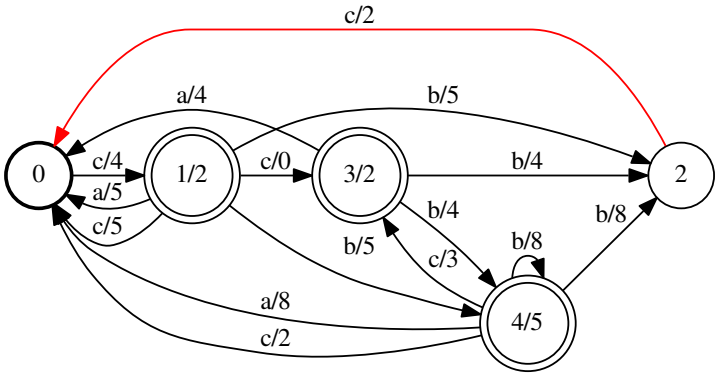
- Key Idea: The removal of each transition in the list processing does not change that A and A' accept the same strings.
- Differing numberings of the states can result in different weighted automata with potentially different sizes after trimming. Nevertheless, all such resulting weighted automata are equivalent.

Disambiguation - Example

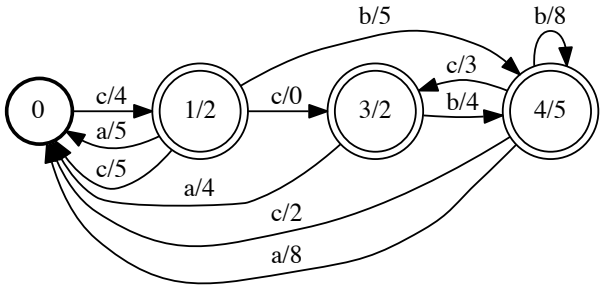
Illustration of the full disambiguation algorithm applied to a cyclic WFA:



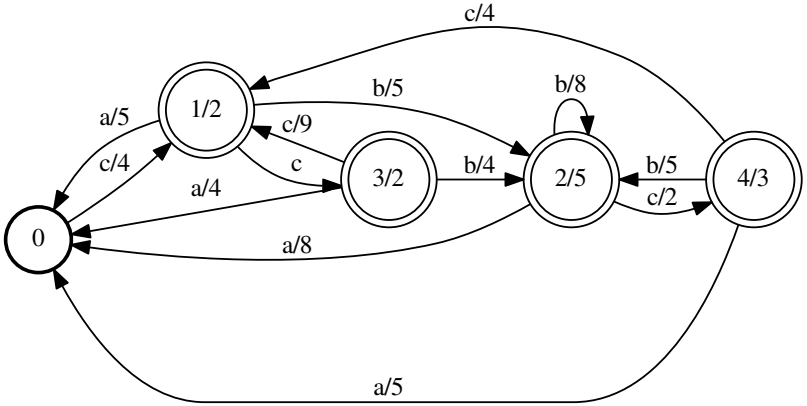
WFA A



A': pre-disambiguation of A



A'': full disambiguation of A

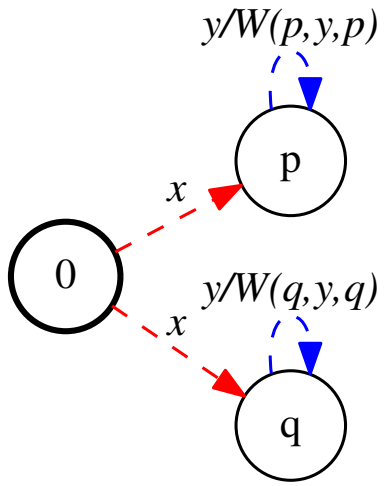


determinizaton of A

WFA A'' is obtained from A' by removal of the red transition from state 2 and trimming.

Disambiguation - Sufficient Conditions I

- Definitions: Two states $p, q \in Q$ of WFA A are:
  - siblings: both reachable from the initial state by paths labeled with $x \in \Sigma^*$ and have cycles labeled with $y \in \Sigma^*$
  - twins: siblings such that $W(p, y, p) = W(q, y, q)$
  - weak twins: twins that are in relation: $p \, R \, q$
  - (weak) twins property: all siblings are (weak) twins



Disambiguation - Sufficient Conditions II

- Tropical semiring:
  $\rightarrow (\mathbb{R}_+ \cup \{+\infty\}, \min, +, +\infty, 0)$
  - *If A has the weak twins property, it is pre-disambiguable*
  - *If A is determinizable, it is pre-disambiguable.*
- More general semirings:
  $\rightarrow$ weakly left-divisible, cancellative
  - If A is acyclic, it is pre-disambiguable.

Experiments - Method

- Implementation: The algorithm was implemented in *OpenFst*, a widely-used C++ library for constructing, combining, optimizing and searching WFAs. Both the weighted disambiguation and determinization algorithms are available for download at www.openfst.org.
- Corpora:
  - Experiment 1: 500 speech acyclic hypothesis sets (lattices) drawn randomly from anonymized Google/Android voice searches
  - Experiment 2: 100 possibly cyclic machine translation hypothesis sets from Chinese to English in the the DARPA Gale task (with Cambridge U.)

Experiments - Results

Measured *size expansion* of algorithms, e.g. $\frac{|disamb(A)|}{|A|}$ where $|A| = |Q| + |E|$:

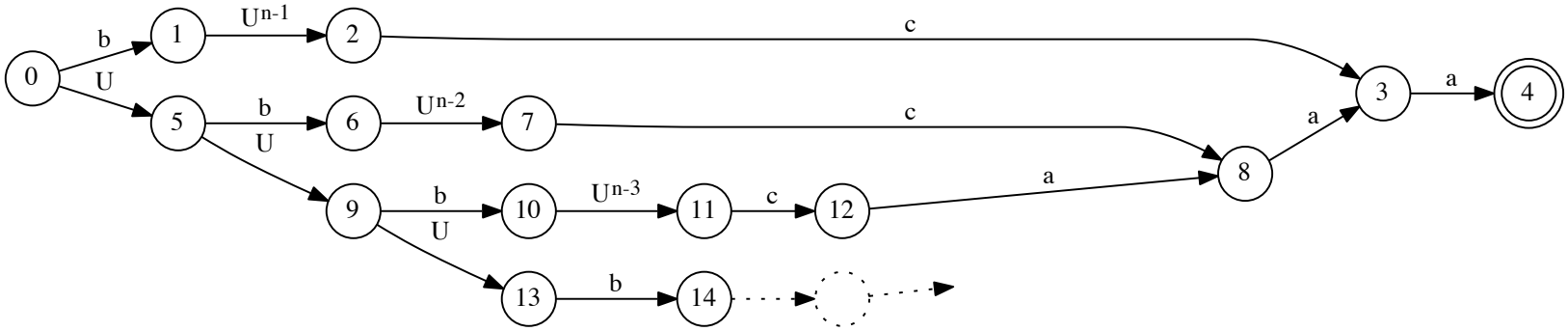
corpus	**weighted disambiguation size expansion**		**weighted determinization size expansion**	
	mean	*std. dev.*	*mean*	*std. dev.*
speech	1.23	0.59	1.31	1.35
translation*	4.53	6.0	54.5	90.5

*Excludes those that failed to determinize or disambiguate (about 2/3 of the former) in the alloted 1GB of memory.

Final Example

An illustration of an acyclic unambiguous (unweighted) automaton over the alphabet $\{a, b, c\}$ whose size is in $O(n^2)$

- Accepts $L = \{(a+b)^{k-1}b(a+b)^{n-k}ca^k : 1 \leq k \leq n\}$
- For any $k \geq 0$, U^k serves as a shorthand for $(a+b)^k$



- No equivalent deterministic automaton can have less than 2^n states since such an automaton must have a distinct state for each of the prefixes of the strings $\{(a+b)^{k-1}b(a+b)^{n-k} : 1 \leq k \leq n\}$, which are prefixes of L

Conclusion

We have presented an algorithm for the disambiguation of WFAs:

- Applies to a family of WFAs defined over the tropical semiring verifying a described sufficient condition
- shows favorable experimental results in some speech and language processing applications

For further study:

- decidability of the weak twins property for arbitrary WFAs
- characterize WFAs that admit an equivalent unambiguous WFA
- characterize the WFAs to which our algorithm applies