

Tekniker för storskalig parsning

Finite automater och transduktorer

Joakim Nivre

Uppsala universitet
Institutionen för lingvistik och filologi
`joakim.nivre@lingfil.uu.se`

Tack till Noah A. Smith för ljusbilder

Finite-State Technology

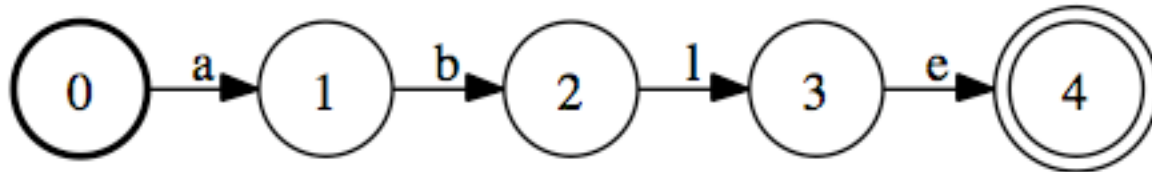
- Formally well-understood
 - Regular languages, rational relations
 - Generalizes n-grams, HMMs
- Many applications in NL technologies
 - Speech recognition
 - Lexical, morphological processing
 - Information extraction
 - Translation
 - Parsing
- Several toolkits
- Often determinizable: **very fast**

Finite-State Automata (Recognizers)

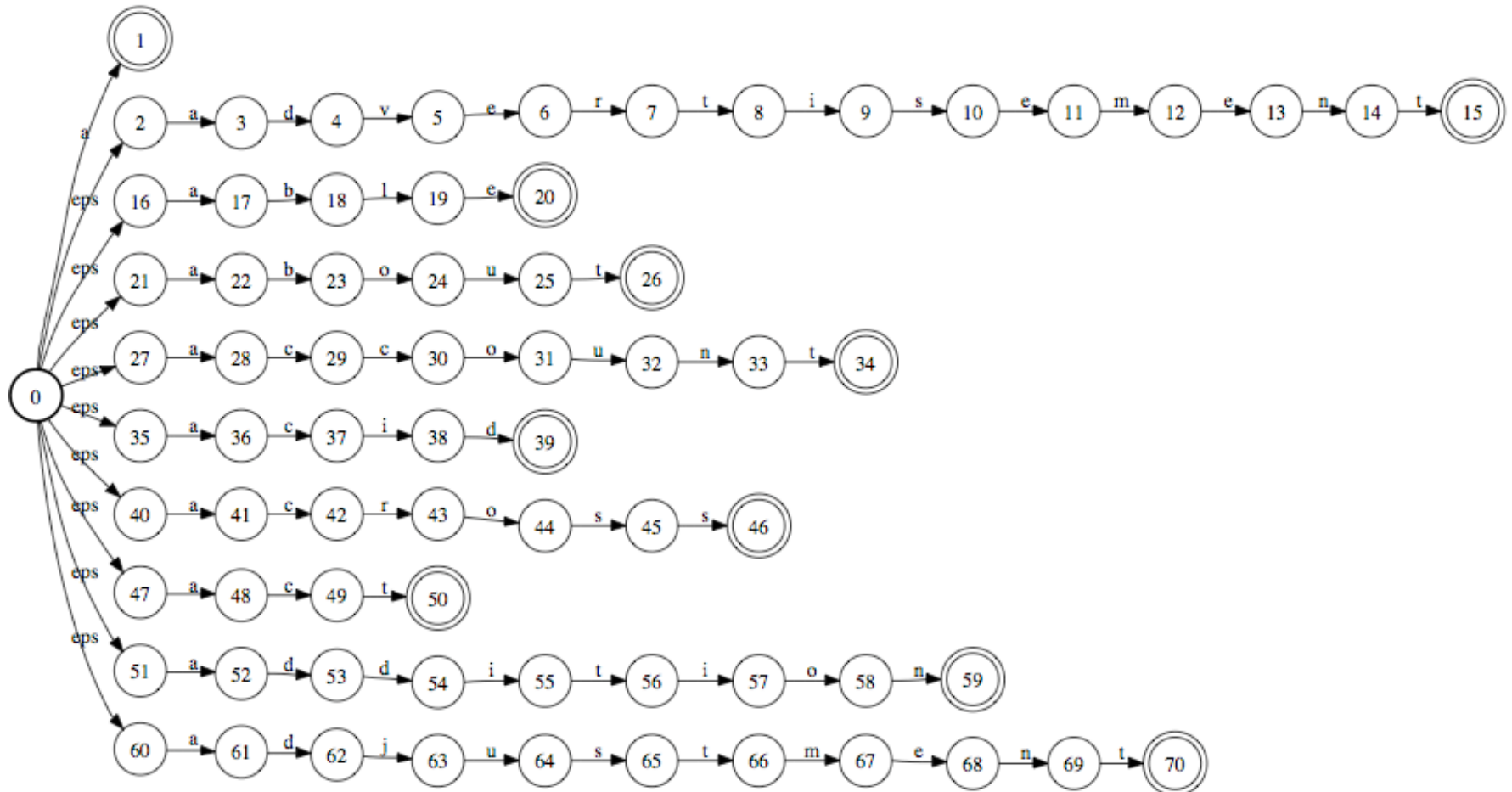
- Automaton that recognizes **regular** language
- Implementation of a **regular** expression
- Regular languages are closed under numerous operations
 - Concatenation, union, intersection, Kleene *, difference, reverse, complement, ...
- Correspond to regular grammars (type 3 in Chomsky hierarchy)

FSM as a Dictionary

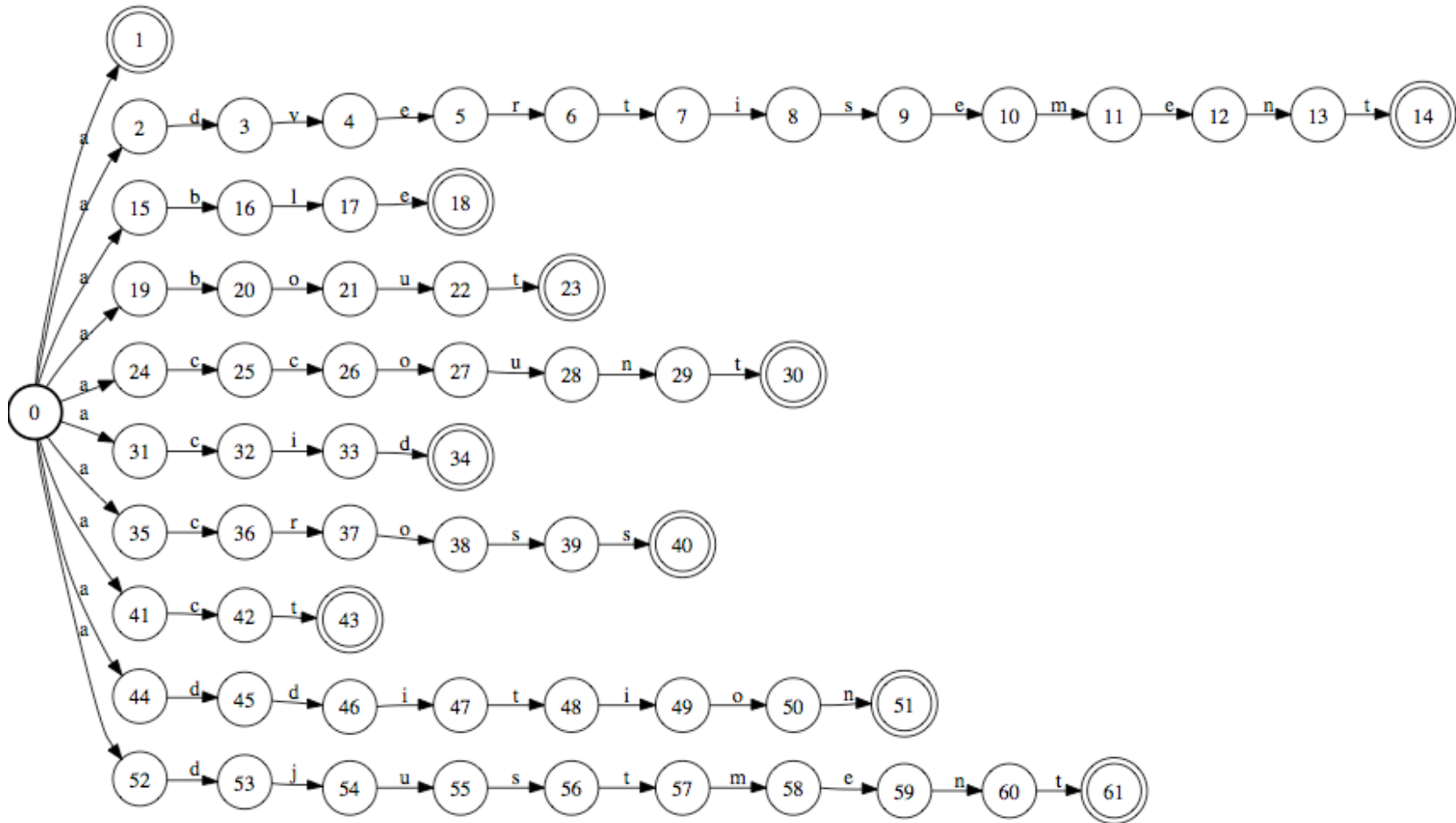
- Example: 850 words in “Basic English”
- Each word is an FSM



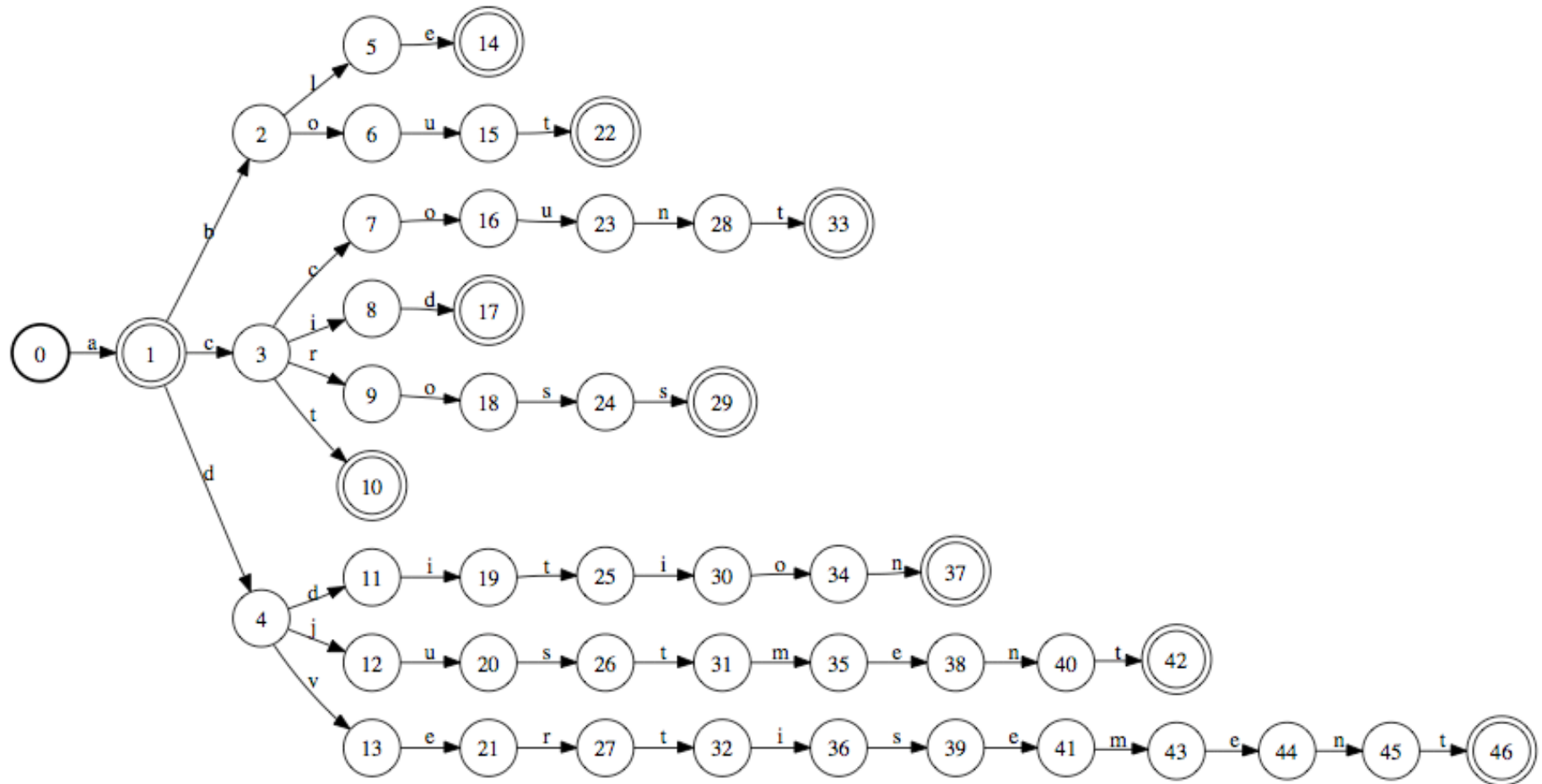
Ten-Word Dictionary



Remove ϵ -transitions



Determinize



Full 850-Word Dictionary

	<i>states</i>	<i>final states</i>	<i>arcs</i>
Union	5303	850	5302
Remove ϵ -transitions	4454	850	4453
Determinize	2609	848	2608
Minimize	744	42	1535

Generalizations

- FS Recognizer is a function from $\Sigma^* \rightarrow \{0,1\}$
 - Meaning: $\text{fsa}(s) = 1 \iff s$ is in the language
- Other **rational relations** ...
 - FS Transducer: $\Sigma^* \rightarrow \Delta^*$
 - Weighted FSA: $\Sigma^* \rightarrow \mathbb{R}$
 - Weighted FST: $\Sigma^* \rightarrow \Delta^* \times \mathbb{R}$
- WFSA and WFSTs can be considered **probabilistic** (but don't have to be)

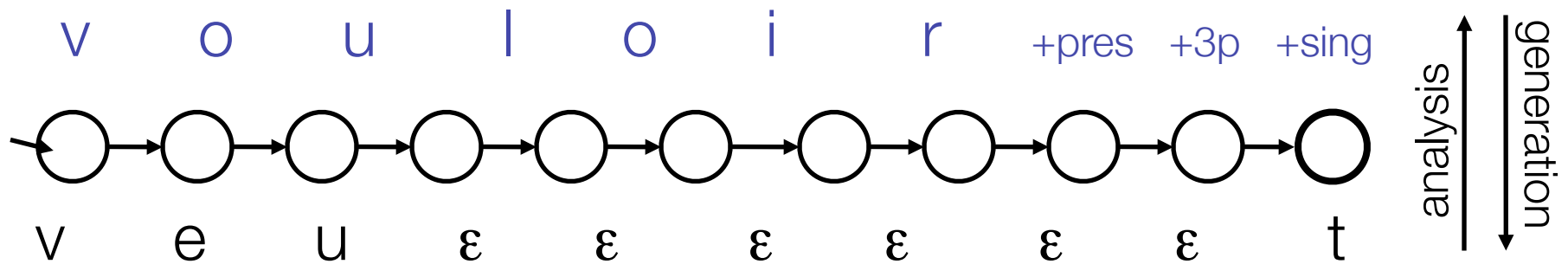
Finite-State Transducers

- Input alphabet Σ
- Output alphabet Δ
- Set of states Q
- Initial state q_0
- Final states $F \subseteq Q$
- Arcs, $Q \times \Sigma \times Q \times \Delta^*$

Finite-State Transducers

- Biggest application: morphology
 - Xerox tools: 20+ languages
- Example ...

This represents one path.



Ambiguity and Optionality

- leaves →

{leaf +N +pl, leave +V +pres +3p +sing}

- advice +maker →

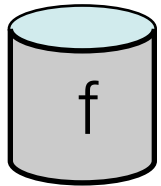
{advisor, adviser}

- inter+ nation +al +ize +ation → {internationalization, internationalisation}

Also, Phonology

- Mapping between pronunciation (phonemes or phonetic symbols) and lexical entries (morpheme sequences or orthography).
- Optionality even more necessary here!

FST Composition



{vouloir → veux,
vouloir → veut,
vouloir → voulons,
vouloir → voulez,
vouloir → veulent,
...}



{veux → vœ,
veut → vœ,
voulons → vulõ,
voulez → vule,
veulent → voel,
...}



{vouloir → vœ,
vouloir → vœ,
vouloir → vulõ,
vouloir → vule,
vouloir → voel,
...}

FST Composition

- Formally, $(x, z) \in f \circ g$ iff there exists y such that $(x, y) \in f$ and $(y, z) \in g$.
- Set and relation:
 $(x, z) \in f \circ g$ iff $x \in f$ and $(x, z) \in g$
- Relation and set:
 $(x, z) \in f \circ g$ iff $(x, z) \in f$ and $z \in g$
- Set and set (intersection):
 $x \in f \circ g$ iff $x \in f$ and $x \in g$

Basically, treat **sets** as identity relations.

Weighted FSAs

- Instead of

Is it grammatical (possible)?

we might ask,

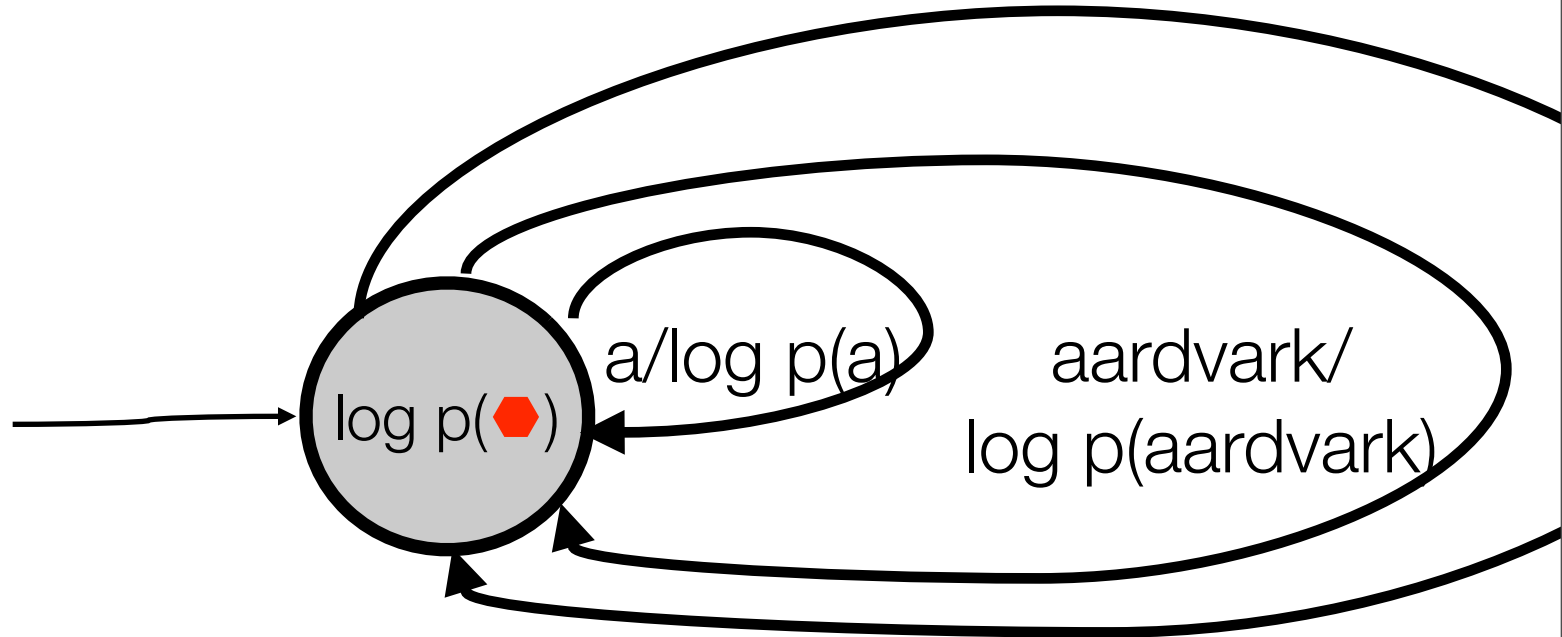
How grammatical (likely) is it?

- Examples:
 - N-gram models
 - HMMs
 - Acoustic lattices

Weighted Finite-State Acceptors

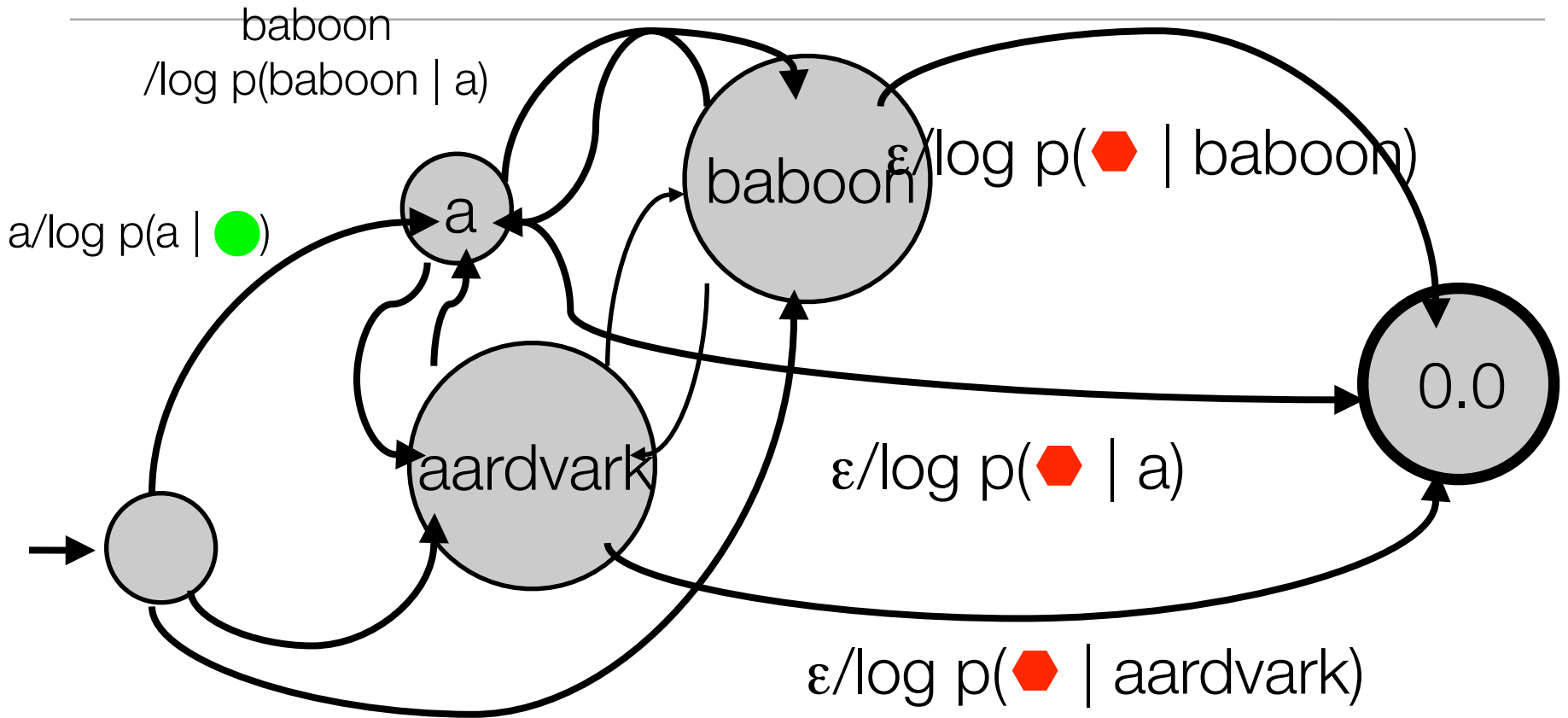
- Alphabet Σ
- Set of states Q
- Initial weight function, $\pi : Q \rightarrow \mathbb{R}$
- Final weight function, $\xi : Q \rightarrow \mathbb{R}$
- Arcs in $Q \times \Sigma \times Q \times \mathbb{R}$

Unigram model as a WFSA



One state.
One arc for every word.

Bigram model as a WFSA



$|\Sigma| + 2$ states.

One arc for every bigram.

Bigram HMM as a WFSA

- Alphabet Σ (HMM's alphabet)
- Set of states Q (HMM's states)
- Initial weight function, $\pi : Q \rightarrow \mathbb{C}$
(0 for start state, $-\infty$ for others)
- Final weight function, $\xi : Q \rightarrow \mathbb{C}$
 $\log \gamma(\blacklozenge \mid q)$
- Arcs in $Q \times \Sigma \times Q \times \mathbb{C}$
 $\log \gamma(q' \mid q) + \log \eta(s \mid q)$

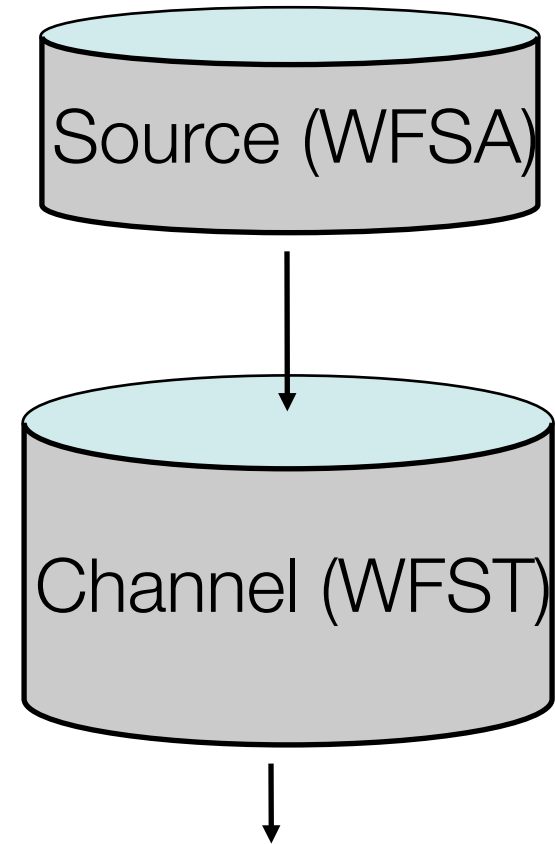
Can you tell how to build a WFSA to implement a **trigram** HMM?

PCFG approximations

- A CFG G can be approximated by a regular grammar G' generating a superset of $L(G)$ with productions of the form:
 - $A \rightarrow aB$
 - $A \rightarrow a$
- From G' we can construct an FSA recognizing $L(G')$ with different paths corresponding to different parses
- A PCFG can therefore be approximated with a WFSA for linear-time parsing

Weighted FSTs

- Weighted relation on $\Sigma^* \times \Delta^*$
- Like FSTs, closed under **composition**
- Examples:
 - Spelling correction
 - Morphological disambiguation
 - Edit distance
 - Machine translation
 - Speech recognition



Toolkits

- FSM libraries (AT&T)
 - Free binaries
 - Implements pretty much everything you need to build weighted and unweighted FS recognizers and transducers ... except training!
- Xerox FS toolkit
 - Web demo; software can be purchased
 - No weights
- RWTH FSA toolkit
 - Newer, open-source
 - Not sure what's implemented
- OpenFST (Google)
 - New incarnation of FSM libraries
 - Free and open source!