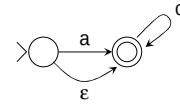


Finite-State Methods

600.465 - Intro to NLP - J. Eisner

1

Finite state acceptors (FSAs)



- Regexprs
- Union, Kleene *, concat, intersect, complement, reversal
- Determinization, minimization
- Pumping, Myhill-Nerode

600.465 - Intro to NLP - J. Eisner

2

slide courtesy of L. Karttunen (modified)

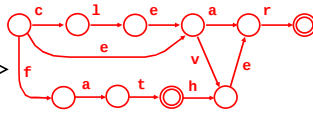
Example: Unweighted acceptor

Wordlist

clear
clever
ear
ever
fat
father

compile

Network



/usr/dict/words
25K words
206K chars

0.6 sec

FSM

17728 states,
37100 arcs

600.465 - Intro to NLP - J. Eisner

3

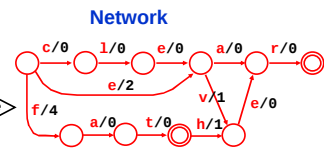
slide courtesy of L. Karttunen (modified)

Example: Weighted acceptor

Wordlist

clear 0
clever 1
ear 2
ever 3
fat 4
father 5

compile



Computes a perfect hash!

600.465 - Intro to NLP - J. Eisner

4

slide courtesy of L. Karttunen (modified)

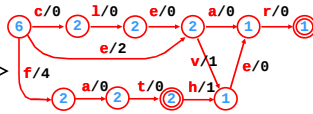
Example: Weighted acceptor

Wordlist

clear 0
clever 1
ear 2
ever 3
fat 4
father 5

compile

Network



- Compute number of paths from each state (Q: how?)
- Successor states partition the path set
- Use offsets of successor states as arc weights
- Q: Would this work for an arbitrary numbering of the words?

600.465 - Intro to NLP - J. Eisner

5

Example: Unweighted transducer

VP [head=vouloir,...]

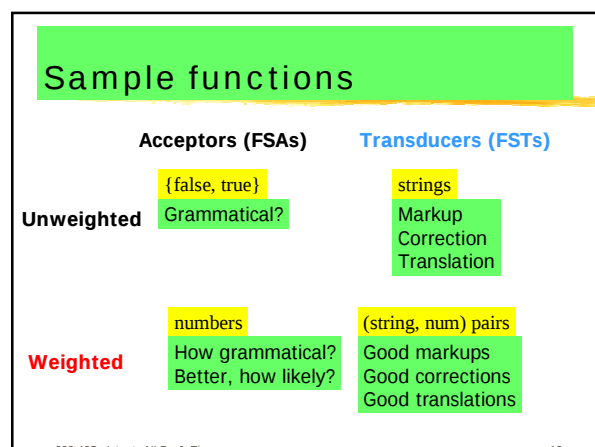
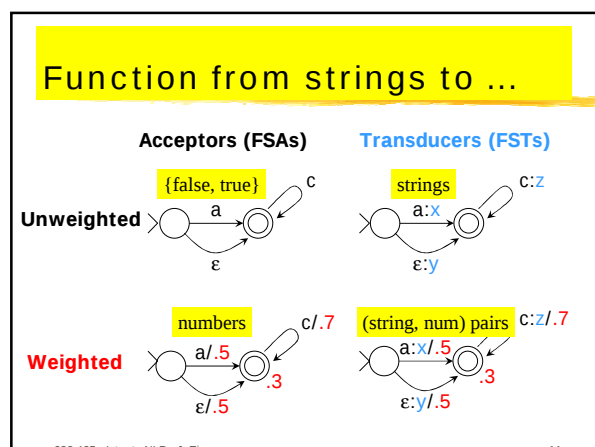
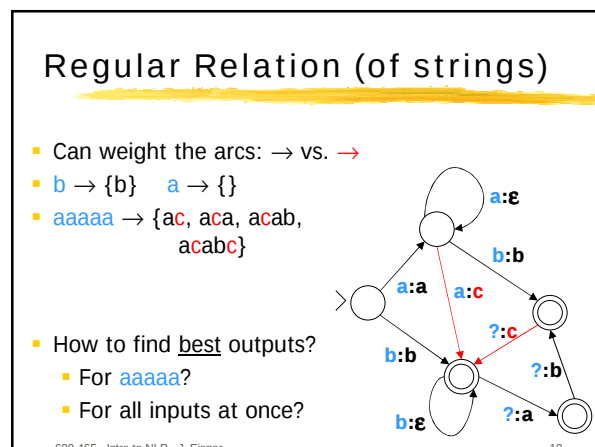
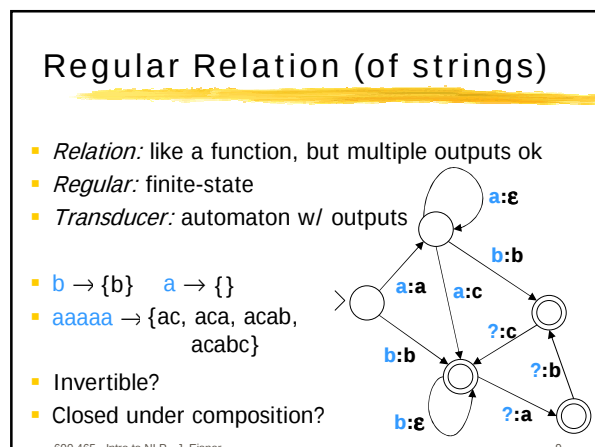
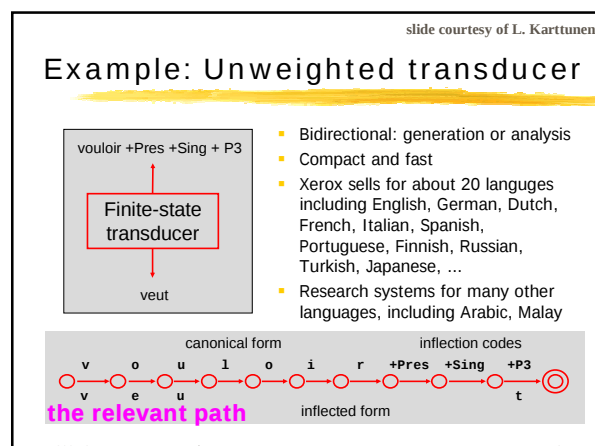
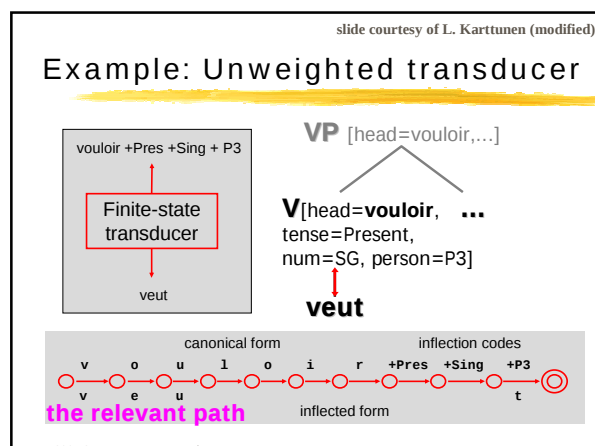
V[head=**vouloir**, ...
tense=Present,
num=SG, person=P3]

veut

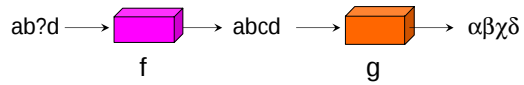
the problem of morphology ("word shape") - an area of linguistics

600.465 - Intro to NLP - J. Eisner

6



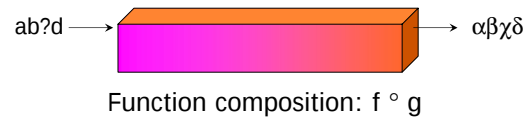
Functions



600.465 - Intro to NLP - J. Eisner

13

Functions

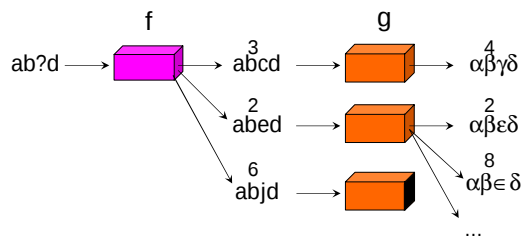


[first f , then g – intuitive notation, but opposite of the traditional math notation]

600.465 - Intro to NLP - J. Eisner

14

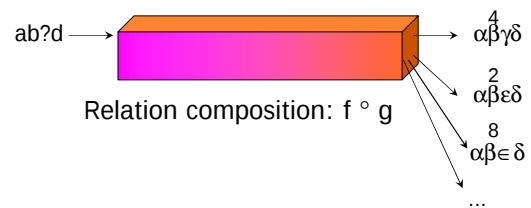
From Functions to Relations



600.465 - Intro to NLP - J. Eisner

15

From Functions to Relations



600.465 - Intro to NLP - J. Eisner

16

From Functions to Relations

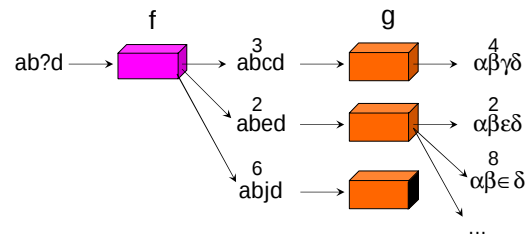


Often in NLP, all of the functions or relations involved can be described as finite-state machines, and manipulated using standard algorithms.

600.465 - Intro to NLP - J. Eisner

17

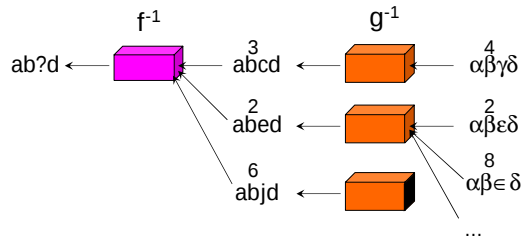
Inverting Relations



600.465 - Intro to NLP - J. Eisner

18

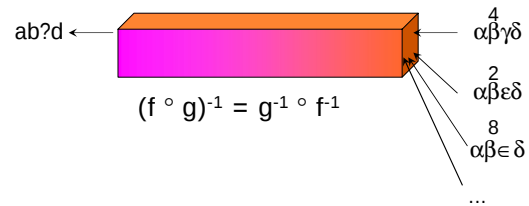
Inverting Relations



600.465 - Intro to NLP - J. Eisner

19

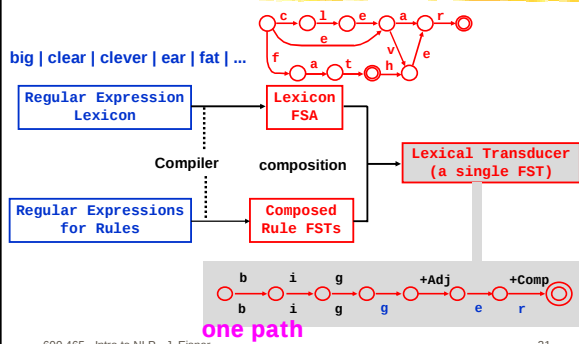
Inverting Relations



600.465 - Intro to NLP - J. Eisner

20

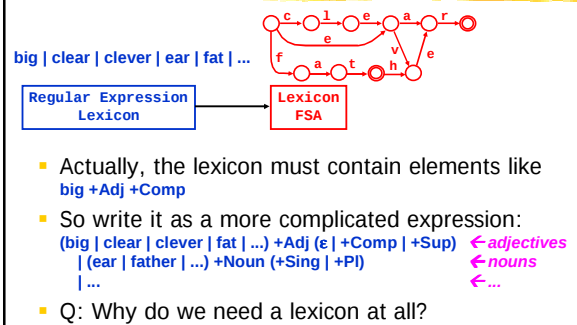
Building a lexical transducer



600.465 - Intro to NLP - J. Eisner

21

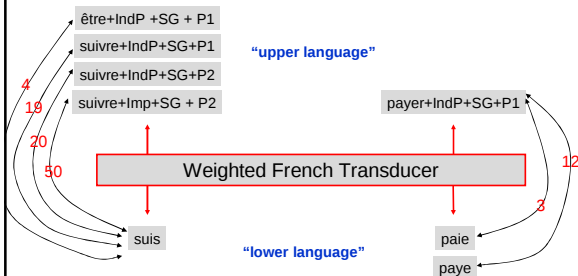
Building a lexical transducer



600.465 - Intro to NLP - J. Eisner

22

Weighted version of transducer: Assigns a weight to each string pair



600.465 - Intro to NLP - J. Eisner

23