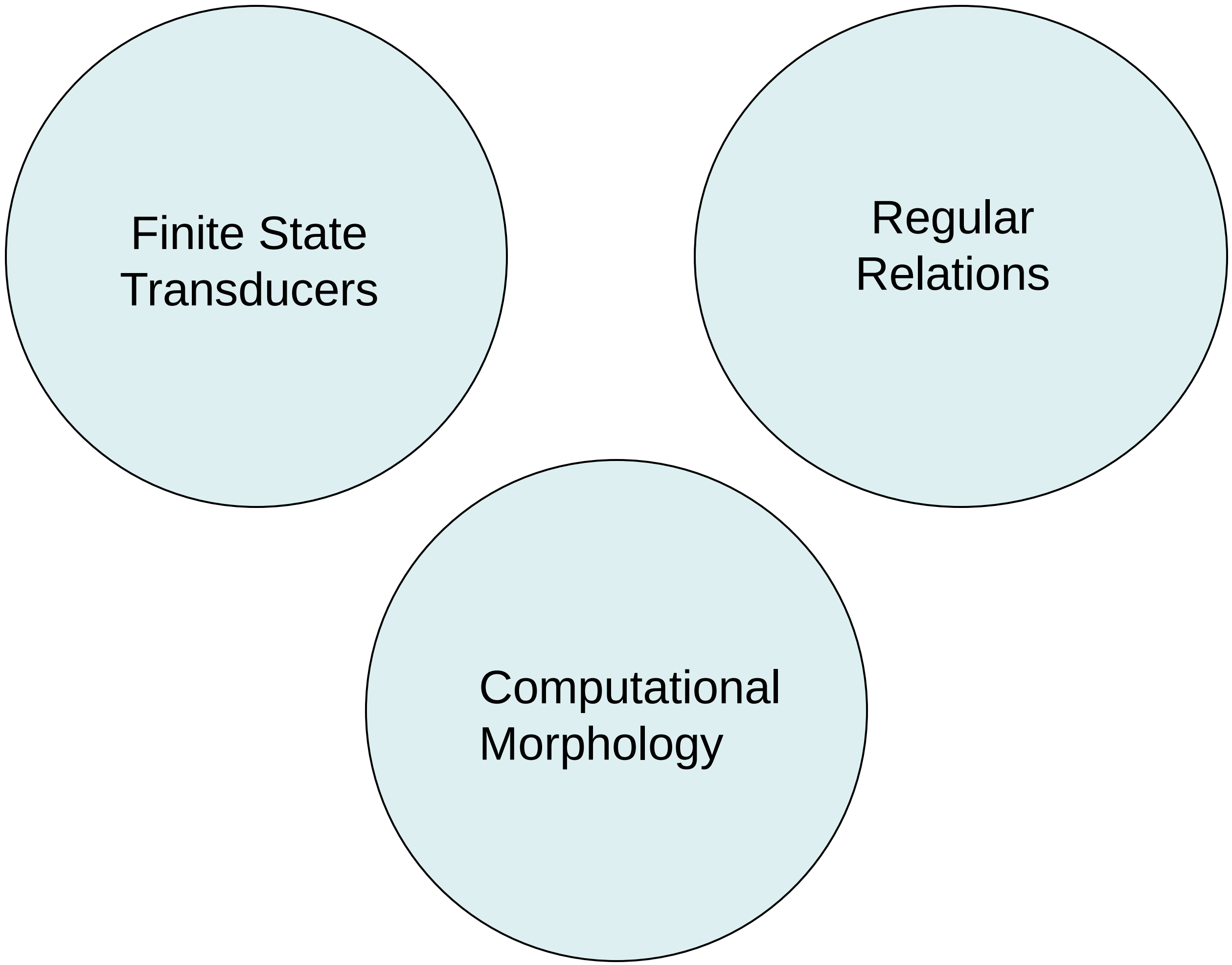
Human Language Technology

Finite State Transducers

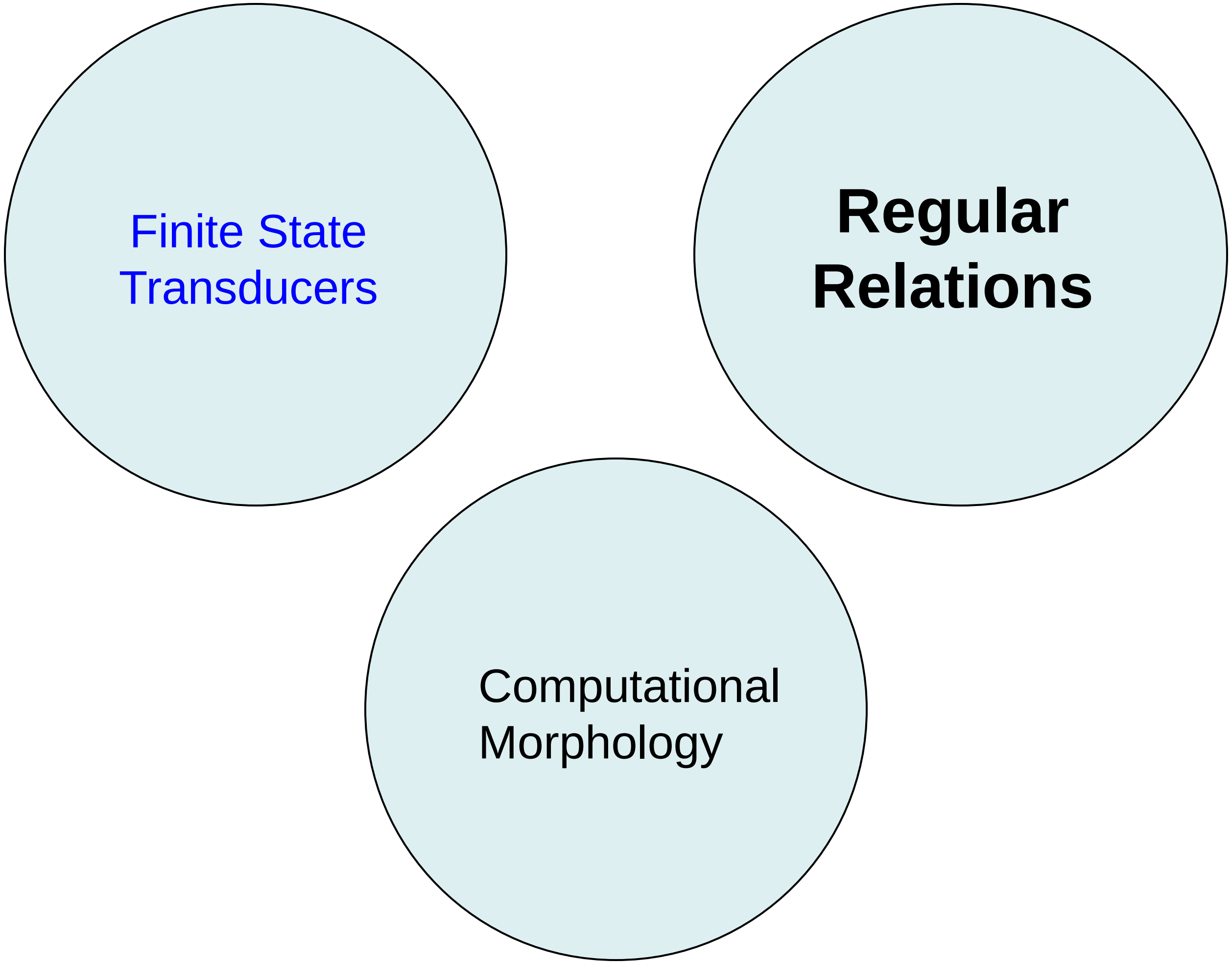
Acknowledgement

- Material in this lecture derived/copied in part from
  - Richard Sproat CL46 Lectures
  - Lauri Karttunen LSA lectures 2005
  - Shuly Wintner 2008 Malta

Three Key Concepts



Three Key Concepts



A Regular Set

L1

ab
abab
ababab
abababab
ababababab
.
.

Two Regular Sets

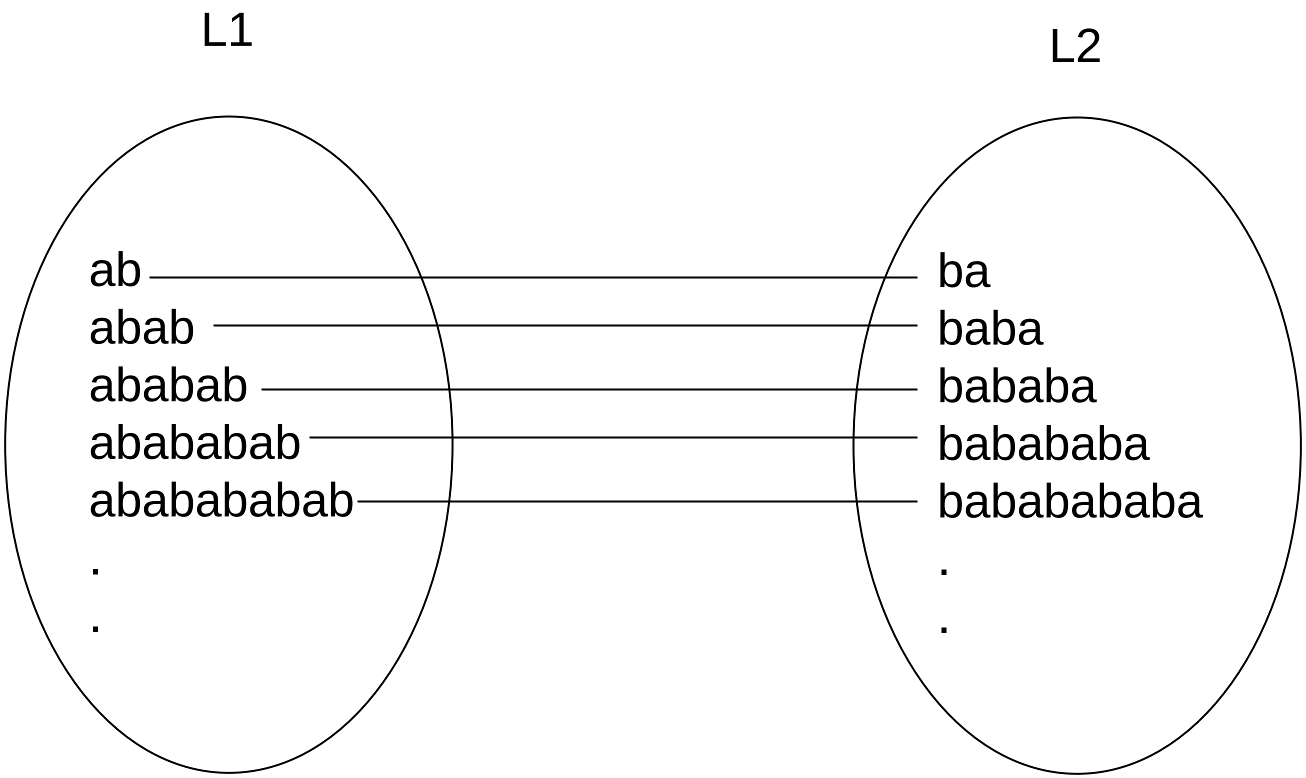
L1

ab
abab
ababab
abababab
ababababab
.
.

L2

ba
baba
bababa
babababa
bababababa
.
.

A Regular Relation $\subseteq$ L1 x L2



or {("ab","ba"), ("abab","baba"),...}

Some closure properties for regular relations

- Concatenation [R1 R2]
- Power (R^n)
- Reversal
- Inversion (R^{-1})
- Composition: $R_1 \circ R_2$

Concatenation and Power

Concatenation

R1 = {("a","b")}

R2 = {("c","d")}

[R1 R2] = {("ac","bd")}

Power

R1+ = {("a","b"),("aa","bb"), ("aaa","bbb"), ...}

Composition

- $R_1 \circ R_2$ denotes the composition of relations R_1 and R_2.
- Definition

  If R_1 contains <x,y>

  And R_2 contains <y,z>

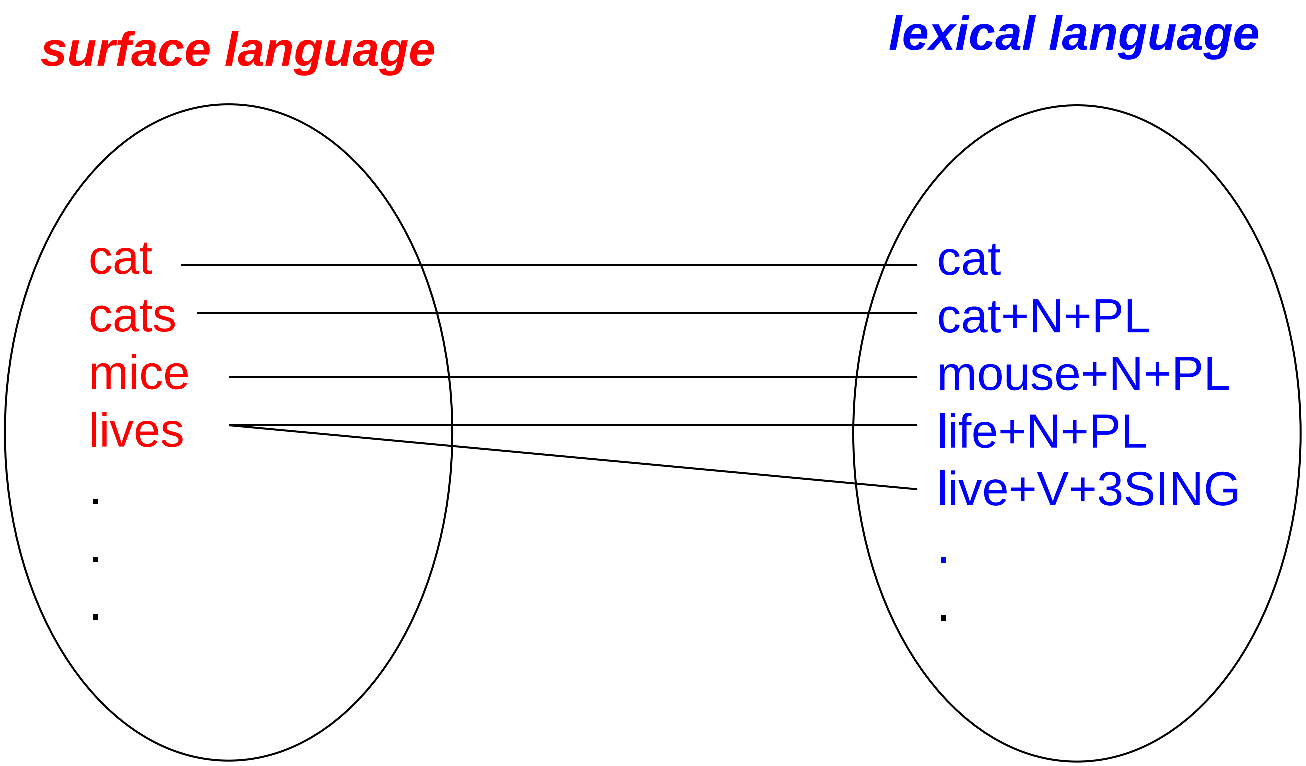
  Then $R_1 \circ R_2$ contains <x,z>
- R_1 and R_2 and B must be *relations*. If either is just a language, it is assumed to abbreviate the identity relation.
- $R_1 \circ R_2$ is written `[R1 .o. R2]` in xfst

61

Closure Properties of Regular Languages and Relations

Operation	**Regular Languages**	**Regular Relations**
Union	yes	yes
Concatenation	yes	yes
Iteration	yes	yes
Intersection	yes	no
Subtraction	yes	no
Complementation	yes	no
Composition	n/a	yes

Morphology as a Regular Relation



or {("cat,cat"),("cats","cat+N+PL"),......}

Part-of-Speech Tagging

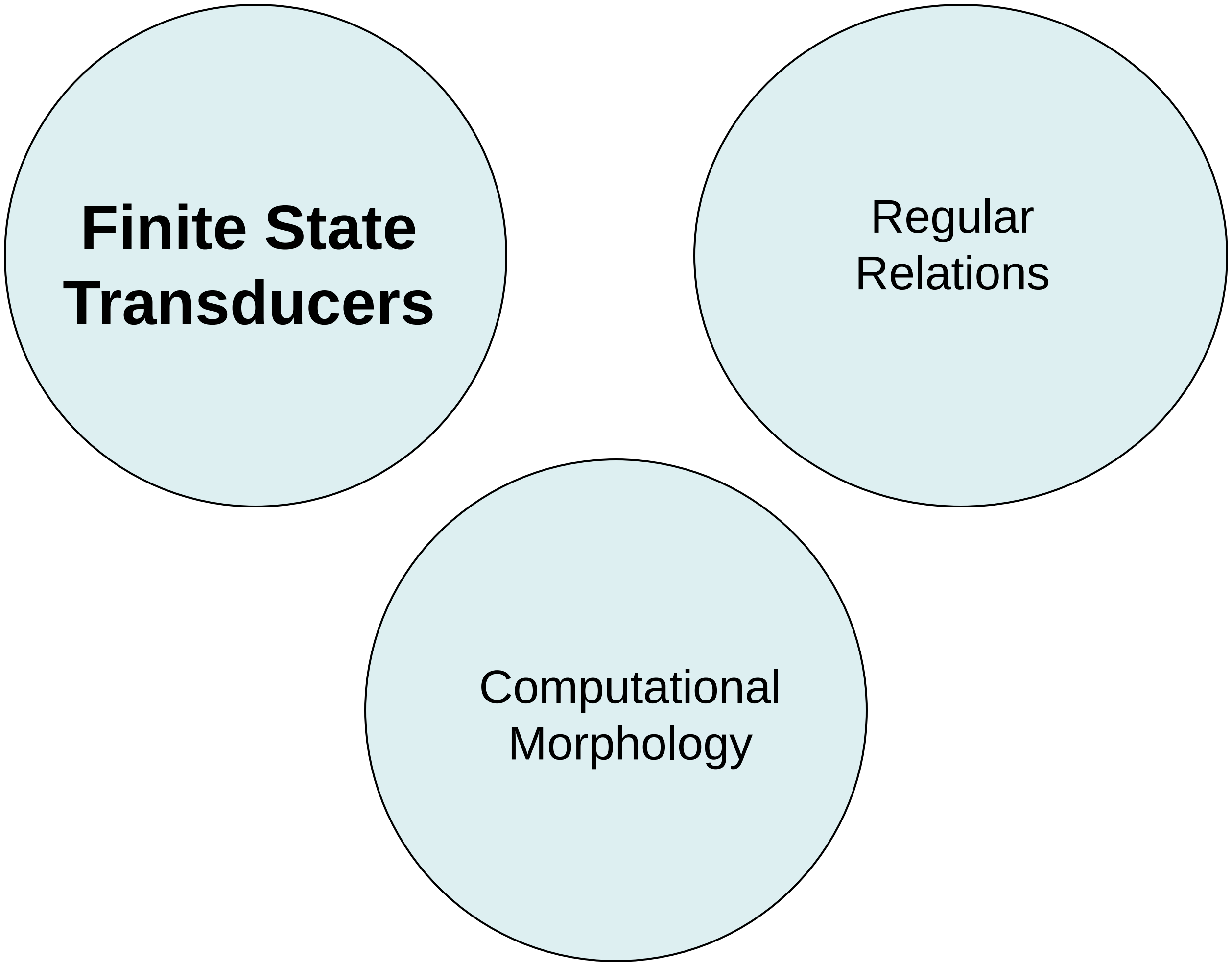
- `I    know some new tricks`
- `PRON V    DET  ADJ N`

- `said the Cat in the Hat`
- `V    DET N   P  DET N`

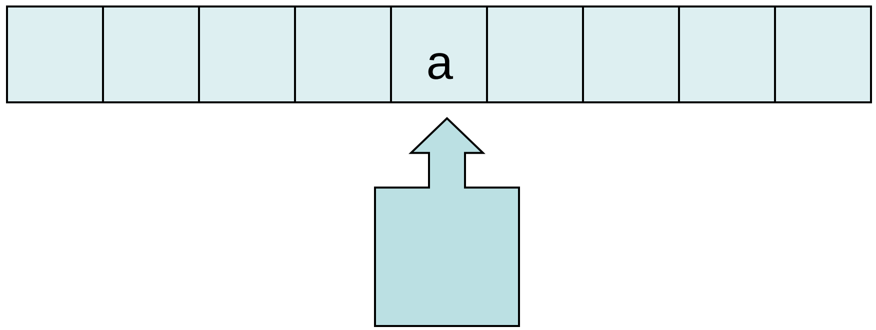
Singular-to-plural mapping:

• `cat`	`hat`	`ox`	`child`	`mouse`	`sheep`
• `cats`	`hats`	`oxen`	`children`	`mice`	`sheep`

Three Key Concepts



FSA



Used for
- Recognition
- Generation

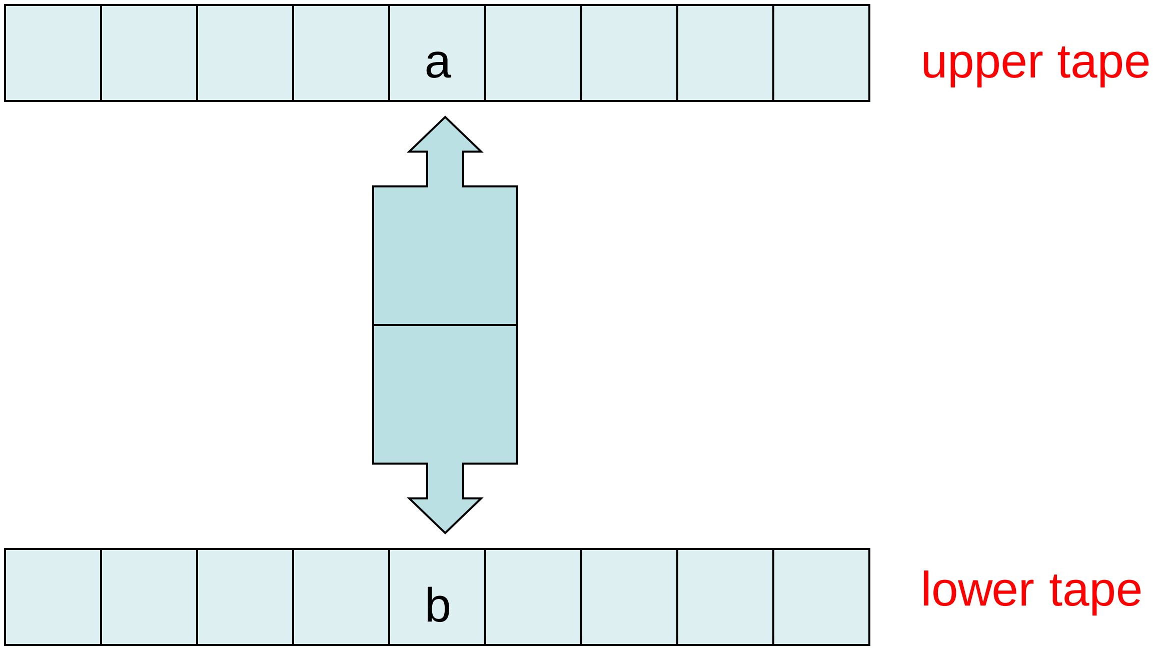
Finite State Transducers

- A finite state transducer (FST) is essentially an FSA finite state automaton that works on two (or more) tapes.
- The most common way to think about transducers is as a kind of translating machine which works by reading from one tape and writing onto the other.

FST Definition

- A 2 way FST is a quintuple $(K,s,F,\Sigma_i x \Sigma_o,\delta)$ where
- Σ_i, Σ_o are input and output alphabets
- K is a finite set of states
- $s \in K$ is an initial state
- $F \subseteq K$ are final states
- δ is a transition relation of type $K \times \Sigma_i \times \Sigma_o \times K$

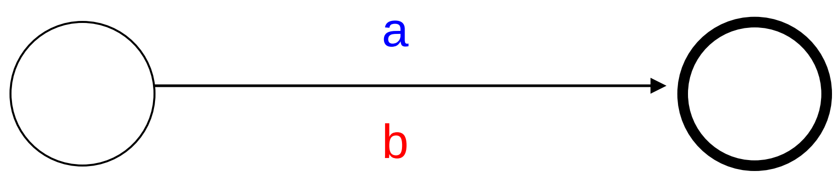
FST



Used for

- Recognition
- Generation
- **Translation**

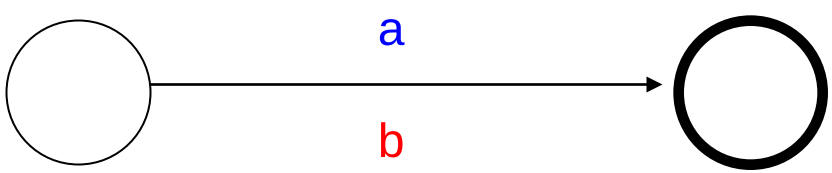
A Very Simple Transducer



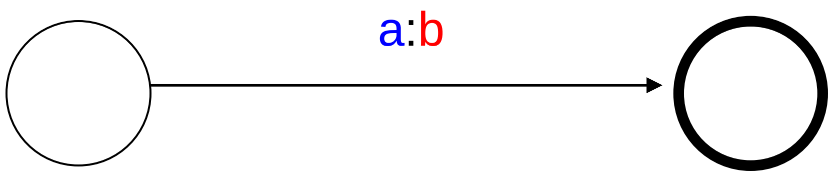
Relation { ("a","b") }

Notation a:b encodes the transition

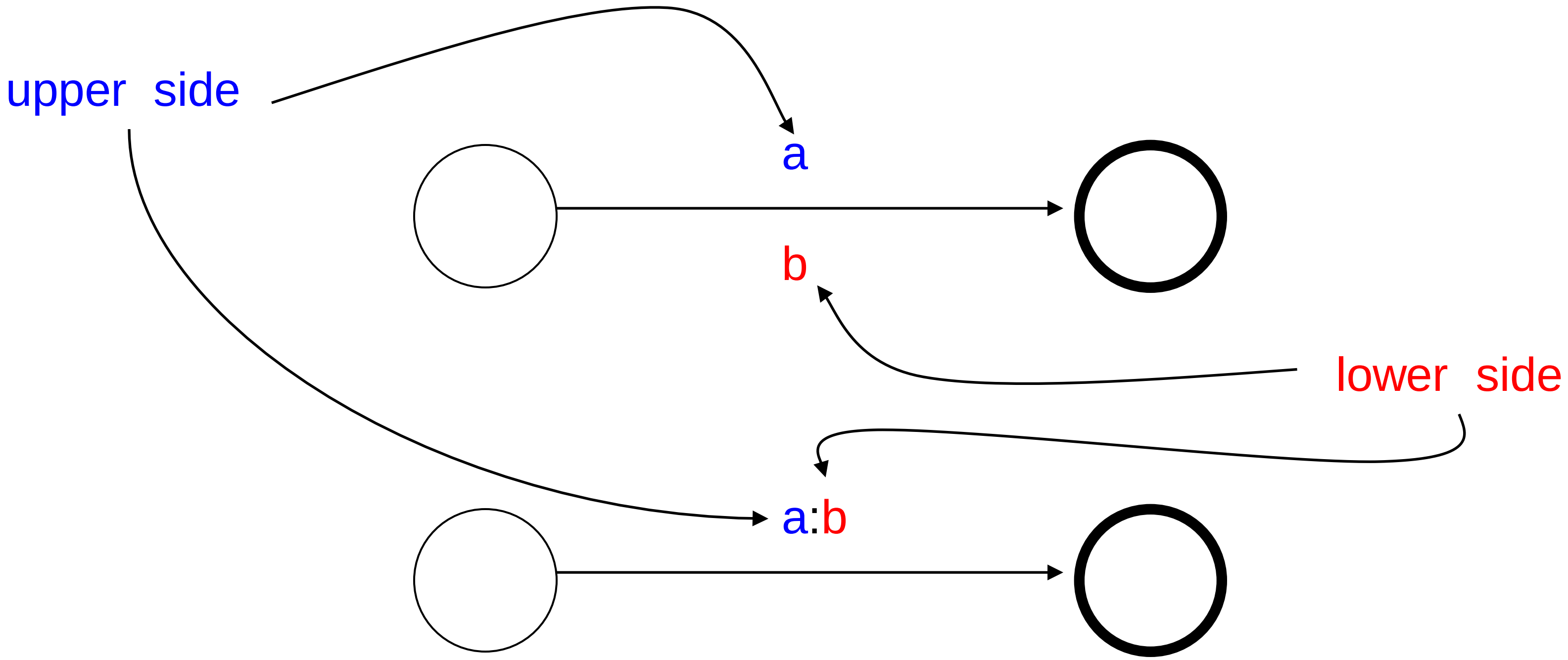
A Very Simple Transducer



also written as



A Very Simple Transducer

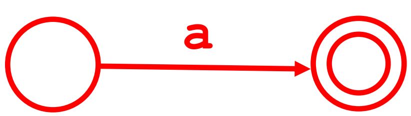


Symbol Pairs

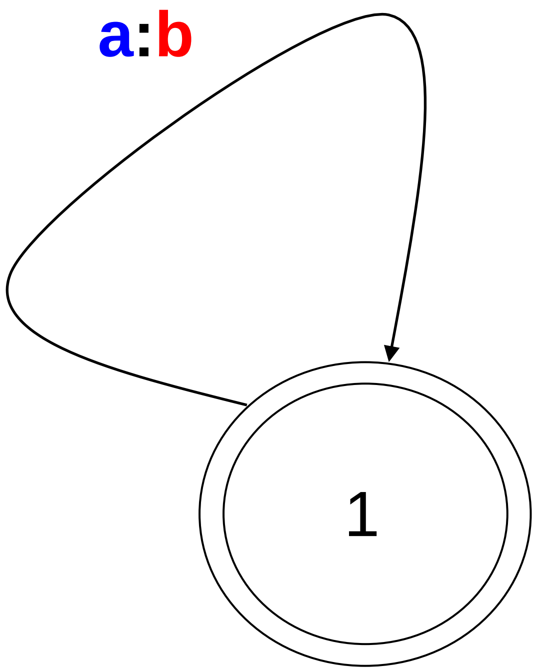
- Symbols vs. symbol pairs
  - In general, no distinction is made in xfst between

    a the language {“a”}

    a:a the identity relation {(“a”, “a”)}



A (more interesting) Transducer



- Relation

  { ("a","b"), ("aa","bb"), ...}

- Notation

  a:b*

- N.B. with this notation a and b must be single symbols

Transducer have Several Modes of Operation

- **generation mode**: It writes on both tapes. A string of **a**s on one tape and a string of **b**s on the other tape. Both strings have the same length.
- **recognition mode**: It accepts when the word on the first tape consists of exactly as many **a**s as the word on the second tape consists of **b**s.
- **translation mode** (left to right): It reads as from the first tape and writes a b for every a that it reads onto the second tape.
- **translation mode** (right to left): It reads **b**s from the second tape and writes an **a** for every **b** that it reads onto the first tape.

The Basic Idea

- Morphology is regular
- Morphology is finite state

Morphology is Regular

- The relation between the **surface forms** of a language and the corresponding **lexical forms** can be described as a **regular relation,** e.g.

  { ("leaf+N+Pl","leaves"),("hang+V+Past","hung"),...}

- Regular relations are **closed** under operations such as **concatenation**, **iteration**, **union**, and **composition**.
- Complex regular relations can be derived from simpler relations.

Morphology is finite-state

- A regular relation can be defined using the metalanguage of **regular expressions**.

- `[{talk} | {walk} | {work}]`
- `[%+Base:0 | %+SgGen3:s | %+Progr:{ing} | %+Past:{ed}];`

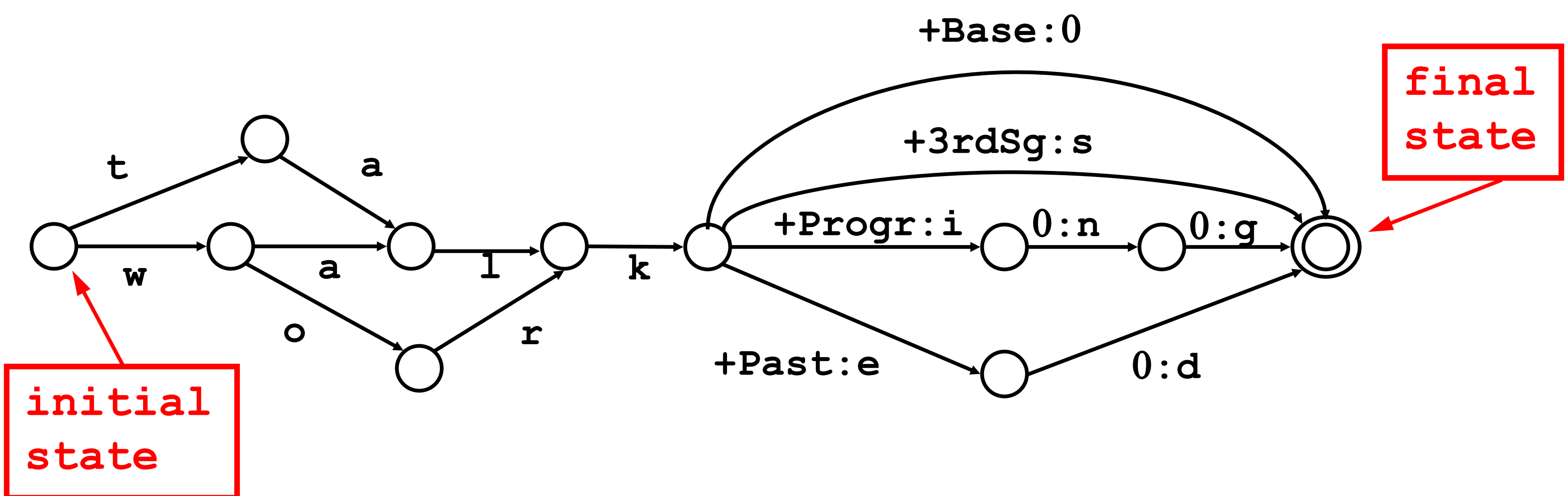
- A regular expression can be compiled into a **finite-state transducer** that implements the relation computationally.

Compilation

Regular expression

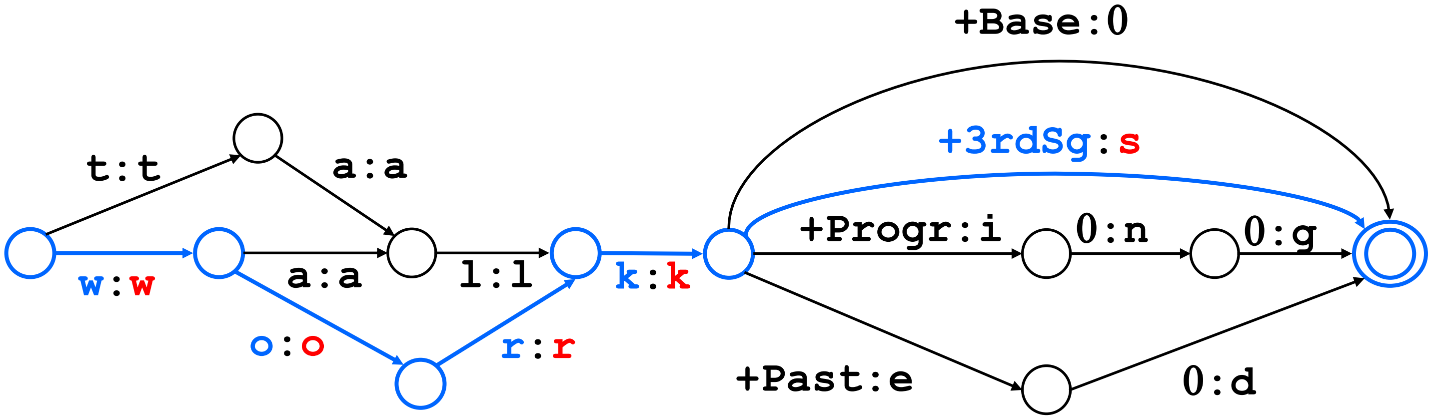
- **[{talk} | {walk} | {work}]**
- **[%+Base:0 | %+SgGen3:s | %+Progr:{ing} | %+Past:{ed}];**

Finite-state transducer

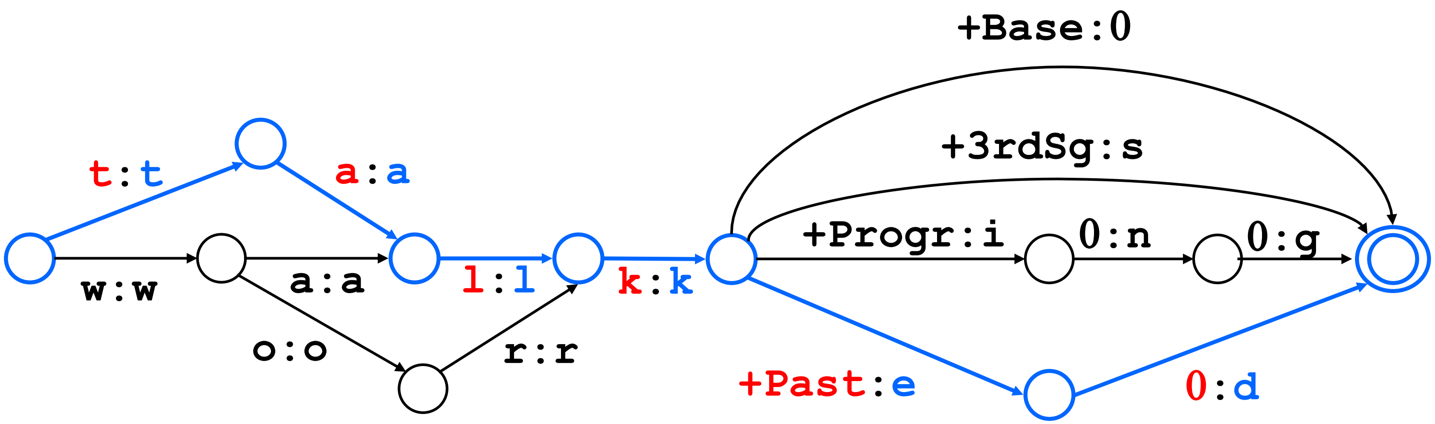


Generation

work+3rdSg --> works



Generation



talked --> talk+Past

XFST Demo 2

```
% xfst
xfst[0]:
```

start xfst

compile a regular expression

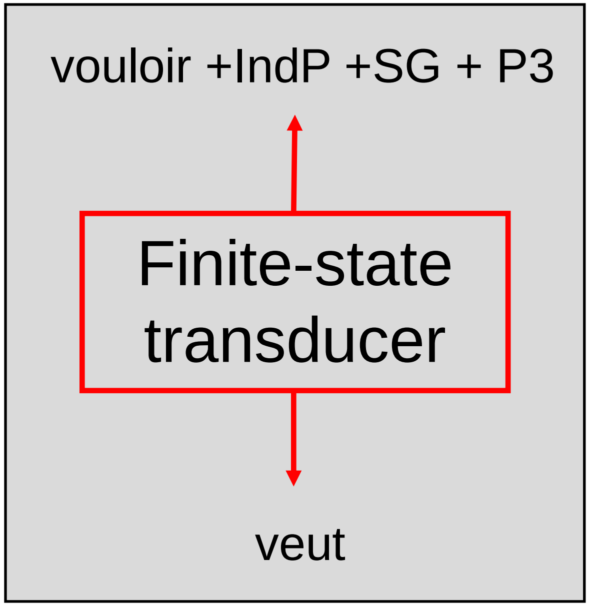
- `xfst[0]: regex`
- `[{talk} | {walk} | {work}]`
- `[% +Base:0 | %+SgGen3:s | %+Progr:{ing} | %+Past:{ed}];`

```
xfst[1]: apply up walked
walk+Past
```

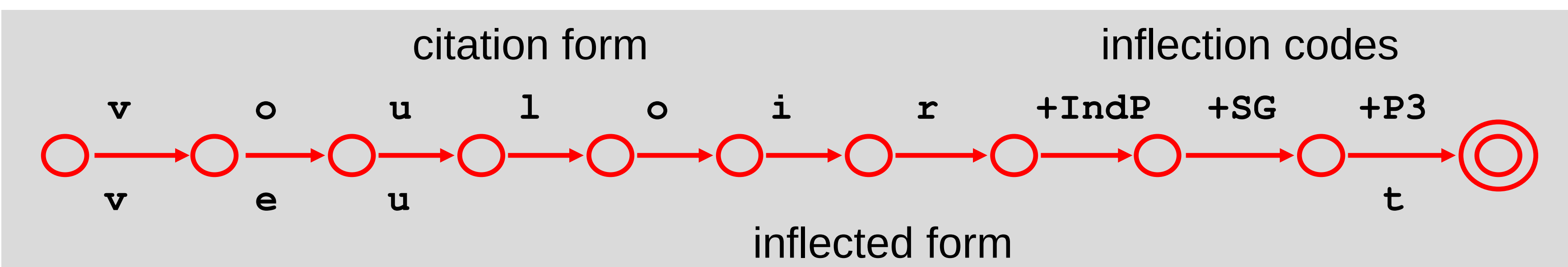
apply the result

```
xfst[1]: apply down talk+SgGen3
talks
```

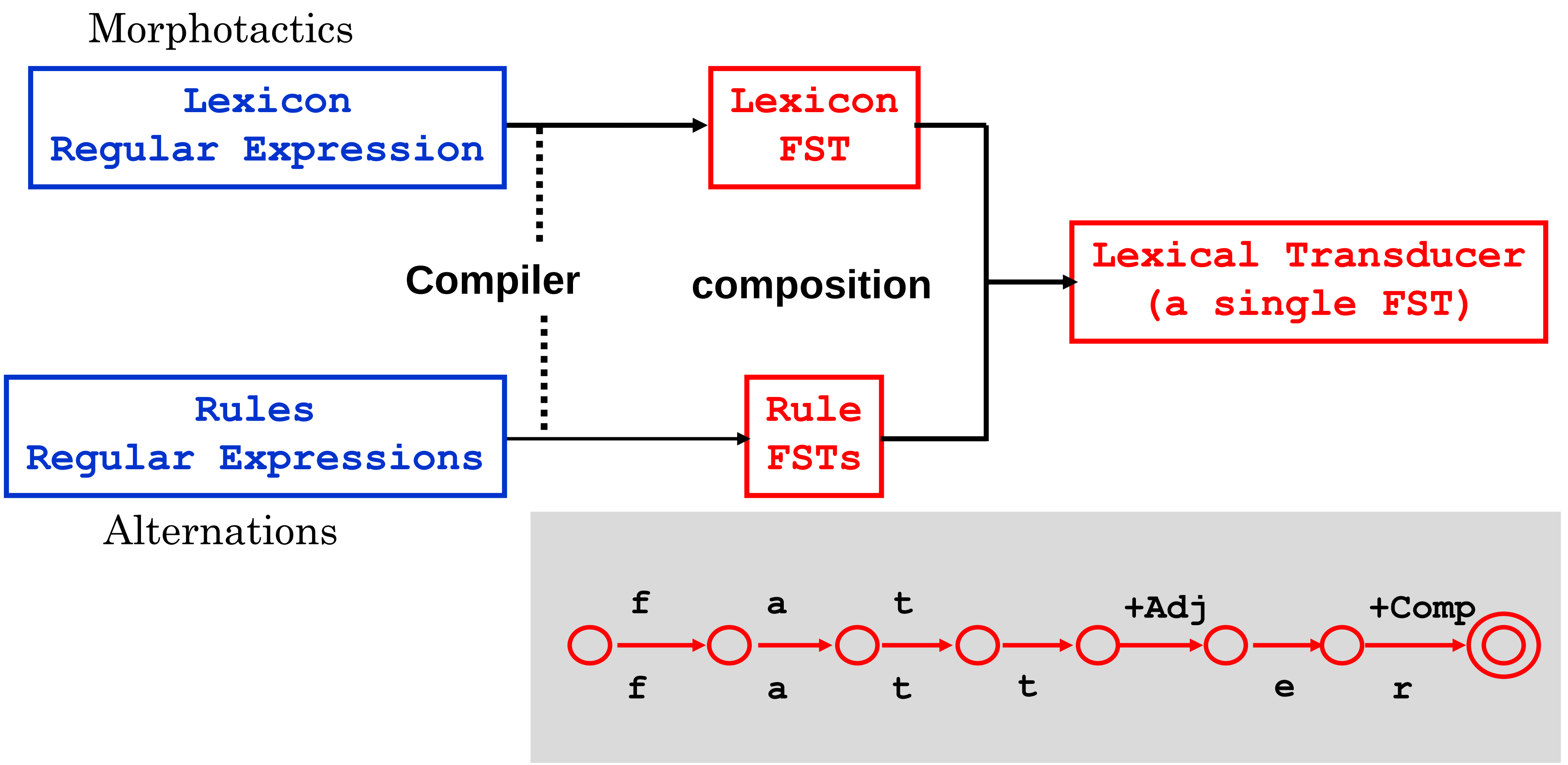
Lexical transducer



- Bidirectional: generation or analysis
- Compact and fast
- Comprehensive systems have been built for over 40 languages:
  – English, German, Dutch, French, Italian, Spanish, Portuguese, Finnish, Russian, Turkish, Japanese, Korean, Basque, Greek, Arabic, Hebrew, Bulgarian, …



How lexical transducers are made



Sequential Model

Lexical form

fst 1

Intermediate form

fst 2

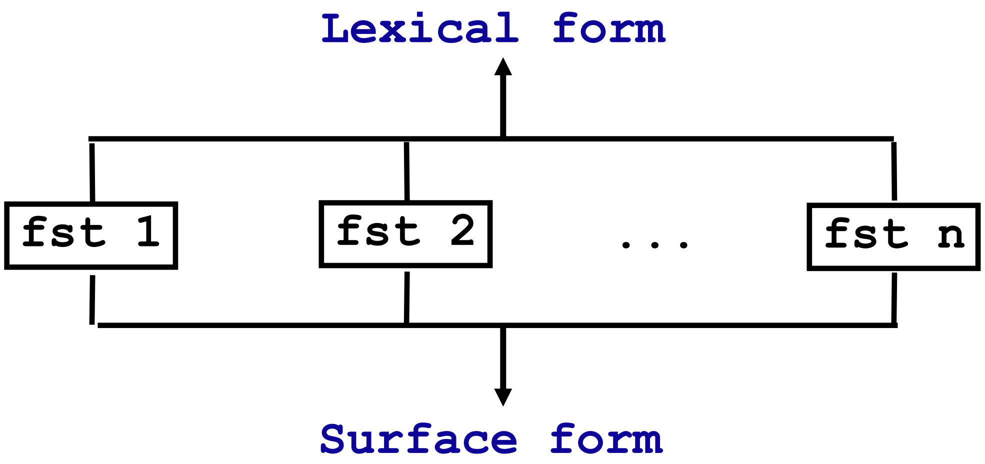
. . .

fst n

Surface form

Ordered sequence
of rewrite rules
(Chomsky & Halle '68)
can be modeled
by a cascade of
finite-state transducers
Johnson '72
Kaplan & Kay '81

Parallel Model

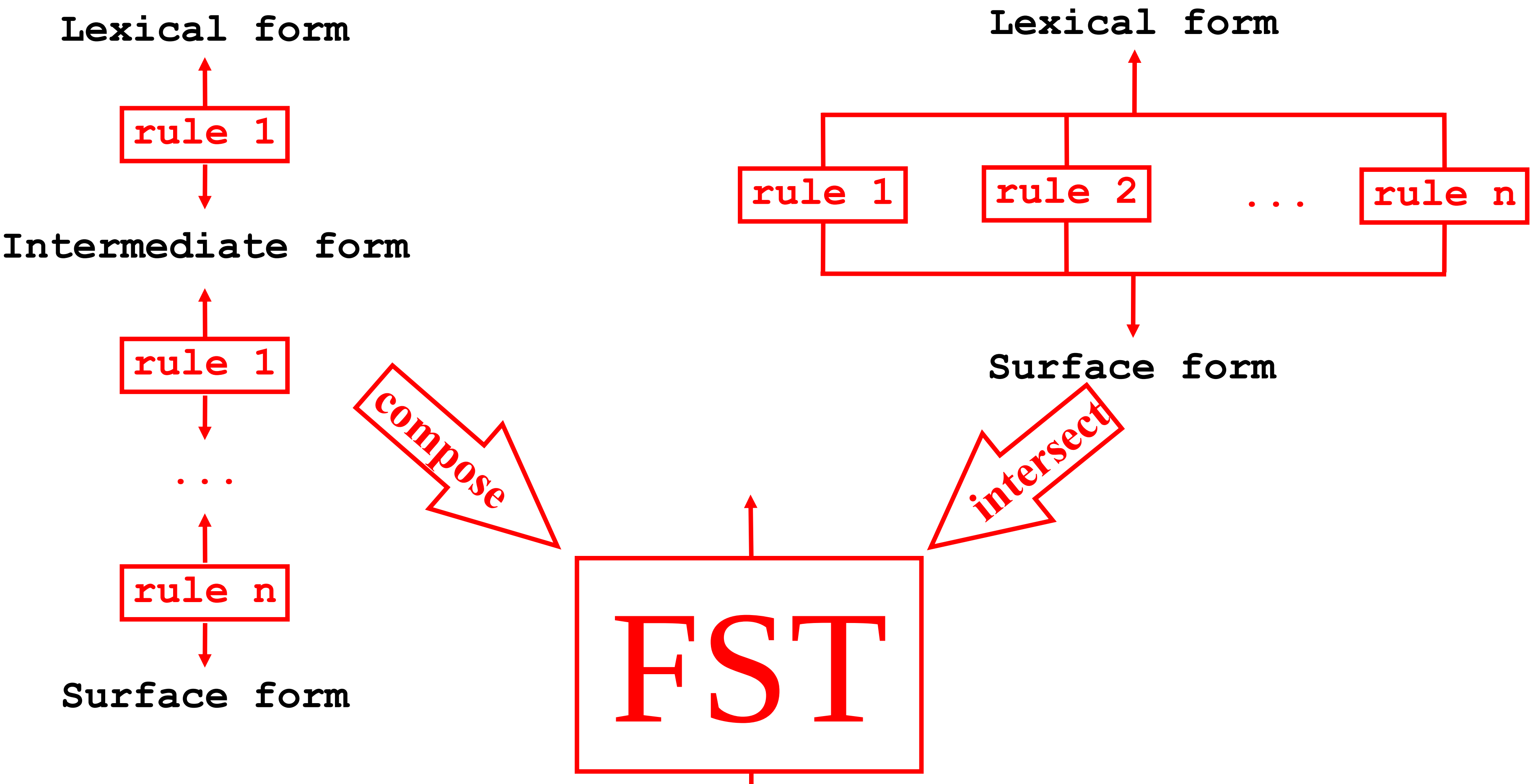


Set of parallel
of two-level rules (constraints)
compiled into finite-state automata
interpreted as transducers
Koskenniemi ‘83

Sequential vs. Parallel rules

Chomsky&Halle 1968

Koskenniemi 1983

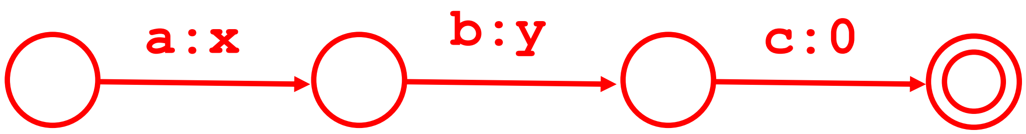


Sequential vs. Parallel Rules

- Sequential rules are combined by means of composition.
- Advantage: FSTs are closed under composition
- Disadvantage: order of operations is sensitive
- Parallel rules are combined by means of intersection
- In general, FSTs are not closed under intersection.
- … but FSTs without ε-transitions are closed under intersection.

Crossproduct

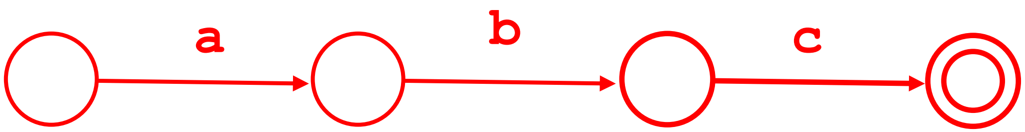
- A .x. B The relation that maps every string in A to every string in B, and vice versa
- A:B Same as [A .x. B].

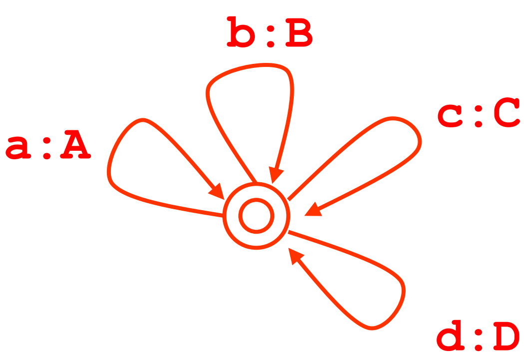


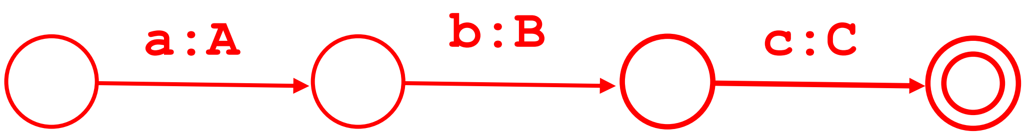
a b c .x. x y **[a b c] : [x y]** **{abc}:{xy}**

Composition

- A .o. B The relation C such that if A maps x to y and B maps y to z, C maps x to z.

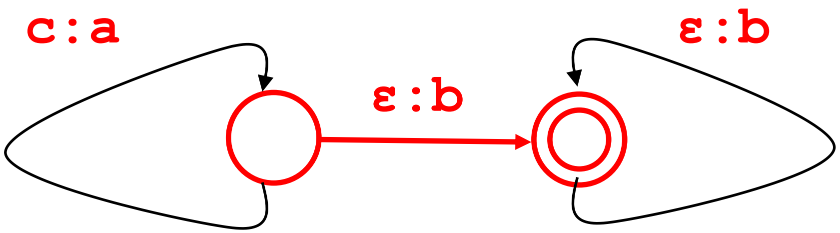






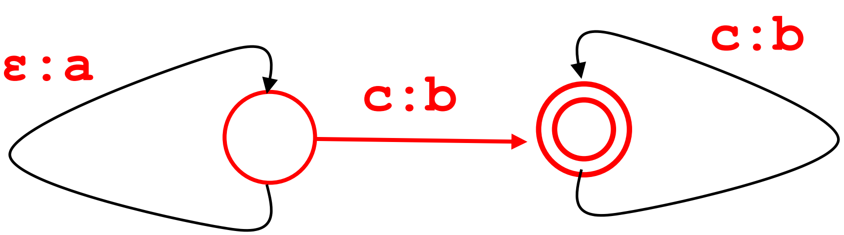
{abc} .o. [a:A | b:B | c:C | d:D]*

Transducers are not closed under intersection



$T_1(C^n) = \{\ a^n b^m \mid m \geq 0\ \}$

$T_1 \cap T_2\ (C^n) = \{\ a^n b^n\ \}$



$T_2(C^n) = \{\ a^m b^n \mid m \geq 0\ \}$

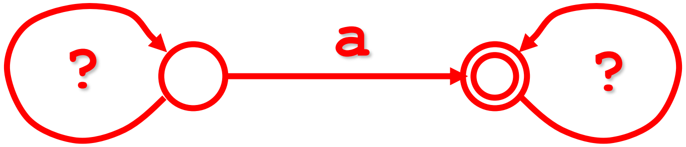
Xerox RE Operators

- **$** containment
- **=>** restriction
- **->** replacement

  – Make it easier to describe complex languages and relations without extending the formal power of finite-state systems.

Containment

`$a`



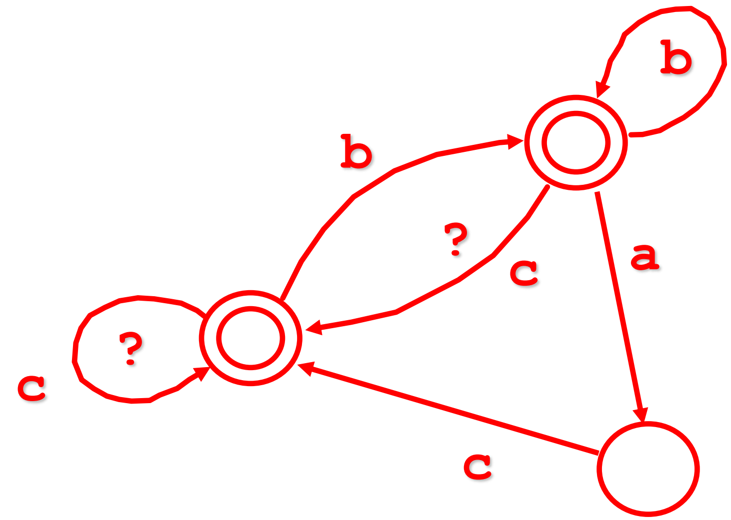
a

`[?* a ?*]`

Restriction

`a => b _ c`

"Any `a` must be preceded by `b` and followed by `c`."



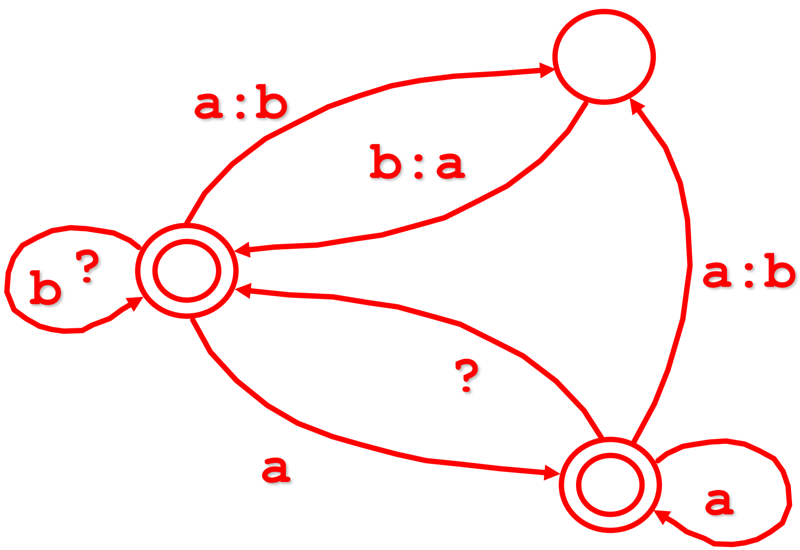
`~[~[?* b] a ?*] & ~[?* a ~[c ?*]]`

Equivalent expression

Replacement

a b -> b a

"Replace 'ab' by 'ba'."

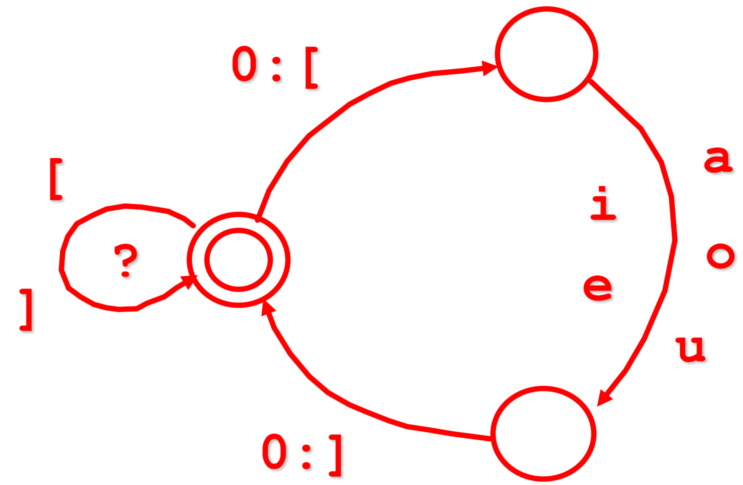


[[~$[a b] [[a b] .x. [b a]]]* ~$[a b]]

Equivalent expression

Replacement + Marking

`a|e|i|o|u -> %[ ... %]`




```
p o t a t o
p[o]t[a]t[o]
```

Conditional Replacement

`A -> B` **`L _ R`**

Replacement Context

The relation that replaces **`A`** by **`B`** between **`L`** and **`R`** leaving everything else unchanged.

Sequential application

k a N p a n

N -> m / _ p

k a m p a n

p -> m / m _

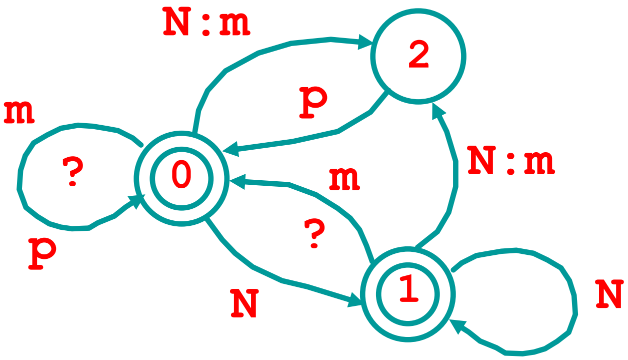
k a m m a n

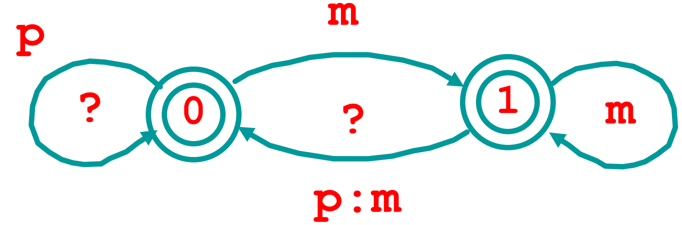
Sequential application in detail

```
 k   a   N   p   a   n
0   0   0 | 2   0   0   0

 k   a   m   p   a   n
0   0   0   1 | 0   0   0

 k   a   m   m   a   n
```





Composition