Introduction to Natural Language Processing

Rule-based NLP and Morphological Analysis

PLAN OF THE LECTURE

- Foundations of Rule-based NLP
- Morphology with FSTs and Tries
- Morphological Analysis: Task Formulations

Definitions

- A **language** is a collection of sentences of finite length all constructed from a finite alphabet of symbols
- A **grammar** can be regarded as a device that enumerates the sentence of a language
- A grammar of language L can be regarded as a function whose range is exactly L

- Jurafsky, D. and Martin, J. H. (2009): Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Second Edition. Pearson: New Jersey. Chapters 2 and 16
- Chomsky, Noam (1959). "On certain formal properties of grammars". Information and Control 2 (2): 137–167

Formal Grammar

A **formal grammar** is a quad-tuple *G = (Φ,Σ,R,S)* where

- Φ is a finite set of **non-terminals**
- Σ a finite set of **terminals**, disjoint from Φ
- R a finite set of **production rules** of the form

$$\alpha \in (\Phi \cup \Sigma)^* \rightarrow \beta \in (\Phi \cup \Sigma)^* \text{ with } \alpha \neq \varepsilon \text{ and } \alpha \notin \Sigma^*$$

- S, Element of Φ: **start symbol**

Derivation, Formal Language, Automaton

Let $G = (\Phi,\Sigma,R,S)$ be a formal grammar and let $u, v \in (\Phi \cup \Sigma)^*$.

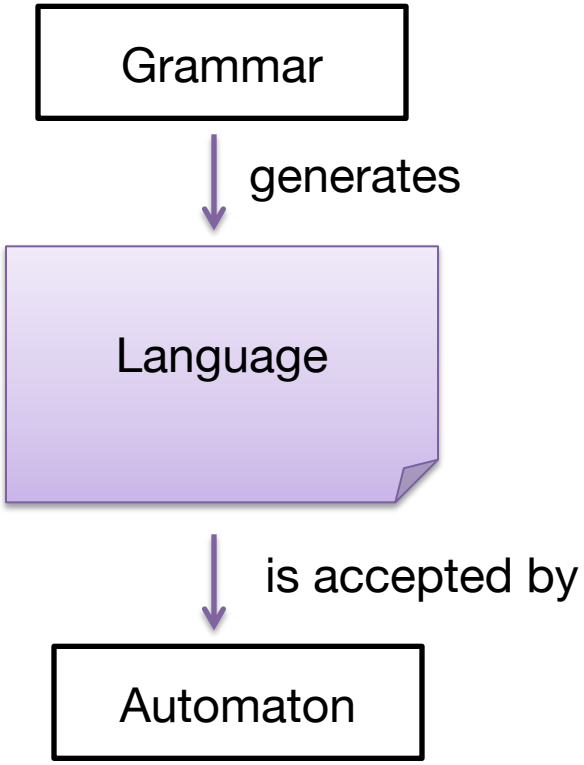
1. v is **directly derivable** from u, noted $u \Rightarrow v$, if $u = awb$, $v = azb$ and $w \rightarrow z$ is a production rule in R.

2. v is **derivable** from u, noted $u \overset{*}{\Rightarrow} v$, if there are words $w_0..w_k$, such that $u \Rightarrow w_0$, $w_{n-1} \Rightarrow w_n$ for all $0 < n \leq k$ and $w_n \overset{*}{\Rightarrow} v$.

Let $G = (\Phi, \Sigma, R, S)$ be a formal grammar. Then,
is the **formal language** generated by G. $L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$

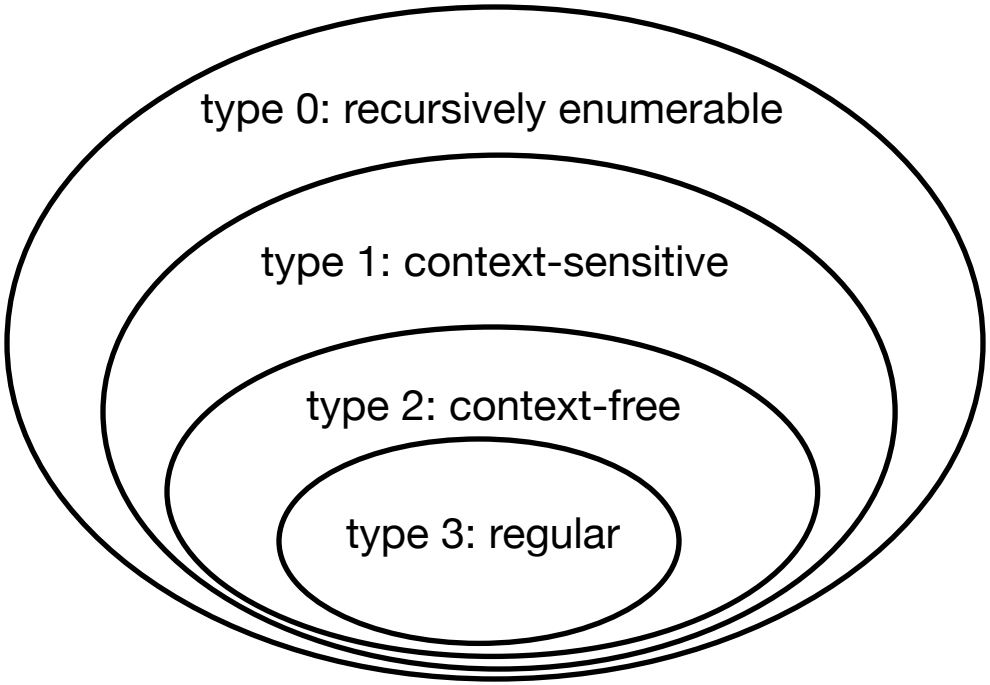
An **automaton** is a device that decides, whether a given sentence belongs to a formal language.

Generation and Acceptance



The complexity of the generating grammar influences the complexity of the accepting automaton

The Chomsky Hierarchy of Formal Languages



Turing Machine (TM)

Linearly bounded TM

Pushdown Automaton (PDA)

Finite State Automaton (FSA)

- The different classes are proper subsets of each other: the expressivity of type-(n) grammars is truly smaller than type-(n-1) grammars.
- Several other classes are known, e.g. corresponding to deterministic context-free grammars, tree adjoining grammars …

The Chomsky's four types of grammar

Grammar	Languages	Recognizing Automaton	Production rules (constraints)*	Examples[5][6]
Type-3	Regular	Finite state automaton	$A \to \mathrm{a}$ and $A \to \mathrm{a}B$	$L = \{a^n \mid n \geq 0\}$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \to \alpha$	$L = \{a^n b^n \mid n > 0\}$
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \to \alpha \gamma \beta$	$L = \{a^n b^n c^n \mid n > 0\}$
Type-0	Recursively enumerable	Turing machine	$\gamma \to \alpha$ (γ non-empty)	$L = \{w \mid w$ describes a terminating Turing machine$\}$

* Meaning of symbols:

- a = terminal
- A, B = non-terminal
- α, β, γ = string of terminals and/or non-terminals

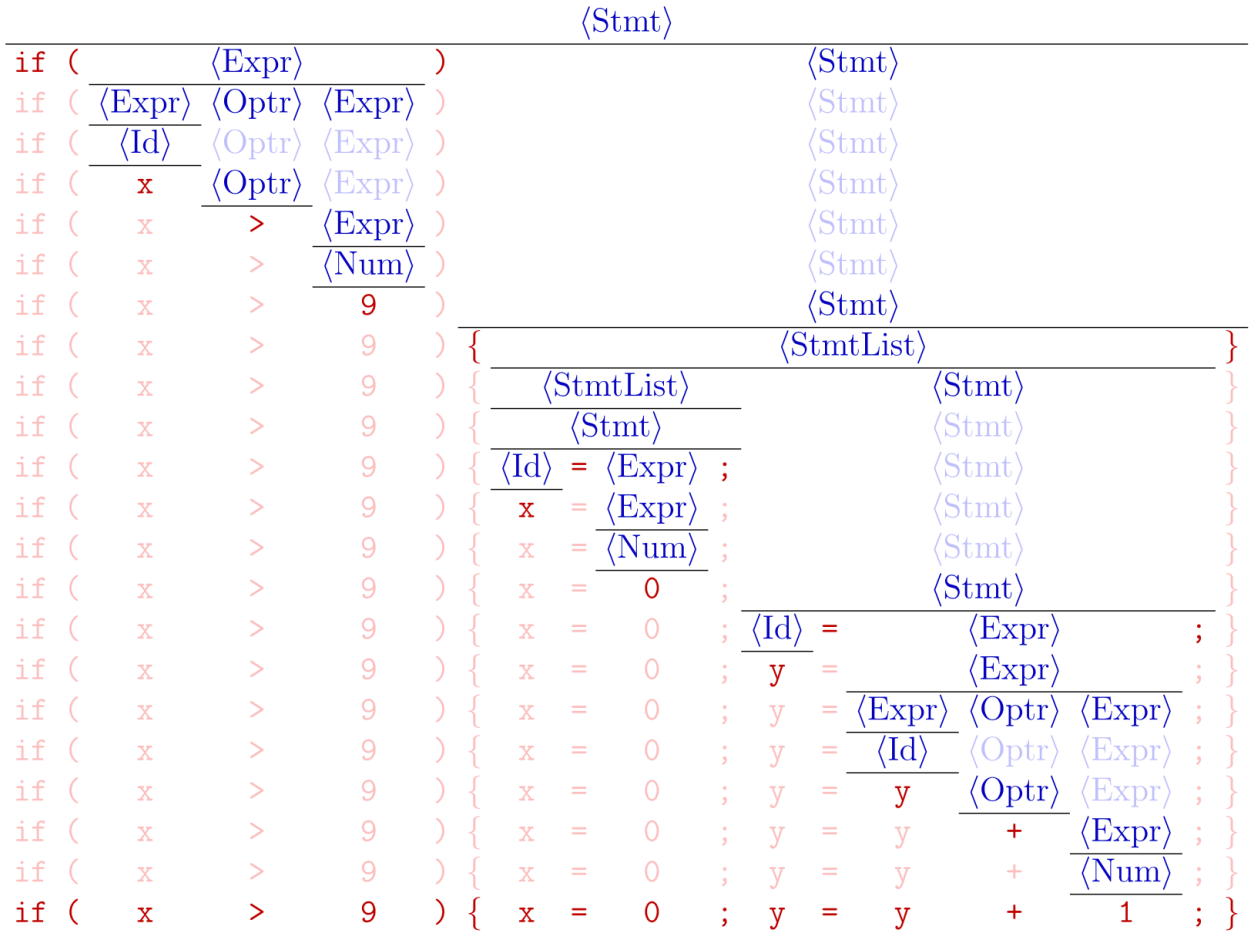
https://en.wikipedia.org/wiki/Chomsky_hierarchy

Regular expressions grammar

TYPE	REGULAR-EXPRESSION	RIGHT-LINEAR-GRAMMAR
SINGLE TERMINAL	e	S --> e
UNION OPERATION	e | f	S --> e | f
CONCATENATION	ef	S --> eA, A --> f
STAR CLOSURE	e*	S --> eS | ^
PLUS CLOSURE	e+	S --> eS | e
STAR CLOSURE ON UNION	(e | f)*	S --> eS | fS | ^
PLUS CLOSURE ON UNION	(e | f)+	S --> eS | fS | e | f
STAR CLOSURE ON CONCATENATION	(ef)*	S --> eA | ^, A --> fS
PLUS CLOSURE ON CONCATENATION	(ef)+	S --> eA, A --> fS | f

Formal languages: part of C language grammar

⟨Stmt⟩ → ⟨Id⟩ = ⟨Expr⟩ ;
⟨Stmt⟩ → { ⟨StmtList⟩ }
⟨Stmt⟩ → if (⟨Expr⟩) ⟨Stmt⟩
⟨StmtList⟩ → ⟨Stmt⟩
⟨StmtList⟩ → ⟨StmtList⟩ ⟨Stmt⟩
⟨Expr⟩ → ⟨Id⟩
⟨Expr⟩ → ⟨Num⟩
⟨Expr⟩ → ⟨Expr⟩ ⟨Optr⟩ ⟨Expr⟩
⟨Id⟩ → x
⟨Id⟩ → y
⟨Num⟩ → 0
⟨Num⟩ → 1
⟨Num⟩ → 9
⟨Optr⟩ → >
⟨Optr⟩ → +



https://upload.wikimedia.org/wikipedia/commons/d/d9/C_grammar_stmt_svg.svg

Natural languages

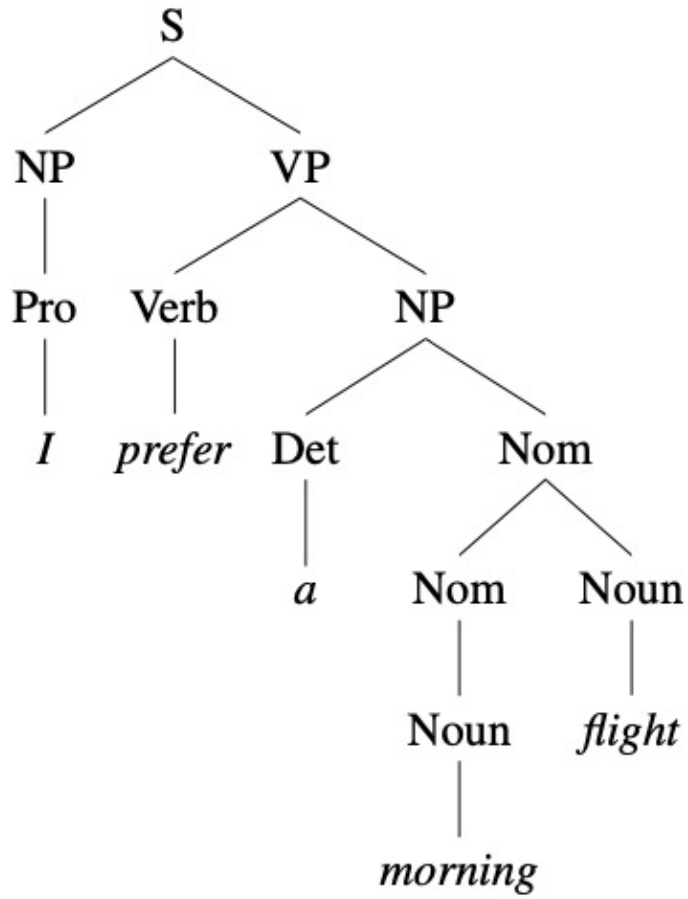
https://web.stanford.edu/~jurafsky/slp3/D.pdf

Noun	→	*flights* | *flight* | *breeze* | *trip* | *morning*
Verb	→	*is* | *prefer* | *like* | *need* | *want* | *fly* | *do*
Adjective	→	*cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun	→	*me* | *I* | *you* | *it*
Proper-Noun	→	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner	→	*the* | *a* | *an* | *this* | *these* | *that*
Preposition	→	*from* | *to* | *on* | *near* | *in*
Conjunction	→	*and* | *or* | *but*

Figure D.2 The lexicon for $\mathscr{L}_0$.

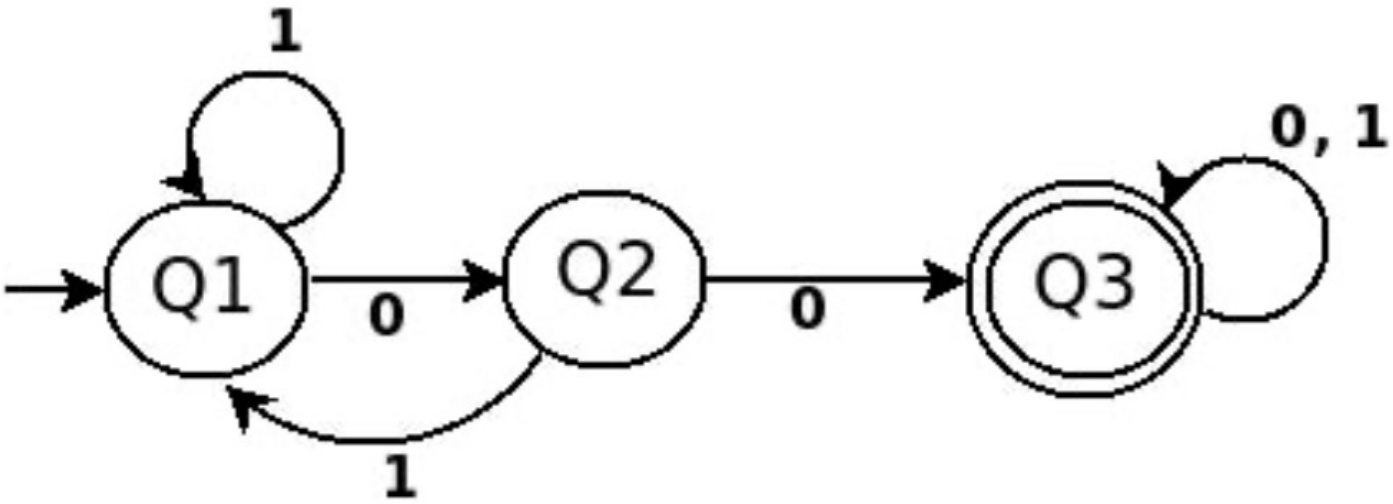
Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

Figure D.3 The grammar for $\mathscr{L}_0$, with example phrases for each rule.



A Sample Regular Expression a.k.a. Regexp

- **Regexp**: (0 | 1)*00(0 | 1)*
- **Grammar**: S --> 0S | 1S | 00A; A --> 0A | 1A | ^
- **Sentences**: 101110**00**11110, 1**00**1, **00**, **00**110



https://stackoverflow.com/questions/13816439/left-linear-and-right-linear-grammars

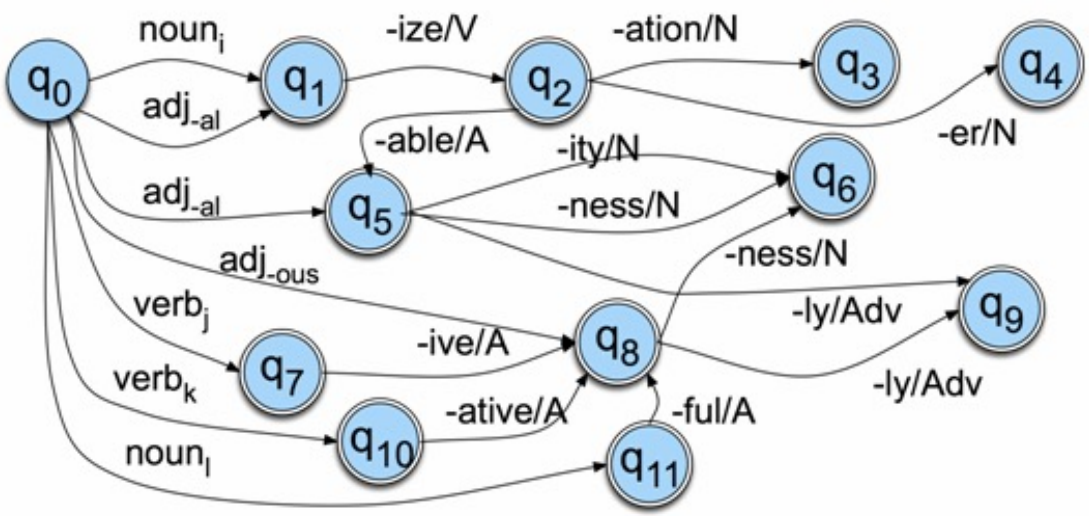
Finite State Automaton

Regular grammars are accepted by finite state automata.

A (deterministic) **finite state automaton** FSA=(Φ,Σ,δ,S,F) consists of:

- set of states Φ
- input alphabet Σ, disjunct with Φ
- transition function $\delta: \Phi \times \Sigma \rightarrow \Phi$
- one start state $S \in \Phi$
- set of final states $F \subset \Phi$

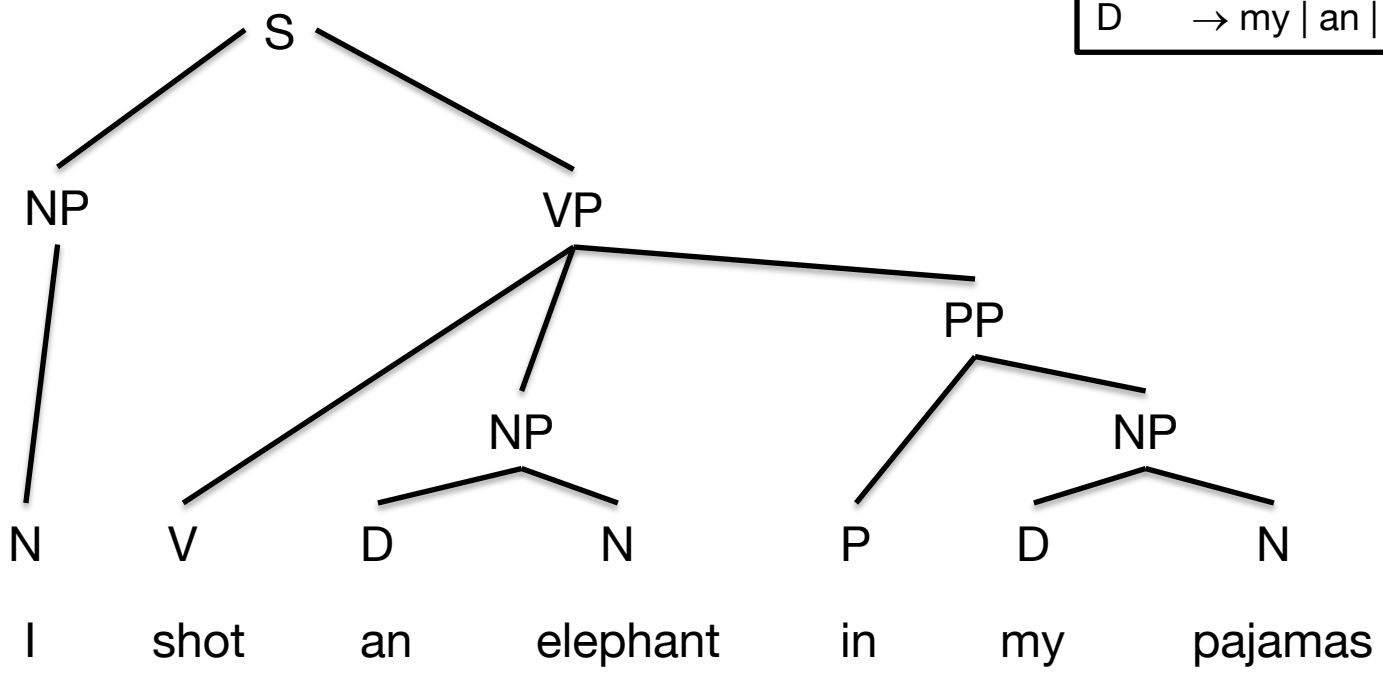
Regular languages cover sub-systems of language, such as morphology and chunk parsing.



Context-free Syntax Trees with Phrase Structure Grammars

- Syntax trees can (almost) be modeled with context-free languages

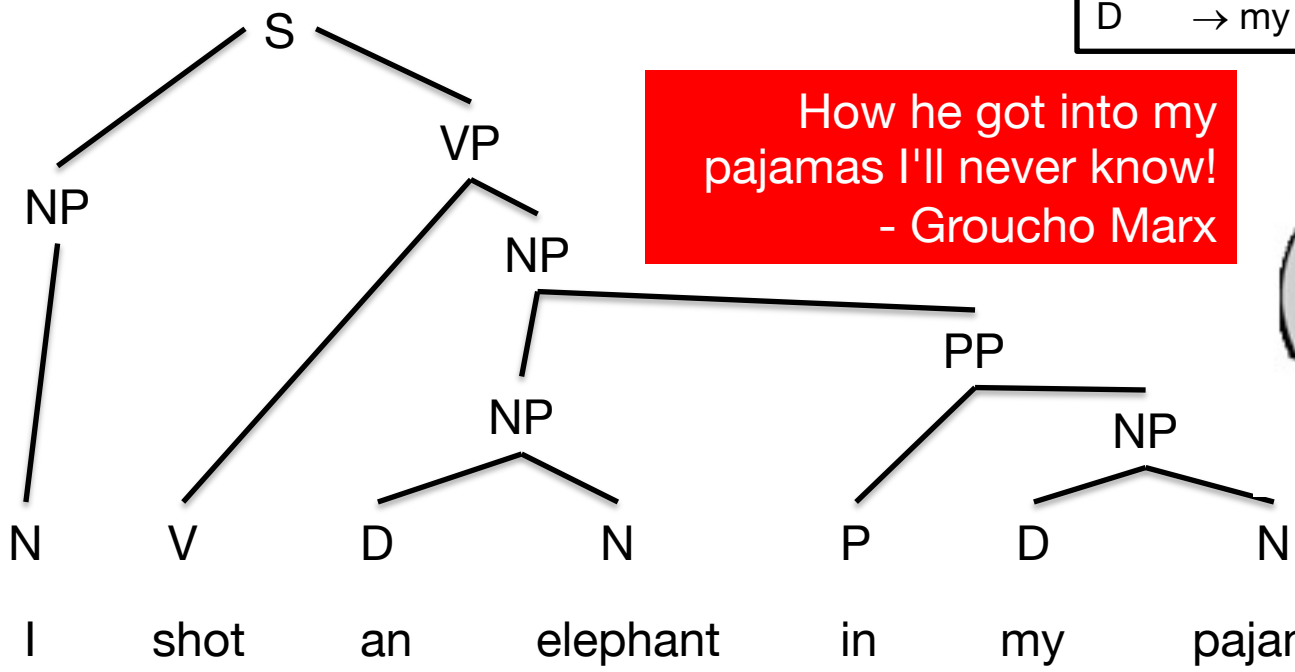
S	→	NP VP
NP	→	N | D N | NP PP
VP	→	V NP | V NP PP
PP	→	P NP
N	→	I | elephant | pajamas
V	→	shot
P	→	in
D	→	my | an | a | the



Context-free Syntax Trees with Phrase Structure Grammars

- Syntax trees can (almost) be modeled with context-free languages
- One surface sentence can have several derivations

S	→	NP VP
NP	→	N | D N | NP PP
VP	→	V NP | V NP PP
PP	→	P NP
N	→	I | elephant | pajamas
V	→	shot
P	→	in
D	→	my | an | a | the



How he got into my pajamas I'll never know!
- Groucho Marx

English Language is not a Regular Language

- Example 1

  Let $L = \{$The cat died, The cat the dog chased died, The cat the dog the rat bit chased died, The cat the dog the rat the elephant admired bit chased died ...$\}$

  The language L can basically be described as: $(\text{noun})^n\ (\text{verb})^n$. This is a nested dependency structure, and it corresponds to $\{0^n1^n \mid n \geq 0\}$. We have already shown above that this language is not regular using the pumping lemma.

- Example 2

  Let $L2 = \{$John and Mary like to eat and sleep, respectively; John, Mary, and Sue like to eat, sleep, and dance, respectively; John, Mary, Sue, and Bob like to eat, sleep, dance, and cook, respectively...$\}$

  This is a cross-serial dependency structure, also described by $(\text{noun})^n\ (\text{verb})^n$, which again corresponds to $\{0^n1^n \mid n \geq 0\}$. Of course, we have already seen that this language is not regular using the pumping lemma.

www.ling.upenn.edu/courses/Spring_2002/ling106/note5.pdf

Programming Language vs. Natural Language

Grammar of Programming languages

- by design: deterministic context free (in most cases)
- which allows efficient parsing
- without ambiguities.
- clearly defined semantics

Grammar of Natural Languages

- somewhere between type-1 and type-2
- many possible parses for a single sentence
- inherent ambiguities
- semantics yet another layer

Shortcomings of the Rule-based Aproaches

- The need to write manually rules.
- Complexity of maintainance and making the rules of derivation consistent.
- Hard to perform adaptation to a new domain and language
- Does not take into account **statistical information**.

- Yet they have some **strong points**:
  - Rules are computationally effective
  - High precision can be obtained
  - If the system of rules is small they may be handy

PLAN OF THE LECTURE

- Foundations of Rule-based NLP
- Morphology with FSTs and Tries
- Morphological Analysis: Task Formulations

UniMorph: a universal cross-lingual morphology annotation scheme

https://unimorph.github.io

Annotated Languages

The following 141 languages have been annotated according to the UniMorph schema. Missing parts of speech will be filled in soon.

	Language	ISO 639-3	Forms	Paradigms	Nouns	Verbs	Adjectives	Source	License
	Adyghe	ady	20475	1666	✔		✔	W	
	Akan	aka	4182	96		✔		L	
	Albanian	sqi	33483	589	✔	✔		W	
	Ancient Greek	grc	41593	2431	✔		✔	W	
	Arabic	ara	140003	4134	✔	✔	✔		
	Armenian	hye	338461	7033	✔	✔	✔	W	
	Ashaninka	cni	10952	407	✔	✔		L	
	Asturian	ast	29797	436	✔	✔	✔	W	
	Aymara	aym	336341	3410	✔	✔		L	
	Azerbaijani	aze	8004	340	✔	✔		W	
	Bashkir	bak	12168	1084	✔		✔	W	
	Basque	eus	11889	26		✔		W	–
	Belarusian	bel	16113	1027	✔	✔	✔	W	
	Bengali	ben	4443	136	✔	✔		W	

UniMorph: a universal cross-lingual morphology annotation scheme

https://raw.githubusercontent.com/unimorph/eng/master/eng

```
3rd     3rded   V;PST
3rd     3rded   V;V.PTCP;PST
3rd     3rding  V;V.PTCP;PRS
3rd     3rds    V;3;SG;PRS
3rd     3rd     V;NFIN
86      86ed    V;PST
86      86ed    V;V.PTCP;PST
86      86es    V;3;SG;PRS
86      86ing   V;V.PTCP;PRS
86      86      V;NFIN
911     911ed   V;PST
911     911ed   V;V.PTCP;PST
911     911ing  V;V.PTCP;PRS
911     911s    V;3;SG;PRS
911     911     V;NFIN
aah     aahed   V;PST
aah     aahed   V;V.PTCP;PST
aah     aahing  V;V.PTCP;PRS
aah     aahs    V;3;SG;PRS
aah     aah     V;NFIN
a       a       V;NFIN
abacinate       abacinated      V;PST
abacinate       abacinated      V;V.PTCP;PST
abacinate       abacinates      V;3;SG;PRS
abacinate       abacinate       V;NFIN
abacinate       abacinating     V;V.PTCP;PRS
abalienate      abalienated     V;PST
abalienate      abalienated     V;V.PTCP;PST
abalienate      abalienates     V;3;SG;PRS
abalienate      abalienate      V;NFIN
abalienate      abalienating    V;V.PTCP;PRS
aband   abanded V;PST
aband   abanded V;V.PTCP;PST
aband   abanding        V;V.PTCP;PRS
aband   abands  V;3;SG;PRS
aband   aband   V;NFIN
abandon abandoned       V;PST
abandon abandoned       V;V.PTCP;PST
abandon abandoning      V;V.PTCP;PRS
abandon abandons        V;3;SG;PRS
abandon abandon V;NFIN
abase   abased  V;PST
abase   abased  V;V.PTCP;PST
abase   abases  V;3;SG;PRS
abase   abase   V;NFIN
abase   abasing V;V.PTCP;PRS
abash   abashed V;PST
abash   abashed V;V.PTCP;PST
abash   abashes V;3;SG;PRS
abash   abashing        V;V.PTCP;PRS
abash   abash   V;NFIN
```

```
modernise       modernised      V;PST
modernise       modernised      V;V.PTCP;PST
modernise       modernises      V;3;SG;PRS
modernise       modernise       V;NFIN
modernise       modernising     V;V.PTCP;PRS
modernize       modernized      V;PST
modernize       modernized      V;V.PTCP;PST
modernize       modernizes      V;3;SG;PRS
modernize       modernize       V;NFIN
modernize       modernizing     V;V.PTCP;PRS
modificate      modificated     V;PST
modificate      modificated     V;V.PTCP;PST
modificate      modificates     V;3;SG;PRS
modificate      modificate      V;NFIN
modificate      modificating    V;V.PTCP;PRS
modify  modified        V;PST
modify  modified        V;V.PTCP;PST
modify  modifies        V;3;SG;PRS
modify  modifying       V;V.PTCP;PRS
modify  modify  V;NFIN
mod     modded  V;PST
mod     modded  V;V.PTCP;PST
mod     modding V;V.PTCP;PRS
mod     mods    V;3;SG;PRS
mod     mod     V;NFIN
```

UniMorph: a universal cross-lingual morphology annotation scheme

https://raw.githubusercontent.com/unimorph/rus/master/rus

аббатский	аббатская	ADJ;NOM;FEM;SG
аббатский	аббатские	ADJ;ACC;INAN;PL
аббатский	аббатские	ADJ;NOM;PL
аббатский	аббатский	ADJ;INAN;ACC;MASC;SG
аббатский	аббатский	ADJ;NOM;MASC;SG
аббатский	аббатским	ADJ;DAT;PL
аббатский	аббатским	ADJ;INS;MASC;SG
аббатский	аббатским	ADJ;INS;NEUT;SG
аббатский	аббатскими	ADJ;INS;PL
аббатский	аббатских	ADJ;ACC;ANIM;PL
аббатский	аббатских	ADJ;ESS;PL
аббатский	аббатских	ADJ;GEN;PL
аббатский	аббатского	ADJ;ANIM;ACC;MASC;SG
аббатский	аббатского	ADJ;GEN;MASC;SG
аббатский	аббатского	ADJ;GEN;NEUT;SG
аббатский	аббатское	ADJ;ACC;NEUT;SG
аббатский	аббатское	ADJ;NOM;NEUT;SG
аббатский	аббатской	ADJ;DAT;FEM;SG
аббатский	аббатской	ADJ;ESS;FEM;SG
аббатский	аббатской	ADJ;GEN;FEM;SG
аббатский	аббатской	ADJ;INS;FEM;SG
аббатский	аббатском	ADJ;ESS;MASC;SG
аббатский	аббатском	ADJ;ESS;NEUT;SG
аббатский	аббатскому	ADJ;DAT;MASC;SG
аббатский	аббатскому	ADJ;DAT;NEUT;SG
аббатский	аббатскую	ADJ;ACC;FEM;SG
аббатство	аббатств	N;GEN;PL
аббатство	аббатства	N;ACC;PL
аббатство	аббатства	N;GEN;SG
аббатство	аббатства	N;NOM;PL
аббатство	аббатствам	N;DAT;PL
аббатство	аббатствами	N;INS;PL
аббатство	аббатствах	N;ESS;PL
аббатство	аббатстве	N;ESS;SG
аббатство	аббатство	N;ACC;SG
аббатство	аббатство	N;NOM;SG
аббатство	аббатством	N;INS;SG
аббатство	аббатству	N;DAT;SG

рэп	рэп	N;ACC;SG
рэп	рэп	N;NOM;SG
рэп	рэпа	N;GEN;SG
рэп	рэпам	N;DAT;PL
рэп	рэпами	N;INS;PL
рэп	рэпах	N;ESS;PL
рэп	рэпе	N;ESS;SG
рэп	рэпов	N;GEN;PL
рэп	рэпом	N;INS;SG
рэп	рэпу	N;DAT;SG
рэп	рэпы	N;ACC;PL
рэп	рэпы	N;NOM;PL
рюкзак	рюкзак	N;ACC;SG
рюкзак	рюкзак	N;NOM;SG
рюкзак	рюкзака	N;GEN;SG
рюкзак	рюкзакам	N;DAT;PL
рюкзак	рюкзаками	N;INS;PL
рюкзак	рюкзаках	N;ESS;PL
рюкзак	рюкзаке	N;ESS;SG
рюкзак	рюкзаки	N;NOM;PL
рюкзак	рюкзаков	N;GEN;PL
рюкзак	рюкзаком	N;INS;SG
рюкзак	рюкзаку	N;DAT;SG

Morphology with FSAs

- Morphology works fairly regular, so FSAs are an appropriate machinery for morphological analysis
- Tasks for automated morphology:
  - analyze a given word into its morphemes
  - generate a full form from a base form + morphological information

Surface	Lexical
runs	run+Verb+Present+3sg run+Noun+Pl
largest	large+Adj+Sup
better	good+Adj+Comp

Surface	Lexical
Boote	boot+Nomen+Plural
verlangsamte	verlangsam+Verb +Imperf+3sg verlangsamt+Adj +NomAkk

- Plain word lists are a possibility, but **redundancies are not utilized** and access can be slow. Further, **no generalization properties**: cannot utilize regularities from inflection classes, cannot guess for unseen words

Morphological Regularity

https://aclanthology.org/W16-2002.pdf

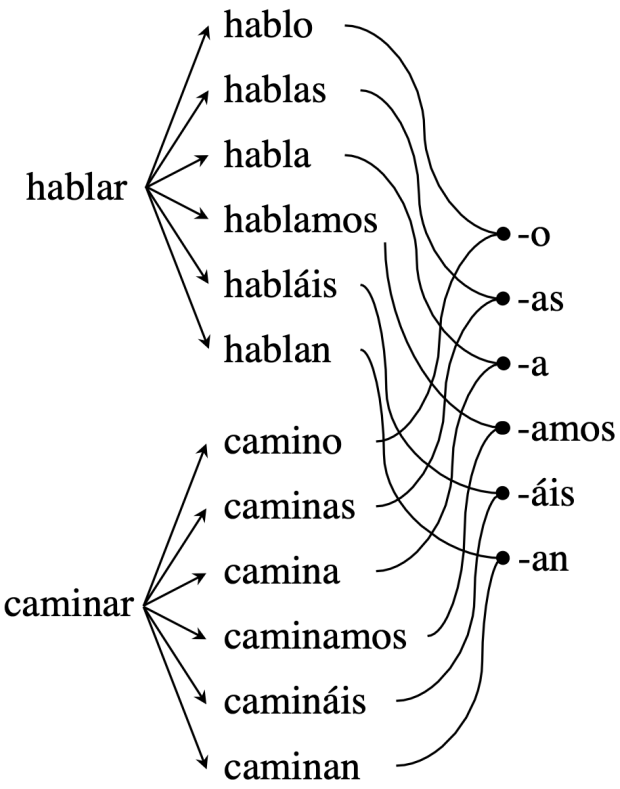


Figure 1: The relatedness of inflected forms, such as the present indicative paradigm of the Spanish verbs *hablar* 'speak' and *caminar* 'walk,' allows generalizations about the shape and affixal content of the paradigm to be extracted.

Finite State Transducer

A **finite state transducer** is a 6-tuple **FST**=($\Phi,\Sigma,\Gamma,\delta,S,F$) and consists of

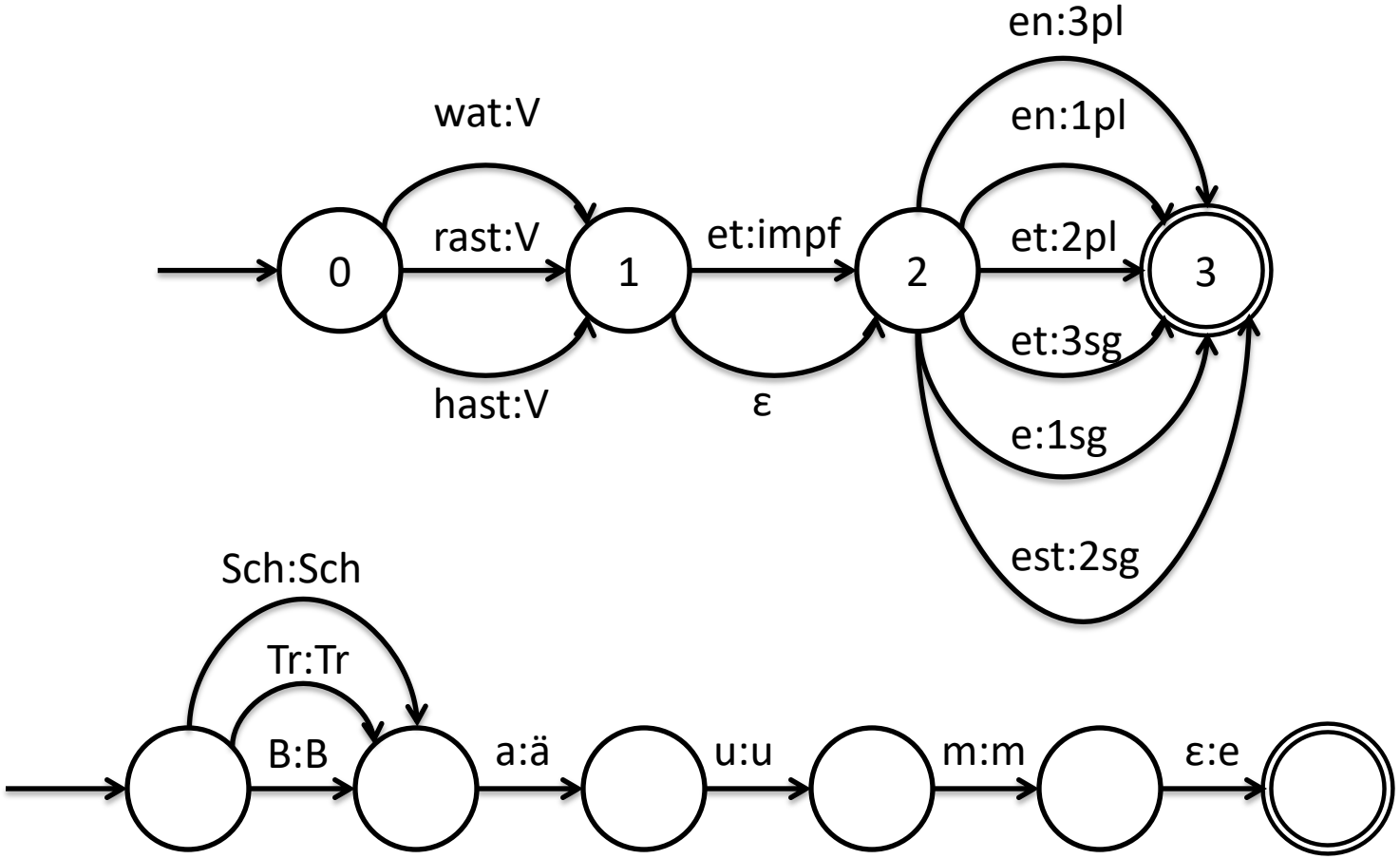
- set of states Φ
- input alphabet Σ, disjunct with Φ
- output alphabet Γ, disjunct with Φ
- set of start states $S \subset \Phi$
- set of final states $F \subset \Phi$
- transition function $\delta \subseteq \Phi \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \times \Phi$

An **FST** is essentially an **FSA with two tapes**. It is useful to think about them as input tape and output tape, or upper tape and lower tape.

An FST transduces an input string x to an output string y if there is a sequence of transitions that starts with a start state and ends with a final state and has x as its input and y as its output string.

FSTs accept **regular relations**.

Examples for Morphology FSTs

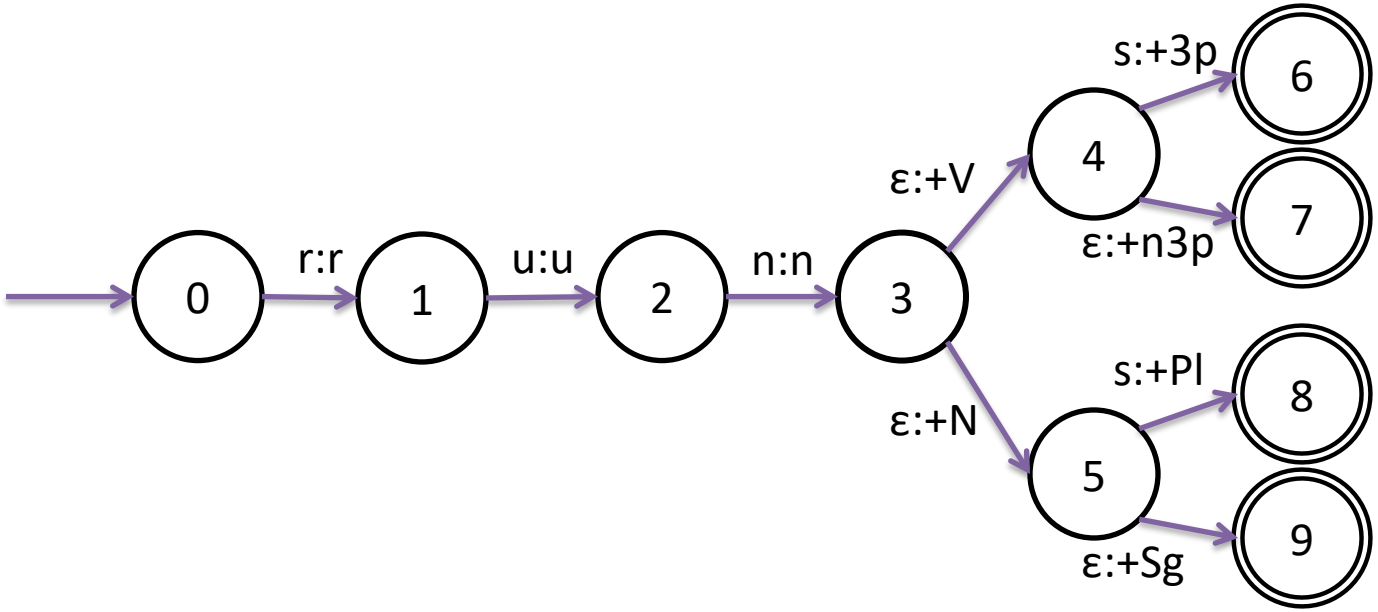


Note that FSTs can be non-deterministic and can have ε-transitions.

Handling nondeterminism and ambiguities

- Since language is ambiguous on many levels, we embrace nondeterminism as a mechanism to reflect that
- As long as we do not know how to resolve ambiguities, we carry along several possibilities
- Nondeterminism for FSA: we don’t know which path we took
- Nondeterminism for FST: different paths produce different output strings
- Nondeterminism requires to keep track of a **set** of current states
- A nondeterministic automaton accepts if there is at least one path to a final state

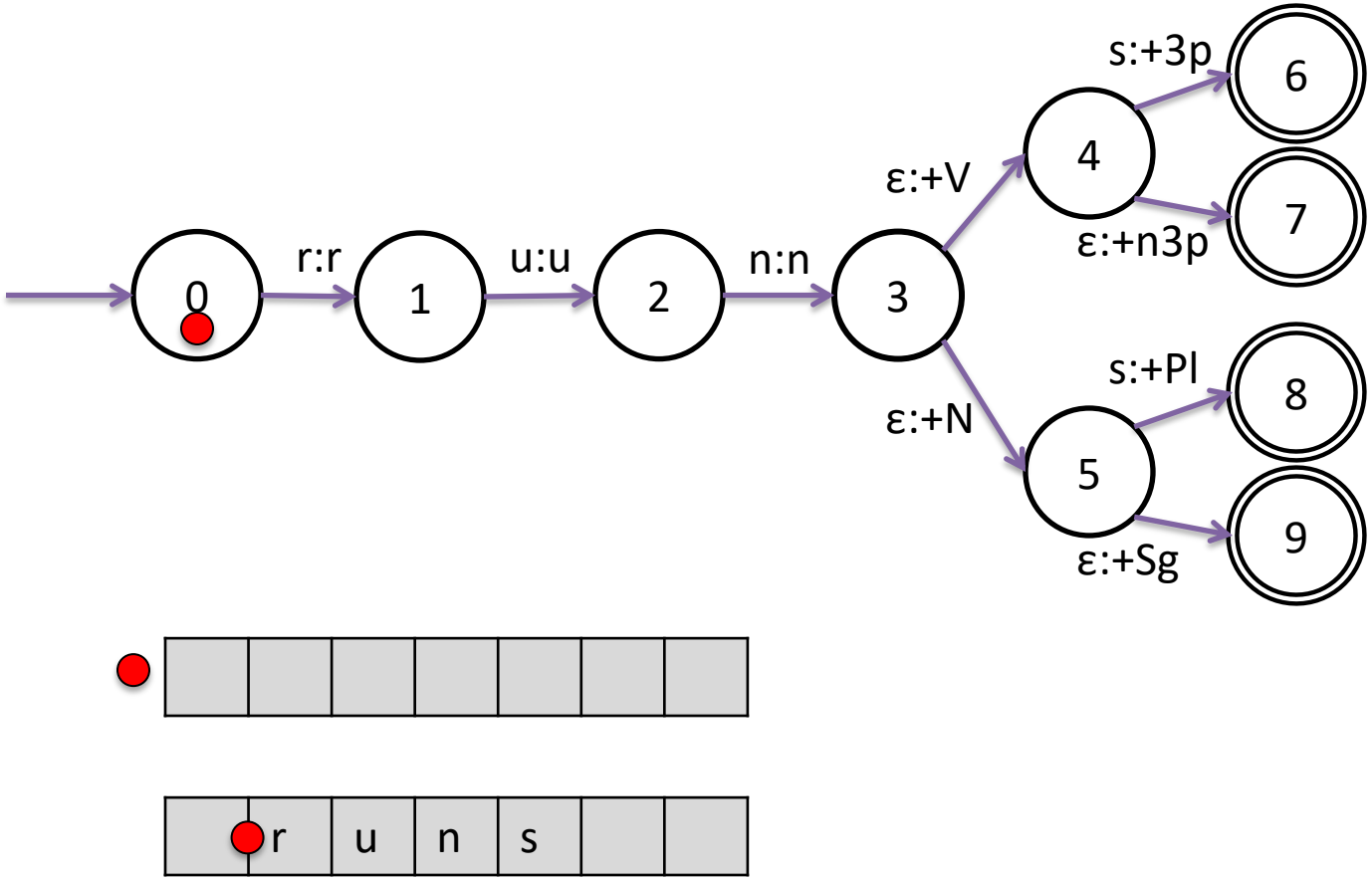
Running Example



input string

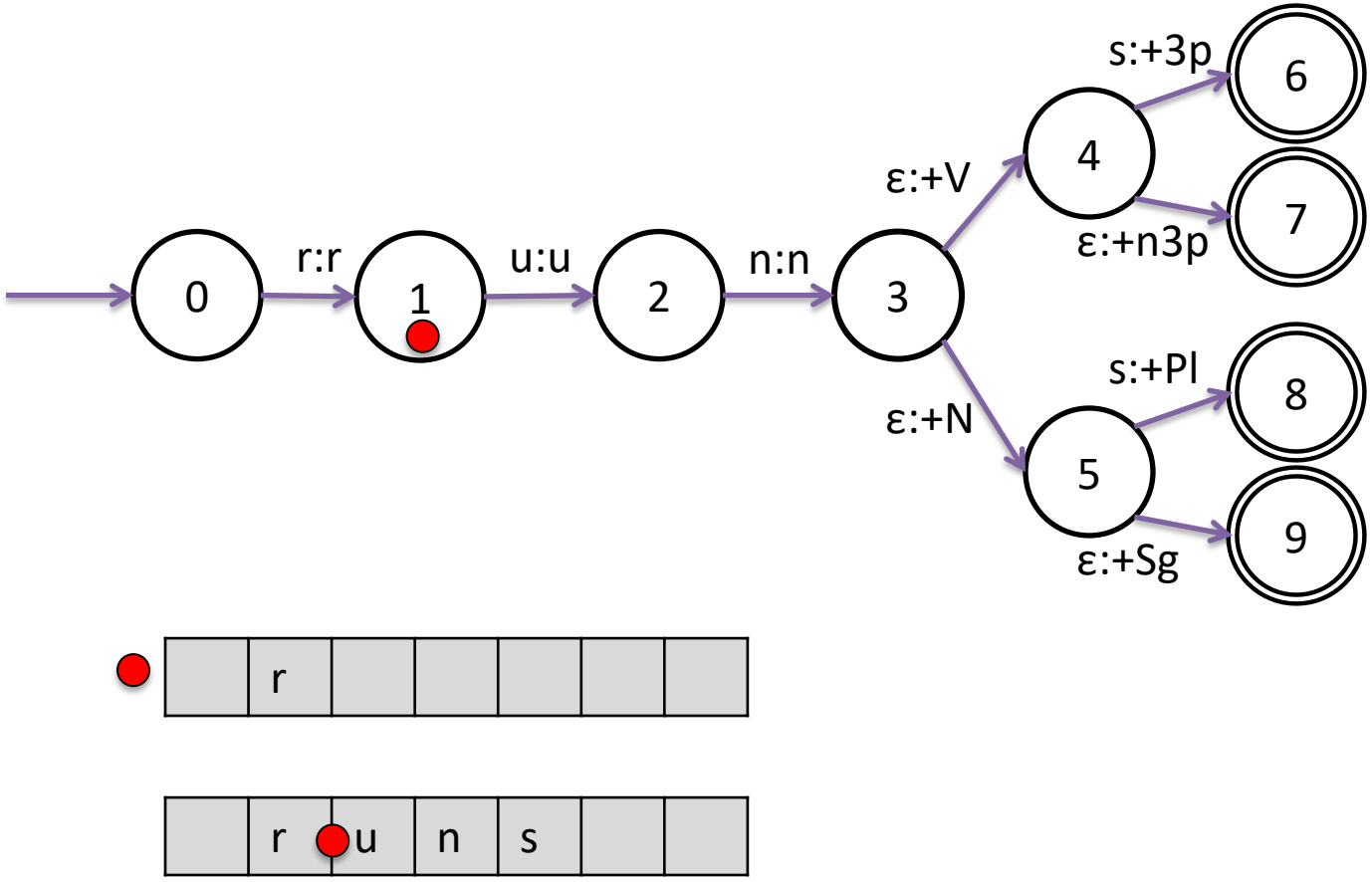
	r	u	n	s		

Running Example



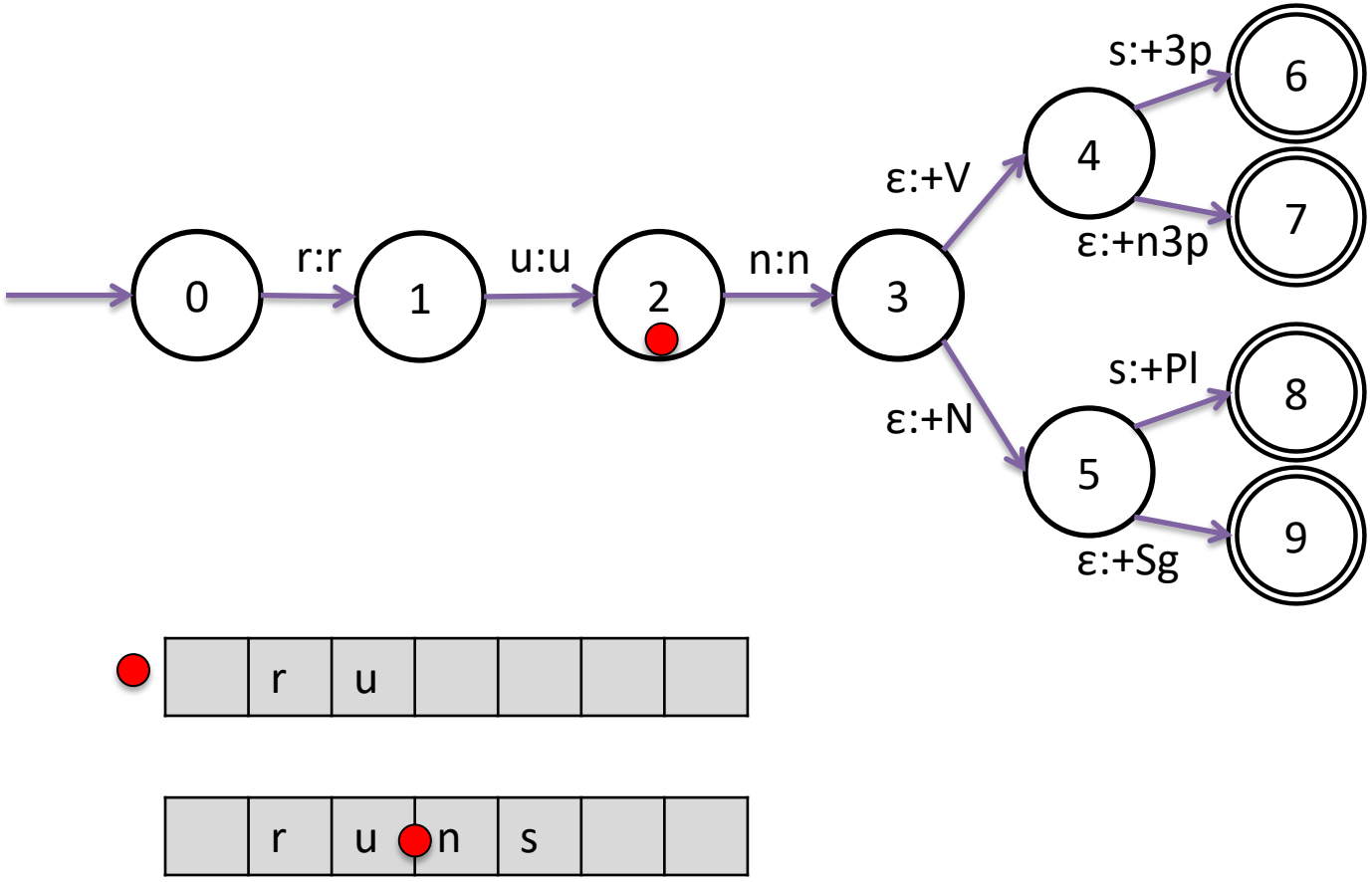
Dots: Keep track current state and output generated so far.

Running Example



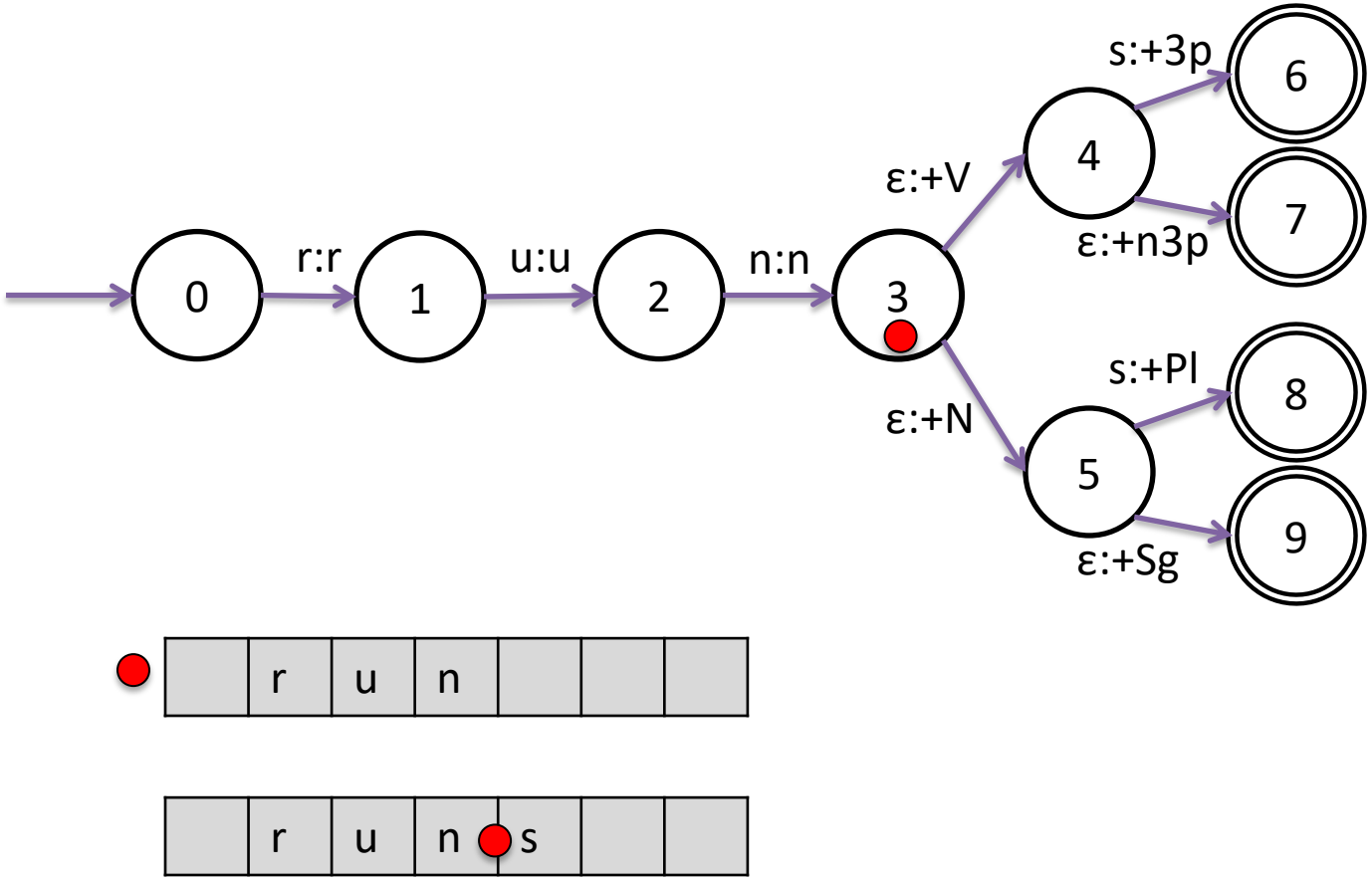
Transition: dot moves on input tape and to next state, generating output

Running Example



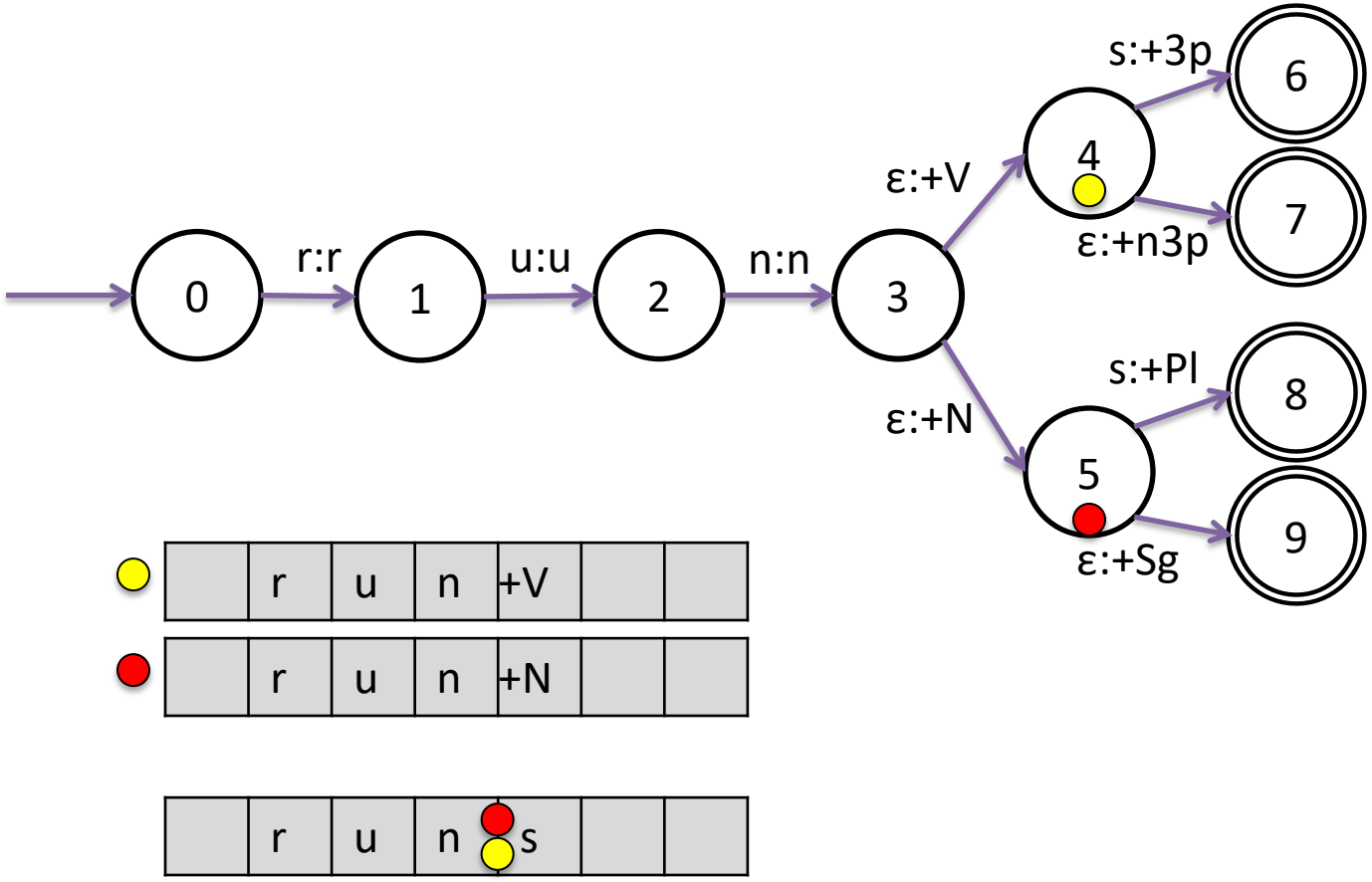
Transition: dot moves on input tape and to next state, generating output

Running Example



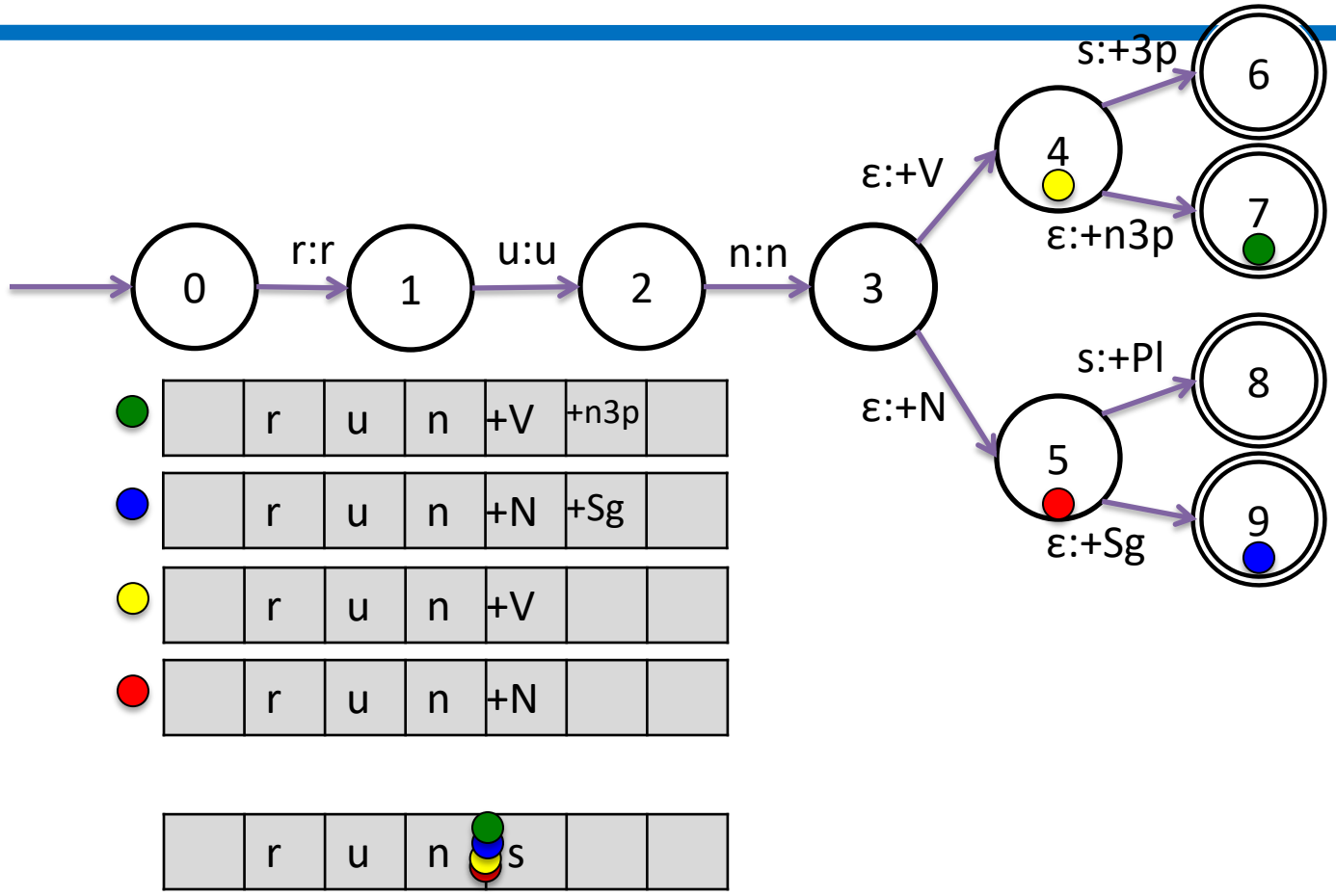
Transition: dot moves on input tape and to next state, generating output

Running Example



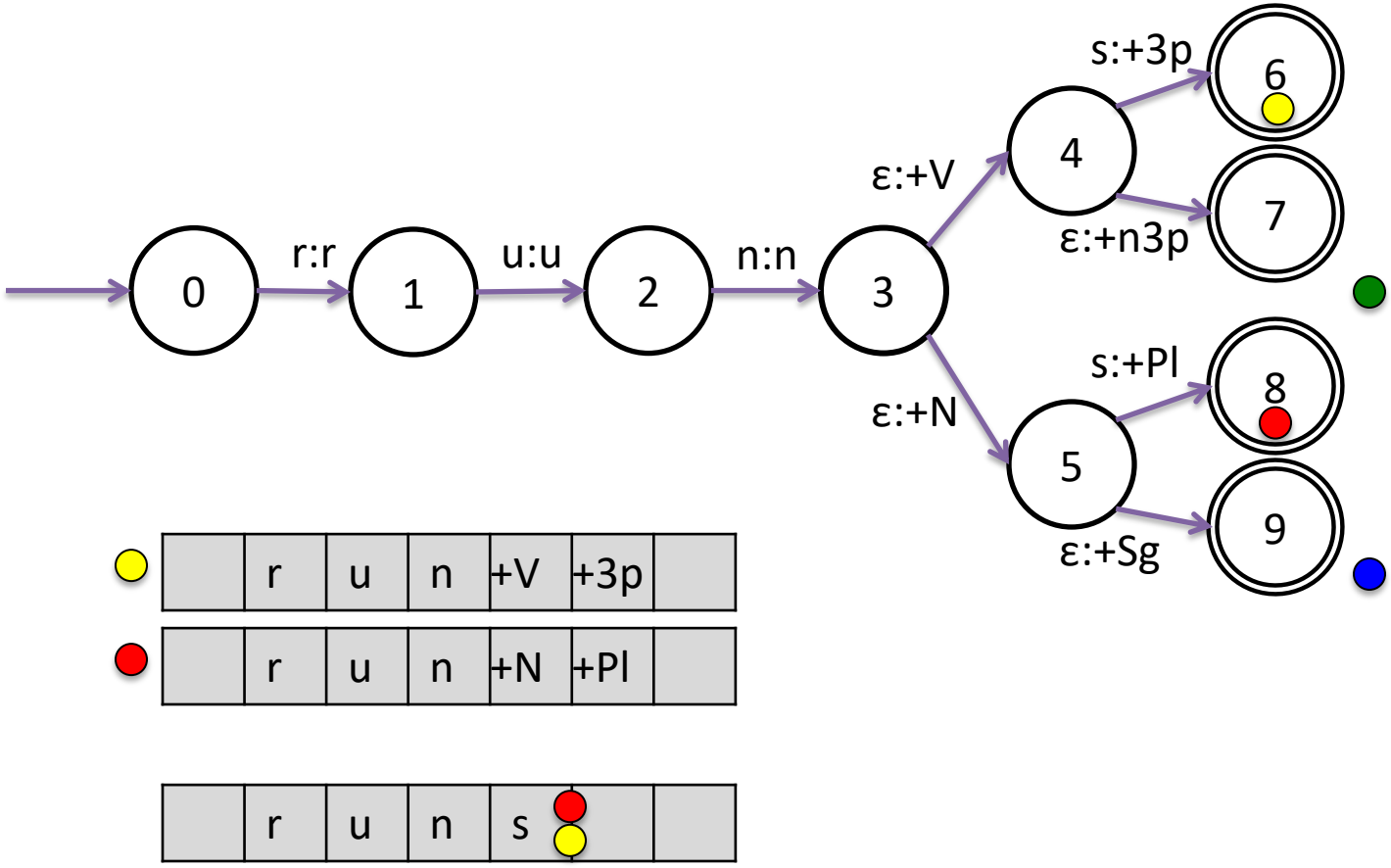
Non-determinism: Dot splits. Output tape is copied.

Running Example



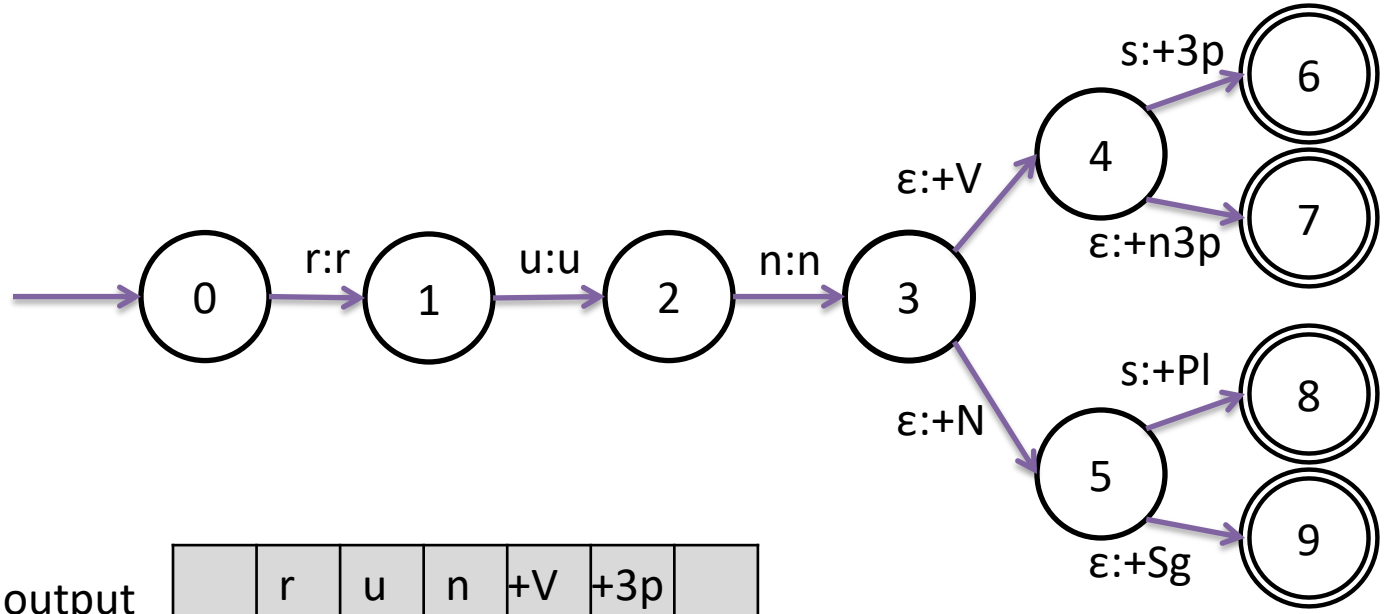
ε-transitions are also non-determinisms.

Running Example



Dots that do not have a follow-up state are abandoned.

Running Example



output strings

	r	u	n	+V	+3p	
	r	u	n	+N	+Pl	

	r	u	n	s		

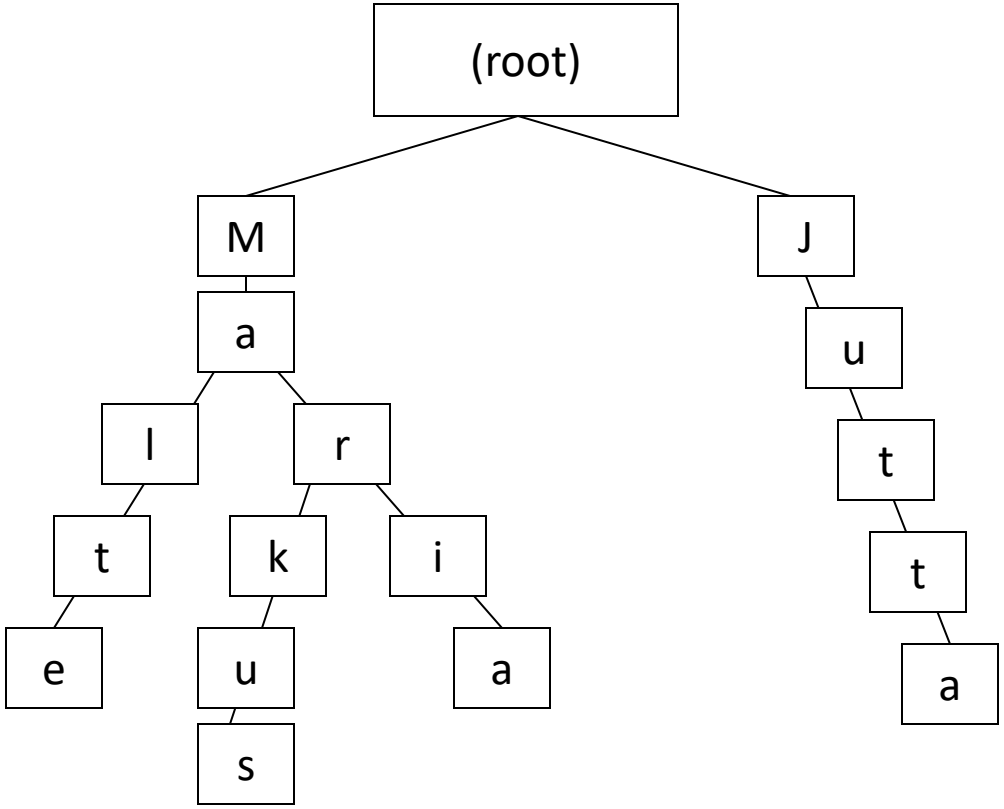
End of input is reached. All dots at final states have successfully transduced.

PLAN OF THE LECTURE

- Formal Languages and Regular Expressions
- Morphology with FSTs and Tries
- Shared Tasks on Morphological Analysis

Tries (a.k.a. Prefix Tree): Combine Common Prefixes

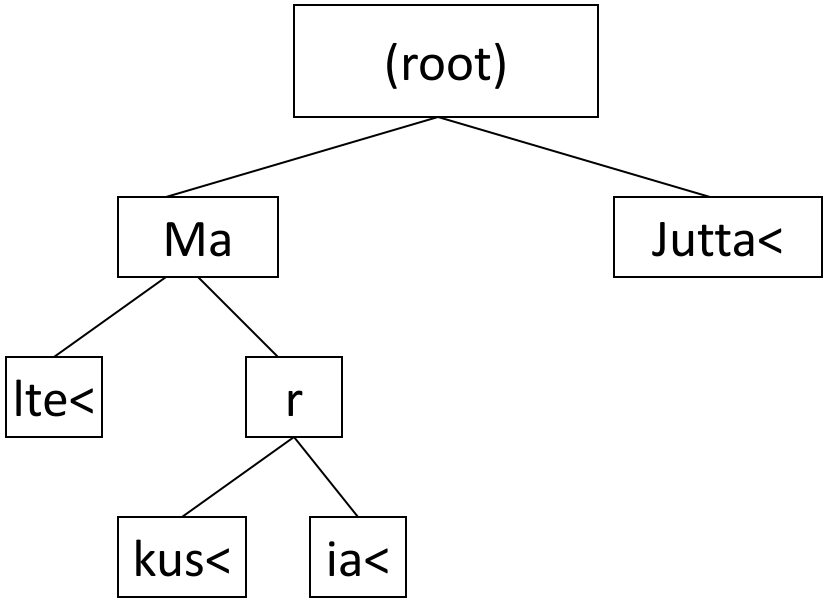
- A trie is a tree structure. The nodes have 0 to N daughters (N number of possible characters in alphabet).
- Example for `Markus, Maria, Jutta, Malte`



17 nodes with
16 characters,
16 edges

Patricia Trie (PT) (a.k.a. Radix Tree)

- Decrease number of edges by putting several characters in one node
- Example for `Markus, Maria, Jutta, Malte`

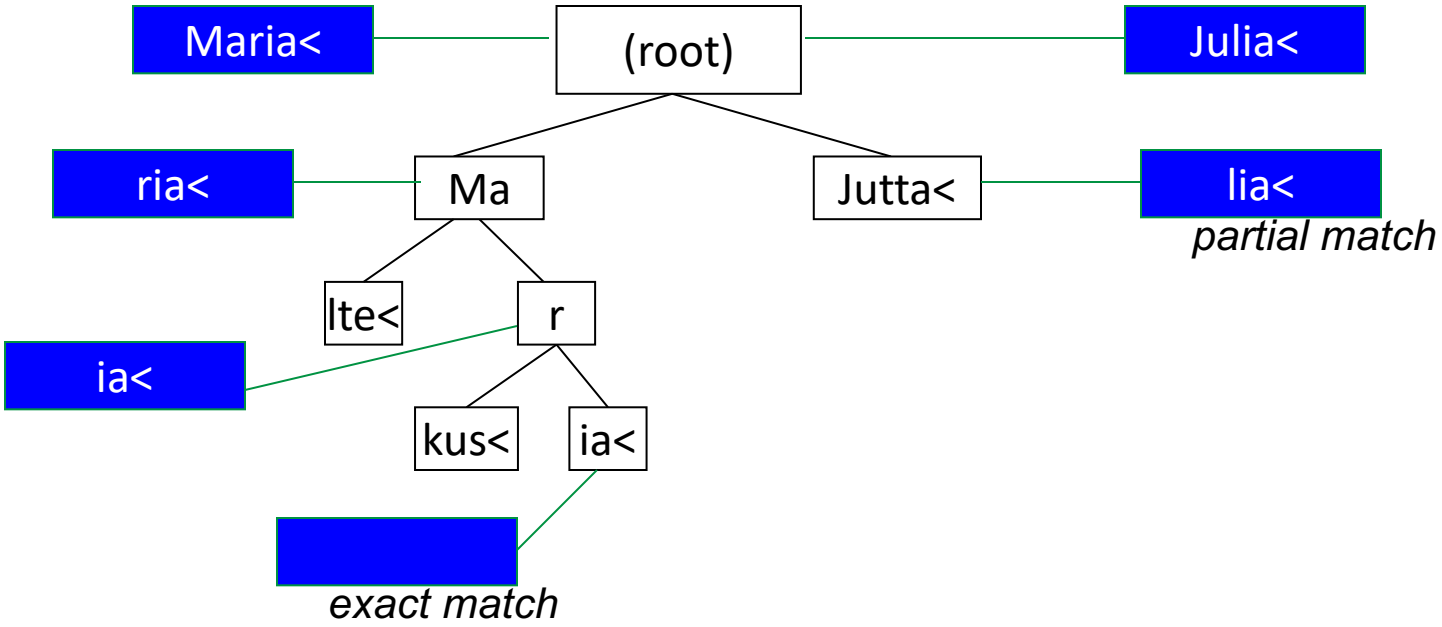


7 nodes,
16 characters,
6 edges.

"<" designates end-of-word

Search in PTs

- Recursively walk down, search word gets eaten up
- Return last reached node.
- If remaining search word is empty: *exact match*, otherwise *partial match*

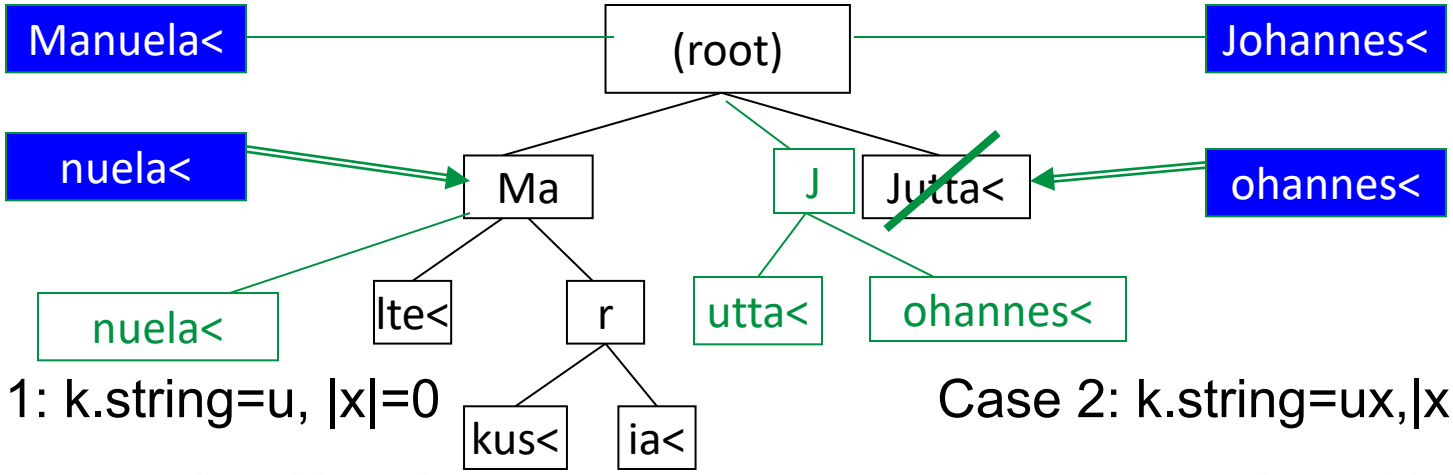


Insert in PTs

Insert of w:

- Search for w returns appropriate node k
- if *exact match*: Word was in PT already
- if *partial match*: Split string contained in k, attach daughter nodes.

In k holds: k: w=uv, k.string=ux



Case 1: k.string=u, |x|=0

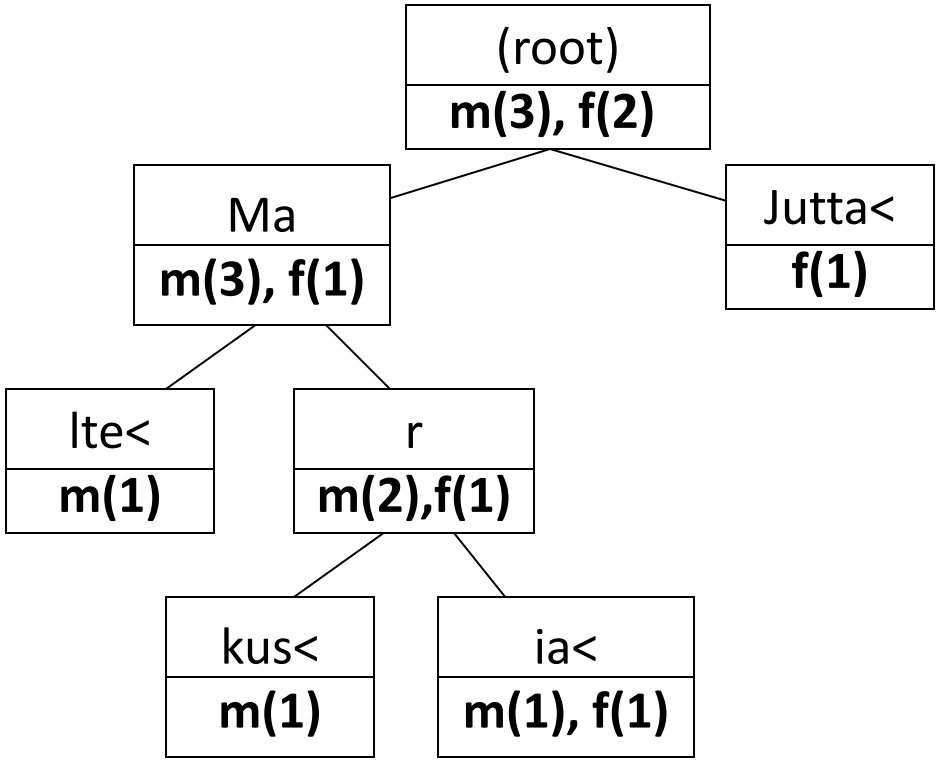
Insert one node with string v as daughter of k

Case 2: k.string=ux,|x|>0

Insert two nodes with strings v and x as daughters of k

Storing Additional Information in Patricia Tries

- Nodes are extended: An additional field stores some information
- Example: Storing the gender of names:

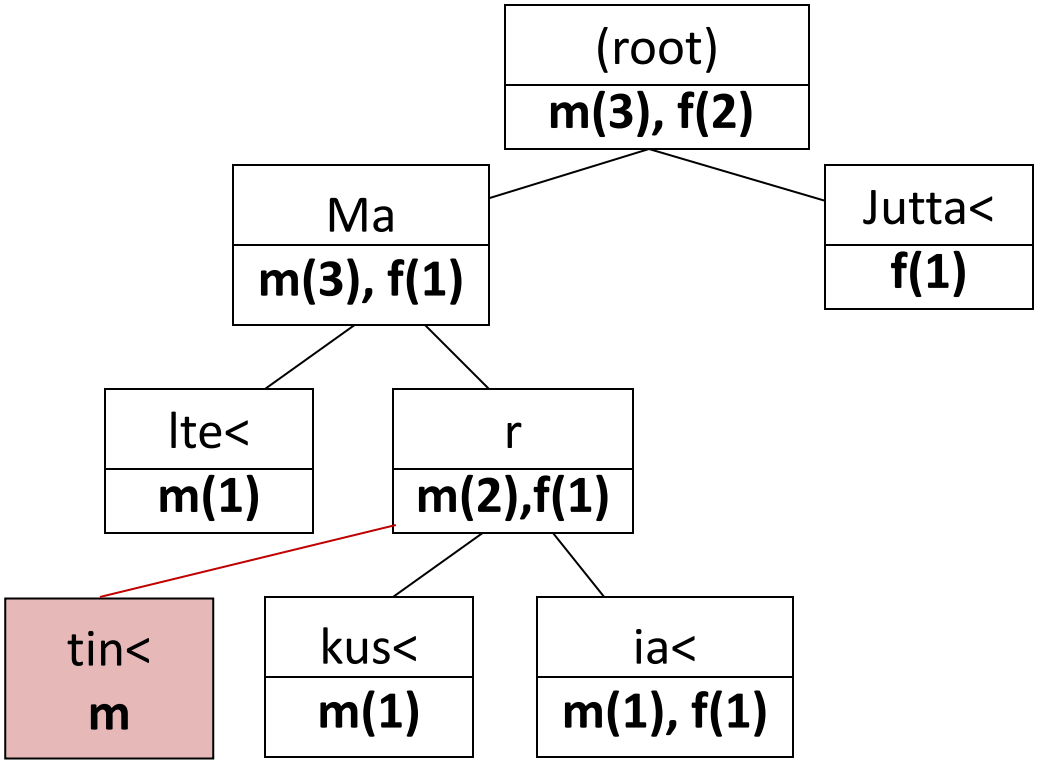


The classes can be found in the leaves.

In intermediate nodes, the additional field stores the sum of the classes in the subtree.

Storing Additional Information in Patricia Tries

- Nodes are extended: An additional field stores some information
- Example: Storing the gender of names:



The classes can be found in the leaves.

In intermediate nodes, the additional field stores the sum of the classes in the subtree.

Application: Base form reduction (Lemmatization)

- Given: List of words with reduction rules

```
Haus            0
Hauses          2
Häuser          5aus
Maus            0
Mäuse           4aus
Bau             0
Baus            1
Aus             0
```

- Reduction: integer *n* and (possible empty) string *x*.
- read: cut *n* characters (bytes) from behind and attach *x*.
- ambiguous cases: multiple instructions:

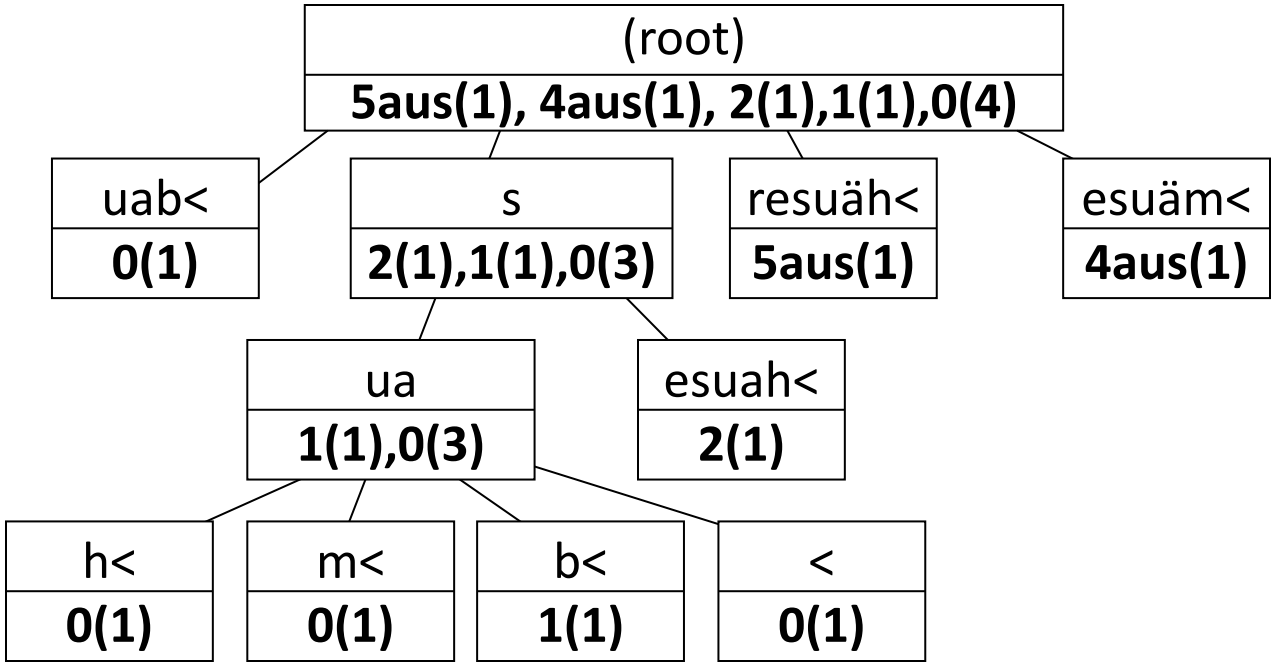
```
Fang                0; 0en
```

- inflection removal for verbs (German-specific): remove first string occurrence after operator #

```
geschienen          5einen#ge
```

Base form reduction II

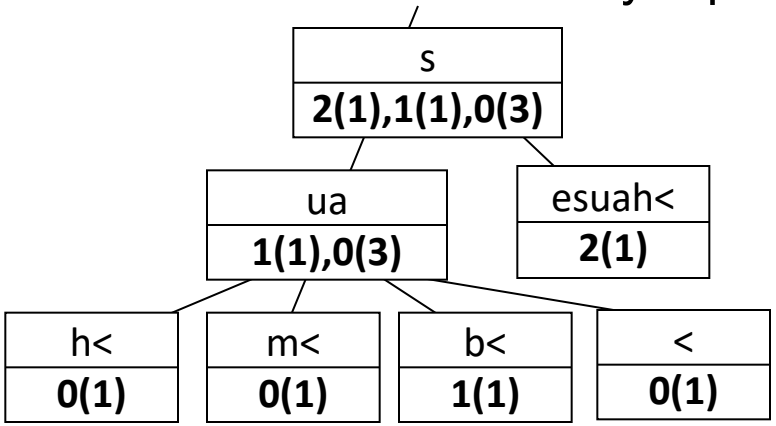
PT is built from reversed words, the reduction rules are stored in the nodes. "<" denotes start-of-word.




```
Haus    0
Hauses  2
Häuser  5aus
Maus    0
Mäuse   4aus
Bau     0
Baus    1
Aus     0
```

Base form reduction III

- For base form reduction, a search with the reversed word is performed, this returns some node (leaf or intermediate node).
- The rule in this node will be applied. If there are several rules, take the one with the highest score and above some threshold
- Under the threshold, return '*undecided*'
- Unknown words receive a morphologically motivated guess
- all known words are fully represented: 100% correct on training



Hochhaus → 0

Spass → 0

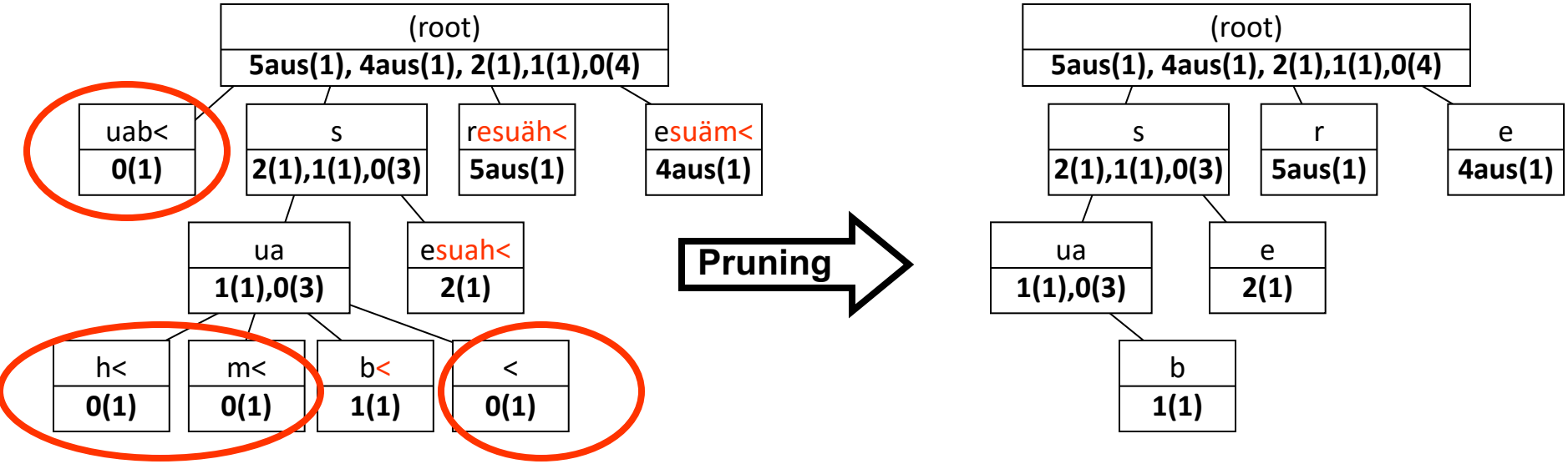
Unterbaus → 1

Pruning to CPT: Memory reduction

If the PT serves merely as classifier and not for storing word lists:

- class-redundant subtrees can be deleted.
- strings in the remaining leaves can be cut to length 1

A pruned PT is called **Compact Patricia Trie (CPT)**



Summary of CPTs

Properties:

- fully reproducing a training set, yet perform educated guesses
- compact data structure for string-based classification
- Trained from training data: word+class
- Insertion and deletion possible (before pruning) without reorganization

Applications:

- morphological classification
- base form reduction
- compound noun decomposition: `Bauhaus` -> `Bau - haus`
- context-independent word class assignment, e.g. Noun, Verb, etc.
- autocomplete / type-ahead

PLAN OF THE LECTURE

- Foundations of Rule-based NLP
- Morphology with FSTs and Tries
- Morphological Analysis: Task Formulations

Morphological Analysis Shared Task (MorphoRuEval-2017)

http://www.dialog-21.ru/en/evaluation/2017/morphology/

- Two tasks. For each word in a sentence:
  - determine the part of speech and a number of grammatical categories (case, gender, number, etc.)
  - perform lemmatization

Table 1. Annotated categories for different parts of speech

Case	nominative—Nom, genitive—Gen, dative—Dat, accusative—Acc, locative—Loc, instrumental—Ins
Gender	masculine—Masc, feminine—Fem, neuter—Neut
Number	singular—Sing, plural—Plur
Animacy	animate—Anim, inanimate—Inan
Tense	past—Past, present or future—Notpast
Person	first—1, second—2, third—3
VerbForm	infinitive—Inf, finite—Fin, gerund—Conv
Mood	indicative—Ind, imperative—Imp
Variant	short form—Brev (no mark for complete form)
Degree	positive or superlative—Pos, comparable—Cmp
NumForm	numeric token—Digit (if the token is written in alphabetic form, no mark is placed).

Morphological Analysis Shared Task (MorphoRuEval-2017)

https://github.com/dialogue-evaluation/morphoRuEval-2017/blob/master/illustration.txt

Example of data:

1	Жалоба	жалоба	NOUN	Animacy=Inan|Case=Nom|Gender=Fem|Number=Sing
2	рассмотрена	рассмотренный	ADJ	Degree=Pos|Gender=Fem|Number=Sing|Variant=Brev
3	во	во	PREP	_
4	второй	второй	ADJ	Case=Loc|Degree=Pos|Gender=Fem|Number=Sing
5	инстанции	инстанция	NOUN	Animacy=Inan|Case=Loc|Gender=Fem|Number=Sing
6	,	,	PUNCT	_
7	ранее	ранее	ADV	Degree=Pos
8	обращение	обращение	NOUN	Animacy=Inan|Case=Acc|Gender=Neut|Number=Sing
9	защиты	защита	NOUN	Animacy=Inan|Case=Gen|Gender=Fem|Number=Sing
10	Ходорковского	Ходорковский	NOUN	Animacy=Anim|Case=Gen|Gender=Masc|Number=Sing
11	и	и	CONJ	_
12	Лебедева	Лебедев	NOUN	Animacy=Anim|Case=Gen|Gender=Masc|Number=Sing
13	отклонил	отклонить	VERB	Gender=Masc|Mood=Ind|Number=Sing|Tense=Past|VerbForm=Fin|Voice=Act
14	собственно	собственно	ADV	Degree=Pos
15	Мосгорсуд	Мосгорсуд	NOUN	Animacy=Inan|Case=Nom|Gender=Masc|Number=Sing
16	.	.	PUNCT	_

MorphoRuEval-2017: a simple yet effective system (Arefyev & Ermolaev'2017)

http://www.dialog-21.ru/media/3966/arefyevnvermolaevpa.pdf

- Bag of character n-grams input representation
  - A window-based classifier treats each window (a target token to be classified with a fixed number of nearby tokens) as a separate example belonging to a single class (the part-of-speech tag of the target token).
  - Window of sizes 1: classification is based on target token only, no context is used)
  - Window of sizes 3 or 5: an example consists of the target token, one token to the left and one to the right

...	^c	^ca	ca	cat	at	at$	t$	...	lower	upper	mixed
...	1	1	1	1	1	1	1	...	1	0	0

Figure 1. Vector representation of the token "cat" for (2–3)-grams. All vector elements not shown are zeros

MorphoRuEval-2017: a simple yet effective system (Arefyev & Ermolaev'2017)

http://www.dialog-21.ru/media/3966/arefyevnvermolaevpa.pdf

Table 2. Influence of category grouping on NB-SVM accuracy

grouping	number of outputs	accuracy
—	10	0.922
Gender+Number+Case, VerbForm+Mood+Tense	6	0.926
Gender+Number	9	0.923
Number+Case	9	0.928
VerbForm+Mood+Tense	8	0.922

Table 1. Accuracy on POS-tagging. NB-SVM (no padding) doesn't add special symbols (^ and $) to the token. NB–SVM (no caps) doesn't use capitalization features

accuracy	model
0.93	Memory baseline
0.97	CRF
0.979	NB-SVM (no padding)
0.98	Tf-idf + linear SVM
0.981	linear SVM
0.983	NB-SVM (no caps)
0.983	NB-SVM

Morphological Analysis Shared Task (MorphoRuEval-2017)

http://www.dialog-21.ru/en/evaluation/2017/morphology/

Team name	team ID	Track	Number of the best try	Accuracy by tags	Accuracyby sentences	Lemmatization, accuracy by wordforms	Lemmatization, accuracy by sentences
MSU-1	C	Closed	2	93.39	65.29		
IQMEN	O	Closed	1	93.08	62.71	92.22	58.21
Sagteam	H	Closed	2	92.64	58.40	80.73	25.01
Aspect	A	Closed	2	92.57	61.01	91.81	56.49
Morphobabushka	M	Closed	2	90.07	48.10		
Pullenti Pos Tagger	G	Closed	4	89.96	47.23	89.32	45.18
	B	Closed	6	89.91	48.2		
	N	Closed	4	89.86	47.13	85.10	29.04
	K	Closed	4	89.46	48.54	88.47	44.78
	F	Closed	2	88.14	39.63	87.27	36.90
	I	Closed	2	86.05	34.62		
	L	Closed	2	71.48	6.48		
ABBYY	E	Open	3	97.11	83.68	96.91	82.13
Aspect	A	Open	4	92.38	60.90	87.66	41.12
	N	Open	5	90.88	51.77	85.91	32.57
	J	Open	1	83.51	29.69		
	D	Open	5	77.13	17.19		

CoNLL format: a universal format for morphological data

ID	FORM	LEMMA	UPOSTAG	XPOSTAG	FEATS	HEAD	DEPREL	DEPS	NER
# newdoc url = http://www.poweredbyosteons.org/2012/01/brief-history-of-bioarchaeological.html									
# newdoc s3 = s3://aws-publicdatasets/common-crawl/crawl-data/CC-MAIN-2016-07/segments...									
...									
# sent_id = http://www.poweredbyosteons.org/2012/01/brief-history-of-bioarchaeological.html#60									
# text = The American Museum of Natural History was established in New York in 1869.									
0	The	the	DT	DT	_	2	det	2:det	O
1	American	American	NNP	NNP	_	2	nn	2:nn	B-Organization
2	Museum	Museum	NNP	NNP	_	7	nsubjpass	7:nsubjpass	I-Organization
3	of	of	IN	IN	_	2	prep	_	I-Organization
4	Natural	Natural	NNP	NNP	_	5	nn	5:nn	I-Organization
5	History	History	NNP	NNP	_	3	pobj	2:prep_of	I-Organization
6	was	be	VBD	VBD	_	7	auxpass	7:auxpass	O
7	established	establish	VBN	VBN	_	7	ROOT	7:ROOT	O
8	in	in	IN	IN	_	7	prep	_	O
9	New	New	NNP	NNP	_	10	nn	10:nn	B-Location
10	York	York	NNP	NNP	_	8	pobj	7:prep_in	I-Location
11	in	in	IN	IN	_	7	prep	_	O
12	1869	1869	CD	CD	_	11	pobj	7:prep_in	O
13	.	.	.	.	_	7	punct	7:punct	O
...									

Morphological (Re-)Inflection

- **SIGMORPHON** Shared Task 2016-2019
  - PLAY +PRESENT PARTICIPLE → playing
  - played+PRESENT PARTICIPLE → playing

Lemma	Tag	Form
RUN	PAST	*ran*
RUN	PRES;1SG	*run*
RUN	PRES;2SG	*run*
RUN	PRES;3SG	*runs*
RUN	PRES;PL	*run*
RUN	PART	*running*

But much less well in low-resource setting

2018 :∼ 96% accuracy on avg. in high-resource setting

https://slideslive.com/38929870/sigmorphon-2020-shared-task-0-typologically-diverse-morphological-inflection

Paradigm Completion

Base	-er/-or	-ee	-ment/-tion	-able/-ible
POS	V↦N	V↦N	V↦N	V↦J
Semantic	AGENT	PATIENT	RESULT	POTENTIAL
animate	*animator*	—	*animation*	*animatable*
attract	*attractor*	*attractee*	*attraction*	*attractable*
—	*aggressor*	*aggressee*	*aggression*	—
employ	*employer*	*employee*	*employment*	*employable*
place	*placer*	—	*placement*	*placeable*
repel	*repeller*	*repelee*	*repellence*	*repellable*
escape	*escapee*	—	—	*escapable*
corrode	*corroder*	—	*corrosion*	*corrosible*
derive	*deriver*	*derivee*	*derivation*	*derivable*

https://arxiv.org/pdf/1708.09151.pdf

Paradigm Discovery

https://arxiv.org/pdf/2005.01630.pdf

Corpus

The cat watched me watching it .
I followed the show but she had n't seen it .
Let 's see who follows your logic .

Lexicon

watching, *seen*, *follows*, *watched*, *followed*, *see*

Gold Grid	cell 1	cell 2	cell 3	cell 4	cell 5
paradigm 1	«*watch*»	«*watches*»	*watching*	*watched*	*watched*
paradigm 2	«*follow*»	*follows*	«*following*»	*followed*	*followed*
paradigm 3	*see*	«*sees*»	«*seeing*»	«*saw*»	*seen*

Table 1: An example corpus, lexicon, and gold analyses. All lexicon entries appear in the corpus and, for our experiments, they will all share a POS, here, verb. The grid reflects all possible analyses of syncretic forms (e.g., *walked*, *followed*), even though these only occur in the corpus as PST realizations, like *saw* in Cell 4, not as PST.PTCP, like *seen* in Cell 5. Bracketed «forms» are paradigm mates of attested forms, not attested in the lexicon.

Morphological Analysis in Context

The $\frac{\text{\_\_\_\_\_}}{\text{(dog)}}$ are barking.

The	dogs	are	barking	.
the	dog	be	bark	.
DET	N;PL	V;PRS;3;PL	V;V.PTCP;PRS	PUNCT

https://arxiv.org/pdf/1910.11493.pdf

Cross-lingual Transfer for Inflection

Low-resource target training data (Asturian)

facer	"fechu"	V;V.PTCP;PST
aguar	"aguà"	V;PRS;2;PL;IND
⋮	⋮	⋮

High-resource source language training data (Spanish)

tocar	"tocando"	V;V.PTCP;PRS
bailar	"bailaba"	V;PST;IPFV;3;SG;IND
mentir	"mintió"	V;PST;PFV;3;SG;IND
⋮	⋮	⋮

Test input (Asturian)

baxar		V;V.PTCP;PRS

Test output (Asturian)

"baxando"

https://arxiv.org/pdf/1910.11493.pdf

Predicting the Growth of Morphological Families

- Growth of Morphological Families

Month	Words
04/2015	"trump"
05/2015	"trump", "trumpish", "trumpster", "trumpy"
06/2015	"trump", "trumpening", "trumper", "trumpish", "trumpness", "trumpology", "trumpster", "trumpy"
07/2015	"trump", "trumpening", "trumper", "trumpic", "trumpification", "trumpiness", "trumpish", "trumpism", "trumpistan", "trumpness", "trumpster", "trumpy"

Table 1: Derivations of "trump" in four subsequent months of the `r/politics` Subreddit.

https://epub.ub.uni-muenchen.de/72198/1/Hofmann_etal_Morphological_Families_ACL2020.pdf

Historical Overview

- 80-early 2000: rule-based morphology
  - Kaplan and Kay (1994)
  - Koskenniemi, Karttunen, Kaplan and Kay
- 2015-2016: UniMorph (unimorph.github.io) by John Sylak-Glassman
  - Cross-lingual morphological annotation schema
- 2016-2020: SIGMORPHON and CoNLL shared tasks
  - Neural approaches outperforms non-neural ones e.g. an ensemble of seq2seq models
    - https://www.aclweb.org/anthology/W16-2002.pdf
  - Morph. Analysis for 52 languages in the 2018 shared task.
  - For a strong neural baselines see: Aharoni & Goldberg (2016) and Makarov & Clematide (2018).
    - https://arxiv.org/pdf/1611.01487
    - https://www.aclweb.org/anthology/K18-3008.pdf