Introduction to Natural Language Processing

a course taught as B4M36NLP at Open Informatics

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE OPEN INFORMATICS FACULTY OF ELECTRICAL ENGINEERING

by members of the Institute of Formal and Applied Linguistics

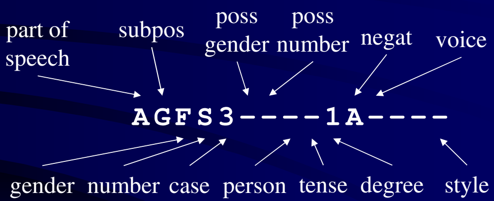
ÚFAL

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Today: **Week 5, lecture**
Today's topic: **Morphological Analysis**
Today's teacher: **Daniel Zeman**

E-mail: zeman@ufal.mff.cuni.cz
WWW: http://ufal.mff.cuni.cz/daniel-zeman

Czech Positional Tags of PDT



part of speech

subpos

poss gender

poss number

negat

voice

AGFS3----1A----

gender

number

case

person

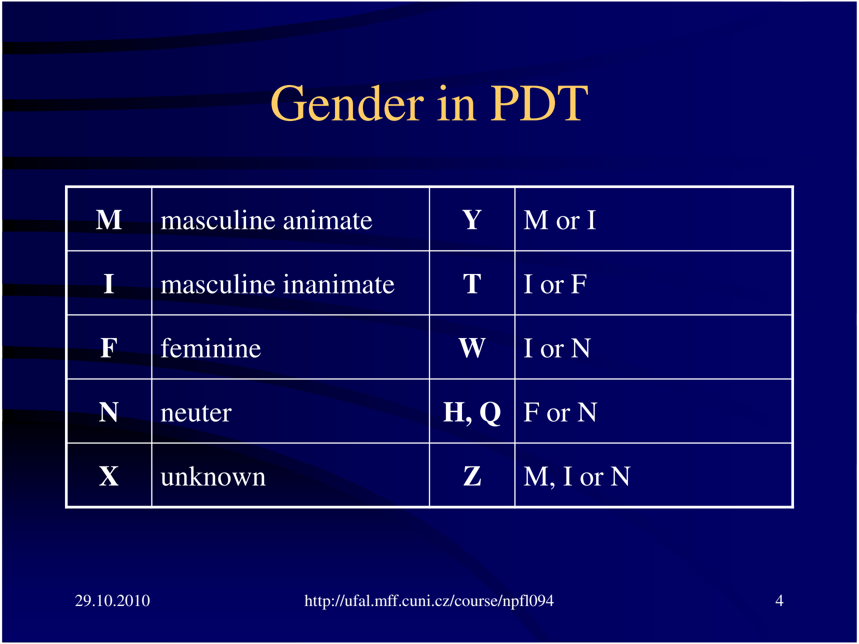
tense

degree

style

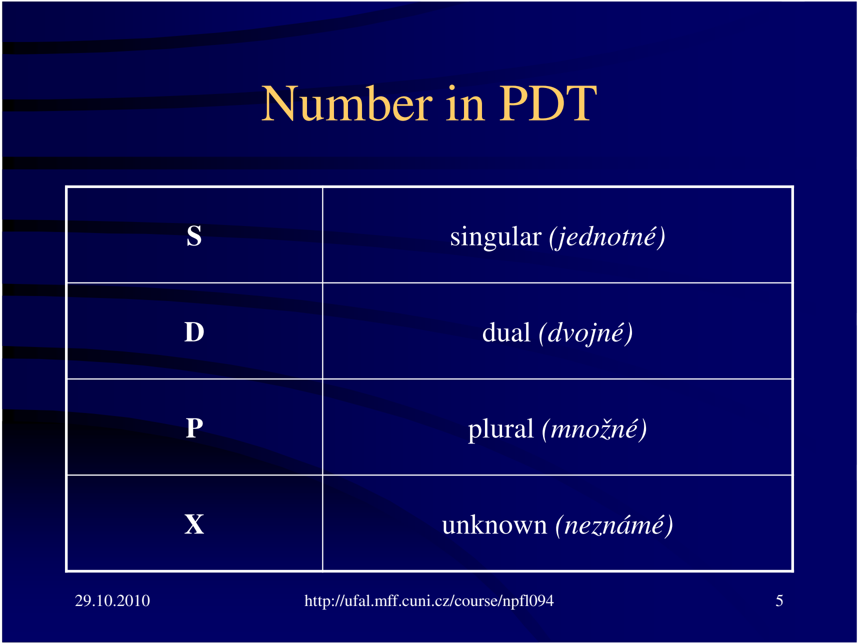
Parts of Speech in PDT

N	noun *(podstatné jméno)*
A	adjective *(přídavné jméno)*
P	pronoun *(zájmeno)*
C	numeral *(číslovka)*
V	verb *(sloveso)*
D	adverb *(příslovce)*
R	preposition *(předložka)*
J	conjunction *(spojka)*
T	particle *(částice)*
I	interjection *(citoslovce)*
Z	special (e.g. punctuation) *(zvláštní, např. interpunkce)*
X	unknown word *(neznámé slovo)*



Gender in PDT

M	masculine animate	**Y**	M or I
I	masculine inanimate	**T**	I or F
F	feminine	**W**	I or N
N	neuter	**H, Q**	F or N
X	unknown	**Z**	M, I or N



Number in PDT

S	singular *(jednotné)*
D	dual *(dvojné)*
P	plural *(množné)*
X	unknown *(neznámé)*

Case in PDT

1	nominative
2	genitive
3	dative
4	accusative
5	vocative
6	locative
7	instrumental
X	unknown

Degree, Negativeness and Person

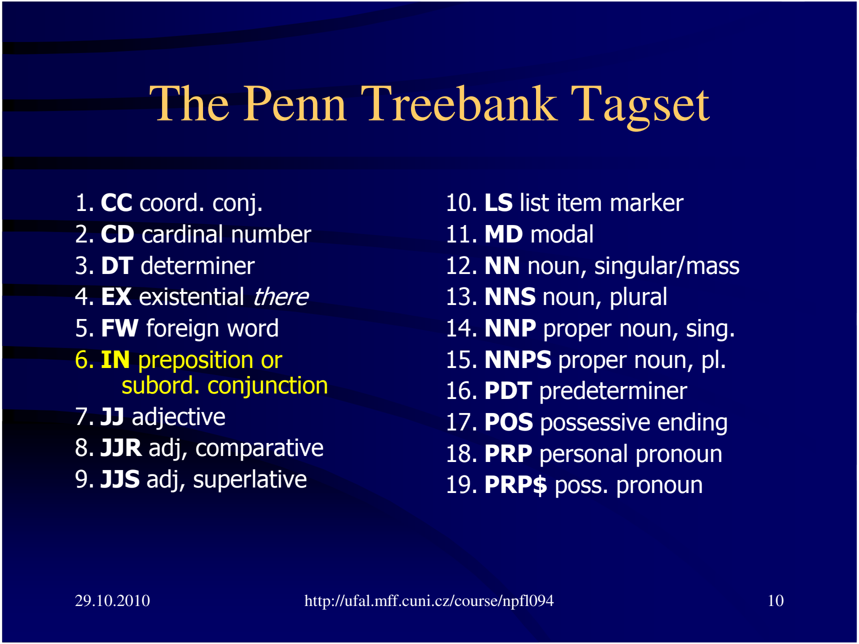
- Degree of comparison of adjectives and adverbs:
  - 1 (positive), 2 (comparative), 3 (superlative)
- Negativeness (for V, A, D and N):
  - A (affirmative), N (negative)
- Person of verbs and pronouns:
  - 1, 2, 3

Mood, Tense and Voice

- Changes relevance of other categories (such as person and number) ⇒ in a sense, these are (sub-)parts of speech
- Tense: present (P), past (M), future (F)
- Voice: active (A), passive (P)
- Mood: indicative (N), imperative (R), conditional (C – e.g. *bych*)

Style and/or Variant

1	other variant, less frequent
2	other variant, very rare, archaic or literary
3	very archaic or colloquial variant
5	colloquial, tolerated in both spoken and written discourse
6	colloquial, inappropriate in written discourse
7	colloquial like 6 but less preferred by speakers
9	special usage (e.g. after some prepositions)

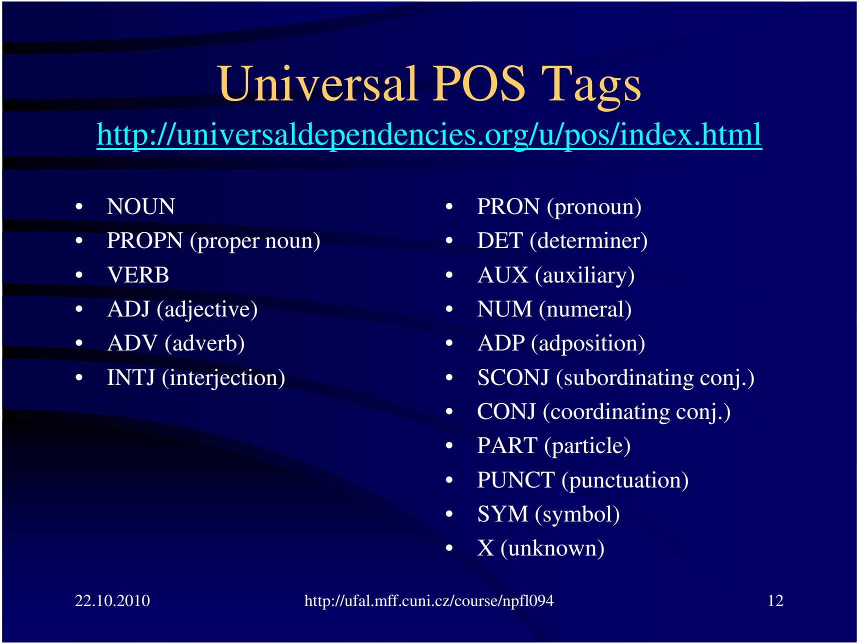


The Penn Treebank Tagset

1. **CC** coord. conj.
2. **CD** cardinal number
3. **DT** determiner
4. **EX** existential *there*
5. **FW** foreign word
6. **IN** preposition or subord. conjunction
7. **JJ** adjective
8. **JJR** adj, comparative
9. **JJS** adj, superlative
10. **LS** list item marker
11. **MD** modal
12. **NN** noun, singular/mass
13. **NNS** noun, plural
14. **NNP** proper noun, sing.
15. **NNPS** proper noun, pl.
16. **PDT** predeterminer
17. **POS** possessive ending
18. **PRP** personal pronoun
19. **PRP$** poss. pronoun

The Penn Treebank Tagset

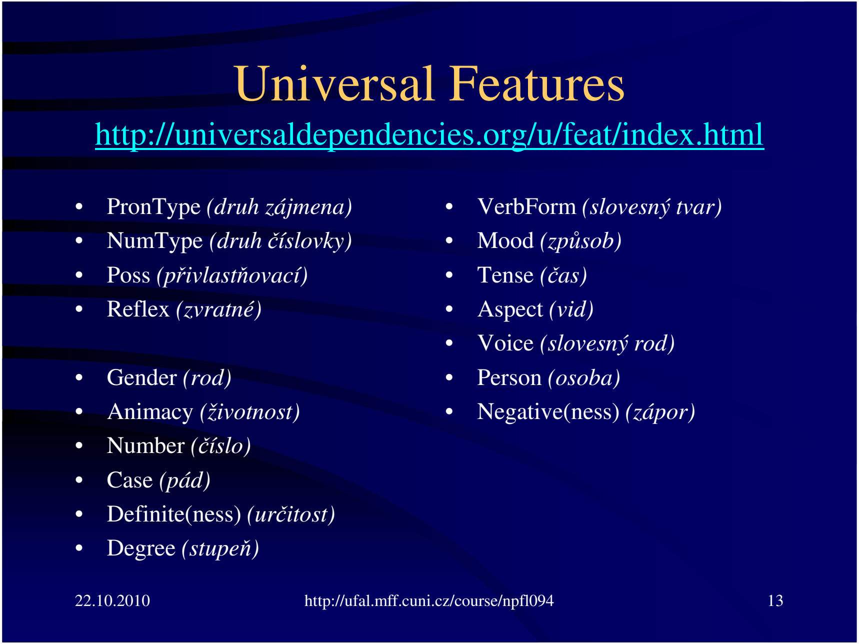
20. **RB** adverb
21. **RBR** adv, comparative
22. **RBS** adv, superlative
23. **RP** particle
24. **SYM** symbol
25. **TO** *to*
26. **UH** interjection
27. **VB** verb, base *(do)*
28. **VBD** verb, past *(did)*
29. **VBG** verb, gerund or pres. participle *(doing)*
30. **VBN** verb, past participle *(done)*
31. **VBP** verb, non-3rd pers. sing. present *(do)*
32. **VBZ** verb, 3rd pers. sing. present *(does)*
33. **WDT** wh-det. *(which?)*
34. **WP** wh-pronoun *(who)*
35. **WP$** possessive wh-pronoun *(whose)*
36. **WRB** wh-adv. *(where)*
37. **.** period…



Universal POS Tags

http://universaldependencies.org/u/pos/index.html

- NOUN
- PROPN (proper noun)
- VERB
- ADJ (adjective)
- ADV (adverb)
- INTJ (interjection)
- PRON (pronoun)
- DET (determiner)
- AUX (auxiliary)
- NUM (numeral)
- ADP (adposition)
- SCONJ (subordinating conj.)
- CONJ (coordinating conj.)
- PART (particle)
- PUNCT (punctuation)
- SYM (symbol)
- X (unknown)



Universal Features

http://universaldependencies.org/u/feat/index.html

- PronType *(druh zájmena)*
- NumType *(druh číslovky)*
- Poss *(přivlastňovací)*
- Reflex *(zvratné)*

- Gender *(rod)*
- Animacy *(životnost)*
- Number *(číslo)*
- Case *(pád)*
- Definite(ness) *(určitost)*
- Degree *(stupeň)*

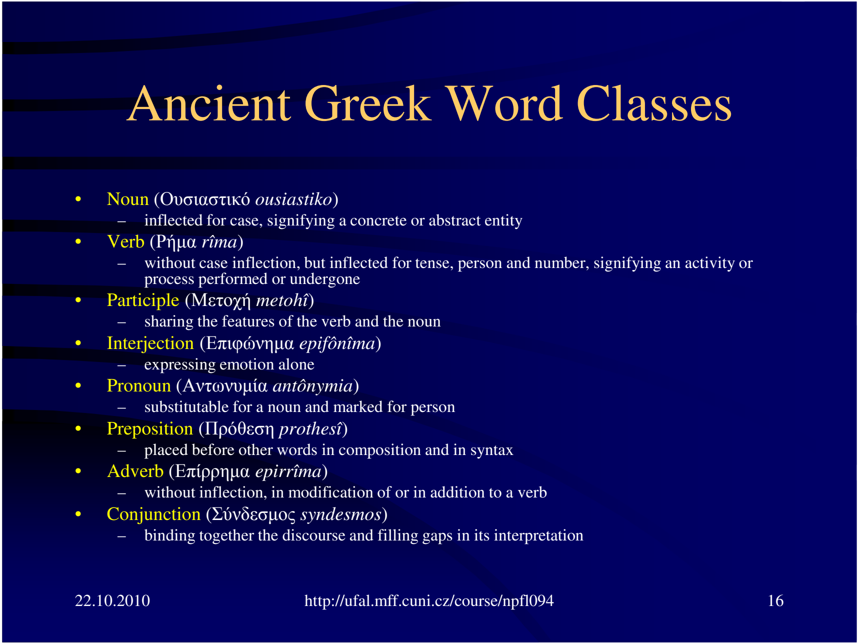
- VerbForm *(slovesný tvar)*
- Mood *(způsob)*
- Tense *(čas)*
- Aspect *(vid)*
- Voice *(slovesný rod)*
- Person *(osoba)*
- Negative(ness) *(zápor)*

Part of Speech

- Vague definitions, criteria of mixed nature
- Looong tradition… (difficult to change)
  - Traditional linguistics:
    - Classification differs cross-linguistically!
    - (Even among established classes, not just endemic minor parts of speech.)
  - Computational linguistics (tagsets):
    - Dozens of classes and subclasses
    - Significant differences even within one language

History

- 4th century BC: Sanskrit
- European tradition (prevailing in modern linguistics): Ancient Greek
  - Plato (4th century BC): sentence consists of nouns and verbs
  - Aristotle added “conjunctions” (included conjunctions, pronouns and articles)
  - End of 2nd century BC: classification stabilized at 8 categories (Διονύσιος ὁ Θρᾷξ: *Τέχνη Γραμματική* / Dionysios o Thrax: *Art of Grammar*)

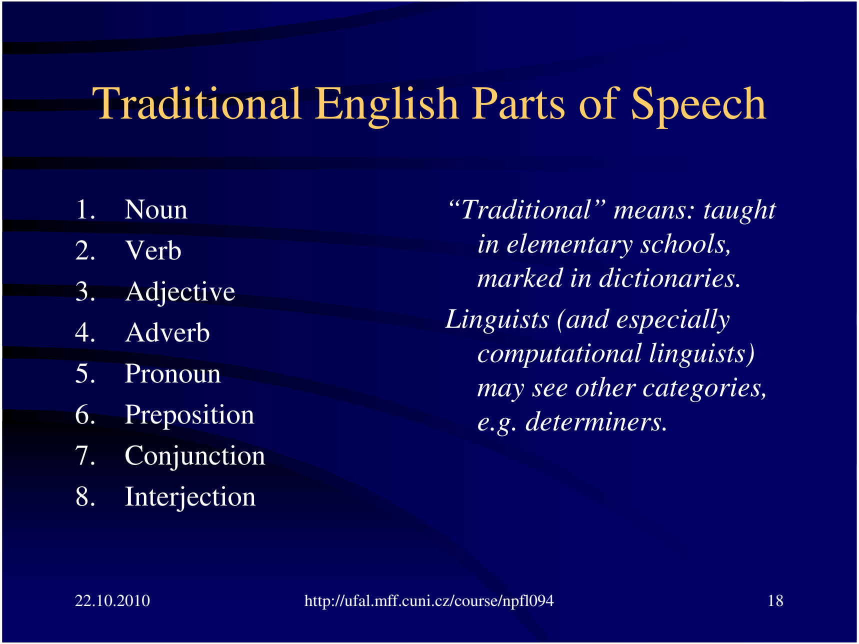


Ancient Greek Word Classes

- Noun (Ουσιαστικό *ousiastiko*)
  - inflected for case, signifying a concrete or abstract entity
- Verb (Ρήμα *rîma*)
  - without case inflection, but inflected for tense, person and number, signifying an activity or process performed or undergone
- Participle (Μετοχή *metohî*)
  - sharing the features of the verb and the noun
- Interjection (Επιφώνημα *epifônîma*)
  - expressing emotion alone
- Pronoun (Αντωνυμία *antônymia*)
  - substitutable for a noun and marked for person
- Preposition (Πρόθεση *prothesî*)
  - placed before other words in composition and in syntax
- Adverb (Επίρρημα *epirrîma*)
  - without inflection, in modification of or in addition to a verb
- Conjunction (Σύνδεσμος *syndesmos*)
  - binding together the discourse and filling gaps in its interpretation

Where Are Adjectives?

- The best matching Ancient Greek definition is that of nouns, and perhaps participles.
- Adjectives are a relatively new (1767) invention from France:
  - Nicolas Beauzée: *Grammaire générale, ou exposition raisonnée des éléments nécessaires du langage.* Paris, 1767



Traditional English Parts of Speech

1. Noun
2. Verb
3. Adjective
4. Adverb
5. Pronoun
6. Preposition
7. Conjunction
8. Interjection

"Traditional" means: taught in elementary schools, marked in dictionaries.

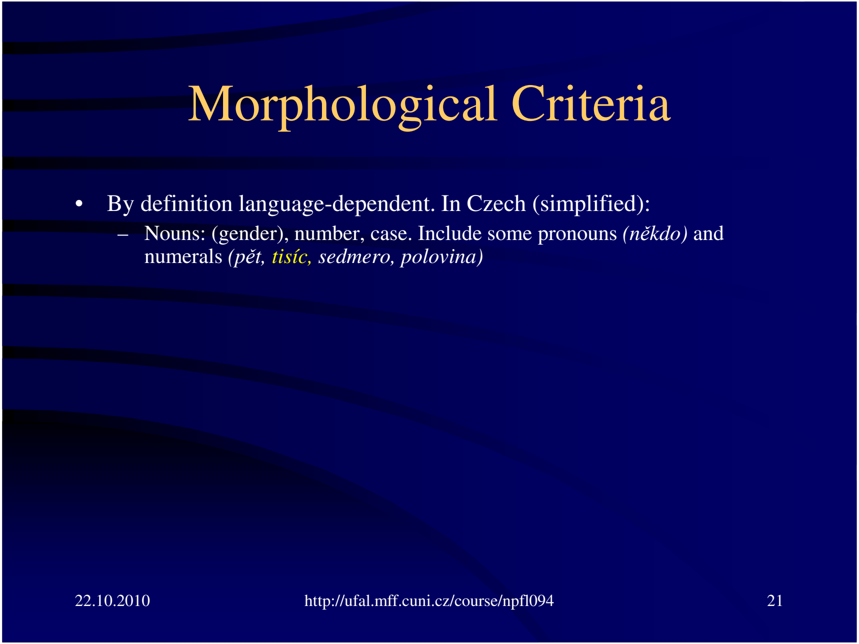
Linguists (and especially computational linguists) may see other categories, e.g. determiners.

Traditional Czech Parts of Speech

1. Noun *(podstatné jméno, substantivum)*
2. Adjective *(přídavné jméno, adjektivum)*
3. Pronoun *(zájmeno)*
4. Numeral *(číslovka)*
5. Verb *(sloveso)*
6. Adverb *(příslovce, adverbium)*
7. Preposition *(předložka)*
8. Conjunction *(spojka)*
9. Particle *(částice)*
10. Interjection *(citoslovce)*

A Mixture of Criteria

- Parts of speech are defined on the basis of morphological, syntactic and semantic criteria
- In many cases they are just rough approximation
- Because of long *tradition* in some languages, it is difficult to redesign the system
- Sets of POS tags strive to
  - keep reasonable consistency with tradition
  - partition the word space systematically



Morphological Criteria

- By definition language-dependent. In Czech (simplified):
  - Nouns: (gender), number, case. Include some pronouns *(někdo)* and numerals *(pět, tisíc, sedmero, polovina)*

Morphological Criteria

- By definition language-dependent. In Czech (simplified):
  - Nouns: (gender), number, case. Include some pronouns *(někdo)* and numerals *(pět, tisíc, sedmero, polovina)*
  - Adjectives: gender, number, case, sometimes degree; agr. with N. Include some pronouns *(který, žádný)* and numerals *(první, druhý, čtverý)*

Morphological Criteria

- By definition language-dependent. In Czech (simplified):
  - Nouns: (gender), number, case. Include some pronouns *(někdo)* and numerals *(pět, tisíc, sedmero, polovina)*
  - Adjectives: gender, number, case, sometimes degree; agr. with N. Include some pronouns *(který, žádný)* and numerals *(první, druhý, čtverý)*
  - Personal pronouns: person, gender, number, case

Morphological Criteria

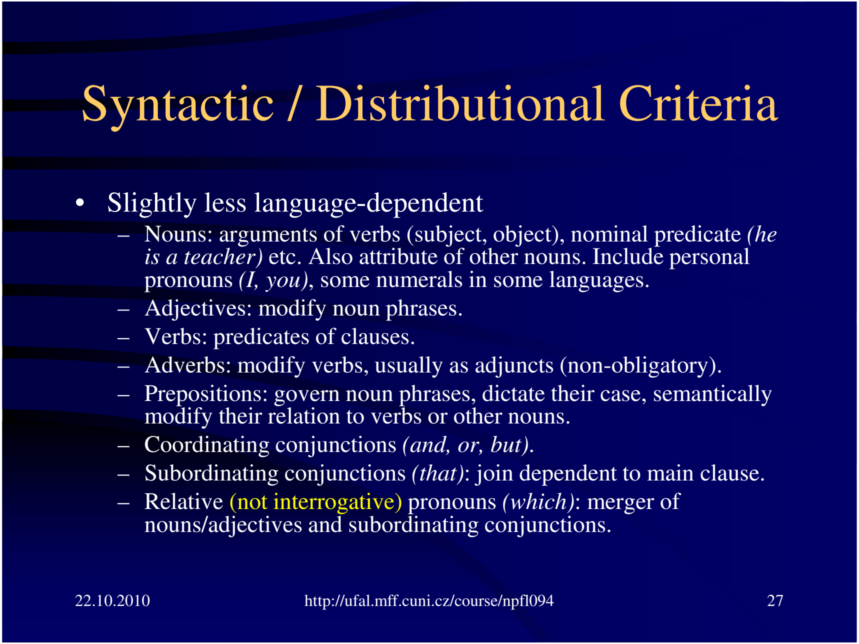
- By definition language-dependent. In Czech (simplified):
  - Nouns: (gender), number, case. Include some pronouns *(někdo)* and numerals *(pět, tisíc, sedmero, polovina)*
  - Adjectives: gender, number, case, sometimes degree; agr. with N. Include some pronouns *(který, žádný)* and numerals *(první, druhý, čtverý)*
  - Personal pronouns: person, gender, number, case
  - Possessive pronouns: possessor's person, gender & number; possessed gender & number

Morphological Criteria

- By definition language-dependent. In Czech (simplified):
  - Nouns: (gender), number, case. Include some pronouns *(někdo)* and numerals *(pět, tisíc, sedmero, polovina)*
  - Adjectives: gender, number, case, sometimes degree; agr. with N. Include some pronouns *(který, žádný)* and numerals *(první, druhý, čtverý)*
  - Personal pronouns: person, gender, number, case
  - Possessive pronouns: possessor's person, gender & number; possessed gender & number
  - Verbs:
    - infinitive
    - finite: mood (indicative/imperative), tense (present/future), person, number
    - participle: voice (active/passive), gender, number
    - transgressive: tense (present/past), gender, number

Morphological Criteria

- By definition language-dependent. In Czech (simplified):
  - Nouns: (gender), number, case. Include some pronouns *(někdo)* and numerals *(pět, tisíc, sedmero, polovina)*
  - Adjectives: gender, number, case, sometimes degree; agr. with N. Include some pronouns *(který, žádný)* and numerals *(první, druhý, čtverý)*
  - Personal pronouns: person, gender, number, case
  - Possessive pronouns: possessor's person, gender & number; possessed gender & number
  - Verbs:
    - infinitive
    - finite: mood (indicative/imperative), tense (present/future), person, number
    - participle: voice (active/passive), gender, number
    - transgressive: tense (present/past), gender, number
  - Non-inflectional words



Syntactic / Distributional Criteria

- Slightly less language-dependent
  - Nouns: arguments of verbs (subject, object), nominal predicate *(he is a teacher)* etc. Also attribute of other nouns. Include personal pronouns *(I, you)*, some numerals in some languages.
  - Adjectives: modify noun phrases.
  - Verbs: predicates of clauses.
  - Adverbs: modify verbs, usually as adjuncts (non-obligatory).
  - Prepositions: govern noun phrases, dictate their case, semantically modify their relation to verbs or other nouns.
  - Coordinating conjunctions *(and, or, but)*.
  - Subordinating conjunctions *(that)*: join dependent to main clause.
  - Relative (not interrogative) pronouns *(which)*: merger of nouns/adjectives and subordinating conjunctions.

Syntactic Nouns

- Arguments of verbs (subject, object), nominal predicate *(he is a* ***teacher****)* etc.
- Attributes of other nouns (`cs:` *auto prezidenta = president's car*)
  - `en:` *Christmas present:* is *Christmas* a syntactic adjective or noun?
  - Even if definitions are purely syntactic, consensus across languages is not guaranteed because every language has its own set of syntactic constructions
- Including
  - pronouns: personal *(I, you, he, we)*, indefinite *(somebody)*, negative *(nothing)*, totality *(everyone)*, some demonstratives *(this* in ***this*** *is ridiculous)*
  - `cs:` some numerals in some cases *(pět, deset, tisíc, miliarda, třetina, sedminásobek, desatero)*

Syntactic Adjectives

- Modify a noun phrase, typically agree with it in gender, number and case. Include:
  - Possessive pronouns (determiners?) *(my, your, his, our)*
  - Demonstrative pronouns in some contexts *(**this** apple is sweet)*
  - Some indefinite and other pronouns in some languages (`cs:` *nějaký (some), každý (every), žádný (no))* (in other languages these may not be traditionally considered pronouns)
  - Cardinal numerals (but see next slide) *(one, two, three)*
  - Adjectival ordinal numerals *(first, second, third)*
  - Adjectivally used participles *(**traveling** salesman, **mixed** feelings)*
  - Possibly even adjectivally used nouns *(**Christmas** present, **car** repair, New **York Times advisory board** member)*

Syntactic Behavior of Czech Cardinal Numerals

- *jeden (one), dva (two), tři (three), čtyři (four)* are syntactic adjectives. They agree in case (and also gender and number) with the counted noun
- *pět (five)* and higher may behave as syntactic nouns
  - whole phrase in nominative / accusative / vocative: the numeral governs the counted noun, forces it to genitive: *pět* **/nom** *židlí (five chairs)* **/gen**, not *pět *židle* **/nom** ⇒ *pět* is syntactic noun
  - whole phrase in other cases: the numeral agrees in case with the counted noun ⇒ it modifies the noun: *k pěti***/dat** *židlím***/dat** *(to five chairs)* ⇒ *pěti* is a syntactic adjective
- *tisíc (thousand), milión (million), miliarda (billion)* in both Czech and English can be used as
  - nouns (morphologically and syntactically): *z banky zmizely milióny = millions vanished from a bank*
  - traditional numerals, syntactic nouns: *dluží mi milión dolarů = he owes me one million dollars*

Syntactic Verbs

- Predicate of a main clause
- Predicate of a dependent clause
- Auxiliary verb, modal verb or another part of a complex verb form:
  - `en:` *would have been willing (to) keep smiling* ☺
  - `cs:` *bych byl býval mohl chtít udělat*
    (= (I) could have wanted to do)
- Copula in nominal predicates:
  - `en:` *he is a teacher*

Syntactic Adverbs

- Modify verbs, optionally specify circumstances such as location, time, manner, extent, cause…
- Can also modify adjectives *(very large)* or other adverbs *(very well)*
- Including:
  - some ordinal numerals: `cs:` *poprvé (for the first time)*
  - multiplicative numerals: `cs:` *dvakrát (twice), pětasedmdesátkrát (seventy-five times)*
  - converbs (transgressives): `cs:` *čekajíc na autobus všimla si ho (she noticed him while waiting for a bus);* `hi:` दरवाज़ा खोलकर वह कमरे में आई *darvāzā kholkar vah kamre mẽ āī (having opened the door she came in)*

Conjunctions

- Coordinating conjunctions join phrases of same or similar type or even whole clauses (independent)
  - single coordinators:
    - *Peter and Paul; today or tomorrow; he wanted to go but she didn't like the idea*
  - paired coordinators:
    - *neither here nor there; the sooner the better; as soon as possible*
- Subordinating conjunctions join dependent clauses or phrases to the governing node, specifying their function
  - single subordinators:
    - *that; so; if; whether; because*
  - paired subordinators:
    - `hi:` जब मैं कहूँगा तब आना *jab maĩ kahū̃gā tab ānā* (lit: *when I tell then come)*

Relative Pronouns, Determiners, Numerals and Adverbs

- Merge properties of syntactic nouns / adjectives / adverbs and of subordinating conjunctions
  - relative syntactic noun: *those who know; a car that never breaks; the man whom I met; who knows what you find*
  - relative syntactic adjective: *the man whose son is this boy; you decide from what time on you work; …which color you like*
    - `cs:` relative numerals: *pověz mi, kolik máš peněz (tell me how much money you have); …kolikátý jsi byl (where did you rank;* lit. *how-many-th you were)*
  - relative syntactic adverb: *I don't know when she came; …where it is; …how to say; …why he's here*
- Interrogative pronouns (adverbs etc.) may have same form (in some languages) but not the same joining function.

Adpositions

- Govern syntactic noun (dictate its case marking), specify its role as argument of
  - a verb *(believe in something)*
  - another noun *(lack of something)*
  - or adjective *(acceptable for me)*
- Appear before, after or around the noun phrase:
  - Preposition: *in the house; under the table; beyond this point*
  - Postposition: `hi:` कमरे में *kamre mẽ* (lit. *room in)*
  - Circumposition: `de:` *von diesem Zeitpunkt an (from this moment on)*

Semantic / Notional Criteria

- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*

Semantic / Notional Criteria

- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*
- Semantic adjective: a quality, property
  - `en:` *cleverly* could be regarded as a form of the semantic adjective *clever*
  - How far should we go? Is *cleverness* an adjective, too? What purpose would such classification serve?

Semantic / Notional Criteria

- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*
- Semantic adjective: a quality, property
  - `en:` *cleverly* could be regarded as a form of the semantic adjective *clever*
  - How far should we go? Is *cleverness* an adjective, too? What purpose would such classification serve?
- Semantic adverb: a circumstance (location, time, manner)
  - `cs:` traditional adjective *zítřejší* could be regarded as a form of the semantic adverb *zítra (tomorrow)*

Semantic / Notional Criteria

- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*
- Semantic adjective: a quality, property
  - `en:` *cleverly* could be regarded as a form of the semantic adjective *clever*
  - How far should we go? Is *cleverness* an adjective, too? What purpose would such classification serve?
- Semantic adverb: a circumstance (location, time, manner)
  - `cs:` traditional adjective *zítřejší* could be regarded as a form of the semantic adverb *zítra (tomorrow)*
- Semantic verb: a state or an action
  - `cs:` deverbative nouns *(dělání = the doing)* and adjectives *(dělající = doing; udělavší = the one that did; udělaný = done)* could be regarded as forms of the semantic verb

Semantic / Notional Criteria

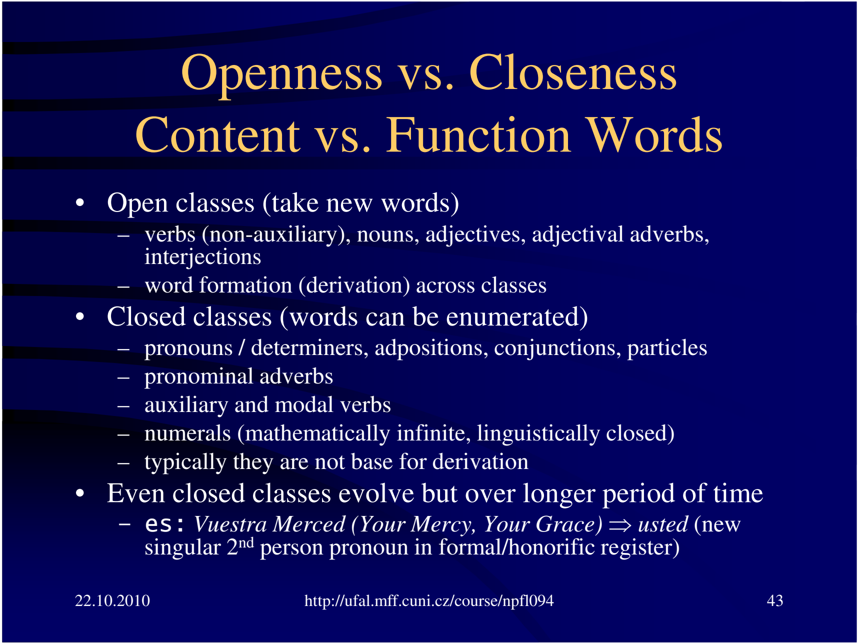
- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*
- Semantic adjective: a quality, property
  - `en:` *cleverly* could be regarded as a form of the semantic adjective *clever*
  - How far should we go? Is *cleverness* an adjective, too? What purpose would such classification serve?
- Semantic adverb: a circumstance (location, time, manner)
  - `cs:` traditional adjective *zítřejší* could be regarded as a form of the semantic adverb *zítra (tomorrow)*
- Semantic verb: a state or an action
  - `cs:` deverbative nouns *(dělání = the doing)* and adjectives *(dělající = doing; udělavší = the one that did; udělaný = done)* could be regarded as forms of the semantic verb
- Pronoun: any referential word (trad. pronoun, determiner, numeral, adverb / personal, possessive, indefinite, absolute, negative, interrogative, relative, demonstrative)

Semantic / Notional Criteria

- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*
- Semantic adjective: a quality, property
  - `en:` *cleverly* could be regarded as a form of the semantic adjective *clever*
  - How far should we go? Is *cleverness* an adjective, too? What purpose would such classification serve?
- Semantic adverb: a circumstance (location, time, manner)
  - `cs:` traditional adjective *zítřejší* could be regarded as a form of the semantic adverb *zítra (tomorrow)*
- Semantic verb: a state or an action
  - `cs:` deverbative nouns *(dělání = the doing)* and adjectives *(dělající = doing; udělavší = the one that did; udělaný = done)* could be regarded as forms of the semantic verb
- Pronoun: any referential word (trad. pronoun, determiner, numeral, adverb / personal, possessive, indefinite, absolute, negative, interrogative, relative, demonstrative)
- Numeral: a number, amount *(one, two, three; first, second, third; once, twice, thrice; twofold; pair, triple, quadruple)*

Semantic / Notional Criteria

- Semantic noun: a concrete or abstract entity
  - `cs:` *otcův (father's)* is traditionally a possessive adjective but could be regarded as a form of the semantic noun *otec (father);* not to confuse with genitive case *otce/otců*
- Semantic adjective: a quality, property
  - `en:` *cleverly* could be regarded as a form of the semantic adjective *clever*
  - How far should we go? Is *cleverness* an adjective, too? What purpose would such classification serve?
- Semantic adverb: a circumstance (location, time, manner)
  - `cs:` traditional adjective *zítřejší* could be regarded as a form of the semantic adverb *zítra (tomorrow)*
- Semantic verb: a state or an action
  - `cs:` deverbative nouns *(dělání = the doing)* and adjectives *(dělající = doing; udělavší = the one that did; udělaný = done)* could be regarded as forms of the semantic verb
- Pronoun: any referential word (trad. pronoun, determiner, numeral, adverb / personal, possessive, indefinite, absolute, negative, interrogative, relative, demonstrative)
- Numeral: a number, amount *(one, two, three; first, second, third; once, twice, thrice; twofold; pair, triple, quadruple)*
- Adpositions + conjunctions + particles + auxiliaries (glue material)



Openness vs. Closeness
Content vs. Function Words

- Open classes (take new words)
  - verbs (non-auxiliary), nouns, adjectives, adjectival adverbs, interjections
  - word formation (derivation) across classes
- Closed classes (words can be enumerated)
  - pronouns / determiners, adpositions, conjunctions, particles
  - pronominal adverbs
  - auxiliary and modal verbs
  - numerals (mathematically infinite, linguistically closed)
  - typically they are not base for derivation
- Even closed classes evolve but over longer period of time
  - `es:` *Vuestra Merced (Your Mercy, Your Grace)* ⇒ *usted* (new singular 2nd person pronoun in formal/honorific register)

The Big Four

- Nouns
  - Proper nouns
- Verbs
  - Participles (between verbs and nouns / adjectives / adverbs)
- Adjectives
  - Modify nouns
- Adverbs
  - Modify verbs, adjectives or adverbs

Common Minors

- Adpositions
  - Prepositions
  - Postpositions
  - Circumpositions
- Conjunctions
  - Subordinators
  - Coordinators
- Interjections
- Particles (often “garbage can category”)

Pronouns vs. Determiners

- In some tagsets clear (but context-dependent) definition:
  - Pronouns *replace* noun phrases
    - *I, you, he, she, it, we, they, who, something…*
    - *This is unbelievable!*
    - *Yours is better.*
  - Determiners *modify* noun phrases (so they include traditional possessive “pronouns”)
    - *This book is John’s.*
    - *Your book is better.*
- Some traditional grammars (and tagsets) refer to all the above as pronouns (e.g. Czech)

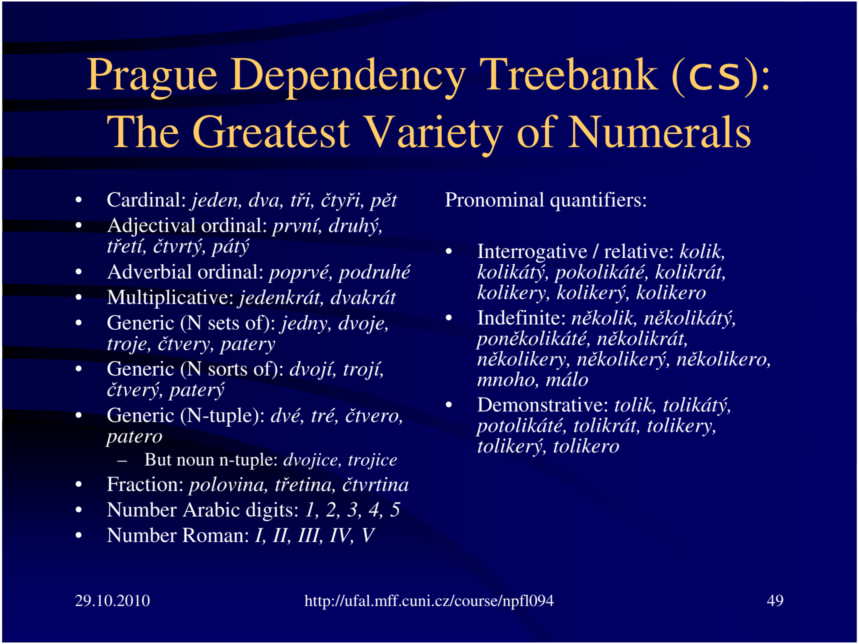
BulTreeBank Tagset (bg): The Broadest Sense of Pronouns

- Subcategories:
  - personal: *аз, ти, той*
  - possessive: *мой, моя, твой, негов*
  - demonstrative: *този, тоя*
  - interrogative: *кой, коя, кое*
  - relative: *който, що*
  - collective: *всеки, всякой*
  - indefinite: *един, някой*
  - negative: *никой, никакъв*

- Referential type:
  - entity: *кой, коя, кое*
  - attribute: *какъв, каква, какво*
  - possession: *чий, чия, чие*
  - quantity: *колко, доколко*
  - location: *къде, где, докъде*
  - time: *кога, докога, откога*
  - manner: *как*
  - cause: *защо*

Numerals vs. Adjectives

- Many tagsets distinguish *cardinal numbers*
  - While some (Danish) take them as special class of adjectives
- Ordinal numbers
  - Sometimes separate POS
  - Sometimes special class of adjectives
  - Sometimes “normal” adjectives (undistinguished)



Prague Dependency Treebank (cs): The Greatest Variety of Numerals

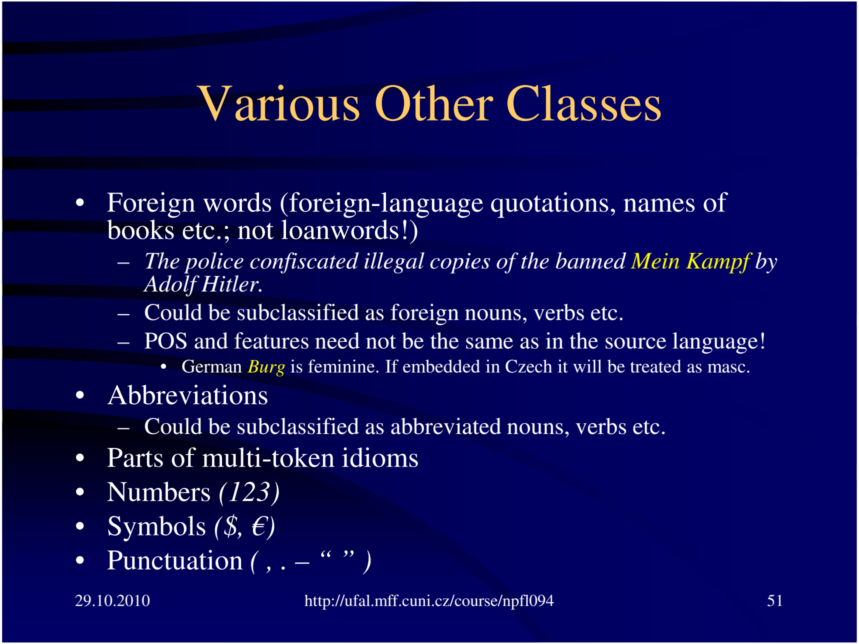
- Cardinal: *jeden, dva, tři, čtyři, pět*
- Adjectival ordinal: *první, druhý, třetí, čtvrtý, pátý*
- Adverbial ordinal: *poprvé, podruhé*
- Multiplicative: *jedenkrát, dvakrát*
- Generic (N sets of): *jedny, dvoje, troje, čtvery, patery*
- Generic (N sorts of): *dvojí, trojí, čtverý, paterý*
- Generic (N-tuple): *dvé, tré, čtvero, patero*
  - But noun n-tuple: *dvojice, trojice*
- Fraction: *polovina, třetina, čtvrtina*
- Number Arabic digits: *1, 2, 3, 4, 5*
- Number Roman: *I, II, III, IV, V*

Pronominal quantifiers:

- Interrogative / relative: *kolik, kolikátý, pokolikáté, kolikrát, kolikery, kolikerý, kolikero*
- Indefinite: *několik, několikátý, poněkolikáté, několikrát, několikery, několikerý, několikero, mnoho, málo*
- Demonstrative: *tolik, tolikátý, potolikáté, tolikrát, tolikery, tolikerý, tolikero*

Some Endemic Classes

- Existential *there* in English
- Infinitival marker: English *to*, German *zu*, Swedish *att*
- Predeterminer: English *both the boys, all the people*
- Response particle: *yes, no, thanks*
- Negative particle: *not, n’t,* Arabic لا *lā*
- Question particle: Polish *czy*, Hindi क्या *kyā*
- Separable verbal prefix: German *vorstellen* ⇒ *stellen Sie sich vor*
- Adjectival particle: German *am besten, zu groß*
- Classifier: Chinese 一個人 = *yí gè rén* = “one (piece) man”



Various Other Classes

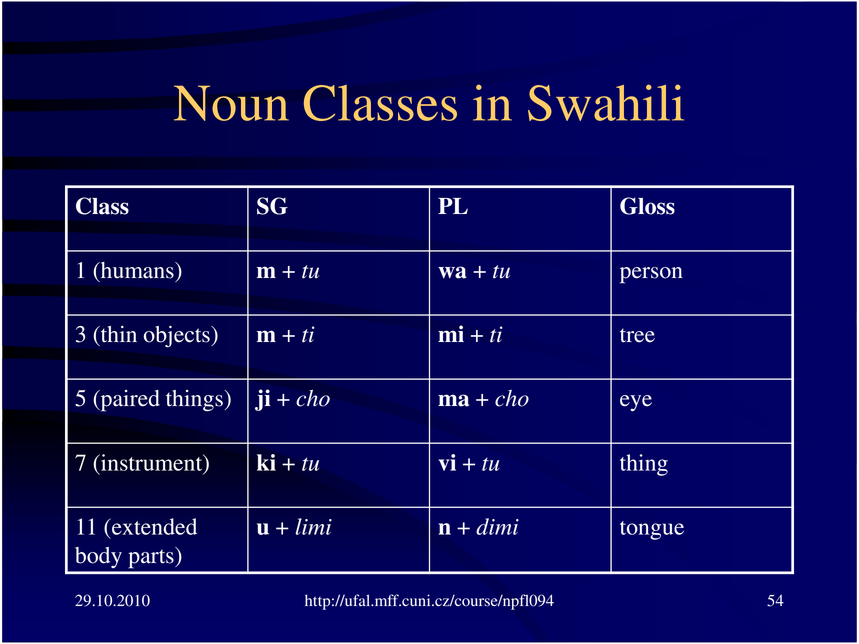
- Foreign words (foreign-language quotations, names of books etc.; not loanwords!)
  - *The police confiscated illegal copies of the banned Mein Kampf by Adolf Hitler.*
  - Could be subclassified as foreign nouns, verbs etc.
  - POS and features need not be the same as in the source language!
    - German *Burg* is feminine. If embedded in Czech it will be treated as masc.
- Abbreviations
  - Could be subclassified as abbreviated nouns, verbs etc.
- Parts of multi-token idioms
- Numbers *(123)*
- Symbols *($, €)*
- Punctuation *(, . – “ ”)*

Clitics

- Clitic is a
  - Syntactically independent word
  - Phonologically / orthographically dependent morpheme
- es: *despiértate = wake yourself; démelo = give me it;*
  ru: *защищаться = zaščiščat’sja = to defend oneself*
- de: *zum = zu dem = to the; am = an dem = on the;*
  fr: *du = de le = of the*
- cs: *proň = pro něj = for him; oč = o co = for what; tys = ty jsi = you have; žes = že jsi = that you have; scvrnkls = scvrnkl jsi = you flicked off; přišelť = neboť přišel = because he came*
- ar: وبالفالوجة = *wabiālfālūjah* = **wa/CONJ + bi/PREP + Al fAlwjp/NOUN_PROP** = *and in al-Falujah*

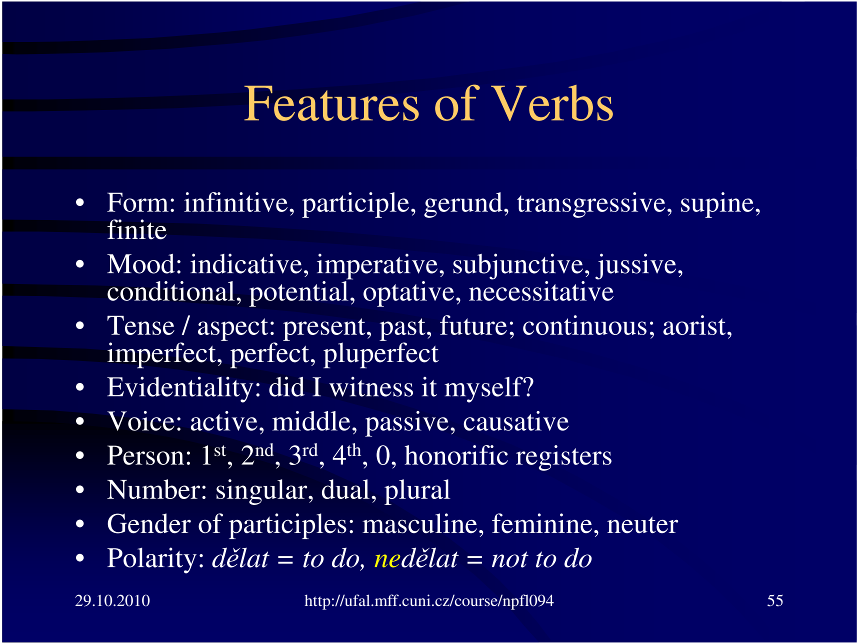
Features of Nouns and Adjectives

- Gender / animateness (lexical for nouns, agreemental for adjectives) or class (Bantu languages)
- Number (singular, dual, plural, trial, paucal)
- Case (`en:` 2 for pronouns; `cs:` 7; `fi:` 14)
- Definiteness (`ro:` *poiană = a meadow, poiana = the meadow)*
- Polarity (`cs:` *schopný = able, neschopný = unable; schopnost = ability, neschopnost = inability)*
- Degree of comparison (positive, comparative, superlative, absolutive)



Noun Classes in Swahili

Class	SG	PL	Gloss
1 (humans)	**m** + *tu*	**wa** + *tu*	person
3 (thin objects)	**m** + *ti*	**mi** + *ti*	tree
5 (paired things)	**ji** + *cho*	**ma** + *cho*	eye
7 (instrument)	**ki** + *tu*	**vi** + *tu*	thing
11 (extended body parts)	**u** + *limi*	**n** + *dimi*	tongue



Features of Verbs

- Form: infinitive, participle, gerund, transgressive, supine, finite
- Mood: indicative, imperative, subjunctive, jussive, conditional, potential, optative, necessitative
- Tense / aspect: present, past, future; continuous; aorist, imperfect, perfect, pluperfect
- Evidentiality: did I witness it myself?
- Voice: active, middle, passive, causative
- Person: 1st, 2nd, 3rd, 4th, 0, honorific registers
- Number: singular, dual, plural
- Gender of participles: masculine, feminine, neuter
- Polarity: *dělat = to do, nedělat = not to do*

Other Features

- Case of adpositions (subcategorization, not inflection)
  - What case must the governed noun phrase be in?
- Possessor's gender and number
  - cs: *jejímu psovi = to her dog:* feminine possessor, masculine possessed
  - cs: *jehož kráva = whose ("of which guy") cow:* singular masculine possessor, singular feminine possessed
  - cs: *jejíž kráva = whose ("of which woman") cow:* singular feminine possessor, singular feminine possessed
  - cs: *jejichž kráva = whose ("of which people") cow:* plural possessor, singular possessed

Morphological and Syntactic Analysis

Two-Level Morphology

Daniel Zeman

http://ufal.mff.cuni.cz/course/npfl094

Two-Level (Mor)Phonology

- Kimmo Koskenniemi: PhD thesis (1983).
- Testable using **`pc-kimmo`** (freely available at http://www.sil.org/pckimmo/).
- Lauri Karttunen (Xerox Grenoble): two-level compiler, finite state technology, **`xfst`**, see http://www.xrce.xerox.com/.
- Morphological “classics”

Finite-State Automaton

- Five-tuple (A, Q, P, q_0, F).
  - A … finite alphabet of input symbols
  - Q … finite set of states
  - P … transition function (set of rules) $A \times Q \rightarrow Q$.
  - $q_0 \in Q$ … initial state
  - $F \subseteq Q$ … set of terminal states

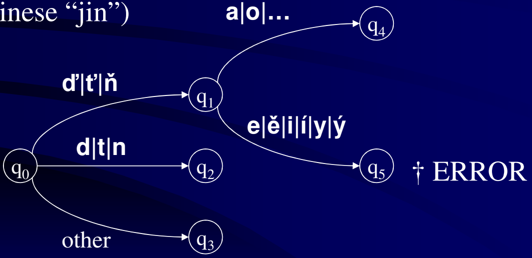
- A word is accepted as correct if we read it as input and we end up in a terminal state.
- An additional action can be bound to the terminal state (output info).

Example of Finite-State Machine

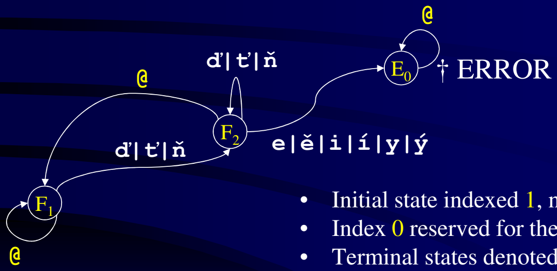
- Checks correct spelling of `cs:` dě, tě, ně…
- Czech orthographical rules:
  - *di, ti, ni* is pronounced *[ďi, ťi, ňi]*
  - *dě, tě, ně* is pronounced *[ďe, ťe, ňe]*
  - Orthography prohibits strings *ďi, ťi, ňi, ďy, ťy, ňy, ďe, ťe, ňe, ďě, ťě, ňě*
  - Note however that long *ďé, ťé* is permitted: these are the names of the letters *Ď, Ť*. (And *ě* cannot be used for them because it is short.)
- Exception: Czech system of transcription of Mandarin Chinese (used for Chinese names in news and encyclopedias):
  - *ťin* … pinyin equivalent is *jin*

Example of Finite-State Machine

- Checks correct spelling of `cs:` dě, tě, ně…
- Ignores official exceptions (“ťin” … Czech transcription of Chinese “jin”)



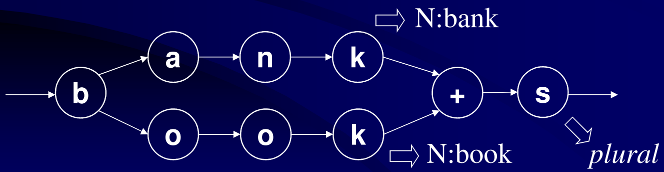
Example of Finite-State Machine (polished, new notation)



- Initial state indexed 1, not 0 (here F$_1$).
- Index 0 reserved for the error state.
- Terminal states denoted by the letter F.
- At sign (“@”) means “other”, i.e. characters not found on other transitions with the same start.

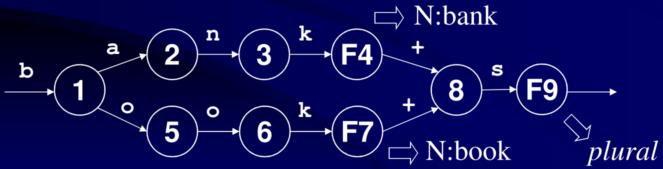
Lexicon

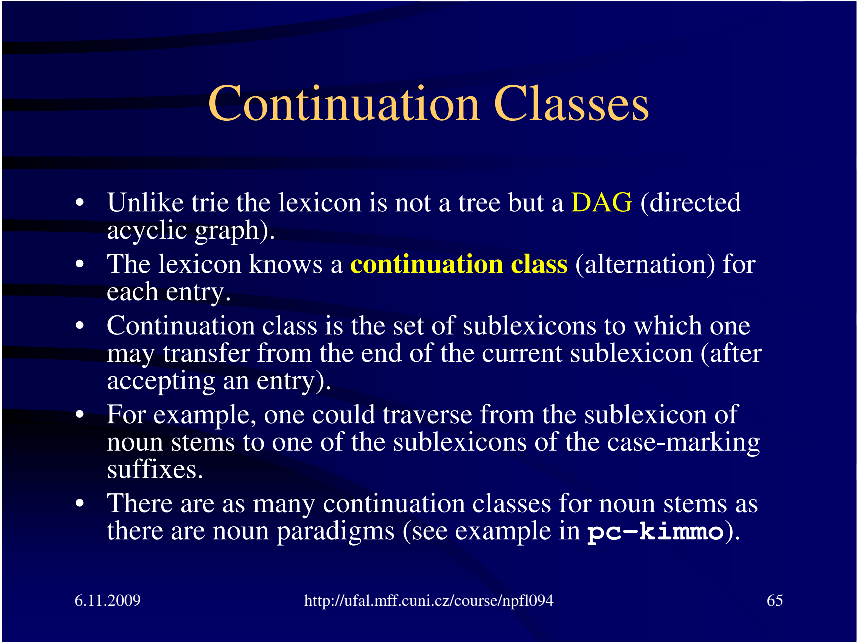
- Implemented as a finite-state automaton (trie) *[tri:]*.
- Compiled from a list of strings and sublexicon references.
- Sublexicons for stems, prefixes, suffixes.
- Notes (glosses) at the end of every sublexicon.
- Example: (edges labeled same way as nodes they lead to)



Lexicon

- Implemented as a finite-state automaton (trie) *[tri:]*.
- Compiled from a list of strings and sublexicon references.
- Sublexicons for stems, prefixes, suffixes.
- Notes (glosses) at the end of every sublexicon.
- Example: (edges labeled same way as nodes they lead to)





Continuation Classes

- Unlike trie the lexicon is not a tree but a DAG (directed acyclic graph).
- The lexicon knows a **continuation class** (alternation) for each entry.
- Continuation class is the set of sublexicons to which one may transfer from the end of the current sublexicon (after accepting an entry).
- For example, one could traverse from the sublexicon of noun stems to one of the sublexicons of the case-marking suffixes.
- There are as many continuation classes for noun stems as there are noun paradigms (see example in **`pc-kimmo`**).

Examples of Lexicons

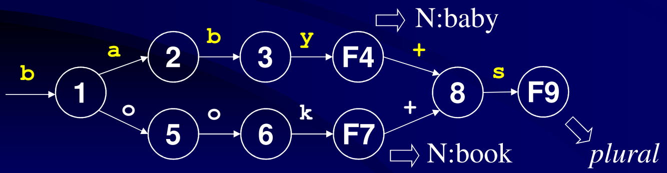
- English noun stems (typically whole words at the same time): *book, bank, car, cat, donut…*
- See also **`pc-kimmo / englex`**.
- Czech stems (not always a whole lemma): *pán, hrad, muž, stroj, (před)sed, soudc, žen, růž, píseň, kost, měst, moř, kuř, staven*
- Czech prefixes: *do, na, od, po, pře, před, při, se, z, za…*
  odpo, dopři, pona… *nej, ne* *dvoj, troj…*

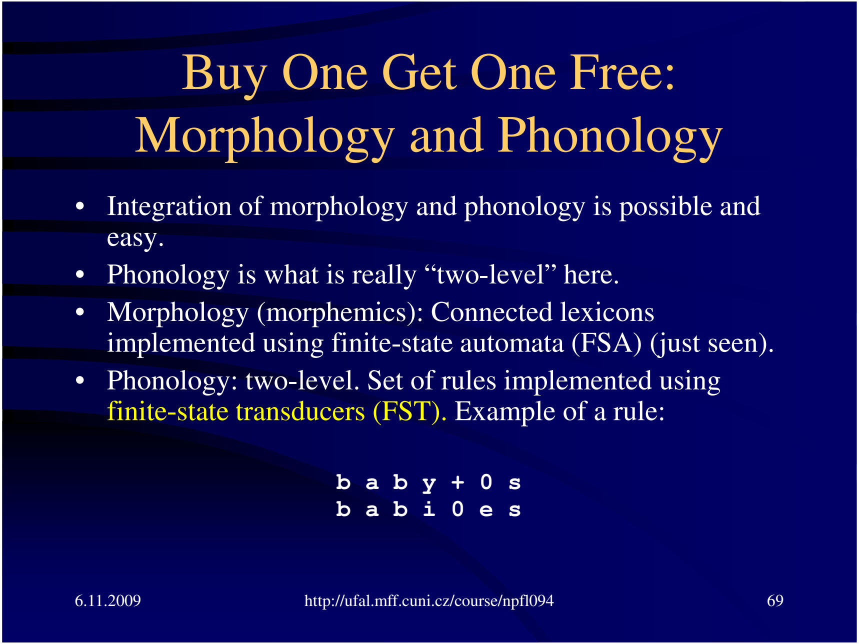
Examples of Lexicons

- Suffixes of Czech nouns
  - *0, a, e, u, ovi, i, o, em, ou, i, ové, y, ů, ům, ech, ích*
  - *a, e, 0, y, i, u, o, ou, í, ám, ím, em, ách, ích, ech, ami, emi, mi*
  - *o, e, í, a, ete, u, i, eti, em, etem, ím, ata, 0, at, ům, ím, atům, ech, ích…*
- Suffixes of Czech adjectives
  - *ý, ého, ému, ým, í, ých, é, ými, á, ou, ém*
  - *í, ího, ímu, ím, ích, ími*
  - *(ej+, ěj+) š + í, ího, …*
- Suffixes of Czech verbs
  - *(n+) u, eš, e, eme, ete, ou*
    ím, íš, í, íme, íte, í
    ám, áš, á, áme, áte, ají
  - *(e+, u+) (j+) 0, me, te*
  - *l, en, t*
    - *0, a, o, i, y, y, a*

A Problem Called Phonology

- Sometimes attaching a suffix causes phoneme or grapheme (spelling) changes!
  - For simplicity I will call both *phonology*.
- Plural of *baby* is not **babys* but *babies*!





Buy One Get One Free: Morphology and Phonology

- Integration of morphology and phonology is possible and easy.
- Phonology is what is really “two-level” here.
- Morphology (morphemics): Connected lexicons implemented using finite-state automata (FSA) (just seen).
- Phonology: two-level. Set of rules implemented using finite-state transducers (FST). Example of a rule:

```
b a b y + 0 s
b a b i 0 e s
```

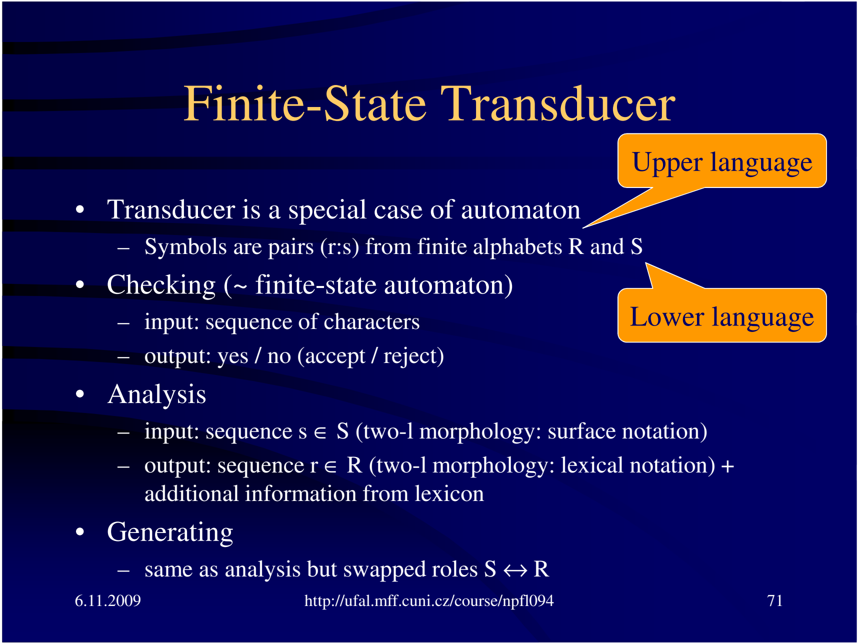
Two-Level Rules

- **Upper and lower language**
  - Upper is also called **lexical**.
  - Lower is also called **surface**.
- Two-line notation is encoded using colons:

```
      b a b y + 0 s
      b a b i 0 e s

b:b a:a b:b y:i +:0 0:e s:s
```

- The + character usually denotes morpheme boundary.
- The 0 character usually denotes an empty position (its counterpart has no realization on this level).
- Other special characters of PC-Kimmo: #, @.



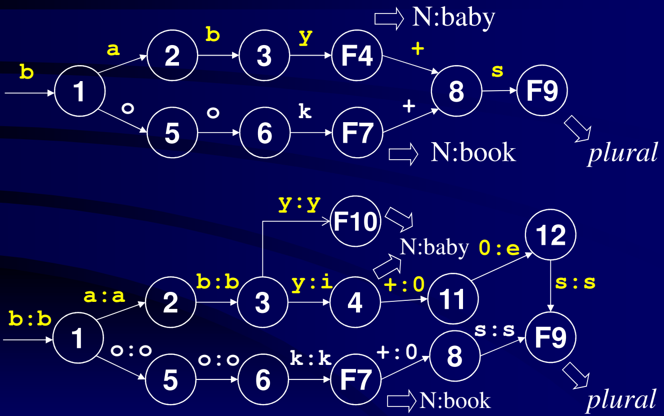
Finite-State Transducer

Upper language

- Transducer is a special case of automaton
  - Symbols are pairs (r:s) from finite alphabets R and S
- Checking (~ finite-state automaton)
  - input: sequence of characters
  - output: yes / no (accept / reject)
- Analysis
  - input: sequence s ∈ S (two-l morphology: surface notation)
  - output: sequence r ∈ R (two-l morphology: lexical notation) + additional information from lexicon
- Generating
  - same as analysis but swapped roles S ↔ R

Lower language

Automaton vs. Transducer



Another Way of Rule Notation: Two-Level Grammar

- If lexical *y* is followed by +*s*, then on surface the *y* must be replaced by *i*.

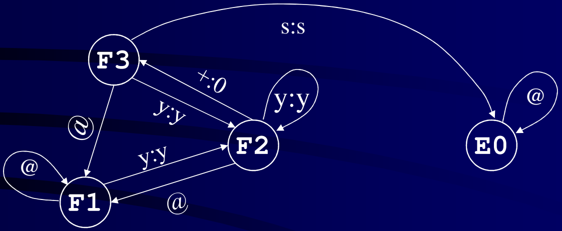
  ```
  y:i <= _ +:0 s:s
  ```

  - We don’t require the reverse implication this time. It is possible that *y* is changed to *i* elsewhere for other reasons.
- At the same time we require that in the same context an *e* is inserted before *s*:

  ```
  0:e <= y:i +:0 _ s:s
  ```

- Create finite-state transducer that converts the lexical layer to the surface one according to the rules.
  - More precisely: a transducer is an automaton that only *checks* that we are converting the layers correctly.

Example of Transducer: *baby+s*

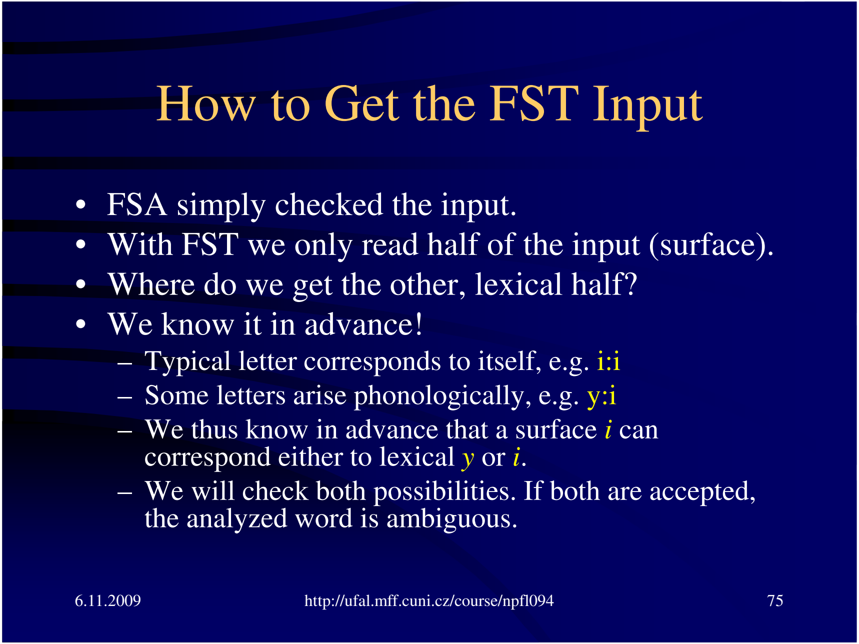


`y:i <= _ +:0 s:s`

N:
non-terminal state

F:
terminal state

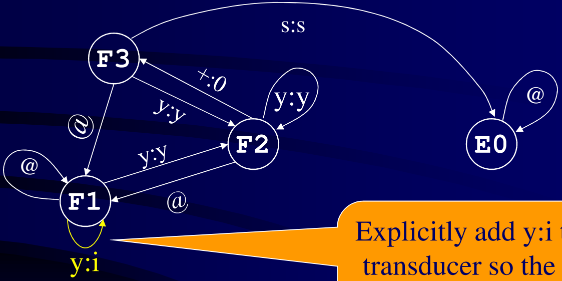
E:
error state



How to Get the FST Input

- FSA simply checked the input.
- With FST we only read half of the input (surface).
- Where do we get the other, lexical half?
- We know it in advance!
  - Typical letter corresponds to itself, e.g. i:i
  - Some letters arise phonologically, e.g. y:i
  - We thus know in advance that a surface *i* can correspond either to lexical *y* or *i*.
  - We will check both possibilities. If both are accepted, the analyzed word is ambiguous.

Example of Transducer: *baby+s*



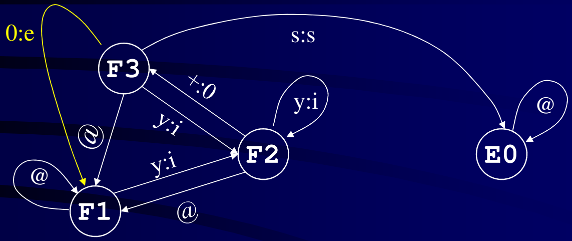
N: non-terminal

Explicitly add y:i to some transducer so the system knows about the the possibility.

error state

`y:i <= _ +:0 s:s`

Example of Transducer: *baby+s*

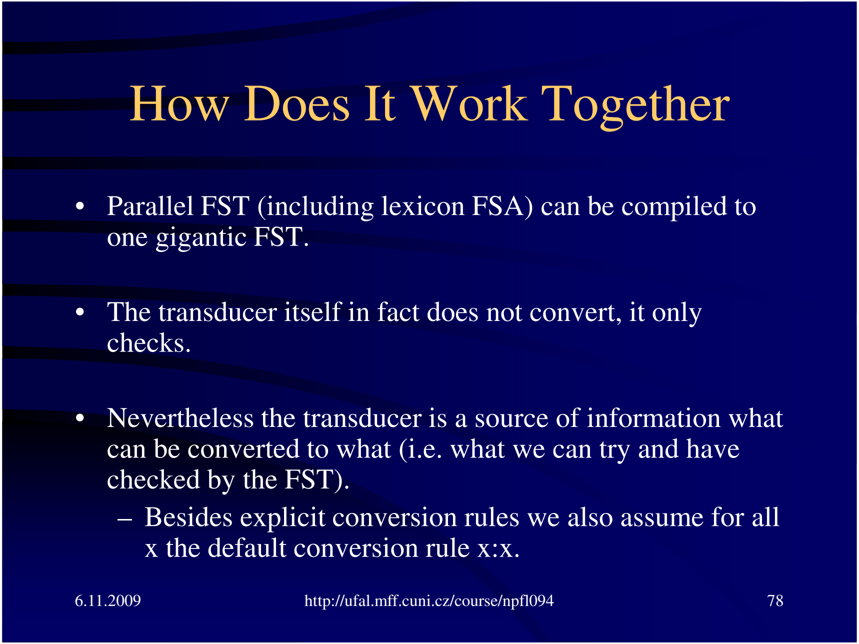


0:e <= y:i +:0 _ s:s

N:
non-terminal
state

F:
terminal
state

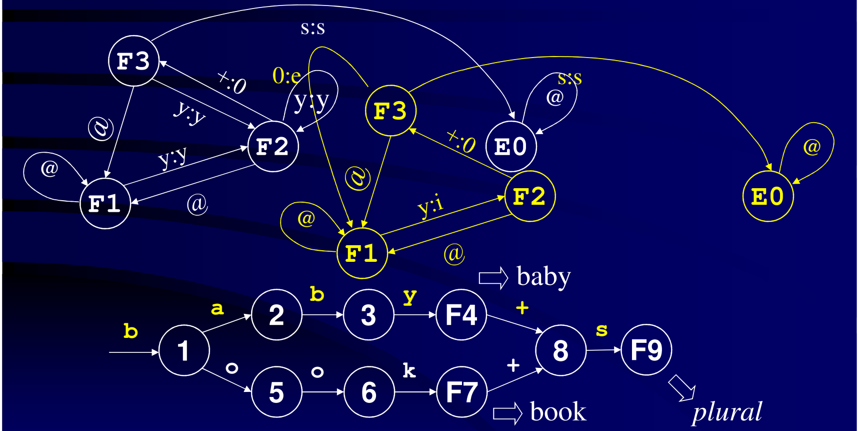
E:
error state

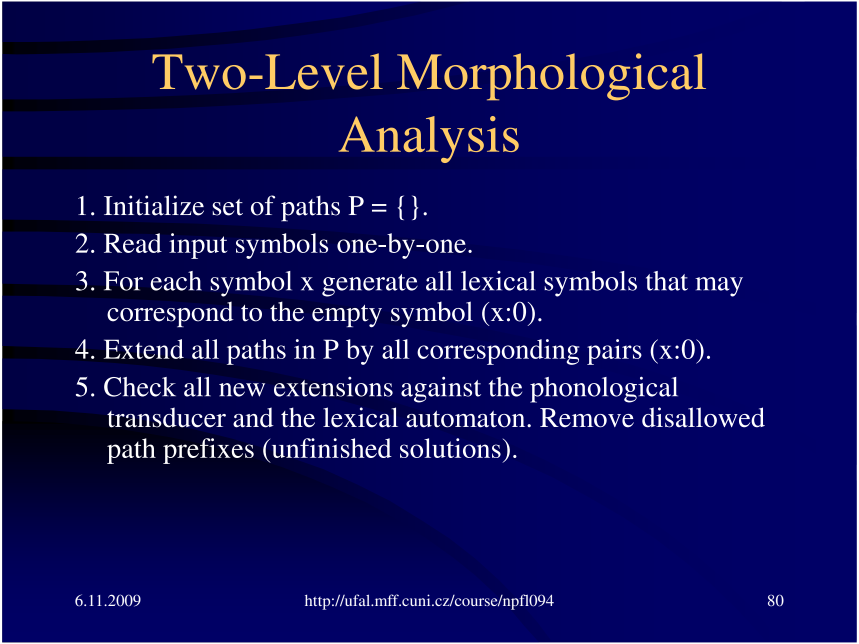


How Does It Work Together

- Parallel FST (including lexicon FSA) can be compiled to one gigantic FST.
- The transducer itself in fact does not convert, it only checks.
- Nevertheless the transducer is a source of information what can be converted to what (i.e. what we can try and have checked by the FST).
  - Besides explicit conversion rules we also assume for all x the default conversion rule x:x.

Lexicon and Rules Together





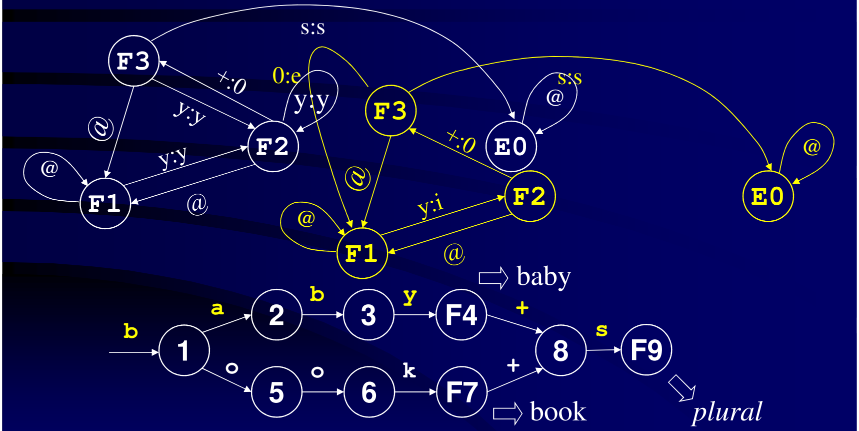
Two-Level Morphological Analysis

1. Initialize set of paths P = { }.
2. Read input symbols one-by-one.
3. For each symbol x generate all lexical symbols that may correspond to the empty symbol (x:0).
4. Extend all paths in P by all corresponding pairs (x:0).
5. Check all new extensions against the phonological transducer and the lexical automaton. Remove disallowed path prefixes (unfinished solutions).

Two-Level Morphological Analysis

6. Repeat 4–5 until the maximum possible number of subsequent zeroes is reached.
7. Generate all possible lexical symbols (of all transducers) for the current symbol. Create pairs.
8. Extend each path in P by all such pairs.
9. Check all paths in P (the next transition in FST/FSA). Remove all impossible paths.
10. Repeat since step 3 until input finishes.
11. Collect glosses from the lexicon from all paths that survived.

Algorithm Example



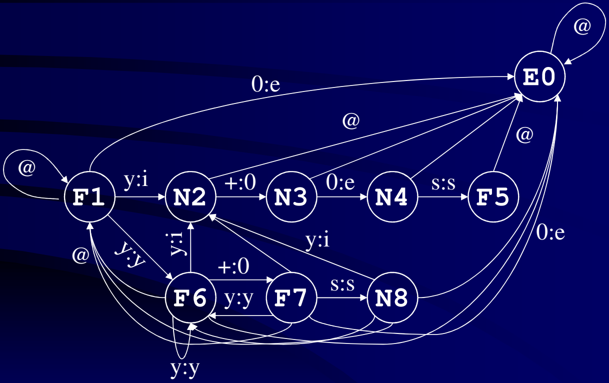
`0:e <=> y:i +:0 _ s:s`

Algorithm Example

- Every letter corresponds to itself
- In addition: `y:i`, `+:0`, `0:e`
- Input: *babies*
- Try inserting lexical + (`+:0`) … blocked by lexicon (no word starts like that)
- Try `b:b` … OK (neither lexicon nor the transducers object)
- `b:b +:0` … lexicon error
- `b:b a:a` … OK
- `b:b a:a +:0` … lexicon error
- `b:b a:a b:b` … OK
- `b:b a:a b:b +:0` … l. error
- `b:b a:a b:b i:i` … l. error
  `b:b a:a b:b y:i` … OK

- … `b:b y:i +:0` … OK
  … `b:b y:i +:0 +:0` … error
- … `y:i e:e` … error
  … `y:i 0:e` … OK
  … `y:i +:0 e:e` … error
  … `y:i +:0 0:e` … OK
- … `0:e +:0` … OK
  … `0:e +:0 +:0` … error
  … `+:0 0:e +:0` … error
- … `0:e s:s` … error
  … `+:0 0:e s:s` … OK
  … `0:e +:0 s:s` … OK
- … `+:0 0:e s:s +:0` … error
  … `0:e +:0 s:s +:0` … error
- One of the hypotheses could be blocked by our FSTs if we designed them better (⇔)

Example of Transducer: *baby+s*



Czech Examples

- Joining stem with suffix may for instance bring together ď and e that normally cannot occur together. *(káď = tun)*

```
k á ď + e
k á ď 0 e
```

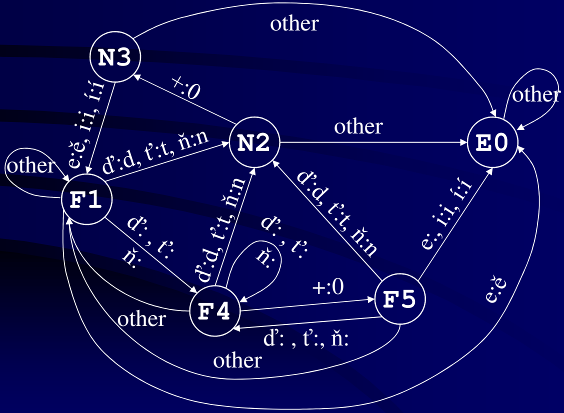
- We need a rule for such cases that will ensure the correct conversion ďe ⟶ dě.

```
k á ď + e
k á d 0 ě
```

Example of Transducer: ď, ť, ň on morpheme boundary

- `ď:d +:0 e:ě` is correct, other possibilities are not.
- Assumption: ďe, ďi could only occur on morpheme boundary (other positions are in the lexicon and should be correct).
- We don't cover ďě. The character ě can appear in the suffix only because of a phonological change, not otherwise:
  - (brzda brzďe, žena žeňe, máta máťe, máma máměe, bába bábje, matka matce, váha váze, sprcha sprše, kůra kůře, mula mule, vosa vose, lůza lůze)
- We further don't cover ďy (which could arise by application of the inflection paradigm to a noun ending in –ďa; it is incorrect and should be changed to –di).

Example of Transducer: ď, ť, ň on morpheme boundary

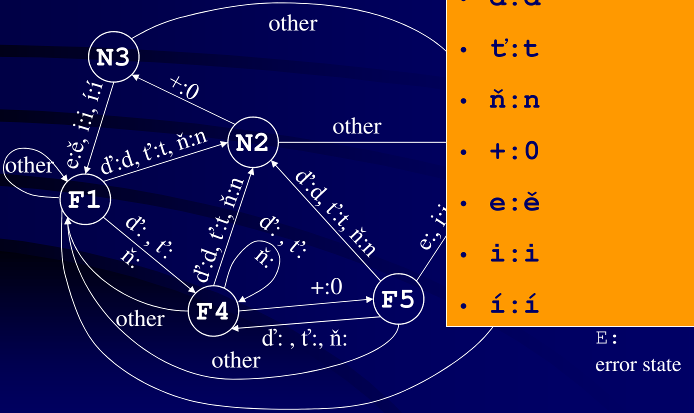


N: non-terminal state

F: terminal state

E: error state

Example of Transducer:
ď, ť, ň on morpheme

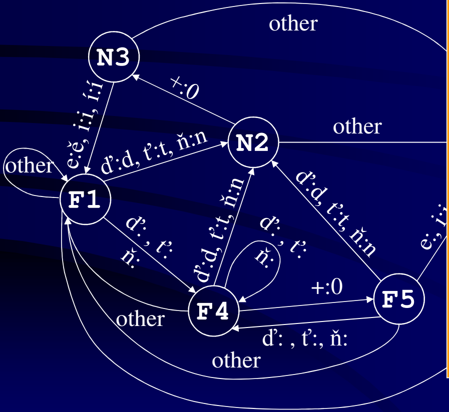


Possible conversions:

- ď:d
- ť:t
- ň:n
- +:0
- e:ě
- i:i
- í:í

E:
error state

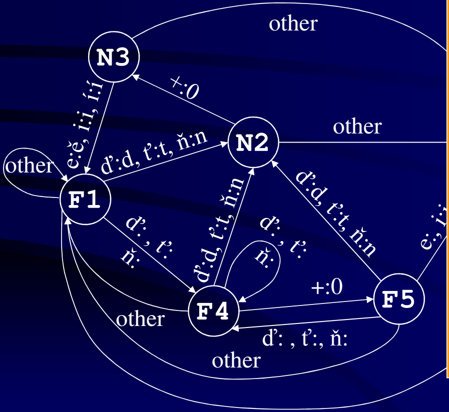
Example of Transducer: ď, ť, ň on morpheme



Possible conversions:

- ď:d @
- ť:t @
- ň:n @
- +:0
- e:ě @
- i:i
- í:í
- @:@

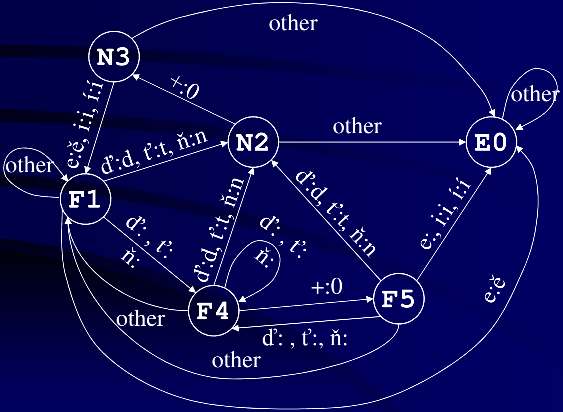
Example of Transducer: ď, ť, ň on morpheme



Add **alphabet:**

- **ď:d ď**
- **ť:t ť**
- **ň:n ň**
- **+:0**
- **e:ě e**
- **i:i**
- **í:í**
- **x:x …**

Example of Transducer: ď, ť, ň on morpheme boundary



N:
non-terminal
state

F:
terminal
state

E:
error state

Transducer Encoding in a Matrix

```
RULE "[ď:d | ň:n | ť:t] <=> _ +:0 [e:ě | i:i
                                    | í:í]" 5 12
```

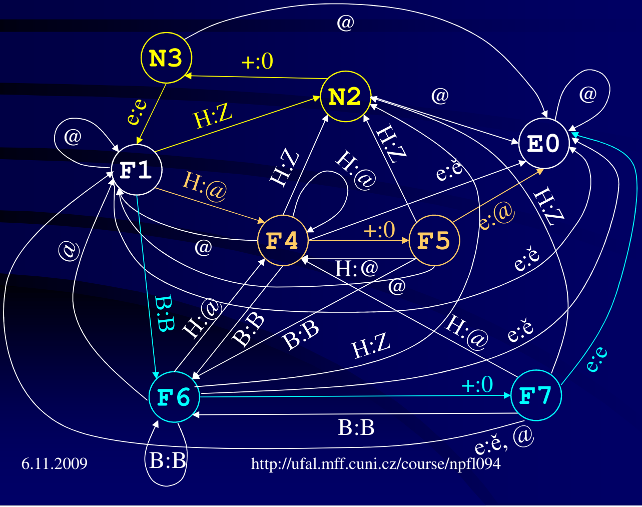
```
    ď ň ť ď ň ť + e i í e @
    d n t @ @ @ 0 ě i í @ @
1:  2 2 2 4 4 4 1 0 1 1 1 1
2.  0 0 0 0 0 0 3 0 0 0 0 0
3.  0 0 0 0 0 0 0 1 1 1 0 0
4:  2 2 2 4 4 4 5 1 1 1 1 1
5:  2 2 2 4 4 4 1 0 0 0 0 1
```

The pairs illustrate various stem-final changes in the paradigm *žena* of Czech feminine nouns. All words are **surface** strings—nominative singular on the left, dative singular on the right.

- váha – váze
- sprcha – sprše
- matka – matce
- kůra – kůře
- Olga – Olze
- vláda – vládě
- máta – mátě
- žena – ženě

- bába – bábě
- karafa – karafě
- máma – mámě
- chrpa – chrpě
- jíva – jívě
- Naďa – Nadě
- Jíťa – Jítě
- Áňa – Áně

Palatalization *žena – ženě*



H:Z = g:z |
h:z | ch:š |
k:c | r:ř

B:B = b:b |
f:f | m:m |
p:p | v:v |
w:w | q:q |
d:d | t:t | n:n
| ď:d | ť:t |
ň:n

Examples of Two-Level Rules in Czech

- Palatalization of stem-final consonants.

```
m a t E K + e
m a t 0 c 0 e
```

- Epenthesis: inserting or deleting of *e*.

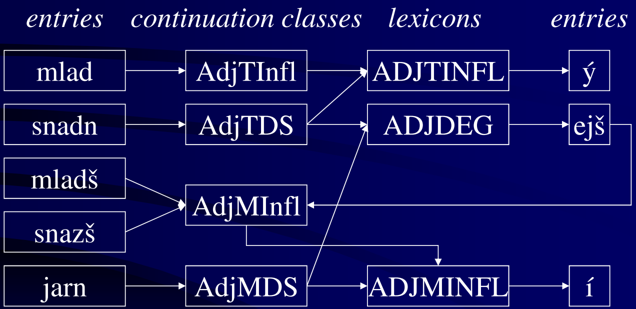
```
m a t E K
m a t e k
```

- Transitions among present, past and infinitival verbal stems.
  - Palatalization of stem-final consonant in imperative.

PC Kimmo: Czech Adjectives

- Two inflection classes:
  - Hard: *černý (black), černého, černému…, černá* [fem], *černé…*
  - Soft: *jarní (spring), jarního, jarnímu…, jarní* [fem], *jarní…*
- Regular comparative:
  - Suffix +*ejš*
  - Comparative is always soft regardless the original class: *černější, černějšího… jarnější, jarnějšího…*
- Irregular comparatives: *mladý (young)* ⇒ *mladší (younger); snadný (easy)* ⇒ *snadnější* | *snazší (easier)*
- Superlative: *nej* + comparative, e.g. *nejmladší (youngest)*

PC Kimmo: Czech Adjectives



entries *continuation classes* *lexicons* *entries*

mlad → AdjTInfl → ADJTINFL → ý

snadn → AdjTDS → ADJTINFL, ADJDEG

ADJDEG → ejš → AdjMInfl

mladš → AdjMInfl

snazš → AdjMInfl

AdjMInfl → ADJMINFL

jarn → AdjMDS → ADJDEG, ADJMINFL

ADJMINFL → í

Long-Distance Dependencies

- Disadvantage of TLM:
  - Capturing of long-distance dependencies is clumsy!

Example from German

- German umlauts (simplified):

u ↔ ü if (not only if) followed by c h e r (Buch → Bücher)

pravidlo: **u:ü** ⇐

_ c:c h:h e:e r:r

FST:

Buch:

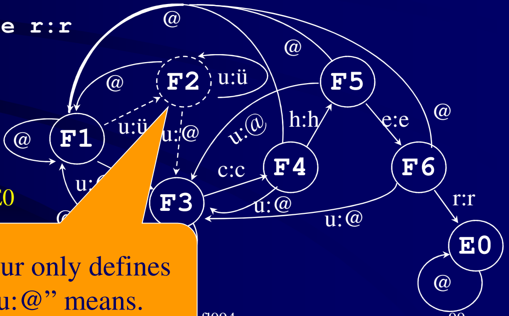
F1 F3 F4 F5

Bucher:

F1 F3 F4 F5 F6 E0

Buck:

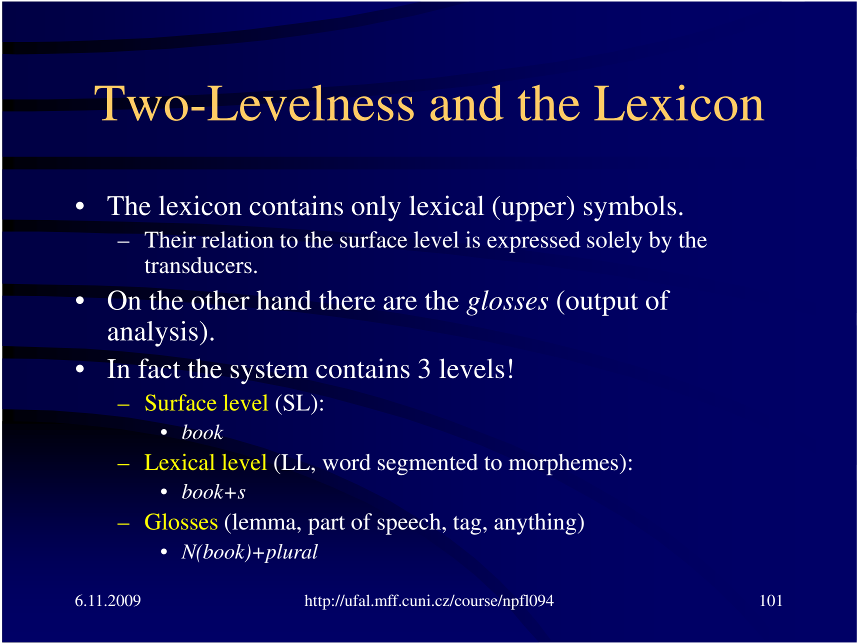
F1 I



This detour only defines what “u:@” means.

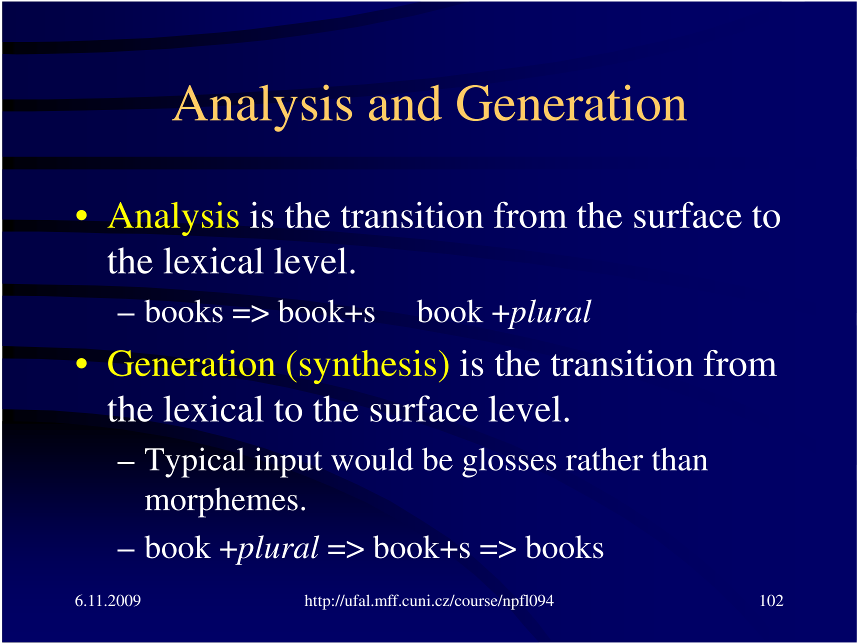
Example from German

- *Buch / Bücher, Dach / Dächer, Loch / Löcher*
- Context should also contain +:0 and perhaps test end of word (#)
  - Otherwise *Sucherei* (searching) will be considered wrong!
  - Not only must we recognize that there is a suffix. It must be a plural suffix and the stem must be marked for plural umlauting.
  - Counterexamples:
    - *Kocher* (cooker), here the *er* suffix only derives from the verb *kochen* (to cook). *Kocher* is identical in singular and plural! We don't want to confuse it with *Köcher* (quiver), nor to consider umlaut-less *Kocher* an error!
    - *Besucher* (visitor), derived from *Besuch* (visit), same singular and plural, there is no **Besücher*!
- Capturing long-distance dependencies is clumsy.
  - E.g. *Kraut / Kräuter* has different intervening symbols so it looks like a different rule.
  - A transducer could be more general and allow anything until +*er* but would it overgenerate?



Two-Levelness and the Lexicon

- The lexicon contains only lexical (upper) symbols.
  - Their relation to the surface level is expressed solely by the transducers.
- On the other hand there are the *glosses* (output of analysis).
- In fact the system contains 3 levels!
  - Surface level (SL):
    - *book*
  - Lexical level (LL, word segmented to morphemes):
    - *book+s*
  - Glosses (lemma, part of speech, tag, anything)
    - *N(book)+plural*

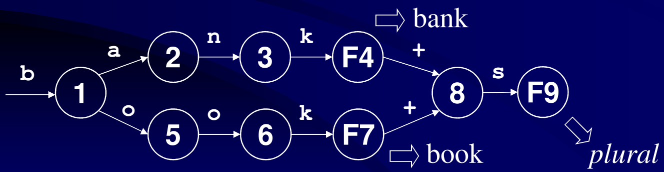


Analysis and Generation

- Analysis is the transition from the surface to the lexical level.
  - books => book+s   book +*plural*
- Generation (synthesis) is the transition from the lexical to the surface level.
  - Typical input would be glosses rather than morphemes.
  - book +*plural* => book+s => books

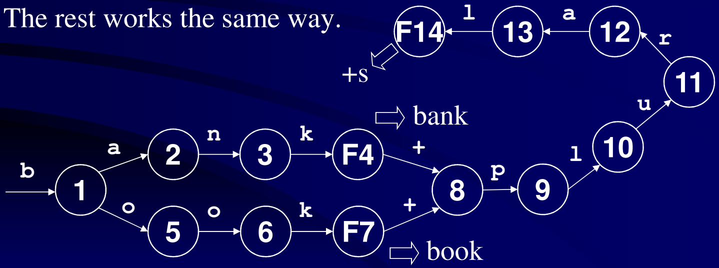
Lexicon for Analysis

- Implemented as FSA (trie).
- Compiled from a list of strings and inter-lexicon links.
- Sublexicons for stems, prefixes, suffixes.
- Notes (glosses) at the end of each sublexicon.



Lexicon for Generation

- Swap surface and lexical levels (glosses).
- Again, it can be automatically compiled from the same list as the lexicon for analysis.
- The rest works the same way.

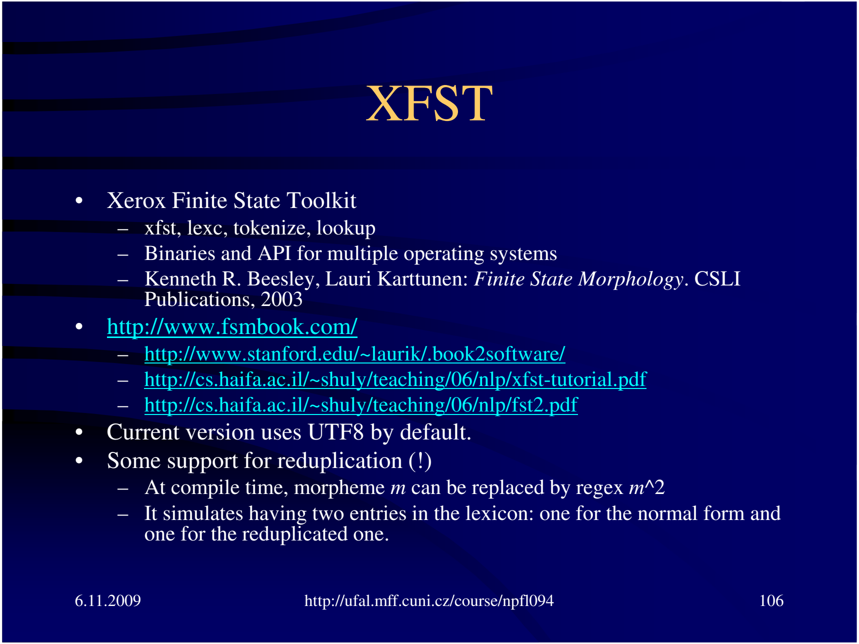


Morphological and Syntactic Analysis

Multi-Level Finite State Rules

Daniel Zeman

http://ufal.mff.cuni.cz/~zeman/



XFST

- Xerox Finite State Toolkit
  - xfst, lexc, tokenize, lookup
  - Binaries and API for multiple operating systems
  - Kenneth R. Beesley, Lauri Karttunen: *Finite State Morphology*. CSLI Publications, 2003
- http://www.fsmbook.com/
  - http://www.stanford.edu/~laurik/.book2software/
  - http://cs.haifa.ac.il/~shuly/teaching/06/nlp/xfst-tutorial.pdf
  - http://cs.haifa.ac.il/~shuly/teaching/06/nlp/fst2.pdf
- Current version uses UTF8 by default.
- Some support for reduplication (!)
  - At compile time, morpheme *m* can be replaced by regex *m*^2
  - It simulates having two entries in the lexicon: one for the normal form and one for the reduplicated one.

Foma

- Open-source finite-state toolkit
  - In contrast, xfst comes without sources and with some copyright restrictions
- Claims compatibility with Xerox tools
  - But also supports Perl-style regular expressions
- Now integrated in Apertium (open-source rule-based machine translation framework)
- Home: https://code.google.com/p/foma/
  - Publication: http://www.aclweb.org/anthology-new/E/E09/E09-2008.pdf

Foma vs. Kimmo

- Multiple levels
  - Sequence of ordered rewrite rules
  - Even lexicon supports two levels (TAG:suffix)
- Regular expressions
  - Instead of directly encoding transducers
  - Supports usual FSM algorithms (minimization etc.)
- Sequence of rules still compiled into one FST
  - We still have one upper and one lower language

Compiling Regular Expressions: regex

- `regex a+;`
- `regex c a t | d o g;`
- `regex ?* a ?*;`

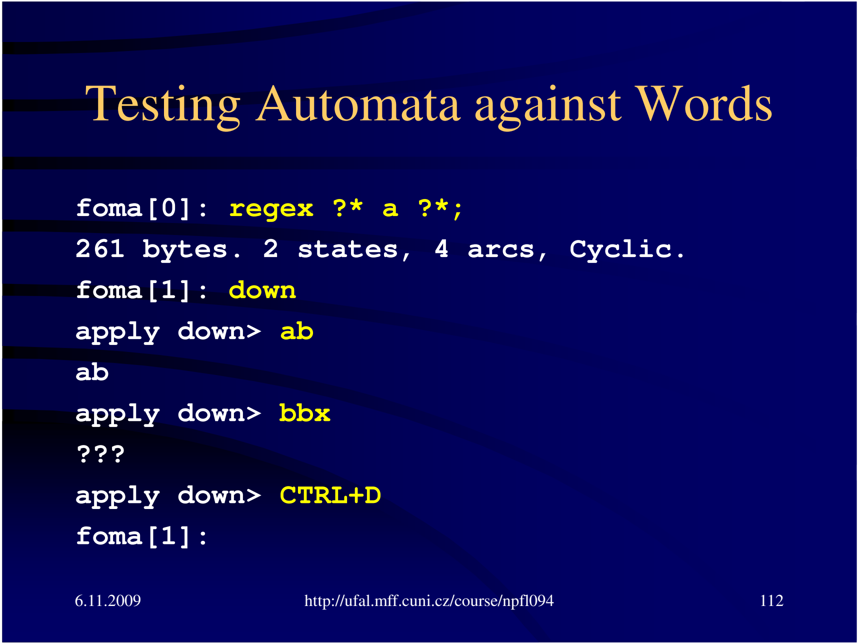
- `regex [a:b | b:a]*;`
- `regex [c a t]:[k a t u a];`
- `regex b -> p, g -> k, d -> t || _ .#.;`

Foma Operators

- (space) … concatenation
- `|` … union
- `*` … Kleene star
- `&` … intersection
- `~` … complement
- Single- and multi-character symbols
  - Supports Unicode
- `0` … empty string (epsilon)
- `?` … any symbol (similar to "." in Perl, grep etc.)
- `( a )` … "a" is optional (as "a?" in Perl)

Difference between Colon “ : ” and Arrow “–>”

- Colon “:” affects a specific position or sequence of positions.
- Regular expressions with colons restrict the set of words that belong to the language.
- Regular expressions with arrows yield transducers that accept any string. If the string contains the searched character, it will be replaced.
- Arrow is implemented with the help of colon.



Testing Automata against Words

```
foma[0]: regex ?* a ?*;
261 bytes. 2 states, 4 arcs, Cyclic.
foma[1]: down
apply down> ab
ab
apply down> bbx
???
apply down> CTRL+D
foma[1]:
```

Labeling FSMs: define

```
foma[0]: define V [a|e|i|o|u];
defined V: 317 bytes. 2 states, 5 arcs, 5
paths.
foma[0]: define StartsWithVowel [V ?*];
defined StartsWithVowel: 429 bytes. 2
states, 11 arcs, Cyclic.
foma[0]:
```

Rewrite Rules

```
foma[0]: regex a -> b;
290 bytes. 1 states, 3 arcs, Cyclic.
foma[1]: down
apply down> a
b
apply down> axa
bxb
apply down> CTRL+D
```

Accepts any input.

Changes *a* to *b*.

Conditional Replacement

```
foma[0]: regex a -> b || c _ d ;
526 bytes. 4 states, 16 arcs, Cyclic.
foma[1]: down cadca
cbdca
foma[1]:
```

Multiple Contexts

```
foma[0]: regex a -> b || c _ d, e _ f;
890 bytes. 7 states, 37 arcs, Cyclic.
foma[1]: down
apply down> cadeaf
cbdebf
apply down> a
a
apply down> CTRL+D
```

Parallel Rules
End-of-Word Symbol

```
foma[0]: regex b -> p, g -> k, d -> t ||
_ .#. ;
634 bytes. 3 states, 20 arcs, Cyclic.
foma[1]: down
apply down> cab
cap
apply down> dog
dok
apply down> dad
dat
```

Composition of Rules

```
foma[0]: define Rule1 a -> b || c _ ;
defined Rule1: 384 bytes. 2 states, 8 arcs, Cyclic.
foma[0]: define Rule2 b -> c || _ d ;
defined Rule2: 416 bytes. 3 states, 10 arcs, Cyclic.
foma[0]: regex Rule1 .o. Rule2;
574 bytes. 4 states, 19 arcs, Cyclic.
foma[1]: down
apply down> cad
ccd
apply down> ca
cb
apply down> ad
ad
```

Review

- **`regex`** regular-expression;
  - compile regular expression and put it on the stack
- **`define`** name regular-expression;
  - name a FST/FSM using regex; do not put it on the stack
- **`view`** (**`view net`**)
  - (Linux only) display the compiled regex from stack graphically in a window
- **`net`** (**`print net`**)
  - textual net description
- **`down`** <word> (apply down)
  - run a lexical word through a transducer (generation)
- **`up`** <word> (apply up)
  - run a surface word through a transducer (analysis)
- **`words`** (print words)
  - print all the words an automaton accepts
- **`lower-words`**
  - only lower side of an FST
- **`upper-words`**
  - only upper side of an FST

Lexicon in lexc Format

- Create the file, then load it to Foma

```
LEXICON Root
cat   Suff;
dog   Suff;
horse Suff;

LEXICON Suff
s #;
  #;
```

Load Lexicon to Foma

```
foma[0]: read lexc simple.lexc
Root…3, Suff…2
Building lexicon…Determinizing…Minimizing…Done!
575 bytes. 13 states, 15 arcs, 8 paths.
foma[1]: print words
horse horses dog dogs cat cats
foma[1]: define Lexicon;
```

Or alternatively:

```
foma[0]: define Lexicon [c a t|d o g|…] (s);
```

Example English lexc File

```
Multichar_Symbols
+N +V +PastPart
+Past +PresPart +3P
+Sg +Pl
LEXICON Root
Noun ;
Verb ;
LEXICON Noun
cat  Ninf;
city Ninf;
```

- `LEXICON Ninf`
- `+N+Sg:0  #;`
- `+N+Pl:^s #;`

  ! ^ is our morpheme boundary

Put It All Together

- Lexical string = `city+N+Pl`
- Lexicon transducer: `city+N+Pl` → `city^s`
- *y* → *ie* rule: `city^s` → `citie^s`
- Remove ^: `citie^s` → `cities`
- Surface string = `cities`

Put It All Together

```
foma[0]: read lexc english.lexc
foma[1]: define Lexicon;
foma[0]: define YRepl y -> i e || _ “^”
s;
foma[0]: define Cleanup “^” -> 0;
foma[0]: regex Lexicon .o. YRepl .o.
Cleanup;
foma[1]: lower-words
cat cats city cities …
```

Irregular Forms

```
LEXICON Verb
beg Vinf;
make+V+PastPart:made #;  ! bypass Vinf
make+V #;
…
```

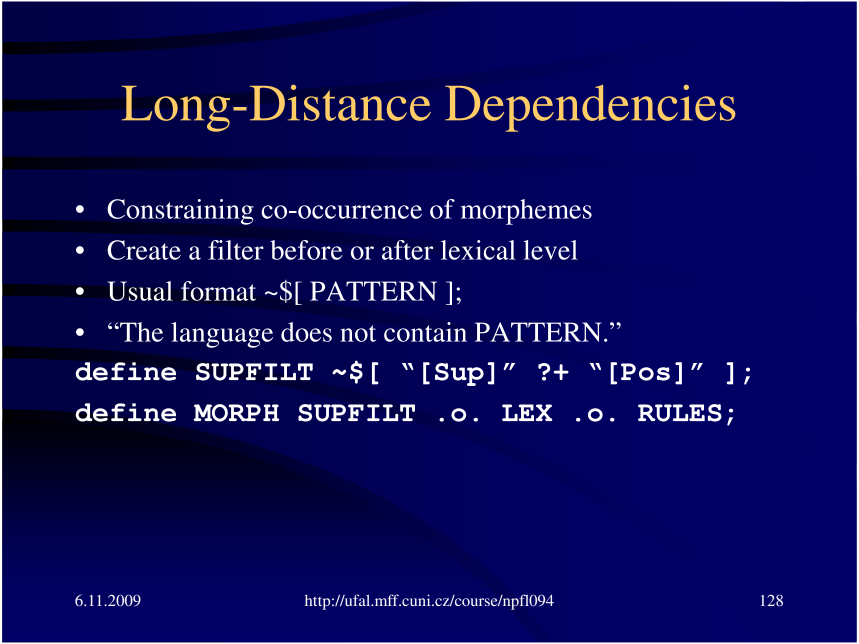
Priority Union

```
foma[1]: define Grammar;
foma[0]: define Exceptions [m a k e “+V”
“+PastPart”]:[m a d e];
foma[0]: regex [Exceptions .P. Grammar];
foma[1]: down
apply down> make+V+PastPart
made
apply down> CTRL+D
```

Alternate Forms

- English: *cactus*+N+Pl → *cactuses*, *cacti*

```
foma[0]: define Parallel [c a c t u s
"+N" "+Pl"]:[c a c t i];
foma[1]: regex Parallel | Grammar;
…
```



Long-Distance Dependencies

- Constraining co-occurrence of morphemes
- Create a filter before or after lexical level
- Usual format ~$[PATTERN];
- “The language does not contain PATTERN.”

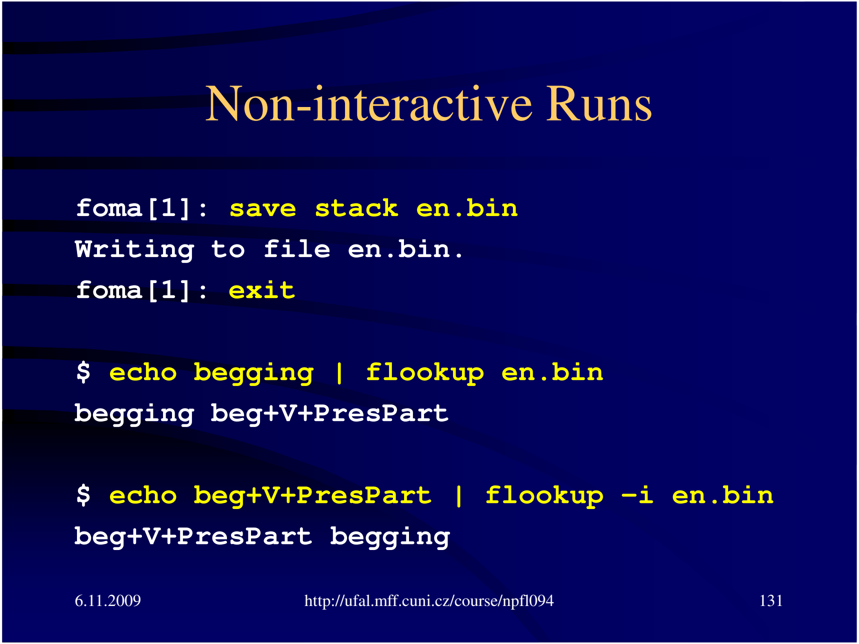
```
define SUPFILT ~$[ “[Sup]” ?+ “[Pos]” ];
define MORPH SUPFILT .o. LEX .o. RULES;
```

Flag Diacritics

- Invisible symbols to control co-occurrence:
  - U … unify features @U.*feature.value*@
  - P … positive set @P.*feature.value*@
  - N … negate @N.*feature.value*@
  - R … require feat/val @R.*feature(.value)*@
  - D … disallow feat/val @D.*feature(.value)*@
  - C … clear feature @C.*feature*@
  - E … require equal feat/val @E.*feature.value*@

Flag Diacritics
to Control Czech Superlatives

- `Multichar_Symbols Sup+ +Pos +Comp`
  `@P.SUP.ON@ @D.SUP@`
- `LEXICON AdjSup`
  `@P.SUP.ON@Sup+:@P.SUP.ON@nej^ Adj;`
- `LEXICON AhardDeg`
  `@D.SUP@+Pos:@D.SUP@ Ahard;`
  `+Comp:^ejš Asoft;`

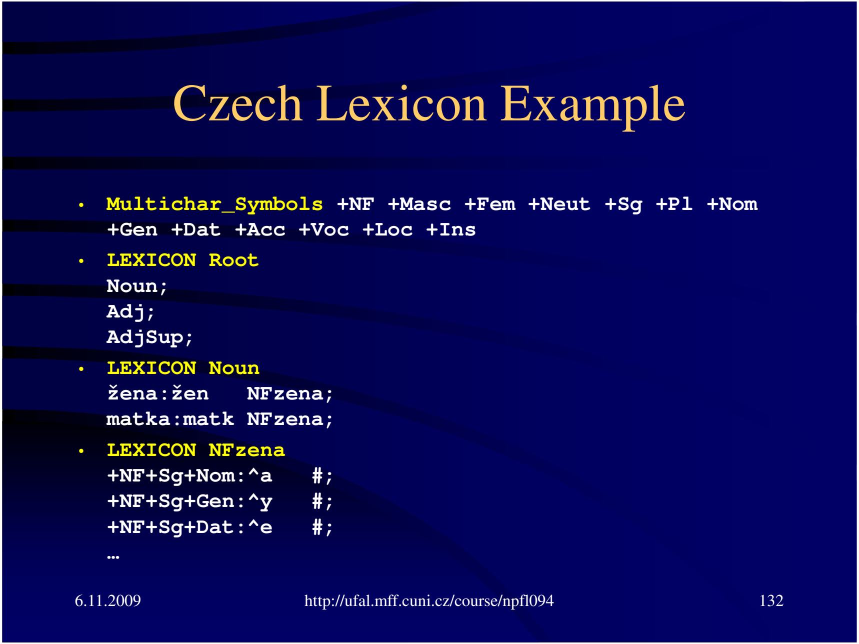


Non-interactive Runs

```
foma[1]: save stack en.bin
Writing to file en.bin.
foma[1]: exit

$ echo begging | flookup en.bin
begging beg+V+PresPart

$ echo beg+V+PresPart | flookup -i en.bin
beg+V+PresPart begging
```



Czech Lexicon Example

- Multichar_Symbols +NF +Masc +Fem +Neut +Sg +Pl +Nom +Gen +Dat +Acc +Voc +Loc +Ins
- LEXICON Root
  ```
  Noun;
  Adj;
  AdjSup;
  ```
- LEXICON Noun
  ```
  žena:žen   NFzena;
  matka:matk NFzena;
  ```
- LEXICON NFzena
  ```
  +NF+Sg+Nom:^a   #;
  +NF+Sg+Gen:^y   #;
  +NF+Sg+Dat:^e   #;
  …
  ```

Czech Rules Example

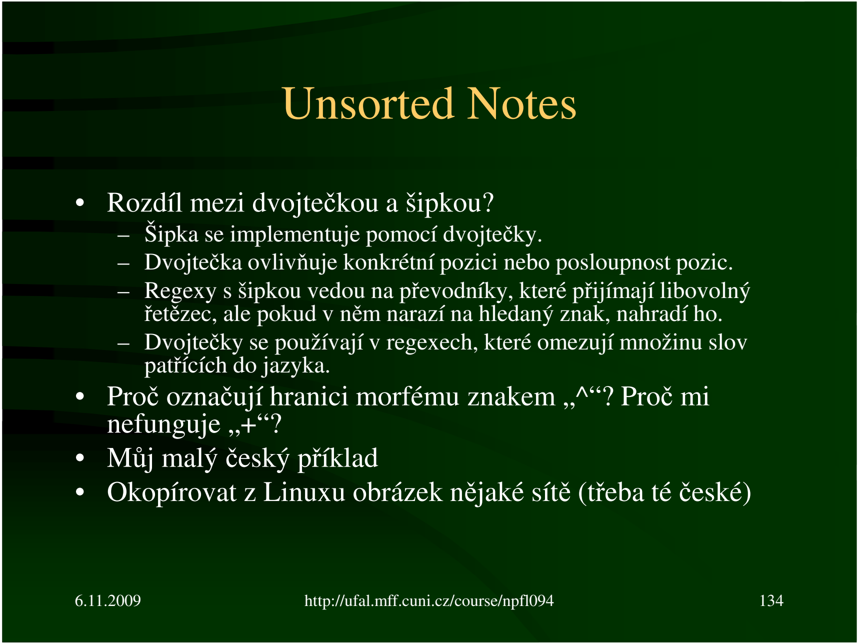
-
  ```
  # matk + ^0 --> matek
  define NFPlGenEInsertion [t k]->[t e k] || _ "^" λ;
  ```

-
  ```
  # matke -> matce, žene -> žeňe
  define NFSgDatPalatalization k->c, n->ň || _ "^" e;
  ```

-
  ```
  # ďe ťe ňe -> dě tě ně
  define DeTeNe [ď "^" e]->[d "^" ě], [ť "^" e]->[t "^"
  ě], [ň "^" e]->[n "^" ě];
  ```

-
  ```
  # Finally erase temporary symbols.
  define Surface "^" -> 0, λ -> 0;
  ```

-
  ```
  read lexc cs.lexc
  define Lexicon;
  regex Lexicon .o. NFPlGenEInsertion .o.
  NFSgDatPalatalization .o. DeTeNe .o. Surface;
  ```



Unsorted Notes

- Rozdíl mezi dvojtečkou a šipkou?
  - Šipka se implementuje pomocí dvojtečky.
  - Dvojtečka ovlivňuje konkrétní pozici nebo posloupnost pozic.
  - Regexy s šipkou vedou na převodníky, které přijímají libovolný řetězec, ale pokud v něm narazí na hledaný znak, nahradí ho.
  - Dvojtečky se používají v regexech, které omezují množinu slov patřících do jazyka.
- Proč označují hranici morfému znakem „^“? Proč mi nefunguje „+“?
- Můj malý český příklad
- Okopírovat z Linuxu obrázek nějaké sítě (třeba té české)