Computational Morphology
LIN 650
Course Review

Jeffrey Heinz

December 10, 2018

This course

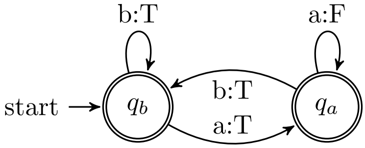
1. Theory (regular grammars)
2. Application to Morphology (Roark and Sproat, chapters 1-3)
3. Programming with Pynini
4. Special Topics

Regular grammars for sets and transformations

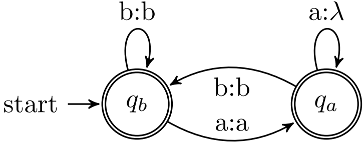
1. Regular expressions
2. Finite-state machines
3. Monadic Second Order (MSO)-definability

Kleene 1956, Scott and Rabin 1959, Büchi 1960, Engelfriedt and Hoogeboom 2001

Computing Functions on Strings: Paths in a Machine

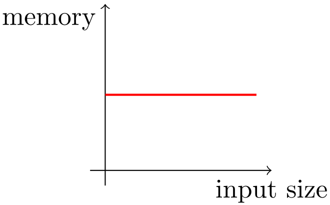


Computing Functions on Strings: Paths in a Machine

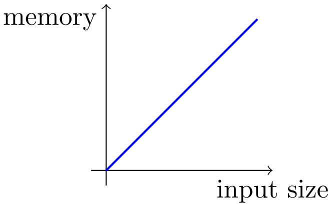


What "Regular" means

A set, relation, or function is regular provided **the memory required for the computation is bounded by a constant,** ***regardless of the size of the input.***



Regular



Non-regular

Product Constructions

We can *multiply* two machines together to create new machines.

- Intersection (sets)
- Union (sets)
- Composition (relations)

Closure operations

Properties	Languages	Relations
concatenation	yes	yes
Kleene star	yes	yes
union	yes	yes
intersection	yes	no
difference	yes	no
composition	–	yes
inversion	–	yes

Application to Morphology

- Morphemes are functions that transform strings.
- Application of a morpheme to a lexical item can be computed via composition.
- More generally, composition lets one build a large lexicon by applying a set of morphemes to an atomic lexicon and iterating.
- Unpredictable forms are managed by removing them from the domain of the predictable morphology, listing their transformations, and adding them back in.

Morphological Theory

1. Roark and Sproat claim that the distinctions between lexical-incremental and realizational-functional dissolve in the light of computational analysis.
2. However, we saw that realizational-functional can encompass lexical-incremental, it was not clear how lexical-incremental can handle cases of multiple exponence.

Programming with Pynini

- Installed from source!
- Became familiar with Pynini's syntax and operations to write scripts which compute morpho-phonologies.
- Studied rule application and how `cdrewrite` builds a transducer.
- Because it is a Python library, the sky is the limit.

Special Topic: Reduplication with 2DFTs

RedTyp: `https://github.com/jhdeov/RedTyp`

- SQL database of reduplicative processes
- Modeled 138 reduplicative processes across 90 languages using 57 2-way FSTs
- Average number of states: 8.8
- Largest number of states: 30
(1000s for 1-way FSTs)

Hossep Dolatian

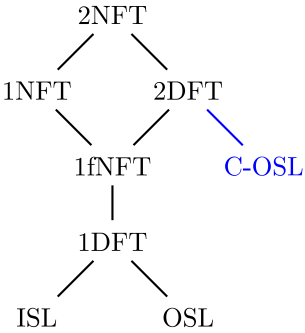
Contributions

1. 2-way FSTs can model virtually all reduplication patterns.
2. ~87% belongs to a subclass which can be described as the "Concatenation of two OSL functions" (C-OSL).
3. Simple learning algorithm for C-OSL which uses OSLFIA but also a boundary-enriched sample.

Dolatian and Heinz 2018 (ICGI, SIGMORPHON)

The Encyclopedia of Categories for Maps

2:	2-way
1:	1-way
N:	Non-deterministic
D:	Deterministic
f:	functional
I:	Input
O:	Output
S:	Strictly
L:	Local



Engelfriedt and Hoogeboom 2001, Chandlee 2014, Filiot and Reynier 2016, Dolatian and Heinz 2018a,b

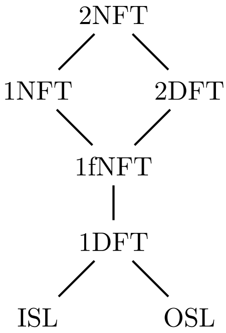
Special Topic: Logical characterizations

We saw how to use First Order logic to define regular sets.

1. $\forall x[\mathtt{a}(x)]$
2. $\exists x[\mathtt{a}(x)]$
3. $\forall x, y[(\mathtt{a}(x) \wedge S(x, y) \rightarrow \neg\, \mathtt{a}(y)]$

Special Topic: Logical characterizations

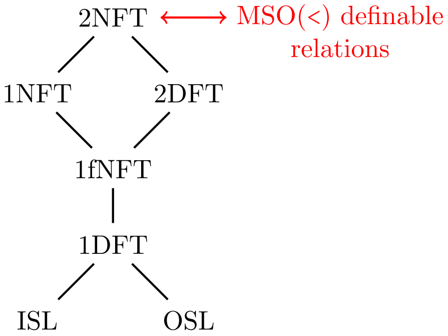
2:	2-way
1:	1-way
N:	Non-deterministic
D:	Deterministic
f:	functional
I:	Input
O:	Output
S:	Strictly
L:	Local



Engelfriedt and Hoogeboom 2001, Chandlee 2014, Filiot and Reynier 2016, Chandlee and Lindell, forthcoming

Special Topic: Logical characterizations

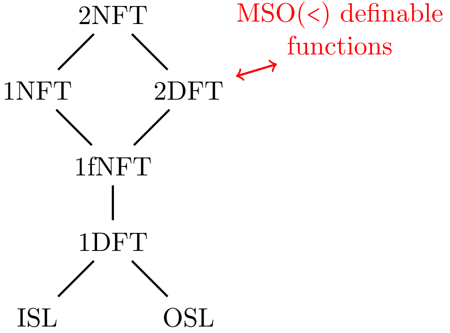
2:	2-way
1:	1-way
N:	Non-deterministic
D:	Deterministic
f:	functional
I:	Input
O:	Output
S:	Strictly
L:	Local



Engelfriedt and Hoogeboom 2001, Chandlee 2014, Filiot and Reynier 2016, Chandlee and Lindell, forthcoming

Special Topic: Logical characterizations

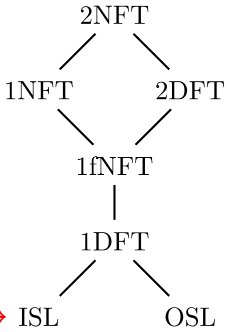
2:	2-way
1:	1-way
N:	Non-deterministic
D:	Deterministic
f:	functional
I:	Input
O:	Output
S:	Strictly
L:	Local



Engelfriedt and Hoogeboom 2001, Chandlee 2014, Filiot and Reynier 2016, Chandlee and Lindell, forthcoming

Special Topic: Logical characterizations

2:	2-way
1:	1-way
N:	Non-deterministic
D:	Deterministic
f:	functional
I:	Input
O:	Output
S:	Strictly
L:	Local



Quantifer Free (◁) definable ⟷ ISL

Engelfriedt and Hoogeboom 2001, Chandlee 2014, Filiot and Reynier 2016, Chandlee and Lindell, forthcoming

Special Topic: Machine Learning

Key Ideas

1. Minimum Description Length
2. Largest Common Structures
3. Bayesian Inference
4. Weighted FSTs
5. Neural Networks

Fun Challenge

- Morphological Re-inflection Challenge
- Unimorph

Studying Linguistic Typology

Requires two books:

- "encyclopedia of categories"
- "encyclopedia of types"

Wilhelm Von Humboldt