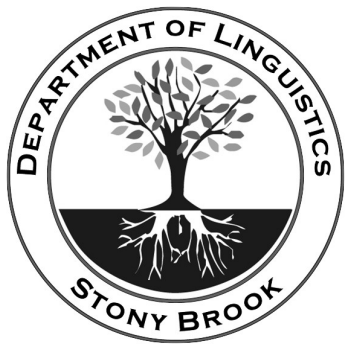
What are Morphological Transformations?

Jeffrey Heinz



Stony Brook University

NYU
November 17, 2020

Points of Agreement with Earlier Speakers

1. The problem of how morphology and morpho-phonology is learned/acquired is extremely interesting and challenging.
2. Productivity, principles of simplicity, and representations, all matter.

Today, I will ask "What is it that is being learned?" with respect to morphological transformations, and try to give some insights from what we know about *categories* of string-to-string functions.

Doing Linguistic Typology

Requires two books:

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

Wilhelm Von Humboldt

Realizational Morphology

A Template for Realization Rules

$$RR_{n,\tau,C}(< X, \sigma >) =_{\text{def}} < Y', \sigma >$$

n	an index for a particular block of rules
τ	the set of morphological features that are realized by the application of the rule
C	the lexical category that the rule is concerned with
X	a phonological input string that is either a part of the lexical representation or has been derived
σ	a set of morpho-syntactic properties (= features)
Y'	the resulting output string

(Stump 2001, Karttunen 2016)

String to String Functions

A Template for Realization Rules

$$RR_{n,\tau,C}(< X, \sigma >) \ =_{\text{def}} \ \ < Y', \sigma >$$

I want to focus on the ways in which input string X can be realized as output string Y ('exponing' in DM).

1. What is the encyclopedia of types?
2. What is the encyclopedia of categories?

Encyclopedia of Types

Morphological Transformations

1. Null affixation
2. Prefixation
3. Suffixation
4. Circumfixation
5. Infixation
6. Truncation
7. Root and pattern
8. Umlaut/Ablaut
9. Partial Reduplication
10. Total Reduplication
11. ...

Encyclopedia of Categories?

$$f : \Sigma^* \rightarrow \Delta^*$$

Questions

1. What is a 'local' transformation?
2. What are 'non-local' transformations?
3. What kinds of transformations require a lot of memory and/or computational resources?
4. What kinds of transformations do not?

Analogy to Real Functions

$$f : \mathbb{R} \to \mathbb{R}$$

Encyclopedia of Categories

1. Linear functions
2. Step functions
3. Polynomial functions (quadratic, cubic, degree n)
4. Exponential functions
5. Logarithmic functions
6. Trigonometric functions (sin, tanh, ...)
7. ...

String-to-String Functions

The established, foundational view (Roark and Sproat 2007)

Rational Relations	non-Rational Relations
Prefixation	Total Reduplication
Suffixation	
Circumfixation	
Infixation	
Truncation	
Root and pattern	
Umlaut/Ablaut	
Partial Reduplication	
...	

String-to-String Functions

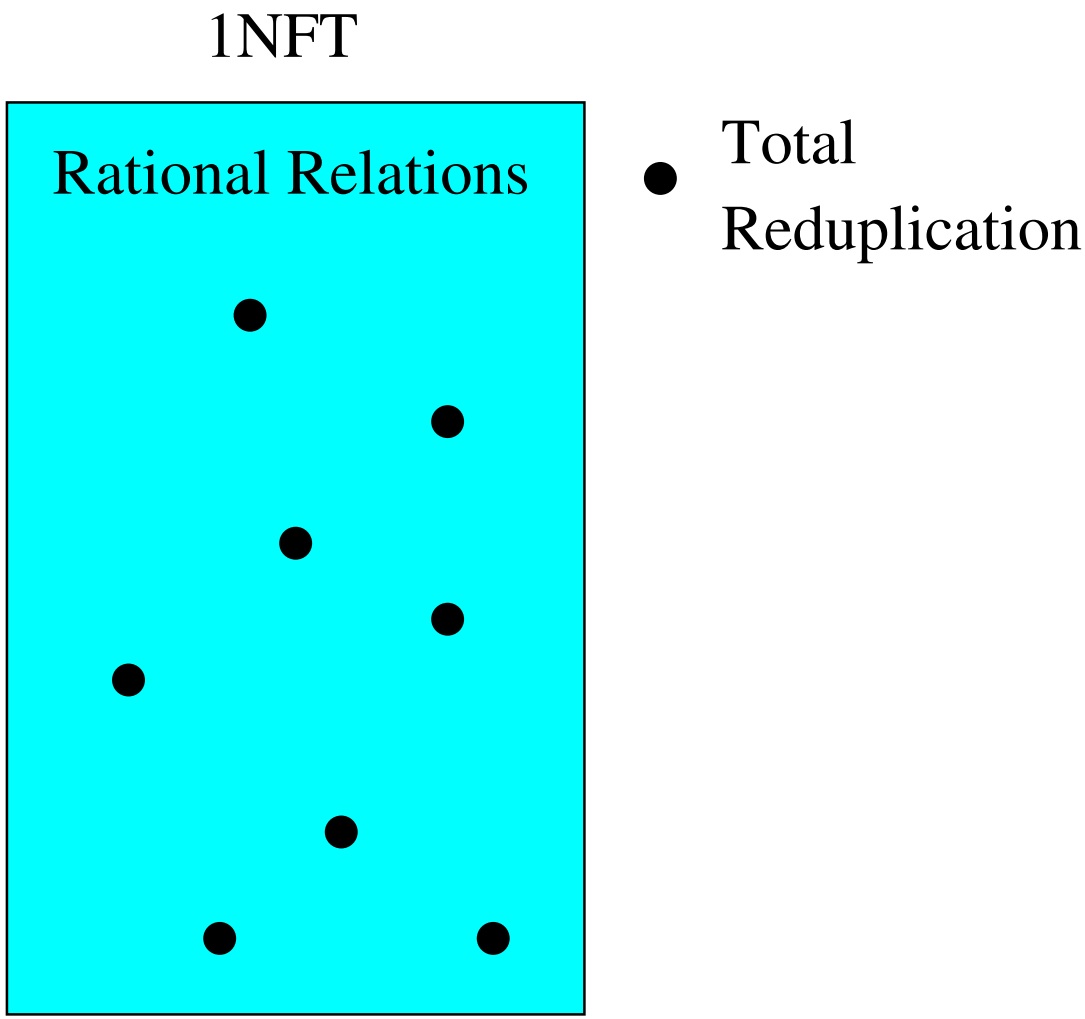
The established, foundational view (Roark and Sproat 2007)

Rational Relations	non-Rational Relations
Prefixation	Total Reduplication
Suffixation	
Circumfixation	
Infixation	
Truncation	
Root and pattern	
Umlaut/Ablaut	
Partial Reduplication	
...	

I think linguistics will be well-served by a more articulated view of this kind of encyclopedia of categories. Formal language theory is not static! Much more to discover.

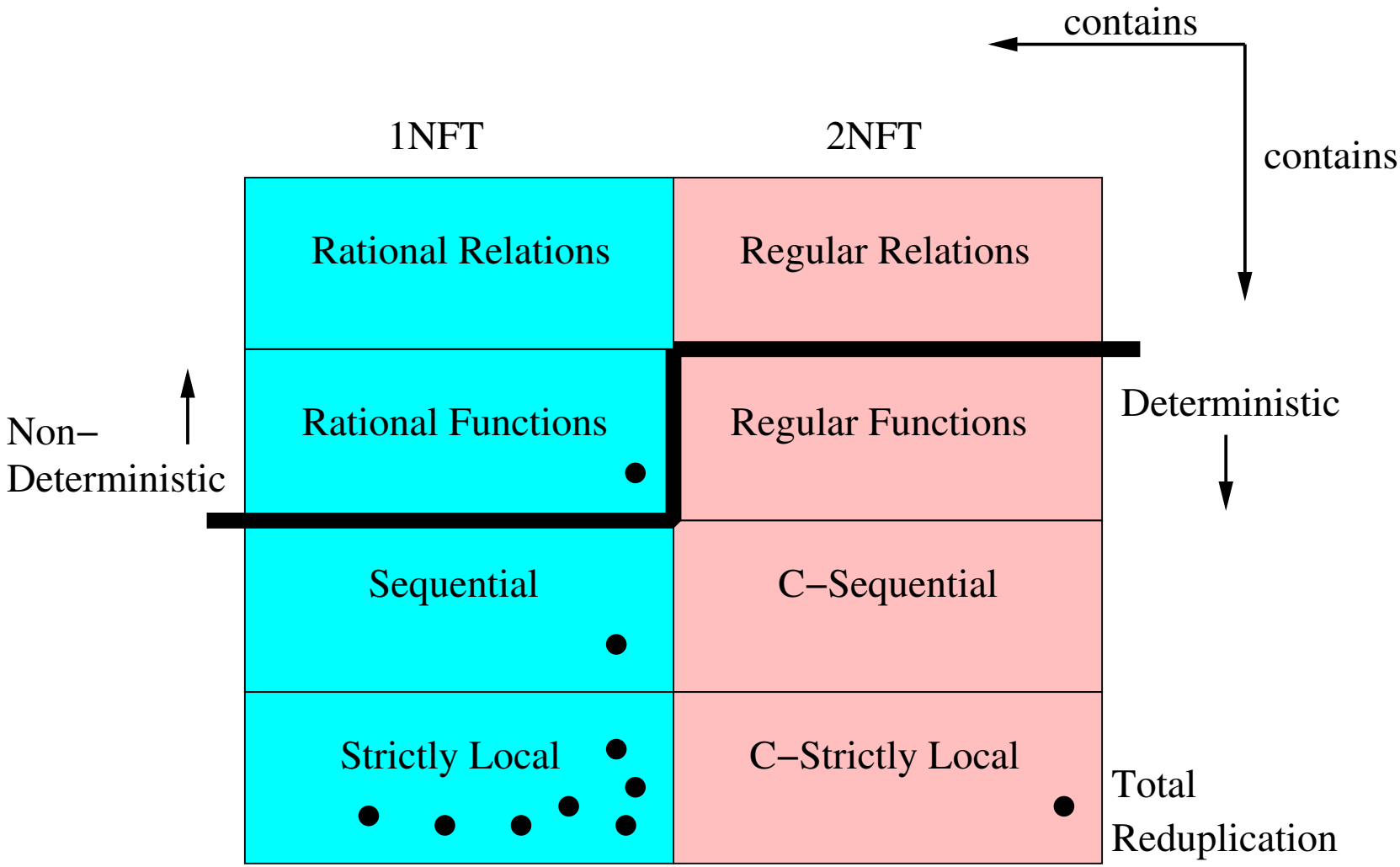
String-to-String Functions

This basic view pictorially



String-to-String Functions

A more articulated view



(Chandlee 2017, Dolatian and Heinz 2020)

Rational vs. Regular

For stringsets (formal languages) there is no distinction.

$$[\![\text{1DFA}]\!] = [\![\text{1NFA}]\!] = [\![\text{2NFA}]\!]$$
$$= [\![\text{RE}]\!] = [\![\text{GRE}]\!]$$
$$= [\![\text{MSO}(+1)]\!] = [\![\text{MSO}(<)]\!]$$

1/2	1-way or 2-way
N/D	Non-deterministic or Deterministic
FA	Finite-state Acceptor
(G)RE	(Generalized) Regular Expressions
MSO	Monadic Second Order with successor (+1) or precedence (<)

Rational vs. Regular

For string-to-string functions, there are!

$$\underbrace{\underbrace{[\![\text{1DFT}]\!] \subsetneq [\![\text{1fNFT}]\!] \subsetneq [\![\text{1NFT}]\!]}_{\text{Rational}} \sim [\![\text{2DFT}]\!] \subsetneq [\![\text{2NFT}]\!]}_{\text{Regular}}$$

Making copies vs. Recognizing copies

It is easier to *make* a copy than to *recognize* a copy.

1. Given w, is there is a v such that $w = vv$?
   (recognizing copies)
2. Given w, return ww. (making copies)

The act of copying is *regular* but not *rational*. Recognizing copies is neither.

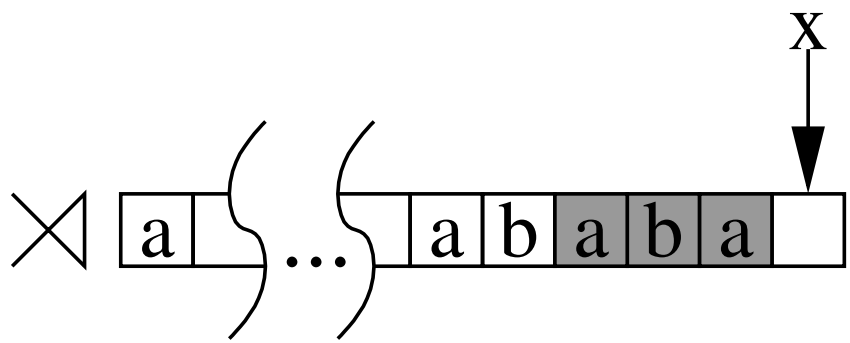
(Filiot and Reynier 2016, Dolatian and Heinz 2020)

Local String-to-String Functions

- What could it mean for a string-to-string function to be local?
- Consider the Markov property.

$$P(a_{n+1} \mid a_1 a_2 \ldots a_n) \approx P(a_{n+1} \mid a_{n-k} a_{n-k+1} \ldots a_n)$$

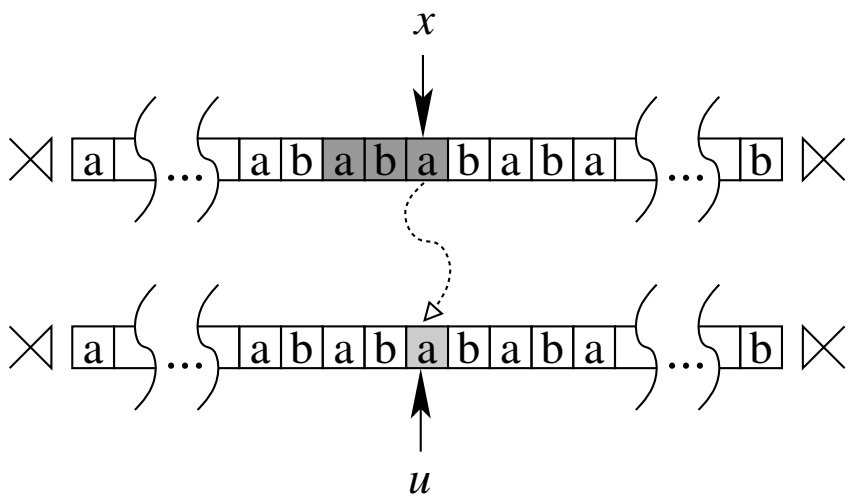
- The probability of the next item only depends on the previous k symbols.



Strictly Local Functions

Chandlee develops the same idea in the context of string rewriting.

Input Strictly Local

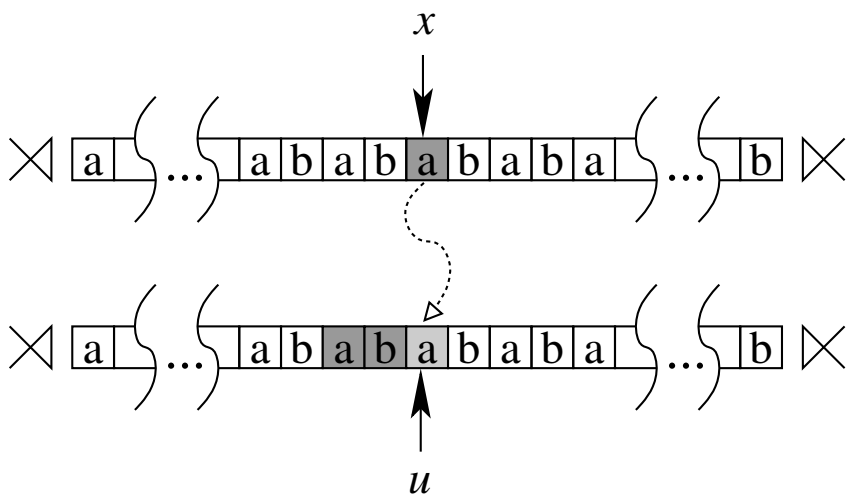


(Chandlee 2014, Chandlee et al. 2014, 2015)

Strictly Local Functions

Chandlee develops the same idea in the context of string rewriting.

Output Strictly Local

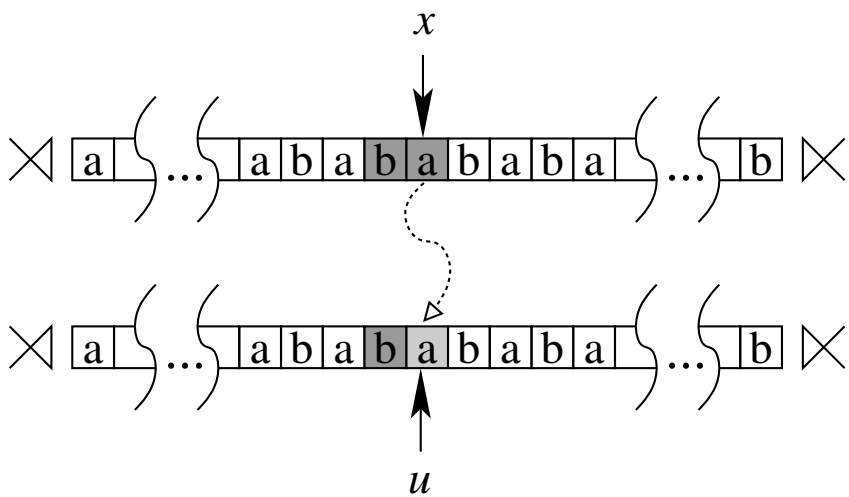


(Chandlee 2014, Chandlee et al. 2014, 2015)

Strictly Local Functions

Chandlee develops the same idea in the context of string rewriting.

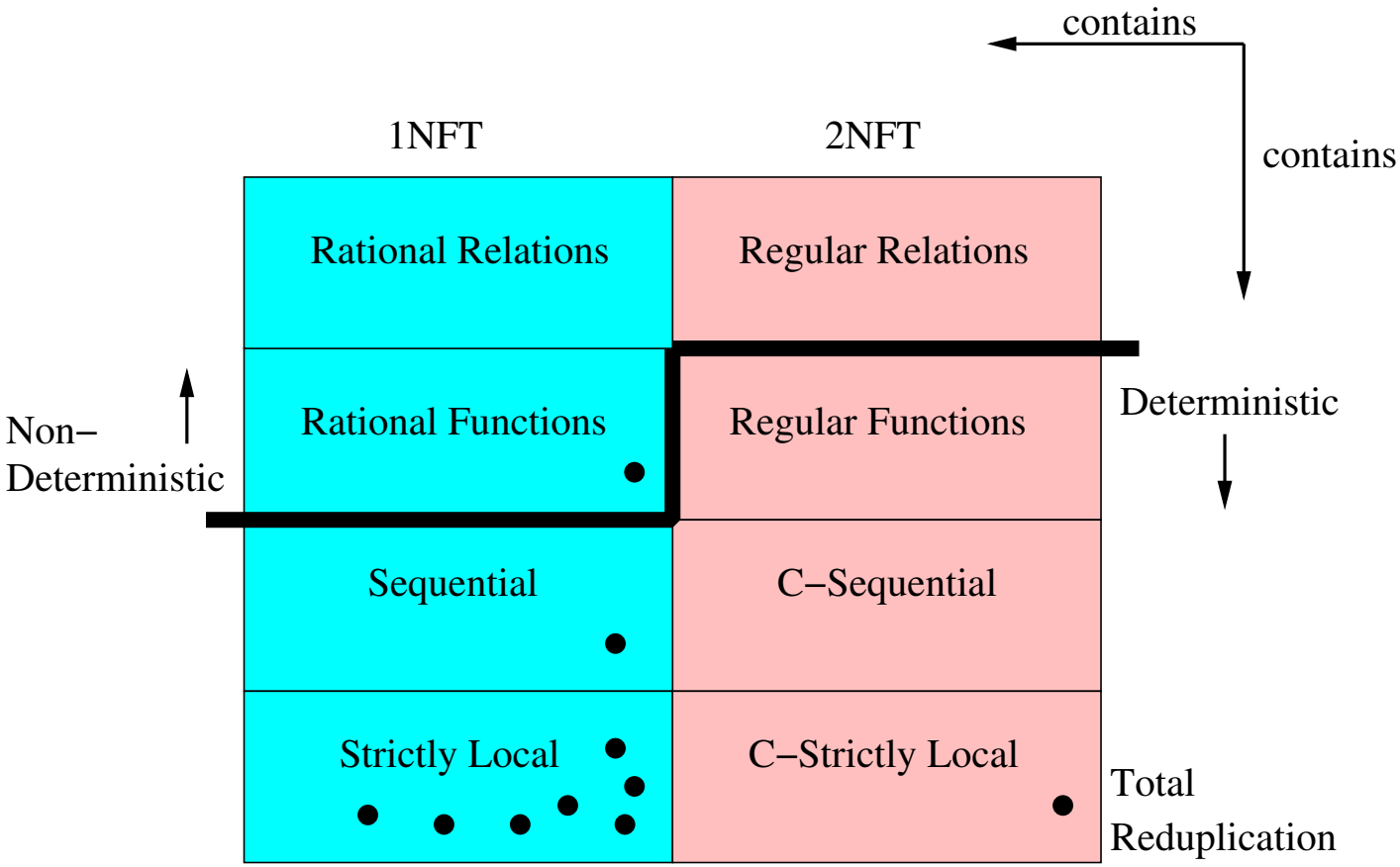
Input-Output Strictly Local



(Chandlee 2014, Chandlee et al. 2014, 2015)

Strictly Local Functions

- I group all of these types here as "strictly local functions."
- They cover quite a bit of the typology of morphology and phonology (Chandlee 2017, Chandlee and Heinz 2018)
- They are learnable more or less the same way n-gram models are (Chandlee et al. 2014, 2015)



More on reduplication

- 2-way DFTs (regular functions) can be thought of as the *concatenation* of 1-way DFTs (rational functions)

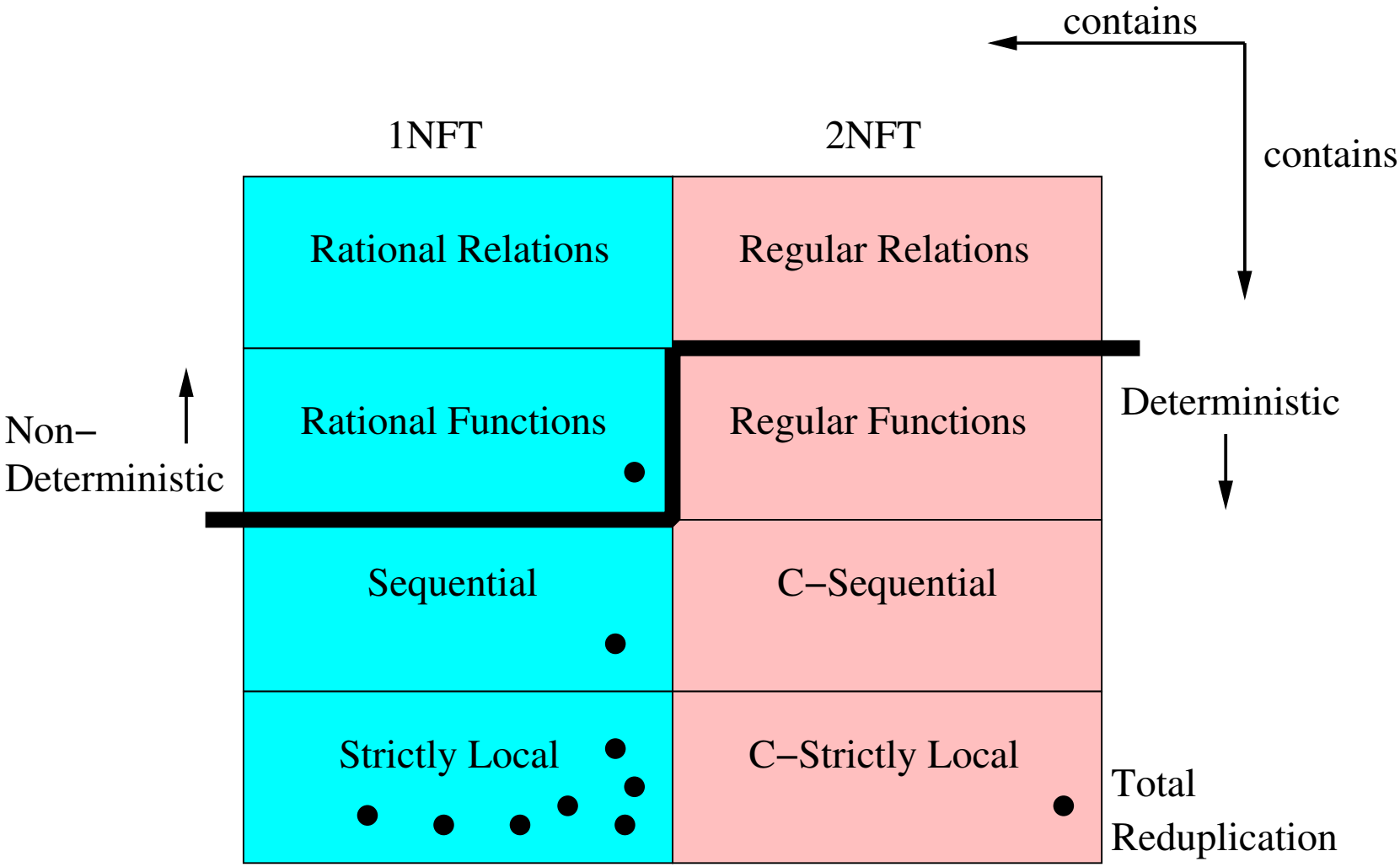
$$f(w) = g(w) \circ h(w)$$

  where g, h are defined with 1-way DFTs.
- Hence we can study classes of regular functions in terms of the concatenation of rational functions.
- Total reduplication is the concatenation of the simplest strictly local functions.
- Partial reduplication is arguably better understood this this way too.

(Dolatian and Heinz 2020)

Summary

1. The types of string-to-string functions matter.
2. Regular functions and rational functions are different.
3. Strictly Local functions are a natural way to understand locality in string transformations.
4. We achieve a better understanding of copying processes.
5. Everything that was done here with strings we can do with *trees*, and other linguistic structures! (using logic)


contains
contains
1NFT
2NFT
Rational Relations
Regular Relations
Rational Functions
Regular Functions
Sequential
C–Sequential
Strictly Local
C–Strictly Local
Non–
Deterministic
Deterministic
Total
Reduplication

Thanks!