Recursivity in phonology – what can it mean below the word?

Julian Bradfield

University of Edinburgh

Recursion

Recursion

II

Recursion

Recursion

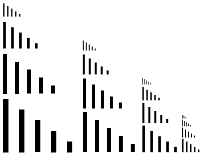
Recursion

Recursion

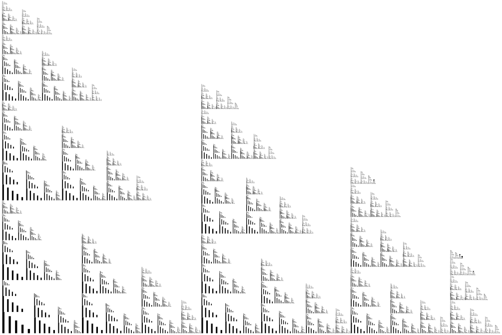
Recursion

Recursion

Recursion



Recursion



Recursion

The Hitchhiker's Guide to

DON'T
PANIC

Recursion

Brief History

Recursive procedures *sensu lato*

- go back millennia: Indian mathematicians approximated $\sqrt{2}$.
- but can mostly be seen as iterative (see later)

Brief History

Recursive procedures *sensu lato*

- go back millennia: Indian mathematicians approximated $\sqrt{2}$.
- but can mostly be seen as iterative (see later)

Recursive definitions

- first appear with Dedekind in the 1890s
- became the foundation of set theory (and thus mathematics:) around that time
- were studied intensively *per se* in the 1930s–50s

Brief History

Recursive procedures *sensu lato*

- go back millennia: Indian mathematicians approximated $\sqrt{2}$.
- but can mostly be seen as iterative (see later)

Recursive definitions

- first appear with Dedekind in the 1890s
- became the foundation of set theory (and thus mathematics:) around that time
- were studied intensively *per se* in the 1930s–50s

Recursive procedures *sensu stricto*

- are themselves recursive definitions
- have a rich theory even confined to procedures on integers
- and became a key part of programming from the 50s

Recursive definitions

(Modernized) Peano's definition of natural numbers:

- 0 is a natural number
- if n is a natural number, so is $\mathsf{S}(n)$
- (nothing else is a natural number)

and then equality and arithmetic are defined recursively too.

Recursive definitions

(Modernized) Peano's definition of natural numbers:

- 0 is a natural number
- if n is a natural number, so is $\mathrm{S}(n)$
- (nothing else is a natural number)

and then equality and arithmetic are defined recursively too.

Syntax:

$$\mathrm{NP} \rightarrow \mathrm{N} \mid \mathrm{Adj\ NP}$$

Recursive definitions

(Modernized) Peano's definition of natural numbers:

- 0 is a natural number
- if n is a natural number, so is $\mathsf{S}(n)$
- (nothing else is a natural number)

and then equality and arithmetic are defined recursively too.

Syntax:

$$\text{NP} \rightarrow \text{N} \mid \text{Adj NP}$$

Binary trees:

$$\text{Tree} \rightarrow \text{Leaf} \mid \text{Tree Tree}$$

$$\text{Tree} := \text{leaf} \mid \text{node}(\text{Tree}, \text{Tree})$$

$$\text{Tree} = \{\emptyset\} \cup (\text{Tree} \times \text{Tree})$$

$$\text{Tree} = \mu X.\{\emptyset\} \cup (X \times X)$$

Recursive definitions

(Modernized) Peano's definition of natural numbers:

- 0 is a natural number
- if n is a natural number, so is $\mathsf{S}(n)$
- (nothing else is a natural number)

and then equality and arithmetic are defined recursively too.

Syntax:

$$\text{NP} \to \text{N} \mid \text{Adj NP}$$

Binary trees:

$$\text{Tree} \to \text{Leaf} \mid \text{Tree Tree}$$

$$\text{Tree} := \text{leaf} \mid \text{node}(\text{Tree}, \text{Tree})$$

$$\text{Tree} = \{\emptyset\} \cup (\text{Tree} \times \text{Tree})$$

$$\text{Tree} = \mu X.\{\emptyset\} \cup (X \times X)$$

More syntax:

$$\text{NP} \to \text{N} \mid \text{Adj NP} \mid \text{NP PP}$$

$$\text{PP} \to \text{P NP}$$

Recursion and Iteration

A natural number is a string of the form I^*

$\text{NP} := \text{Adj}^*\ N$

$\text{Tree} = \{T \subseteq 2^* : T \text{ is prefix-closed}\}$

Recursion and Iteration

A natural number is a string of the form I^*

$\text{NP} := \text{Adj}^* \; N$

$\text{Tree} = \{T \subseteq 2^* : \; T \text{ is prefix-closed}\}$

Recursive definitions and computations can be *implemented* using iteration, arithmetic or pointers, and **unbounded** memory.

Iterative definitions basically give regular expressions – much less than recursion. (Cf. Jeff Heinz *et al.* on subregular phonology.)

Recursion entails infinity!

There are infinitely many natural numbers, NPs, binary trees, etc.
This is *required* by the definitions.

Recursion entails infinity!

There are infinitely many natural numbers, NPs, binary trees, etc.
This is *required* by the definitions.

'$\text{Tree}_2 =$ The binary trees of depth at most two':

Recursion entails infinity!

There are infinitely many natural numbers, NPs, binary trees, etc. This is *required* by the definitions.

'$\text{Tree}_2 =$ The binary trees of depth at most two':

- define Tree, define depth d (by recursion), say $\{T \in \text{Tree} \colon d(T) \leq 2\}$. This **requires** infinity (and counting).

Recursion entails infinity!

There are infinitely many natural numbers, NPs, binary trees, etc.
This is *required* by the definitions.

'Tree_2 = The binary trees of depth at most two':

- define Tree, define depth d (by recursion), say $\{T \in \text{Tree}\colon d(T) \leq 2\}$. This **requires** infinity (and counting).
- Finitely by enumeration: $\{\,\emptyset, (\emptyset,\emptyset), ((\emptyset,\emptyset),\emptyset), (\emptyset,(\emptyset,\emptyset)), ((\emptyset,\emptyset),(\emptyset,\emptyset))\,\}$. This is **not recursive**.

Recursion entails infinity!

There are infinitely many natural numbers, NPs, binary trees, etc. This is *required* by the definitions.

'Tree_2 = The binary trees of depth at most two':

- define Tree, define depth d (by recursion), say $\{\, T \in \text{Tree} \colon d(T) \leq 2 \}$. This **requires** infinity (and counting).
- Finitely by enumeration: $\{\, \emptyset, (\emptyset, \emptyset), ((\emptyset, \emptyset), \emptyset), (\emptyset, (\emptyset, \emptyset)), ((\emptyset, \emptyset), (\emptyset, \emptyset)) \,\}$. This is **not recursive**.
- Finitely by level:
  $\text{Tree}_0 = \{\, \emptyset \,\} \quad \text{Tree}_1 = \text{Tree}_0 \cup \text{Tree}_0 \times \text{Tree}_0$
  $\text{Tree}_2 = \text{Tree}_1 \cup \text{Tree}_1 \times \text{Tree}_1$
  This is **not recursive**.

Recursion entails infinity!

There are infinitely many natural numbers, NPs, binary trees, etc.
This is *required* by the definitions.

'$\text{Tree}_2 =$ The binary trees of depth at most two':

- define Tree, define depth d (by recursion), say $\{ T \in \text{Tree} \colon d(T) \leq 2 \}$. This **requires** infinity (and counting).
- Finitely by enumeration: $\{ \emptyset, (\emptyset, \emptyset), ((\emptyset, \emptyset), \emptyset), (\emptyset, (\emptyset, \emptyset)), ((\emptyset, \emptyset), (\emptyset, \emptyset)) \}$. This is **not recursive**.
- Finitely by level:
  $\text{Tree}_0 = \{ \emptyset \} \quad \text{Tree}_1 = \text{Tree}_0 \cup \text{Tree}_0 \times \text{Tree}_0$
  $\text{Tree}_2 = \text{Tree}_1 \cup \text{Tree}_1 \times \text{Tree}_1$
  This is **not recursive**.

No (potential) infinity = no recursion!
Bounded recursion is not real recursion!

Recursion in language

Does *natural* language have unbounded recursion *at all*?

Recursion in language

Does *natural* language have unbounded recursion *at all*?

The grammar in a speaker's head is not the grammar in a textbook. Humans have very small working memory.

Recursion in language

Does *natural* language have unbounded recursion *at all*?

The grammar in a speaker's head is not the grammar in a textbook. Humans have very small working memory.

Language games are outwith the bounds of natural language.

Bounds on recursion

Fred Karlsson (2007) argues that no *natural* language uses more than two (maybe three) levels of centre-embedding (recursive construct requiring unbounded memory).

Bounds on recursion

Fred Karlsson (2007) argues that no *natural* language uses more than two (maybe three) levels of centre-embedding (recursive construct requiring unbounded memory).

Jim Hurford (2012) and students say maybe four is comprehensible:

Bounds on recursion

Fred Karlsson (2007) argues that no *natural* language uses more than two (maybe three) levels of centre-embedding (recursive construct requiring unbounded memory).

Jim Hurford (2012) and students say maybe four is comprehensible:

Entweder
die Sprache,
die Kinder
von ihren,
sich an den Haaren zerrenden
Eltern
lernen,
ist Deutsch,
oder sie sind dumm.

Unbounded right embedding = iteration

This is the farmer sowing the corn,
That kept the cock that crowed in the morn.
That waked the priest all shaven and shorn,
That married the man all tattered and torn,
That kissed the maiden all forlorn,
That milked the cow with the crumpled horn,
That tossed the dog,
That worried the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

Unbounded right embedding = iteration

This is the farmer sowing the corn,
That kept the cock that crowed in the morn.
That waked the priest all shaven and shorn,
That married the man all tattered and torn,
That kissed the maiden all forlorn,
That milked the cow with the crumpled horn,
That tossed the dog,
That worried the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

Everybody's happy with iteration in supra-word phonology, right?

Unbounded right embedding = iteration

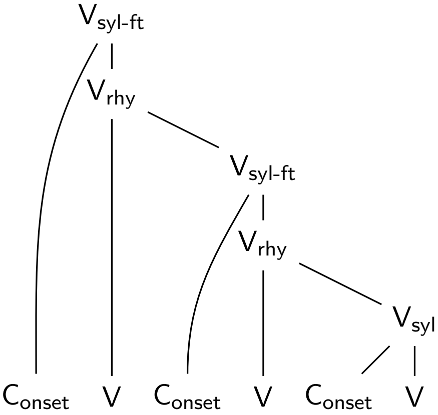
This is the farmer sowing the corn,
That kept the cock that crowed in the morn.
That waked the priest all shaven and shorn,
That married the man all tattered and torn,
That kissed the maiden all forlorn,
That milked the cow with the crumpled horn,
That tossed the dog,
That worried the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

Everybody's happy with iteration in supra-word phonology, right?

Unbounded left/centre-embedding in supra-word phonology???

Infra-word potential recursion: prosody

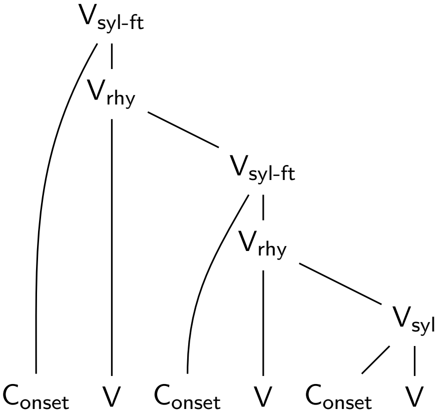
For example, van der Hulst (2010) claims:



for the English dactyl (*seren**dipity***).

Infra-word potential recursion: prosody

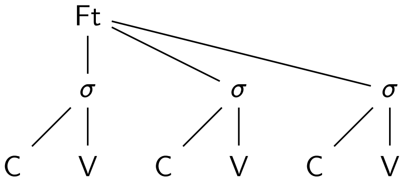
For example, van der Hulst (2010) claims:



for the English dactyl (*seren**dipity***).

Level 5 recursion in feet when we can't even do that in syntax?

Infra-word ~~potential recursion:~~ prosody

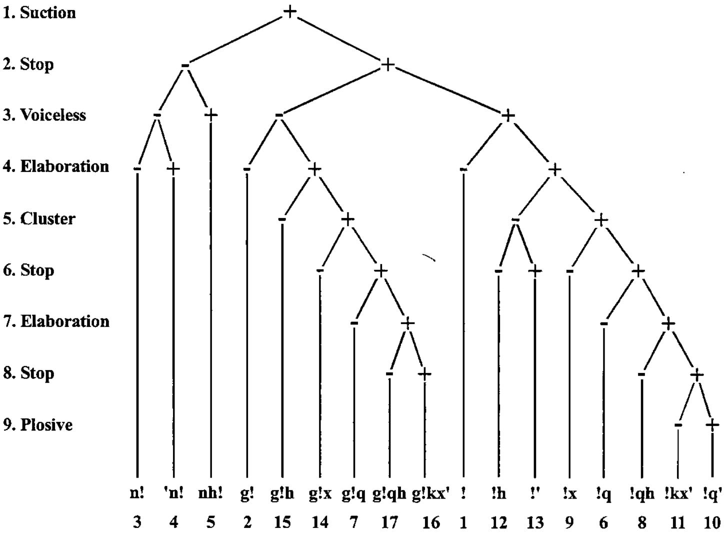


Feet longer than dactyls are also iterative in most theories (perhaps excepting grid–mark phonology) and tend to be supra-word.

(Is ‘characterlessness’ really one foot?)

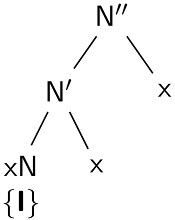
Feature geometry and friends

Tom Güldemann's feature geometry of clicks:



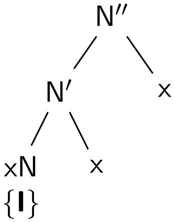
Element geometry and friends

Markus will (almost certainly:) show us things like



Element geometry and friends

Markus will (almost certainly:) show us things like



If you buy the idea that sounds are represented and distinguished by configurations of trees of sets of elements, this might be recursion . . . but how far? Level three? So not recursion.

Does infra-word phonology have recursivity, or does it not?

Does infra-word phonology have recursivity, or does it not?

mu

Does infra-word phonology have recursivity, or does it not?

mu

μ