

# INFLECTED ++ : Rethinking the Customization of Morphotactic Systems with the Grammar Matrix

Michael Wayne Goodman and Emily M. Bender  
{goodmami, ebender}@uw.edu

University of Washington

July 6, 2010



## Lexical Rules in the Matrix

- Non-branching rules
- lexeme-to-lexeme, lexeme-to-word, word-to-lexeme
- Phrasal rules require words (not lexemes) as input

## Slots vs. Morphemes

- Slots are supertypes of morphemes in tdl
- Functionally, slots:
  - Specify the inputs for lexical rules
  - Specify morphotactic constraints between slots and lexical types
- Conceptually, slots are similar to morphological paradigms
- Morphemes:
  - Are lexical rule leaf-types
  - Apply syntactic features
  - Are linked to orthographic rules

## Morphotactics

- The ordering and co-occurrence restrictions of morphemes
- Separate from morphophonology and morphosyntax

## Previous System (O'Hara 2008)

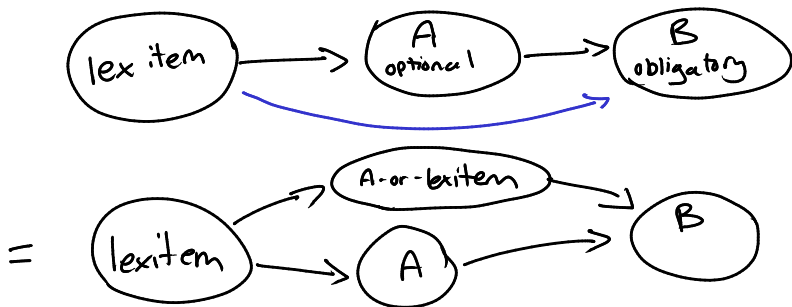
- INFLECTED is a boolean value
- Phrasal rules require INFLECTED +
- Optional/Non-optional rules
- Long-distance morphotactic constraints
  - Require
  - Force
  - Disrequire

## Phrasal rule requirements

$$\left[ \begin{array}{l} \textit{binary-phrase} \\ \text{ARGS} \left\langle \begin{array}{l} \left[ \text{INFLECTED} \quad + \right] \\ \left[ \text{INFLECTED} \quad + \right] \end{array} \right\rangle \end{array} \right]$$

## Optional / Non-Optional Rules

- Last non-optional rule (or lexical item) sets INFLECTED to +
- Optional rules get grouped in a type hierarchy that encompasses all possible inputs for a given lexical rule
- Non-optional rules are not grouped unless explicitly described as a secondary input to another lexical rule





## TRACK for Long-distance and Complex Constraints

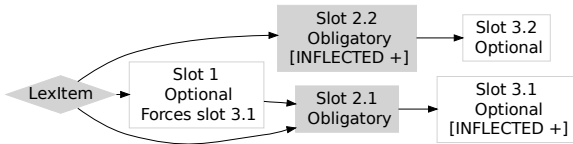
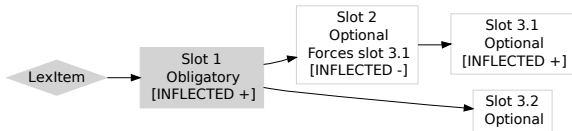
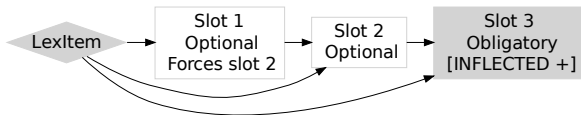
- TRACK is an avm added as a top-level feature on *word-or-lexrule*
- When necessary, a boolean feature (a **flag**) is added to TRACK to keep track of a rule's occurrence
- In order to force, require, or disrequire a slot, the TRACK variable is set both in the affected rule and the constraining rule

Problems:

- Inelegant solutions for some configurations
- Cannot put constraints on lexical types
- Cannot easily handle disjunctive requirements

These problems end up in much duplication of lexical types.

## Examples of problematic configurations:



## New System: INFLECTED avm

- (Essentially) move TRACK into INFLECTED
- Phrasal-rules require a satisfactory AVM
- Allow morphotactic constraints on lexical types
- Allow explicit disjunction of constraints

## Phrasal Rule Requirements

$$\left[ \begin{array}{l} \textit{binary-phrase} \\ \text{ARGS} \left\langle \begin{array}{l} \left[ \text{INFLECTED} \quad \textit{infl-satisfied} \right] \\ \left[ \text{INFLECTED} \quad \textit{infl-satisfied} \right] \end{array} \right\rangle \end{array} \right]$$

## *inflected* Hierarchy

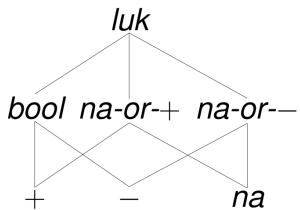
In matrix.tdl:

```
inflected := avm.  
infl-satisfied := inflected.
```

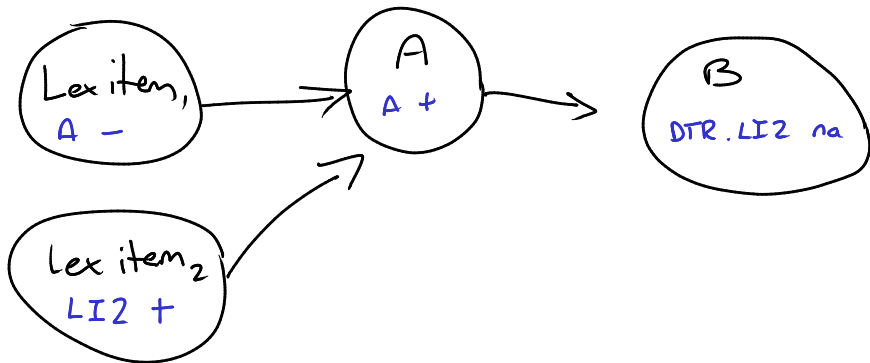
In language specific tdl file:

```
inflected :+ [ X luk,  
                Y luk,  
                Z luk ].  
  
infl-satisfied :+ [ X na-or-+,  
                   Y na-or-+,  
                   Z na-or-+ ].
```

The *luk* hierarchy:



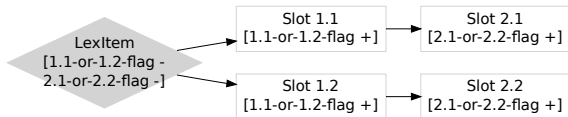
## Constraints on Lextypes



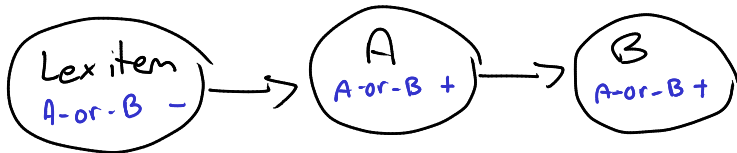


## Disjunctive requirements

- Implicit disjunction: non-sequential slots

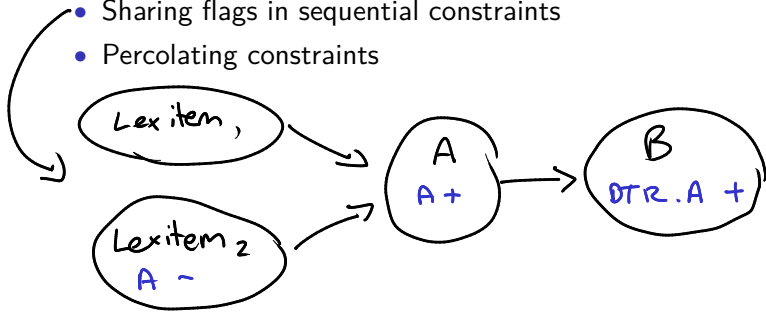


- Explicit disjunction: sequential



## Optimizations

- Reducing flags in implicit disjunctions
- Sharing flags in sequential constraints
- Percolating constraints



## Remaining Problems

- Direct-inverse not yet working with the new system
  - Need more levels than just slots and morphemes?
- Percolating constraints
- Too many flags?