A comprehensive view on inflectional classification

S. Beniamine, O. Bonami
Laboratoire de Linguistique Formelle
Université Paris Diderot
sbeniamine@linguist.univ-paris-diderot.fr
LAGB 2016, York
## Inflection classes

- Classification of lexemes according to inflectional behavior.

<table>
<thead>
<tr>
<th>Case</th>
<th>1st declension</th>
<th>2nd declension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SINGULAR</td>
<td>PLURAL</td>
</tr>
<tr>
<td>Nominative</td>
<td>rosa</td>
<td>rosae</td>
</tr>
<tr>
<td>Vocative</td>
<td>rosa</td>
<td>rosae</td>
</tr>
<tr>
<td>Accusative</td>
<td>rosam</td>
<td>rosas</td>
</tr>
<tr>
<td>Genitive</td>
<td>rosa</td>
<td>rosarum</td>
</tr>
<tr>
<td>Dative</td>
<td>rosa</td>
<td>rosis</td>
</tr>
<tr>
<td>Ablative</td>
<td>rosa</td>
<td>rosis</td>
</tr>
</tbody>
</table>

Table: Latin declension classes
What is this talk about?

Table: Several types of classification structures
What is this talk about?

<table>
<thead>
<tr>
<th>(1) Partition</th>
<th>(2) Tree</th>
<th>(3) Lattice</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
</tbody>
</table>

Table: Several types of classification structures

- “Inflection classes” usually refers to either (1) or (2).
What is this talk about?

Table: Several types of classification structures

- “Inflection classes” usually refers to either (1) or (2).
- We argue that these overlook important relations between lexemes,
What is this talk about?

<table>
<thead>
<tr>
<th>(1) Partition</th>
<th>(2) Tree</th>
<th>(3) Lattice</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
</tbody>
</table>

Table: Several types of classification structures

- “Inflection classes” usually refers to either (1) or (2).
- We argue that these overlook important relations between lexemes, and hide structural properties that are in fact pervasive.
What is this talk about?

<table>
<thead>
<tr>
<th>(1) Partition</th>
<th>(2) Tree</th>
<th>(3) Lattice</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
<td><img src="#" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Table: Several types of classification structures

- “Inflection classes” usually refers to either (1) or (2).
- We argue that these overlook important relations between lexemes,
- and hide structural properties that are in fact pervasive.
- Taking advantage of automation to work on large datasets,
What is this talk about?

<table>
<thead>
<tr>
<th>(1) Partition</th>
<th>(2) Tree</th>
<th>(3) Lattice</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
<tr>
<td>c1</td>
<td>c2</td>
<td>c3</td>
</tr>
</tbody>
</table>

Table: Several types of classification structures

- “Inflection classes” usually refers to either (1) or (2).
- We argue that these overlook important relations between lexemes,
  and hide structural properties that are in fact pervasive.
- Taking advantage of automation to work on large datasets,
  we argue that lattices (3) are a more faithful model of IC.
Outline

1. Specificities of this approach

2. Background on Inflection Classes

3. How to model IC system as lattices

4. Towards a typology of IC lattices
Table of Contents

1. Specificities of this approach

2. Background on Inflection Classes

3. How to model IC system as lattices

4. Towards a typology of IC lattices
Inflectional realizations as patterns

- We take *inflectional behaviour* to be relations between word-forms, or *alternation patterns* (not morphemes). (Blevins, 2006)

\[
X_{\text{PStem}} \leftrightarrow X_{\text{Part}}
\]

/ fəɡɔtɛn /
PPART ‘forgotten’

/ fəɡɛtɪŋ /
PRESPART ‘forgetting’
Specificities of this approach

Inflectional realizations as patterns

- We take **inflectional behaviour** to be relations between word-forms, or **alternation patterns** (not morphemes). (Blevins, 2006)
- Patterns take surface alternation at face value and do not require to choose between stem or exponent alternation.

```
XωCη ⇌ XεCη
```

```
/ fəɡɔtn̩ /
PPART ‘forgotten’

/ fəɡɛtɪŋ /
PRESPART ‘forgetting’
```
Specificities of this approach

Inflectional realizations as patterns

- We take inflectional behaviour to be relations between word-forms, or alternation patterns (not morphemes). (Blevins, 2006)
- Patterns take surface alternation at face value and do not require to choose between stem or exponent alternation.
- We infer those automatically.

\[ X_{\text{aCt}} \iff X_{\text{eCt}} \]

/fəɡɑtŋ/  
PPART ‘forgotten’

/fəɡɛtnŋ/  
PRESPART ‘forgetting’
Inflectional realizations as patterns

- We take **inflectional behaviour** to be relations between word-forms, or **alternation patterns** (not morphemes). (Blevins, 2006)
- Patterns take surface alternation at face value and do not require to choose between stem or exponent alternation.
- We infer those automatically.
  - We wrote an language-independent algorithm which relies on phonology-aware alignment (Frisch, 1997) of wordform pairs, inspired by Albright and Hayes, 2006.

\[
X_{\text{dCη}} \iff X_{\varepsilon\text{Cη}}
\]

/ fəɡɔtŋ /  
PPART ‘forgotten’

/ fəɡɛtiŋ /  
PRESPART ‘forgetting’
Inflectional paradigms

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3s</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>drəʊv</td>
<td>drɪvŋ</td>
<td>draɪv</td>
<td>draɪvz</td>
<td>draɪvŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>rəʊd</td>
<td>rɪdŋ</td>
<td>rɑɪd</td>
<td>rɑɪdz</td>
<td>rɑɪdŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>bɪt</td>
<td>bɪtŋ</td>
<td>bɑɪt</td>
<td>bɑɪts</td>
<td>bɑɪtŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>fəɡɑt</td>
<td>fəɡɑtŋ</td>
<td>fəɡɛt</td>
<td>fəɡɛts</td>
<td>fəɡɛtŋ</td>
</tr>
</tbody>
</table>

Table: Inflectional paradigm for some English verbs.
The pattern table

Lexemes are characterized by their collection of patterns (All pairwise alternations).

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST⇔PPART</th>
<th>PAST⇔PRES3S</th>
<th>PRES3S⇔PRESNOT3S</th>
<th>PPART⇔PRESNOT3S</th>
<th>PPART⇔PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>XəʊC ⇌ XɪCŋ</td>
<td>XəʊC_ ⇌ XaiCz</td>
<td>Xz ⇌ X</td>
<td>XɪCŋ ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>XəʊC ⇌ XɪCŋ</td>
<td>XəʊC ⇌ XaiCz</td>
<td>Xz ⇌ X</td>
<td>XɪCŋ ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>X ⇌ Xŋ</td>
<td>XɪC ⇌ XaiCs</td>
<td>Xs ⇌ X</td>
<td>XɪCŋ ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>X ⇌ Xŋ</td>
<td>XdC ⇌ xeCs</td>
<td>Xs ⇌ X</td>
<td>XdCŋ ⇌ xeC</td>
<td>XdCŋ ⇌ xeCŋ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PRES3S⇔PRESNOT3S</th>
<th>PPART⇔PRES3S</th>
<th>PRESNOT3S⇔PRESPART</th>
<th>PAST⇔PRESPART</th>
<th>PRES3S⇔PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>XəʊC ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCz</td>
<td>X ⇌ Xŋ</td>
<td>XəʊC ⇌ XaiCŋ</td>
<td>Xz ⇌ Xŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>Xəʊ ⇌ Xai</td>
<td>XɪCŋ ⇌ XaiCz</td>
<td>X ⇌ Xŋ</td>
<td>XəʊC ⇌ XaiCŋ</td>
<td>Xz ⇌ Xŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>XɪC ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCs</td>
<td>X ⇌ Xŋ</td>
<td>XVt ⇌ XaiVŋ</td>
<td>Xs ⇌ Xŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>XdC ⇌ XeC</td>
<td>XdCŋ ⇌ XeCs</td>
<td>X ⇌ Xŋ</td>
<td>XdC ⇌ XeCŋ</td>
<td>Xs ⇌ Xŋ</td>
</tr>
</tbody>
</table>

Table: Pattern table for some English verbs.
Lexemes are characterized by their collection of patterns (All pairwise alternations).

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST&lt;-&gt;PPART</th>
<th>PAST&lt;-&gt;PRES3S</th>
<th>PRES3S&lt;-&gt;PRESNOT3S</th>
<th>PPART&lt;-&gt;PRESNOT3S</th>
<th>PPART&lt;-&gt;PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>XəʊC ⇌ XɪCŋ</td>
<td>XəʊC_ ⇌ XaiCz</td>
<td>Xz ⇌ X</td>
<td>XɪCŋ ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>XəʊC ⇌ XɪCŋ</td>
<td>XəʊC ⇌ XaiCz</td>
<td>Xz ⇌ X</td>
<td>XɪCŋ ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>X ⇌ Xŋ</td>
<td>XɪC ⇌ XaiCs</td>
<td>Xs ⇌ X</td>
<td>XɪCŋ ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>X ⇌ Xŋ</td>
<td>XdC ⇌ XɛCs</td>
<td>Xs ⇌ X</td>
<td>XdCŋ ⇌ XɛC</td>
<td>XdCŋ ⇌ XɛCŋ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST&lt;-&gt;PRESNOT3S</th>
<th>PPART&lt;-&gt;PRES3S</th>
<th>PRESNOT3S&lt;-&gt;PRESPART</th>
<th>PAST&lt;-&gt;PRESPART</th>
<th>PRES3S&lt;-&gt;PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>XəʊC ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCz</td>
<td>X ⇌ Xŋ</td>
<td>XəʊC ⇌ XaiCŋ</td>
<td>Xz ⇌ Xŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>Xəʊ ⇌ Xai</td>
<td>XɪCŋ ⇌ XaiCz</td>
<td>X ⇌ Xŋ</td>
<td>XəʊC ⇌ XaiCŋ</td>
<td>Xz ⇌ Xŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>XɪC ⇌ XaiC</td>
<td>XɪCŋ ⇌ XaiCz</td>
<td>X ⇌ Xŋ</td>
<td>XVt ⇌ XaiVŋ</td>
<td>Xs ⇌ Xŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>XdC ⇌ XɛC</td>
<td>XdCŋ ⇌ XɛCs</td>
<td>X ⇌ Xŋ</td>
<td>XdC ⇌ XɛCŋ</td>
<td>Xs ⇌ Xŋ</td>
</tr>
</tbody>
</table>

Table: Pattern table for some English verbs.
The pattern table

Lexemes are characterized by their collection of patterns (All pairwise alternations).

![Pattern table]

**Table**: Pattern table for some English verbs.

---

S. Beniamine, O. Bonami

---
Automated approach

- Patterns and classification are generated by language-independent algorithms.
- This approach requires formal and quantifiable definitions of linguistic concepts,
- and allows us to work on large lexical databases,
- which paves the way for quantitative typological analysis of Inflection classes.
Table of Contents

1. Specificities of this approach

2. Background on Inflection Classes

3. How to model IC system as lattices

4. Towards a typology of IC lattices
Inflection Classes: identity or similarity?

“a set of lexemes whose members each select the same set of inflectional realizations”.

Inflection Classes: identity or similarity?

- “a set of lexemes whose members each select the same set of inflectional realizations”.
  

- Applied to realistic datasets, this leads to a large number of (mostly) very small classes.
Inflection Classes: identity or similarity?

- “a set of lexemes whose members each select the same set of inflectional realizations”.
  

- Applied to realistic datasets, this leads to a large number of (mostly) very small classes.

- In practice, Carstairs-McCarthy and many other authors focus on larger but not fully coherent classes.
Varying degrees of similarity

- Dressler and Thornton’s (1996) terminology:

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>drəvən</td>
<td>drəvən</td>
<td>draiv</td>
<td>draivz</td>
<td>draivnə</td>
</tr>
<tr>
<td>RIDE</td>
<td>rəvəd</td>
<td>rəvəd</td>
<td>rəid</td>
<td>rəidz</td>
<td>rəidnə</td>
</tr>
<tr>
<td>BITE</td>
<td>bɪt</td>
<td>bɪtən</td>
<td>bət</td>
<td>bəts</td>
<td>bətən</td>
</tr>
<tr>
<td>FORGET</td>
<td>fəɡət</td>
<td>fəɡət</td>
<td>fəɡət</td>
<td>fəɡəts</td>
<td>fəɡətən</td>
</tr>
</tbody>
</table>
Varying degrees of similarity

- Dressler and Thornton’s (1996) terminology:
  - Microclasses are based on identity

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>drɔuv</td>
<td>drɪnŋ</td>
<td>draɪv</td>
<td>draɪvz</td>
<td>draɪvŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>rəud</td>
<td>rɪdŋ</td>
<td>raið</td>
<td>raiðz</td>
<td>raiðŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>bɪt</td>
<td>bɪtŋ</td>
<td>baið</td>
<td>baiðz</td>
<td>baiðŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>fəɡət</td>
<td>fəɡətŋ</td>
<td>fəɡət</td>
<td>fəɡəts</td>
<td>fəɡətŋ</td>
</tr>
</tbody>
</table>
Varying degrees of similarity

- Dressler and Thornton’s (1996) terminology:
  - **Microclasses are based on identity**
  - **Macroclasses are based on similarity**
    (see Beniamine, Bonami, and Sagot, 2015 for automatical inference of macroclasses)

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>drəʊv</td>
<td>drɪvŋ</td>
<td>draɪv</td>
<td>draɪvz</td>
<td>draɪvŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>rəʊd</td>
<td>rɪdŋ</td>
<td>raɪd</td>
<td>raɪdz</td>
<td>raɪdŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>bɪt</td>
<td>bɪtŋ</td>
<td>bɑɪt</td>
<td>bɑɪts</td>
<td>bɑɪtŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>fəɡɑt</td>
<td>fəɡɑtŋ</td>
<td>fəɡɛt</td>
<td>fəɡɛts</td>
<td>fəɡɛtŋ</td>
</tr>
</tbody>
</table>
Varying degrees of similarity

- Dressler and Thornton’s (1996) terminology:
  - Microclasses are based on identity
  - Macroclasses are based on similarity
    (see Beniamine, Bonami, and Sagot, 2015 for automatical inference of macroclasses)

- Can form levels in a tree-shaped hierarchy.
  - Corbett and Fraser, 1993; Dressler and Thornton, 1996; Brown and Evans, 2012

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIVE</td>
<td>drəʊv</td>
<td>drɪvŋ</td>
<td>draɪv</td>
<td>draɪvz</td>
<td>draɪvŋ</td>
</tr>
<tr>
<td>RIDE</td>
<td>rəʊd</td>
<td>rɪdŋ</td>
<td>raɪd</td>
<td>raɪdz</td>
<td>raɪdŋ</td>
</tr>
<tr>
<td>BITE</td>
<td>bɪt</td>
<td>bɪtŋ</td>
<td>bɑɪt</td>
<td>bɑɪts</td>
<td>bɑɪtŋ</td>
</tr>
<tr>
<td>FORGET</td>
<td>fəɡɒt</td>
<td>fəɡɒtŋ</td>
<td>fəɡet</td>
<td>fəɡets</td>
<td>fəɡetŋ</td>
</tr>
</tbody>
</table>
Typology of inflection classes

- Evaluate the variation in IC systems relatively to a canonical point of comparison.
- “Canonical IC are fully comparable and are distinguished as clearly as possible”.

Corbett’s (2009), Principle I

- Internally homogeneous
- Structurally identical
- Maximally different
Internal homogeneity

*Within a canonical inflectional class each member behaves identically.*

By definition, it is always true of microclasses.

Corbett (2009), criterion 3
Internal homogeneity

*Within a canonical inflectional class each member behaves identically.*

Corbett (2009), criterion 3

- By definition, it is always true of microclasses.
- By definition, it is always false of any other classes.
Internal homogeneity

*Within a canonical inflectional class each member behaves identically.*

Corbett (2009), criterion 3

- By definition, it is always true of microclasses.
- By definition, it is always false of any other classes.
- A system where microclasses and macroclasses coincide is the most canonical.
Identical structure

*Canonical inflectional classes realize the same morphosyntactic or morphosemantic distinctions (they are of the same structure).*

Corbett (2009), criterion 2

- Two main deviations:

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>beware</td>
<td>-</td>
<td>-</td>
<td>бїв</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>abide</td>
<td>әбайд;</td>
<td>әбайд;</td>
<td>әбайд</td>
<td>әбайдз</td>
<td>әбайдің</td>
</tr>
</tbody>
</table>
Identical structure

*Canonical inflectional classes realize the same morphosyntactic or morphosemantic distinctions (they are of the same structure).*

Corbett (2009), criterion 2

- Two main deviations:
- **Defective** microclasses lack forms for certain cells in the paradigm.

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>beware</td>
<td>-</td>
<td>-</td>
<td>ḯῳε</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>abide</td>
<td>ḯ봐 jade;</td>
<td>ḯ봐 jade;</td>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
</tr>
<tr>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
<td>ḯ봐 abide</td>
</tr>
</tbody>
</table>
Identical structure

Canonical inflectional classes realize the same morphosyntactic or morphosemantic distinctions (they are of the same structure).

Corbett (2009), criterion 2

- Two main deviations:
  - Defective microclasses lack forms for certain cells in the paradigm.
  - Overabundant microclasses have several forms for certain cells in the paradigm.

<table>
<thead>
<tr>
<th>lexeme</th>
<th>PAST</th>
<th>PPART</th>
<th>PRES</th>
<th>PRES3S</th>
<th>PRESPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>beware</td>
<td>-</td>
<td>-</td>
<td>bɪwɛə</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>abide</td>
<td>əbaɪdɪd;</td>
<td>əbaɪdɪd;</td>
<td>əbaɪd</td>
<td>əbaɪdz</td>
<td>əbaɪdɪŋ</td>
</tr>
<tr>
<td></td>
<td>əbaɪd</td>
<td>əbaɪŋ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pattern sharing

*In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.*

Corbett (2009), criterion 1

- A canonical system is a partition of microclasses.
Pattern sharing

In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.

Corbett (2009), criterion 1

- A canonical system is a partition of microclasses.
- Any pattern sharing across ICs is non canonical.
Pattern sharing

In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.

Corbett (2009), criterion 1

- A canonical system is a partition of microclasses.
- Any pattern sharing across ICs is non canonical.
- The typological extreme is a system where microclasses display maximal sharing of patterns.
Pattern sharing: Heteroclisis

“a small number of items showing combinations of forms from other classes can be treated as heteroclites”

Corbett, 2009

- How to assess what small and big means quantitatively is uncertain.
Pattern sharing: Heteroclisis

- a microclass that shares patterns with at least two microclasses.
Pattern sharing: Heteroclisis

- better represented by a **lattice** structure than by a tree.

- **Behavior** represented by a lattice structure than by a tree.
- Subtype of overlapping. Classes can also be overlapping because of overabundance.

\[
\begin{align*}
\text{drive} & \quad \text{drove} & \quad \text{driven} \\
\text{ride} & \quad \text{rode} & \quad \text{ridden} \\
\text{bite} & \quad \text{bit} & \quad \text{bitten} \\
\text{forget} & \quad \text{forgot} & \quad \text{forgotten}
\end{align*}
\]
Pattern sharing: Heteroclisis

- better represented by a lattice structure than by a tree.
- subtype of overlapping. Classes can also be overlapping because of overabundance.
Table of Contents

1. Specificities of this approach

2. Background on Inflection Classes

3. How to model IC system as lattices

4. Towards a typology of IC lattices
Lattices

- More accurate representation of non canonical phenomena.
- Every node in the lattice is an IC.
Formal concept analysis: context

- Formal concept analysis: a branch of applied mathematics which deals with lattices. (Wille, 1984; Ganter, 1998; Bank, 2013-2016)
- Context: incidence table between objects and attributes.

<table>
<thead>
<tr>
<th>Context:</th>
<th>BSE~PPART</th>
<th>BSE~PST</th>
<th>PST~PPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ride</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>bite</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>forget</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

S. Beniamine, O. Bonami
Formal concept analysis: concept

- **Concept**: A set of objects and a set of attributes, all objects have in common exactly these attributes, all attributes are shared by exactly these objects.

**Concept**: \( \langle \{\text{bite, forget}\}, \{X \sim X_\eta\} \rangle \)

<table>
<thead>
<tr>
<th>Context:</th>
<th>BSE~PPART</th>
<th>BSE~PST</th>
<th>PST~PPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive</td>
<td>(X_aC \sim XiC_\eta)</td>
<td>(X \sim X_\eta)</td>
<td></td>
</tr>
<tr>
<td>ride</td>
<td>(X_eC \sim XdC_\eta)</td>
<td>(X \sim X_\eta)</td>
<td></td>
</tr>
<tr>
<td>bite</td>
<td>(XaC \sim XeC_\eta)</td>
<td>(X \sim X_\eta)</td>
<td></td>
</tr>
<tr>
<td>forget</td>
<td>(XbC \sim Xn)</td>
<td>(X \sim X_\eta)</td>
<td></td>
</tr>
</tbody>
</table>

S. Beniamine, O. Bonami
Formal concept analysis: concept

- **Concept**: A set of objects and a set of attributes, all objects have in common exactly these attributes, all attributes are shared by exactly these objects.

**Concept**: \( \langle \{ \text{drive, ride} \}, \{ \text{XaiC} \sim \text{XiC}\eta, \text{XaiC} \sim \text{XeoC}, \text{XeoC} \sim \text{XiC}\eta \} \rangle \)

<table>
<thead>
<tr>
<th></th>
<th>BSE~PPART</th>
<th>BSE~PST</th>
<th>PST~PPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>XaiC~XiC\eta</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>XeoC~XaoC</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>XaiC~XiC</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>XeoC~XaoC</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>XaiC~XiC</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>XeoC~XaoC</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>X~X\eta</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
</tbody>
</table>

**Context**:

<table>
<thead>
<tr>
<th></th>
<th>drive</th>
<th>ride</th>
<th>bite</th>
<th>forget</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>ride</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>bite</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td>forget</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
</tbody>
</table>
Formal concept analysis: concept

- **Concept**: A set of objects and a set of attributes, all objects have in common exactly these attributes, all attributes are shared by exactly these objects.

**Concept**: $\langle \{\text{drive, ride, bite}\}, \{X_{a\text{i}C}\sim X_{iC}\eta\}\rangle$

<table>
<thead>
<tr>
<th></th>
<th>BSE~PPART</th>
<th>BSE~PST</th>
<th>PST~PPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{a\text{i}C}\sim X_{iC}\eta$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$X_{e\text{C}}\sim X_{dC}\eta$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$X_{a\text{i}C}\sim X_{\text{eC}}$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$X_{a\text{i}C}\sim X_{\text{dC}}$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$X_{e\text{C}}\sim X_{\text{dC}}$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$X_{eS_{\text{C}}}\sim X_{\text{iC}}\eta$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
<tr>
<td>$X\sim X\eta$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
</tr>
</tbody>
</table>

- drive
- ride
- bite
- forget
Formal concept analysis: concept

- **Concept**: A set of objects and a set of attributes, all objects have in common exactly these attributes, all attributes are shared by exactly these objects.

<table>
<thead>
<tr>
<th>Concept:</th>
<th>( \langle { \text{bite} }, { X_{\text{airC}} \sim X_{\text{IrC}n}, X_{\text{airC}} \sim X_{\text{IrC}}, X \sim X_{\eta} } \rangle )</th>
</tr>
</thead>
</table>

**Context:**

<table>
<thead>
<tr>
<th></th>
<th>BSE(\sim)PPART</th>
<th>BSE(\sim)PST</th>
<th>PST(\sim)PPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ride</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>bite</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>forget</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
Formal concept analysis: concept

- **Concept**: A set of objects and a set of attributes, all objects have in common exactly these attributes, all attributes are shared by exactly these objects.

<table>
<thead>
<tr>
<th>Concept:</th>
<th>(\langle{\text{forget}}, {X_{\varepsilon}C \sim X_{\delta}C, X_{\varepsilon}C \sim X_{\psi}C, X \sim X_{\eta}}\rangle)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Context:</th>
<th>BSE~PPART</th>
<th>BSE~PST</th>
<th>PST~PPART</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>ride</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>bite</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>forget</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formal concept analysis: lattice

- Lattice: Set of concepts ordered by inclusion:
  \[ \langle x, y \rangle < \langle x_1, y_1 \rangle \text{ iff } x \subset x_1 \iff y \supset y_1 \]

\[
\begin{align*}
\top & \quad \text{supremum} \\
\langle \{\text{drive, ride, bite}\}, \{\text{XaiC} \sim \text{XtiCn}\} \rangle & \quad \langle \{\text{bite, forget}\}, \{\text{X} \sim \text{Xn}\} \rangle \\
\langle \{\text{drive, ride}\}, \{\text{XaiC} \sim \text{XtiCn}, \text{XaiC} \sim \text{XoC}, \text{XoC} \sim \text{XtiCn}\} \rangle & \quad \langle \{\text{bite}\}, \{\text{XaiC} \sim \text{XtiCn}, \text{XaiC} \sim \text{XoC}, \text{X} \sim \text{Xn}\} \rangle \\
\langle \{\text{forget}\}, \{\text{XeC} \sim \text{XoCn}, \text{XeC} \sim \text{XoC}, \text{X} \sim \text{Xn}\} \rangle
\end{align*}
\]

\[ \bot \quad \text{infimum} \]
Formal concept analysis: lattice

- Lattice: Set of concepts ordered by inclusion:
  \[ \langle x, y \rangle < \langle x_1, y_1 \rangle \text{ iff } x \subset x_1 \iff y \supset y_1 \]
- For legibility, we usually omit the \textit{infimum} (but not the \textit{supremum}) and label nodes without repeating information.
Formal concept analysis: lattice

- Lattice: Set of concepts ordered by inclusion:
  \[ \langle x, y \rangle < \langle x_1, y_1 \rangle \text{ iff } x \subset x_1 \iff y \supset y_1 \]

- For legibility, we usually omit the \textit{infimum} (but not the \textit{supremum}) and label nodes without repeating information.

- This reads like a monotonic multiple inheritance hierarchy.
# Table of Contents

1. Specificities of this approach

2. Background on Inflection Classes

3. How to model IC system as lattices

4. Towards a typology of IC lattices
Datasets

- Data: Paradigm tables contain phonemically transcribed forms.
- **English**: CELEX2 dataset (Baayen, Piepenbrock, and Gulikers, 1995), with partial manual validation (6064 verbal entries).
- **French**: Flexique (Bonami, Caron, and Plancq, 2014) (5258 verbal entries).
The lattices

Excerpt of the English data for: bite, forget, beget, ride, drive, abide
The lattices
The lattices
Identical structure

*Canonical inflectional classes realize the same morphosyntactic or morphosemantic distinctions (they are of the same structure).*

Corbett (2009), criterion 2

- **Defective** classes might otherwise be identical to other microclasses and thus be placed higher in the hierarchy.
Identical structure

*Canonical inflectional classes realize the same morphosyntactic or morphosemantic distinctions (they are of the same structure).*

Corbett (2009), criterion 2

- **Defective** classes might otherwise be identical to other microclasses and thus be placed higher in the hierarchy.
- **Overabundant** classes might share these patterns with other microclasses, and thus be placed lower in the hierarchy.
Chains and atoms

- An IC system that only deviates from the canonical ideal by presenting overabundance and/or defectivity can take the form of a chain.

<table>
<thead>
<tr>
<th></th>
<th>p₁</th>
<th>p₂</th>
<th>p₃</th>
<th>p₂'</th>
<th>p₃'</th>
</tr>
</thead>
<tbody>
<tr>
<td>class 1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>defective</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overabundant</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

defective
class 1
overabundant
Chains and atoms

- An IC system that only deviates from the canonical ideal by presenting overabundance and/or defectivity can take the form of a **chain**.
- **Atoms**: nodes that are right above the infimum.

Because of overabundance and defectivity, microclasses are not always atoms.

```
<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p2'</th>
<th>p3'</th>
</tr>
</thead>
<tbody>
<tr>
<td>class1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>defective</td>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>overabundant</td>
<td>×</td>
<td>×</td>
<td></td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
```
Microclasses

- Proportion of microclasses that are atoms

<table>
<thead>
<tr>
<th>Data</th>
<th>Lexemes</th>
<th>Microclasses</th>
<th>Atoms</th>
<th>Defective</th>
<th>Overabundant</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>6064</td>
<td>123</td>
<td>91</td>
<td>9</td>
<td>88</td>
</tr>
<tr>
<td>French</td>
<td>5258</td>
<td>109</td>
<td>85</td>
<td>39</td>
<td>35</td>
</tr>
</tbody>
</table>

S. Beniamine, O. Bonami
Pattern sharing

*In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.*

Corbett (2009), criterion 1

- Canonical situation: a partition of microclasses (plus supremum).

Canonical inflection classes

![Diagram of canonical inflection classes](image-url)
**Pattern sharing**

*In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.*

Corbett (2009), criterion 1

- Canonical situation: a partition of microclasses (plus supremum).
- The maximum possible lattice given some atoms corresponds to the power set over the atoms.

---

Canonical inflection classes

```
  +---+
  |   |
  +---+
```

Maximum pattern sharing across classes

```
  +---+  +---+  +---+  +---+  +---+  +---+  +---+  +---+  +---+  +---+
  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
  +---+  +---+  +---+  +---+  +---+  +---+  +---+  +---+  +---+  +---+
```

---
Pattern sharing: node density

- We evaluate the amount of sharing across microclasses by counting the number of nodes in the lattice.

<table>
<thead>
<tr>
<th>Data</th>
<th>Atoms</th>
<th>nodes</th>
<th>Min</th>
<th>Max</th>
<th>density</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>91</td>
<td>256</td>
<td>93</td>
<td>$&gt; 2 \times 10^{27}$</td>
<td>$6.58 \times 10^{-26}$</td>
</tr>
<tr>
<td>French</td>
<td>85</td>
<td>4027</td>
<td>87</td>
<td>$&gt; 3 \times 10^{25}$</td>
<td>$1.01 \times 10^{-22}$</td>
</tr>
</tbody>
</table>
Pattern sharing: structural properties

*In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.*

Corbett (2009), criterion 1

- **Overlapping**: A node of the lattice that inherits patterns from at least two nodes that are not themselves in hierarchical relation.
Pattern sharing: structural properties

In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.

Corbett (2009), criterion 1

- **Overlapping**: A node of the lattice that inherits patterns from at least two nodes that are not themselves in hierarchical relation.

- **Heteroclite**: A node with overlapping for patterns of distinct pairs of cells.
Pattern sharing: structural properties

*In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.*

Corbett (2009), criterion 1

- **Overlapping**: A node of the lattice that inherits patterns from at least two nodes that are not themselves in hierarchical relation.
- **Heteroclite**: A node with overlapping for patterns of distinct pairs of cells.
- **Sharing without overlapping** is tree-shaped
Pattern sharing: structural properties

In the canonical situation, forms differ as consistently as possible across inflectional classes, cell by cell.

Corbett (2009), criterion 1

- **Overlapping**: A node of the lattice that inherits patterns from at least two nodes that are not themselves in hierarchical relation.
- **Heteroclite**: A node with overlapping for patterns of distinct pairs of cells.
- Sharing without overlapping is tree-shaped
Microclasses

- For each microclass: is it canonical, part of a chain, a tree or overlapping?

<table>
<thead>
<tr>
<th>Data</th>
<th>Microclasses</th>
<th>Canonical</th>
<th>Chain</th>
<th>Tree</th>
<th>Overlapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>125</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>122</td>
</tr>
<tr>
<td>French</td>
<td>109</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>107</td>
</tr>
</tbody>
</table>
A quantitative interpretation: overlapping

- A tree has exactly one parent for each node (indegree: 1), and 0 for its root.
- We quantify the difference between the shape of our lattices and that of a tree by counting the mean indegree (with scaling, assuming constant number of nodes).
- The English datasets has 227 more arcs than if it was a tree.
- The French datasets has 11230 more arcs than if it was a tree.
Conclusion

▶ This view of IC is comprehensive and belongs to the abstractive perspective on morphology (Blevins, 2006).
Conclusion

- This view of IC is comprehensive and belongs to the abstractive perspective on morphology (Blevins, 2006).
- **Heteroclisis** and more generally **multiple inheritance (overlapping)** seem to be the general case rather than the exception.
Conclusion

- This view of IC is comprehensive and belongs to the *abstractive* perspective on morphology (Blevins, 2006).
- **Heteroclisis** and more generally *multiple inheritance* (overlapping) seem to be the general case rather than the exception.
  - Inflection class systems, even when traditionally analyzed as trivial, are more faithfully represented by lattices than by other classification structures.
Conclusion

- This view of IC is comprehensive and belongs to the *abstractive* perspective on morphology (Blevins, 2006).
- **Heteroclisis** and more generally *multiple inheritance (overlapping)* seem to be the general case rather than the exception.
  - Inflection class systems, even when traditionally analyzed as trivial, are more faithfully represented by lattices than by other classification structures.
- At the same time, systems we studied are far less complex than the theoretical maximum.
Conclusion

- This view of IC is comprehensive and belongs to the abstractive perspective on morphology (Blevins, 2006).
- **Heteroclisis** and more generally **multiple inheritance (overlapping)** seem to be the general case rather than the exception.
  - Inflection class systems, even when traditionally analyzed as trivial, are more faithfully represented by lattices than by other classification structures.
- At the same time, systems we studied are far less complex than the theoretical maximum.
  - This converges with research on inflectional complexity, see Carstairs-McCarthy (1991), Ackerman and Malouf (2015) and Blevins (2006).
Conclusion

▶ This view of IC is comprehensive and belongs to the abstractive perspective on morphology (Blevins, 2006).

▶ **Heteroclisis** and more generally multiple inheritance (overlapping) seem to be the general case rather than the exception.
  ▶ Inflection class systems, even when traditionally analyzed as trivial, are more faithfully represented by lattices than by other classification structures.

▶ At the same time, systems we studied are far less complex than the theoretical maximum.
  ▶ This converges with research on inflectional complexity, see Carstairs-McCarthy (1991), Ackerman and Malouf (2015) and Blevins (2006).

▶ Perspective: use our tools on a wide range of data to elaborate a typological analysis.
Thank You!
References I


Beniamine, Sarah, Olivier Bonami, and Benoît Sagot (2015). “Information-theoretic inflectional classification”. *International Quantitative Morphology Meeting 1 (IQMM1).*


Towards a typology of IC lattices

References III

Multiple inheritance hierarchies

- Phonological hierarchies of natural classes (Chomsky and Halle, 1968; Frisch, 1997).
Multiple inheritance hierarchies

- Phonological hierarchies of natural classes (Chomsky and Halle, 1968; Frisch, 1997).
- HPSG type hierarchies (Flickinger, 1987; Ginzburg and Sag, 2000).
Multiple inheritance hierarchies

- Phonological hierarchies of natural classes (Chomsky and Halle, 1968; Frisch, 1997).
- HPSG type hierarchies (Flickinger, 1987; Ginzburg and Sag, 2000).
- Nodes are ordered: (semi)-Lattices.
Comparison with default hierarchies

- Default hierarchies (Brown and Hippisley, 2012, ex. from Corbett and Fraser, 2002).

```
Bird
  has feathers, can fly
  Penguins
  cannot fly
      Percy
    |      |
    |      |
    |      |
  Robin
  Eagle
      Roderick
      Edwina
```
Comparison with default hierarchies

- Default hierarchies (Brown and Hippisley, 2012, ex. from Corbett and Fraser, 2002).
- Monotonic hierarchies: Attributes shared by all descendants, all relevant sets are explicit.