AmbiDexter

Practical Ambiguity Detection

Bas Basten
Tijs van der Storm

Centrum Wiskunde & Informatica
Why Ambiguity Detection?

- Generalized parsing (GLR, GLL, Earley, ...)
  - Modular grammar development
  - Problem: Ambiguity!

- Possible solution: disambiguation constructs
  - Priority, associativity, longest match, etc.

- How do you know all ambiguities are covered?
Ambiguity Detection

- Undecidable in general

- Various approaches
  - Approximative
  - Exhaustive

- Trade-off precision/performance

- Practical: detailed reports fast
AmbiDexter

Approximative Test

'potentially ambiguous'

Harmless Production Filter

Potentially ambiguous productions

Grammar reconstruction

Smaller grammar

Derivation Generator

'Done'

String + trees

'ambiguous'

'unambiguous'
Experimental Results*

- Real world grammars
  - Seeded ambiguity
- Time to find first ambiguity:

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Deriv gen</th>
<th>Deriv gen + filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL</td>
<td>28m26s</td>
<td>0.5s</td>
</tr>
<tr>
<td>Pascal</td>
<td>32s</td>
<td>4s</td>
</tr>
<tr>
<td>C</td>
<td>4h30m</td>
<td>8.0s</td>
</tr>
<tr>
<td>Java</td>
<td>25h</td>
<td>22m52s</td>
</tr>
</tbody>
</table>

* Excerpt from Basten & Vinju – LDTA 2010
Figures of derivation generator AMBER