AmbiDexter

Practical Ambiguity Detection

Bas Basten
Tijs van der Storm

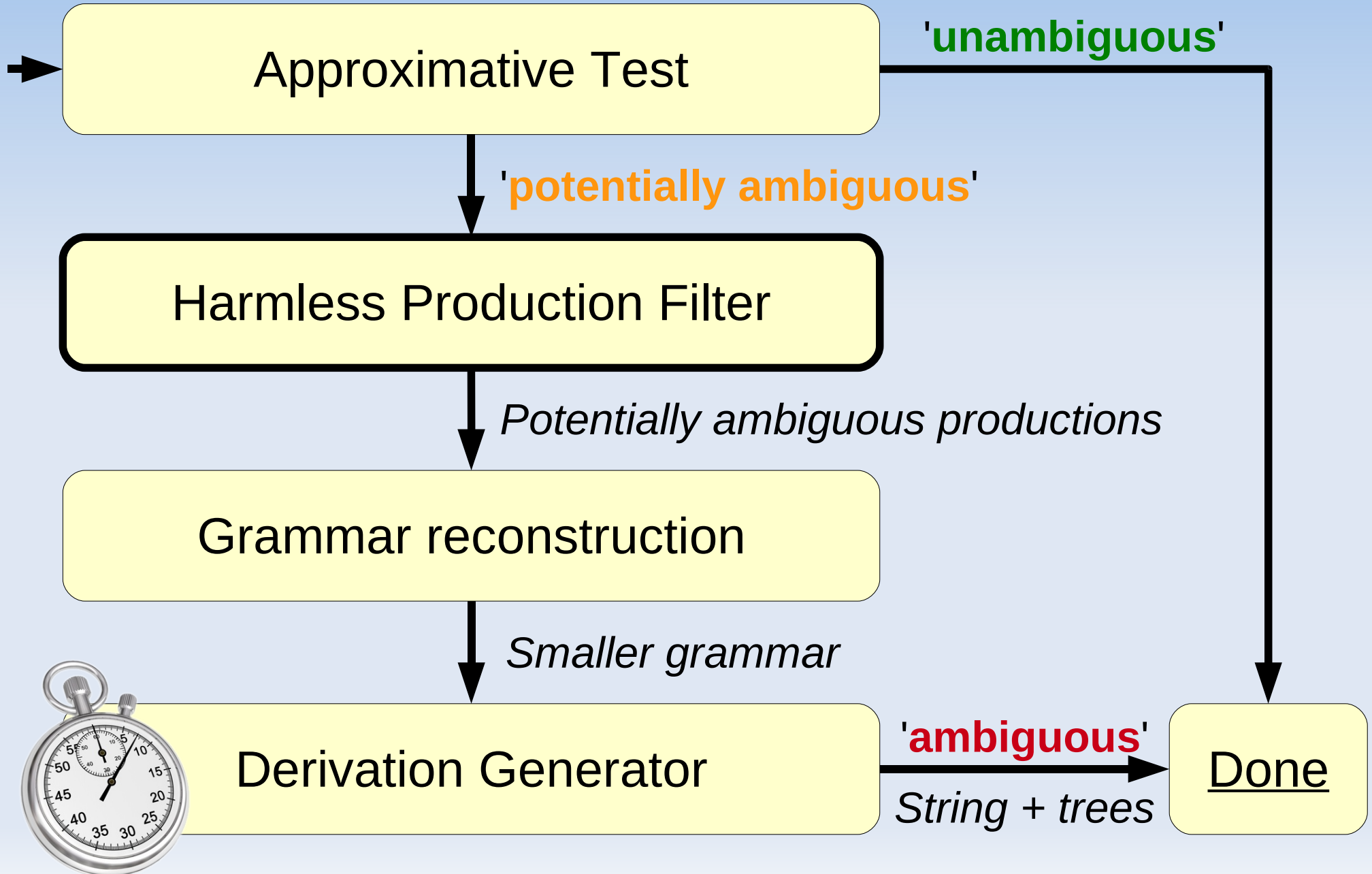Centrum Wiskunde & Informatica

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: <u>detailed</u> reports <u>fast</u>

AmbiDexter

Approximative Test

'**unambiguous**'

'**potentially ambiguous**'

Harmless Production Filter

Potentially ambiguous productions

Grammar reconstruction

Smaller grammar

Derivation Generator

'**ambiguous**'

String + trees

Done

Experimental Results*

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

Grammar	Deriv gen	Deriv gen + filtering
SQL	28m26s	0.5s
Pascal	32s	4s
C	4h30m	8.0s
Java	25h	22m52s

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