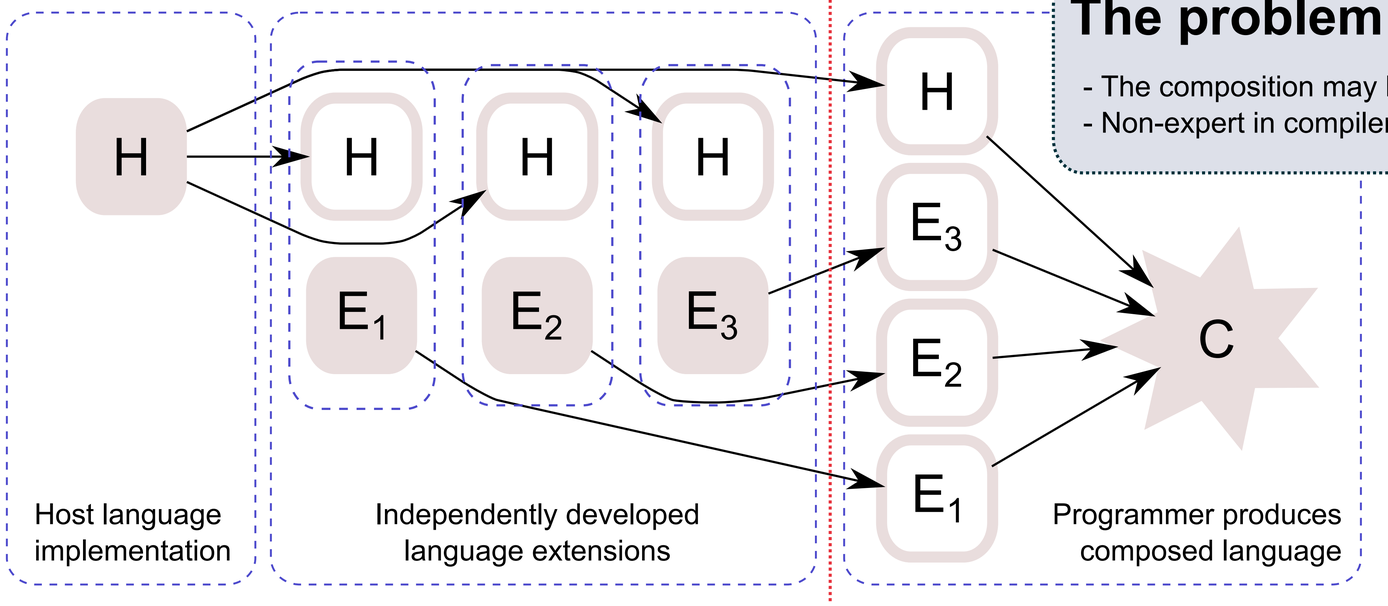
Modular well-definedness analysis for attribute grammars

Ted Kaminski Eric Van Wyk University of Minnesota

The language extension model



The problem

- The composition may be broken!
- Non-expert in compilers can't fix it.

See our talk on Friday

The full solution consists of:

- Forwarding

Van Wyk. de Moor. Backhouse. Kwiathowski. "Forwarding in attribute grammars for modular language design." CC '02

- Copper's modular analysis for syntax

Schwerdfeger. Van Wyk. "Verifiable composition of deterministic grammars." PLDI '09

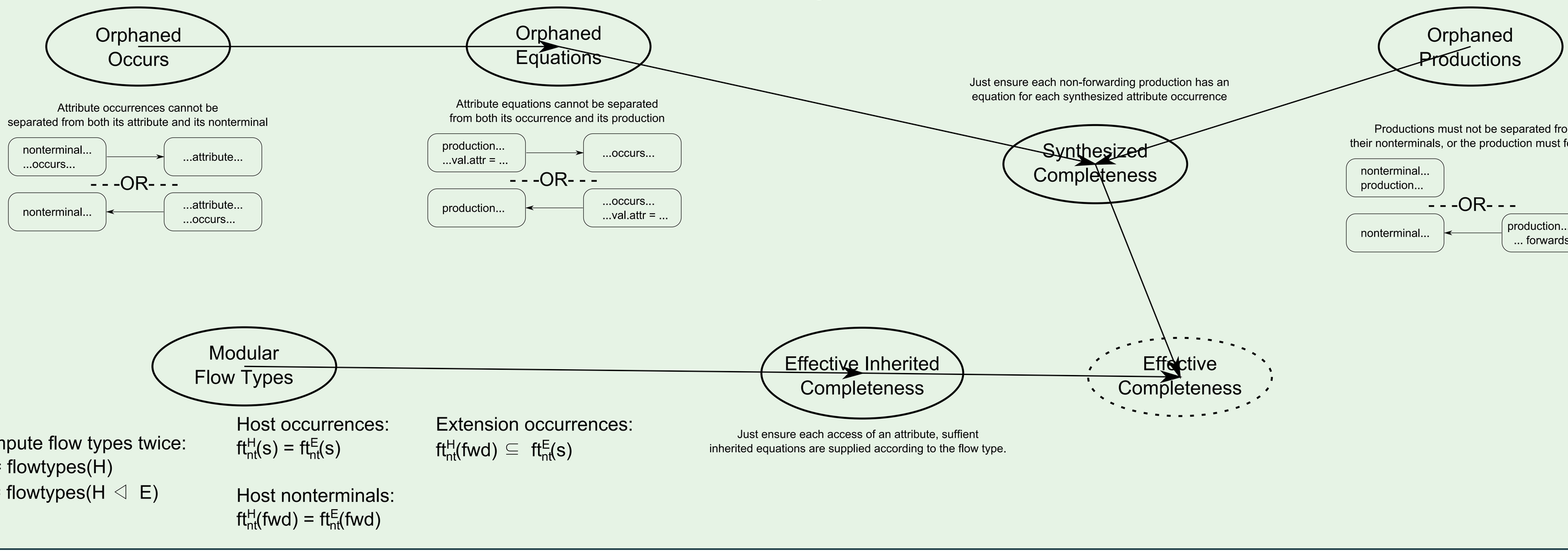
- Silver's modular analysis for semantics

Kaminski. Van Wyk. "Modular well-definedness analysis for attribute grammars." SLE '12 (also our topic on this poster)

The solution

- Analyze each extension individually
  - $H \triangleleft E_1$
  - $H \triangleleft E_2$
  - $H \triangleleft E_3$
- Analysis must ensure properties about the composed language
  - $H \triangleleft (E_1 \uplus_{\emptyset} E_2 \uplus_{\emptyset} E_3)$
- Extension developers deal with errors
- User has no conflict errors

The analysis



Compute flow types twice:

ft^H = flowtypes(H)

ft^E = flowtypes(H $\triangleleft$ E)

Host occurrences:

$ft^H_{nt}(s) = ft^E_{nt}(s)$

Host nonterminals:

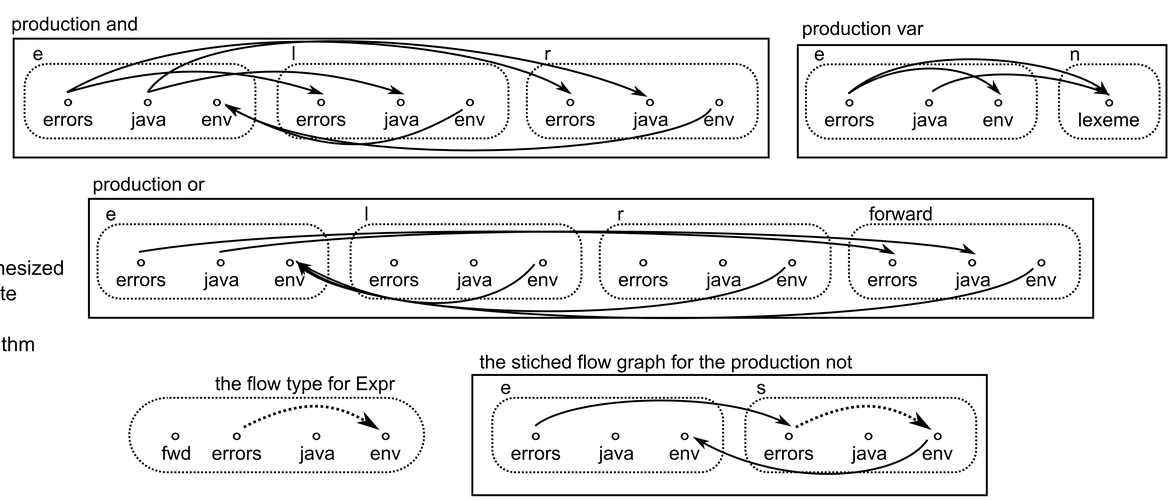
$ft^H_{nt}(fwd) = ft^E_{nt}(fwd)$

Extension occurrences:

$ft^H_{nt}(fwd) \subseteq ft^E_{nt}(s)$

Flow Types

- A flow type f_{nt} is a function mapping synthesized attributes to a set of their inherited attribute dependencies
- Flow types can be computed by an algorithm similar to Knuth's algorithm for testing non-circularity.



Evaluation

Questions:

1. Can we get grammars to pass this analysis?
2. Can we still extend syntax and semantics given the restrictions?

Tested by getting Silver itself to pass the analysis

Silver's extensions

- Unit testing (syntactic)
- Convenience (syntactic)
- Java translation (semantic)
- "Easy terminals" (both)

Results: Yes to both!

- Biggest problem was reference attributes. Worked around with error equations.
- Analysis identified many bits of code that were already in need of refactoring.