A Taste of Stratego/XT

Karl Trygve Kalleberg

karltk@ii.uib.no

University of Bergen

What is Stratego/XT?

- Stratego/XT is a *language* and *toolset* for constructing program transformation systems.
- Stratego – the language – provides *rewrite rules* and *strategies* for implementing program transformation *components*.
- XT – the toolset – contains a *collection of reusable components*, and *small, declarative languages* for generating custom components.
- Together, they provide a *framework* for constructing *modular*, *stand-alone* program transformation systems in with *precise*, *high-level* languages that are compiled to *efficient*, *portable* binaries.

What is it Good For?

- Stratego/XT has been applied in compiler construction, interpretation, static analysis, partial evaluation, construction of extensible languages, implementing AOP, “classical” optimization, implementing domain-specific languages, domain-specific (high-level) optimization, active libraries, generative programming, document transformation, vectorization, and more.
- Used by research groups (at e.g. Universitetet i Bergen, Universiteit Utrecht, University of Waterloo, EPITA Research and Development Labs, Université René Descartes (Paris 5), LIP6
- Some use in industry (at e.g. Philips, Lucent, Lockheed)

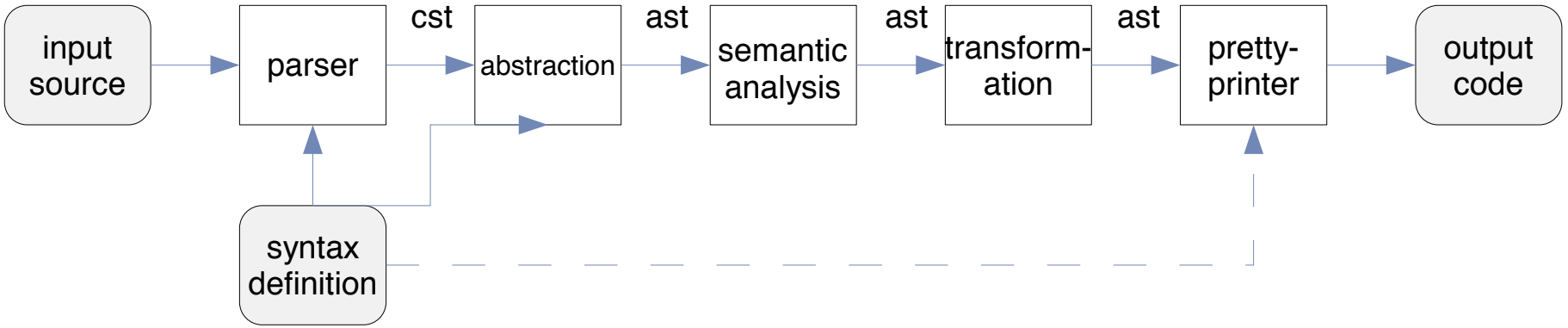
Why Bother?

- *Because we can!*
- We* wanted a *high-level*, *efficient*, *precise, rule-based* language for program transformation that provided *exact control* over the rule applications ⇒ Stratego.
- We wanted a *scalable, componentized* architecture for constructing and *evolving* program transformation systems from *reusable* components ⇒ XT.
- Both problems contained many interesting research questions.

**) we = Eelco Visser*

How is it Realized?

- Stratego/XT is typically (but not only) used to derive *pipelined* transformation systems, where
  - each box is an *XT component;*
  - each line is an *XTC composition;*
  - and components communicate by passing structured data in the form of *annotated terms (= compressed trees)*.



XT

- XT is a collection of
  - Tools
    - `xml-tools`; Converters to/from XML.
    - `format-check`; Well-formedness checker of terms, akin to DTDs/schemas.
    - `xtc`; Component repository manager.
    - `pp-aterm`, `visamb`; ATerm pretty-printer and ambiguation visualizer.
  - Declarative Languages
    - SDF; a *modular*, *declarative* syntax definition formalism, that comes with a *scannerless*, *generalized* LR parser.
    - Box; a *layout language* for converting terms (trees, ASTs) to readable program text.
    - Parse Unit; a tiny language for writing *parser unit tests*.

Stratego

- Important language features:
  - The unit of transformation is the *conditional rewrite rule*.
  - These are composed using *programmable rewrite strategies*.
  - Context information is captured using *dynamic rules*.
  - *Concrete syntax* is optionally used for writing patterns
    - E.g., rule left- and right-hand sides.
  - Sets of rules and strategies are bundled together in *modules*.
- The result:
  - a *high-level*, *domain-specific* language for rewriting of trees, compilable to efficient binaries.

Stratego Example

```
EvalBinOp :
  |[ i + j ]| -> |[ k ]| where <add>(i,j) => k

EvalIf :
  |[ if 0 then e1 else e2 ]| -> |[ e2 ]|

constfold =
  all(constfold); try(EvalBinOp <+ EvalIf)

pe = PropConst <+ pe-assign <+ pe-declare
     <+ pe-let <+ pe-if <+ pe-while <+ pe-for
     <+ all(pe); try(EvalBinOp)

pe-assign =
  |[ x := <pe => e> ]|
  ; if <is-value> e
    then rules( PropConst.x :  |[ x ]| -> |[ e ]| )
    else rules( PropConst.x :- |[ x ]| ) end

pe-declare =
  ? |[ var x ta ]|
  ;  rules( PropConst+x :- |[ x ]| )

pe-if =
  |[ if <pe> then <id> else <id> ]|
  ; (EvalIf; pe
     <+ (|[ if <id> then <pe> else <id> ]|
         /PropConst\ |[ if <id> then <id> else <pe> ]|))
```

Highlights

- The Stratego language is about 8 years old.
- Several larger systems have been built with Stratego/XT
  - CodeBoost, Transformers, Proteus are transformation frameworks for C++.
  - Dryad is a transformation framework for Java 1.5 (in progress).
  - OctaveC is a compiler for Octave (Matlab clone).
- Additional support tools are available
  - Spoofax (Stratego editor for Eclipse), Emacs mode, xDoc source code documentation generator, interactive Stratego interpreter.
  - Tutorial, collection of detailed examples.
  - On line reference manual (work in progress), API docs.

Interested?

- Stratego/XT 0.16 was released last Friday
  - Free (as in both speech and beer)
  - Available for Linux, Unix, OSX and Windows.
  - Comes with tutorial, examples and API documentation.
- Stratego/XT tutorial Tuesday (tomorrow!) afternoon.