DeFacto
Relational Fact Extraction from Source Code

“Fact Extraction without the Boilerplate”

Fact Extraction for:
- Reverse Engineering
- Software Analysis
- Refactoring
- Visualization
- ...

Features:
- Relations as Uniform Representation
- Concise & Declarative Notation
- Local Fact Extraction
- Global Enrichment & Analysis

Software System
- C code
- Java code
- Python code

Fact Extraction

Implicit Variable Declaration & Matching
Powerful Convenience Functions on Lists and Optionals

Fact Annotations on EBNF Grammar Rules

```
{" Statement* " }" -> Block {
    fact(ENTRY, Block, first(Statement-list)),
    fact(EXIT, Block, last(Statement-list)),
    fact(SUCC, next(Statement-list))
}

"while" "{ Expr "}" Statement -> Statement {
    fact(ENTRY, Statement0, Expr),
    fact(EXIT, Statement0, Expr),
    fact(SUCC, next(Expr, Statement1, Expr))
}
...```

Implicit Tree Traversal

List Construction

Relational Facts

Fact Enrichment

Relational Calculations on Extracted Facts

```
type rel[Statement, Statement] Graph

Graph ControlFlow = { <S1, S4> |<Statement S2, Statement S3> : SUCC,
Statement S1 : reachBottom(EXIT, S2),
Statement S4 : reachBottom(ENTRY, S3) }
```

Global Scope on Facts

Strongly Typed

Built-in Functions for Reachability Queries

Analysis

Analysis Results