KeY Version for MISRA C

Daniel Larsson

KeY Symposium

Göteborg, June 2005

CEDES

- CEDES
  (**C**ost **E**fficient **D**ependable **E**lectronic **S**ystems)
- Software-based methods for fault tolerance
  & fault handling
- “Our” work package:
  - KeY version for MISRA C programs
  - Symbolic error propagation
  - Formal verification of exception handling routines

Symbolic Error Propagation

- Would complement fault injection methods
- Main problem with fault injection: **coverage**
- Idea:
  - Represent whole classes of errors in logic
  - Perform symbolic execution to ...
    - verify properties in the presence of errors
    - calculate consequences (strongest postcondition)

KeY Version for MISRA C

- Refactoring of KeY + addition of C datastructures
- Finding and integrating C front-end
- Writing parser for schemaC
- Develop and implement dynamic logic and calculus for MISRA C

Front-end for C

- Cetus
  - Implemented in Java
  - Uses ANTLR parser generator
  - Is an active project

Refactoring of Datastructures

- Should as much as possible be re-used/shared?
  - Save a lot of work
  - Avoid duplicated code

... **or** ...

- Should structures for different languages be kept separate?
  - Java semantics implicitly built-in
  - $\Rightarrow$ Bugs that are hard to find

Refactoring of Datastructures cont'd

- Should as much as possible be re-used/shared?
  - Save a lot of work
  - Avoid duplicated code

... **or** ...

- Should structures for different languages be kept separate?
  - Java semantics implicitly built-in
  - $\Rightarrow$ Bugs that are hard to find

Decision: Go for 1st approach

Refactoring of Datastructures cont'd

How general?

- First plan: Structure that allowed for addition of arbitrary OO language with imperative core
- Not worth the effort

- Existing datastructures already fairly general
- $\Rightarrow$ Go for ad-hoc approach

Refactoring of Datastructures cont'd

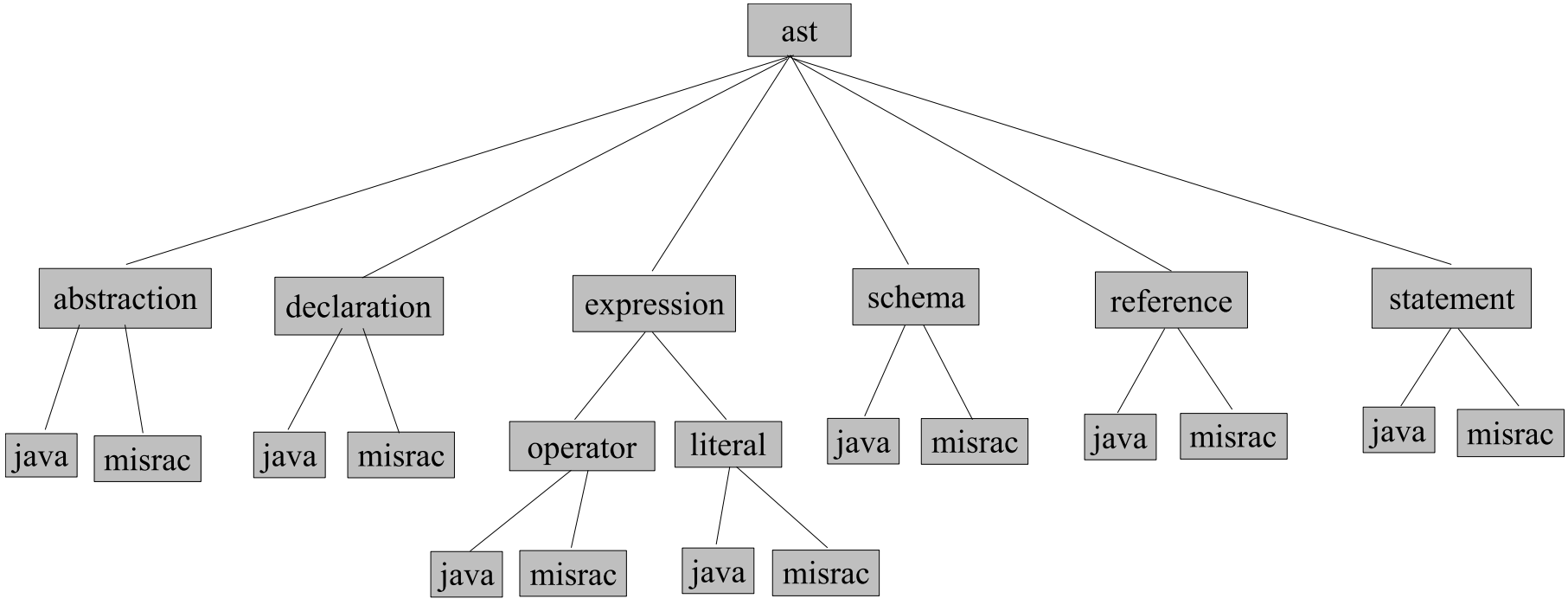
How general?

- First plan: Structure that allowed for addition of arbitrary OO language with imperative core
- Not worth the effort

- Existing datastructures already fairly general
- $\Rightarrow$ Go for ad-hoc approach

Decision: Minimal refactoring to be able to add C constructs

New Package Structure



New Package Structure cont'd