The Spoofax Name Binding Language

Name Binding

Name binding comprises the association of uses of names — such as variables, methods, and classes — to their definitions. The purpose is performing static name analysis and providing IDE services such as error checking, code navigation and code completion.

At source code level the structure of a program communicates the associations and named scopes intuitively. For example, classes open a scope for methods and fields, methods open a scope for local variables.

Navigation

Reference resolution permits IDE users to navigate code from variable, method or class use-site to definition location. Name analysis results in navigable associations.

interface Env
{
  Env add(string id, int val);
}
interface Expr
{
  int eval(Env env);
}
class Plus : Expr
{
  Expr r1; Expr r2;
  int eval(Env env) {
    return l.eval(env) + r.eval(env);
  }
}

Name Resolution

Name resolution is a program analysis that resolves names in abstract syntax trees, resulting in (some representation of) a tree with references from uses to definitions.

Error checking

Usage of names that require associations to missing or invisible definitions are marked in the editor and reported as errors.

class Let : Exp
{
  string name;
  Expr t;
  Expr body;
  class add(Env env) {
    Env newEnv = env.add(name, t.eval(env));
    return body.eval(newEnv);
  }
}

Code completion

Inline code completion provides suggestions while editing. The inline code completion service provides suggestions while editing. Name analysis is required to determine the defined entities that are applicable at the completion location.

class Let : Exp
{
  string r;
  int eval(Env env) {
    return l.eval(env) + r.eval(env);
  }
}

Classical Approaches

Existing approaches for definition of name binding and scope, use programmatic encodings of name resolution algorithms, which hide binding and scope concepts. DSLs for compiler construction have focused on reducing the overhead and accidental complexity of these programmatic encodings. Some examples:

Inference Rules

The inference rules featured in mathematical language definitions encode binding and scope by carrying around name binding environments.

Refers to

Name Binding Language

The Spoofax Name Binding Language (NBL) [5] is a metalanguage for declaratively specifying name binding and scope rules. Language engineers need not be concerned with the mechanics of name binding algorithms, but can focus on the name binding concepts of the language. NBL is integrated in the Spoofax Language Workbench (4), but aspires to be a universal language for name binding, as BNF is for syntax.

NBL currently generates language specific implementations in Stratego of a language-parametric algorithm for static name resolution, error checking and contextual content completion.


defines Field x of type t

Field(t, x) :
import Field from Class x {transitive}
defines Field x of type t

FieldAccess(expr, f) :
import Field from Class x {transitive}

Var(t, x, ...) :
defines Variable x of type t in subsequent scope

VarRef(x) :
refers to Variable x

otherwise refers to Field x

References