

AST MANIPULATION

- Replace constant variables by their value
- Simplify operations if known. E.g:
 - $a + 0 \rightarrow a$
 - $-(-a) \rightarrow a$
 - $1 * a \rightarrow a$
 - $0 * a \rightarrow 0$
 - $a^1 \rightarrow a$
- Replace references to identifiers by reference to symbol table entry in accordance with scoping rules.
- Add type casting instructions to handle implicit type casting for non-standard type combinations.
- Other insertions of meaning into AST:
 - e.g: no by statement \rightarrow by 1
 - e.g.: while() \rightarrow while(true)

SEMANTIC ERROR DETECTION

Timing of Error Detection

- **statically**: during compilation (before evaluation)
- **dynamically**: during run-time (or evaluation)

Types of Errors

- Declaration problems
 - undeclared identifiers
 - redeclared identifiers
 - identifiers out of scope
- Wrong sequences:
 - invalid # of parameters in function calls
 - invalid # of indices in array references
 - invalid fields (for structs, or methods/properties for classes)
- Flow of control problems:
 - Break/continue not inside loops
 - Return not inside functions
- Type errors:
 - in assignments
 - in return statements
 - in operations
 - in indexing and function calls
- Name-matching problems:
 - e.g in ADA:

```
begin block-name
end block-name
```
 - e.g:

```
for i from 1 to 10 do
    ...
next i
```
- Array index out of range
- Division by 0