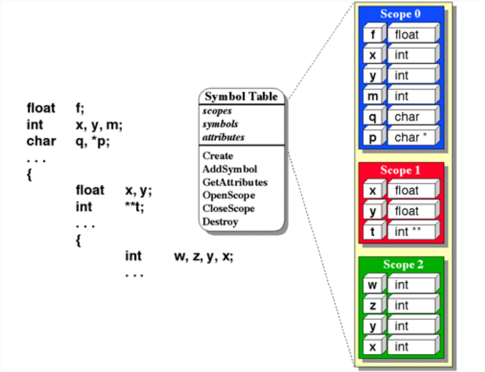
Symbol Tables in Language Translation

Brian A. Malloy

November 16, 2016



```
float   f;
int     x, y, m;
char    q, *p;
. . .
{
        float   x, y;
        int     **t;
        . . .
        {
                int     w, z, y, x;
                . . .
```

Symbol Table

scopes
symbols
attributes

Create
AddSymbol
GetAttributes
OpenScope
CloseScope
Destroy

Scope 0	
f	float
x	int
y	int
m	int
q	char
p	char *

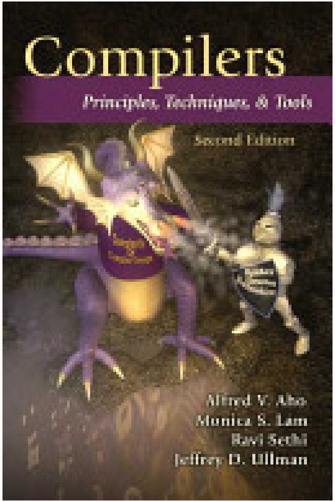
Scope 1	
x	float
y	float
t	int **

Scope 2	
w	int
z	int
y	int
x	int

Resources
Symbol Table . . .
Who Creates an Entry
Scope
Chained Symbol Tables
Implementation
Use of Symbol Tables

Slide 1 of 11
Go Back
Full Screen
Quit

1. Resources


Compilers
Principles, Techniques, & Tools
Second Edition
Alfred V. Aho
Monica S. Lam
Ravi Sethi
Jeffrey D. Ullman

2. Symbol Table Definition

Data structure used by compilers to hold information about source-program constructs.[1]

1. Information is collected incrementally during analysis phase.
2. Symbol table entries contain information about an identifier: string, type, storage location, . . .
3. Typically need to support multiple declarations of the same identifier.

[1] Compilers Principles, Techniques, & Tools, Aho, Lam, Sethi, Ullman, 2007

Resources
Symbol Table . . .
Who Creates an Entry
Scope
Chained Symbol Tables
Implementation
Use of Symbol Tables
◀◀ ▶▶
◀ ▶
Slide 3 of 11
Go Back
Full Screen
Quit

3. Who Creates an Entry

- The parser knows about the semantic structure of a program, and is in better position to distinguish different decls of an id.
- In some cases, a scanner can create a symbol table entry as soon as it sees the name.
- Only parser knows about type & scope.

4. Scope

The scope of a decl is the portion of a program to which the decl applies.[2]

- Typical implementation: separate symbol table for each scope.
- A program block has its own symbol table, with an entry for each decl in the block.
- A class has its own symbol table, with an entry for each data attribute and method.

[2] Compilers Principles, Techniques, & Tools, Aho, Lam, Sethi, Ullman, 2007

CLEMSON
UNIVERSITY

Resources

Symbol Table . . .

Who Creates an Entry

Scope

Chained Symbol Tables

Implementation

Use of Symbol Tables

◀◀ ▶▶

◀ ▶

Slide 5 of 11

Go Back

Full Screen

Quit

4.1. Block Structure

Consider the following code segment:

```
{ int x; char y; { bool y; x; y; } x; y; }
```

Let's rewrite the above segment with decls removed and each use is an identifier followed by its type:

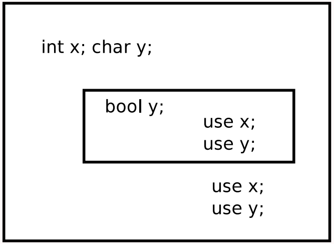
```
{ { x:int; y:bool; } x:int; y:char; }
```

- This example illustrates nested scopes: the same identifier can appear in different scopes.
- Common names, such as i & x, can have multiple uses; subclasses can override method names.

4.2. Symbol Tables & Scopes

- Symbol tables must permit multiple variable use

- Most closely nested rule: an id x is in the scope of the most-closely nested decl of x.



int x; char y;

bool y;
use x;
use y;

use x;
use y;

Resources
Symbol Table . . .
Who Creates an Entry
Scope
Chained Symbol Tables
Implementation
Use of Symbol Tables

◀◀ ▶▶
◀ ▶

Slide 7 of 11

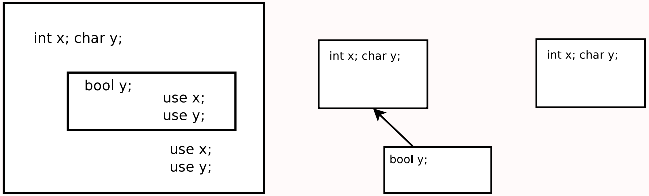
Go Back

Full Screen

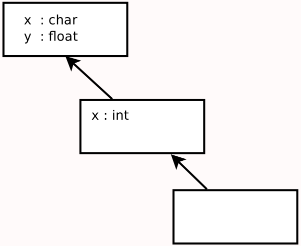
Quit

5. Chained Symbol Tables

```
{ int x; char y; { bool y; x; y; } x; y; }
```

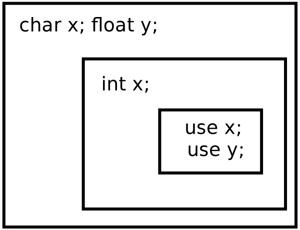


5.1. Another example of chain



x : char
y : float

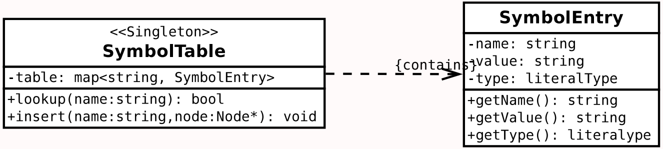
x : int



```
char x; float y;
    int x;
        use x;
        use y;
```

Resources
Symbol Table . . .
Who Creates an Entry
Scope
Chained Symbol Tables
Implementation
Use of Symbol Tables
Go Back
Full Screen
Quit

6. Implementation



<<Singleton>>
SymbolTable

-table: map<string, SymbolEntry>

+lookup(name:string): bool
+insert(name:string,node:Node*): void

{contains}

SymbolEntry

-name: string
-value: string
-type: literalType

+getName(): string
+getValue(): string
+getType(): literalype

Caution: implementation of symbol table as `map<string, Node*>`

CLEMSON UNIVERSITY

Resources

Symbol Table . . .

Who Creates an Entry

Scope

Chained Symbol Tables

Implementation

Use of Symbol Tables

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Quit

7. Use of Symbol Tables

- Role is to pass info from decl to use
- A semantic action puts info about an identifier into the symbol table
- A production such as factor $\rightarrow$ id gets information about an *id* from the symbol table
- Translation of E_1 op E_2 may depend on type of E_1 and E_2.

Resources
Symbol Table . . .
Who Creates an Entry
Scope
Chained Symbol Tables
Implementation
Use of Symbol Tables
◀◀ ▶▶
◀ ▶
Slide 11 of 11
Go Back
Full Screen
Quit