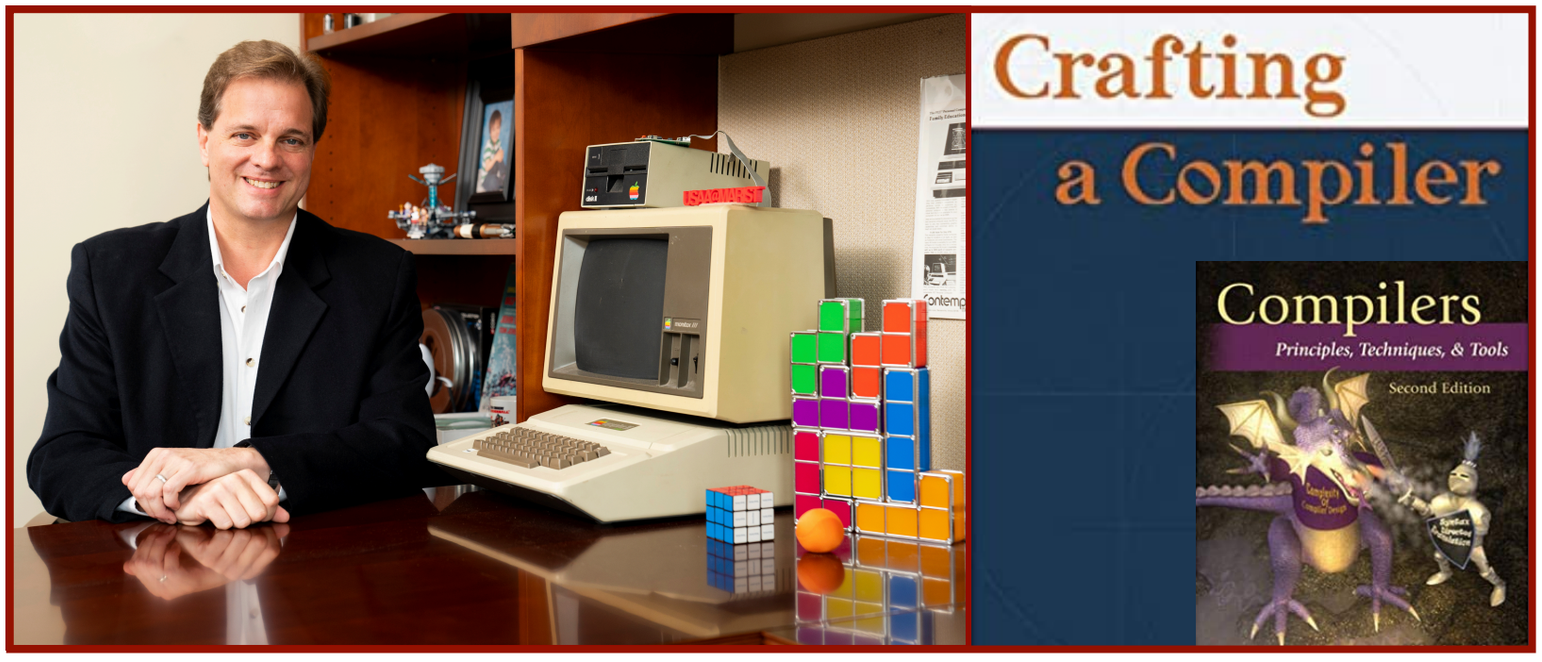
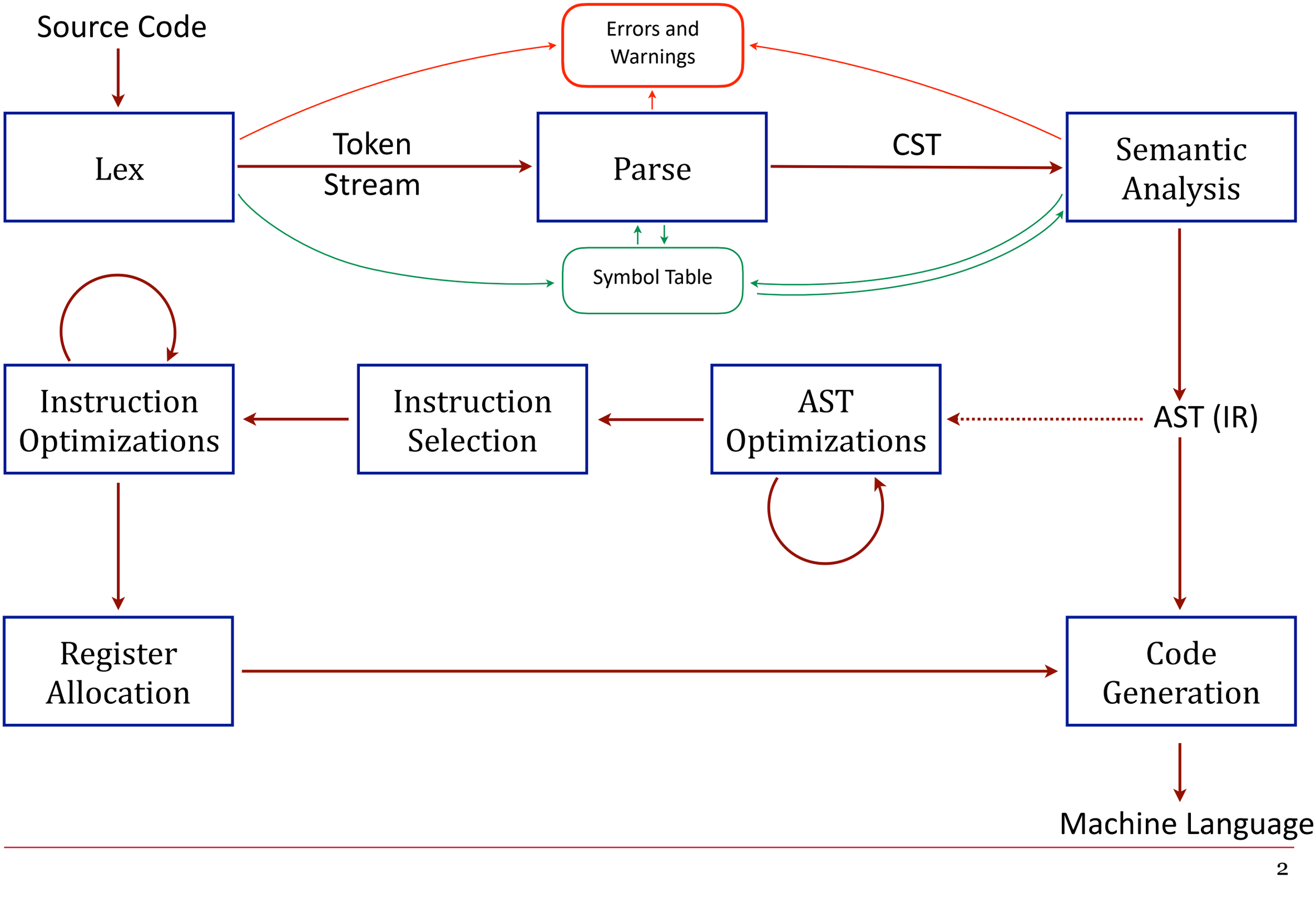
Code Generation

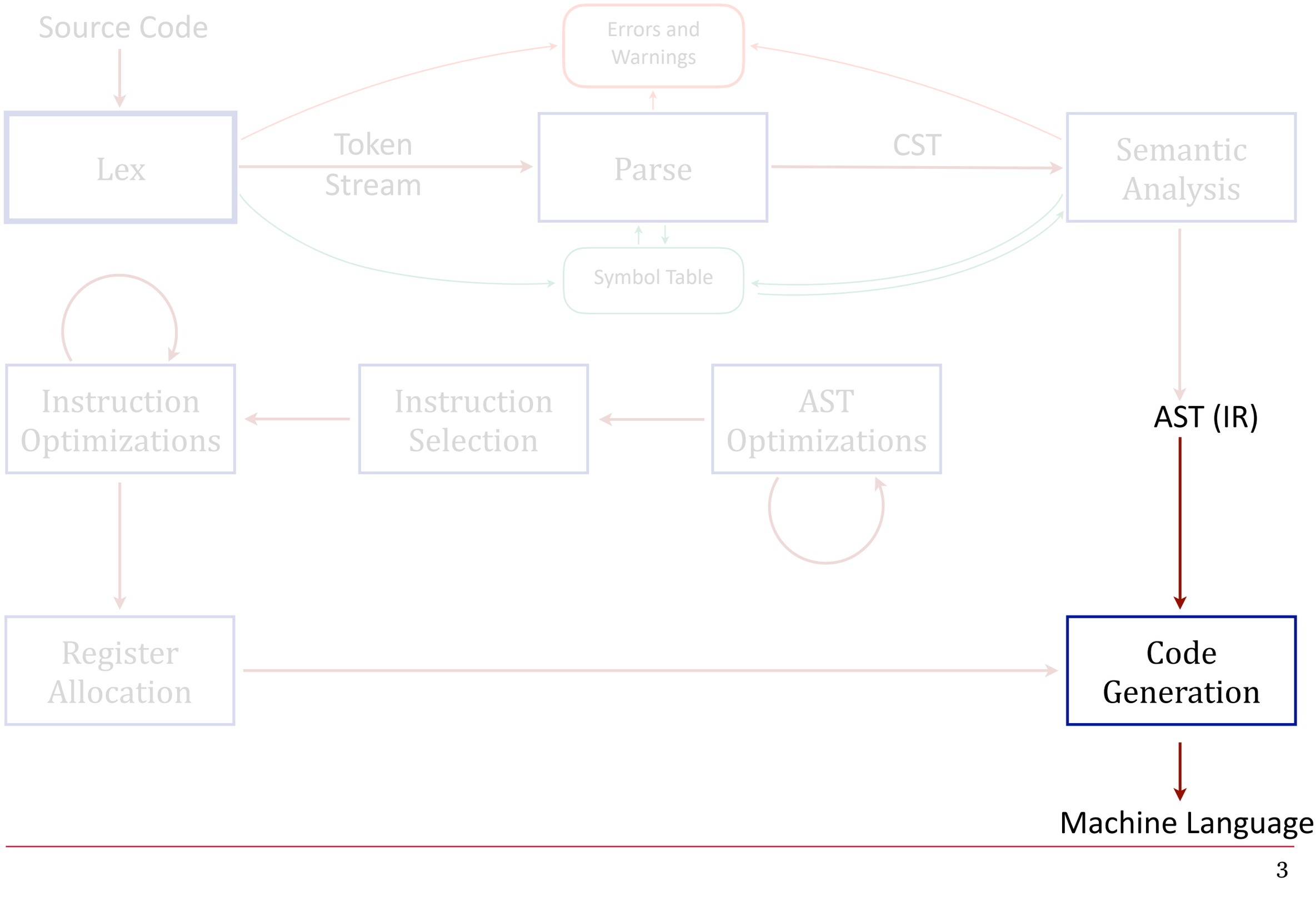


Alan G. Labouseur, Ph.D.
Alan.Labouseur@Marist.edu

Compiler — High Level View



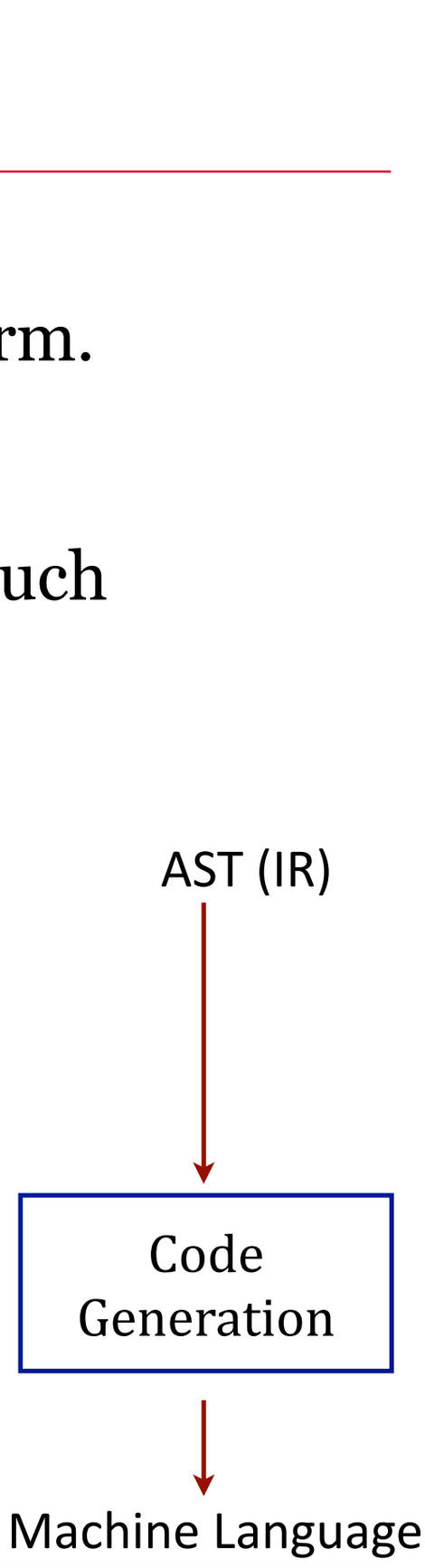
High Level View: Code Generation



High Level View: Code Generation

Code Generation

- Generate machine language code for the target platform.
- We're limited to 256 bytes
- Aside from the size limit, there are no errors to catch because lex, parse, and semantic analysis have done such a great job.
- Focus on meaning: machine language code.



Code Generation

Target Platform: a tiny subset of 6502 Machine Language

Description	Op Code	Mnemonic	Example Assembly	Example Disassembly
Load the accumulator with a constant	A9	LDA	LDA #$07	A9 07
Load the accumulator from memory	AD	LDA	LDA $0010	AD 10 00
Store the accumulator in memory	8D	STA	STA $0010	8D 10 00
Add with carry Adds contents of an address to the contents of the accumulator and keeps the result in the accumulator	6D	ADC	ADC $0010	6D 10 00
Load the X register with a constant	A2	LDX	LDX #$01	A2 01
Load the X register from memory	AE	LDX	LDX $0010	AE 10 00
Load the Y register with a constant	A0	LDY	LDY #$04	A0 04
Load the Y register from memory	AC	LDY	LDY $0010	AC 10 00
No Operation	EA	NOP	EA	EA
Break (which is really a system call)	00	BRK	00	00
Compare a byte in memory to the X reg Sets the Z (zero) flag if equal	EC	CPX	EC $0010	EC 10 00
Branch *n* bytes if Z flag = 0	D0	BNE	D0 $EF	D0 EF
Increment the value of a byte	EE	INC	EE $0021	EE 21 00
System Call #$01 in X reg = print the integer stored in the Y register. #$02 in X reg = print the 00-terminated string stored at the address in the Y register.	FF	SYS		FF

Code Generation

Target Platform

We saw this in our first class: before and after compilation.

Before: Three higher-level languages

```
10 A = 3
20 X = 1
30 print X
40 X = X + 1
50 IF X <> A THEN GOTO 30
60 PRINT "DONE"
```

```
int limit = 3;
int val = 1;
repeat
{
   console.write(val);
  ++val;
} until (val == limit)
console.write("DONE");
```

```
var limit = 3;
var val = 1;
do
{
   alert(val);
   ++val;
} while (val <> limit);
alert("DONE");
```

During: Intermediate Representation (6502a Assembly. Ms-IL and Java bytecodes are similar.)

```
lda #$3          Acc = 3                  0000        LDA #$03        A9 03
sta $0041        Mem[41] = 3              0002        STA $0041       8D 41 00

lda #$1          Acc = 1                  0005        LDA #$01        A9 01
sta $0040        Mem[40] = 1              0007        STA $0040       8D 40 00

loop ldy $0040   Y = Mem[40]              000A LOOP   LDY $0040       AC 40 00
     ldx #$01    X = 1                    000D        LDX #$01        A2 01
     sys         System Call              000F        SYS             FF

     inc $0040   Mem[40]++                0010        INC $0040       EE 40 00
     ldx $0040   X = Mem[40]              0013        LDX $0040       AE 40 00

     cpx $0041   Z bit = (x == Mem[41])   0016        CPX $0041       EC 41 00
     bne loop    if z == 0 goto loop      0019        BNE LOOP        D0 EF

lda #$44         Acc = $44 ("D")          001B        LDA #$44        A9 44
sta $0042        Mem[42] = $44            001D        STA $0042       8D 42 00
lda #$4F         Acc = $4F ("O")          0020        LDA #$4F        A9 4F
sta $0043        Mem[43] = $4F            0022        STA $0043       8D 43 00
lda #$4E         Acc = $4E ("N")          0025        LDA #$4E        A9 4E
sta $0044        Mem[44] = $4E            0027        STA $0044       8D 44 00
lda #$45         Acc = $45 ("E")          002A        LDA #$45        A9 45
sta $0045        Mem[45] = $45            002C        STA $0045       8D 45 00
lda #$00         Acc = $00 (null)         002F        LDA #$00        A9 00
sta $0046        Mem[46] = $00            0031        STA $0046       8D 46 00
ldx #$02         X = 2                    0034        LDX #$02        A2 02
ldy #$42         Y = $42 (address)        0036        LDY #$42        A0 42
sys              System call              0038        SYS             FF

brk              Break                    0039        BRK             00
```

After: Machine Language

```
A9 03 8D 41 00 A9 01 8D 40 00 AC 40 00 A2 01 FF EE 40 00 AE 40 00 EC 41 00 D0
EF A9 44 8D 42 00 A9 4F 8D 43 00 A9 4E 8D 44 00 A9 45 8D 45 00 A9 00 8D 46 00
A2 02 A0 42 FF 00
```

Code Generation

Target Platform: a tiny subset of 6502 Machine Language

Get the full document with examples at
https://www.labouseur.com/commondocs/6502alan-instruction-set.pdf

Description	Op Code	Mnemonic	Example Assembly	Example Disassembly
Load the accumulator with a constant	A9	LDA	LDA #$07	A9 07
Load the accumulator from memory	AD	LDA	LDA $0010	AD 10 00
Store the accumulator in memory	8D	STA	STA $0010	8D 10 00
Add with carry Adds contents of an address to the contents of the accumulator and keeps the result in the accumulator	6D	ADC	ADC $0010	6D 10 00
Load the X register with a constant	A2	LDX	LDX #$01	A2 01
Load the X register from memory	AE	LDX	LDX $0010	AE 10 00
Load the Y register with a constant	A0	LDY	LDY #$04	A0 04
Load the Y register from memory	AC	LDY	LDY $0010	AC 10 00
No Operation	EA	NOP	EA	EA
Break (which is really a system call)	00	BRK	00	00
Compare a byte in memory to the X reg Sets the Z (zero) flag if equal	EC	CPX	EC $0010	EC 10 00
Branch *n* bytes if Z flag = 0	D0	BNE	D0 $EF	D0 EF
Increment the value of a byte	EE	INC	EE $0021	EE 21 00
System Call #$01 in X reg = print the integer stored in the Y register. #$02 in X reg = print the 00-terminated string stored at the address in the Y register.	FF	SYS		FF

Code Generation

Executable Image
256 bytes of code and data.

Code (aka “text”)

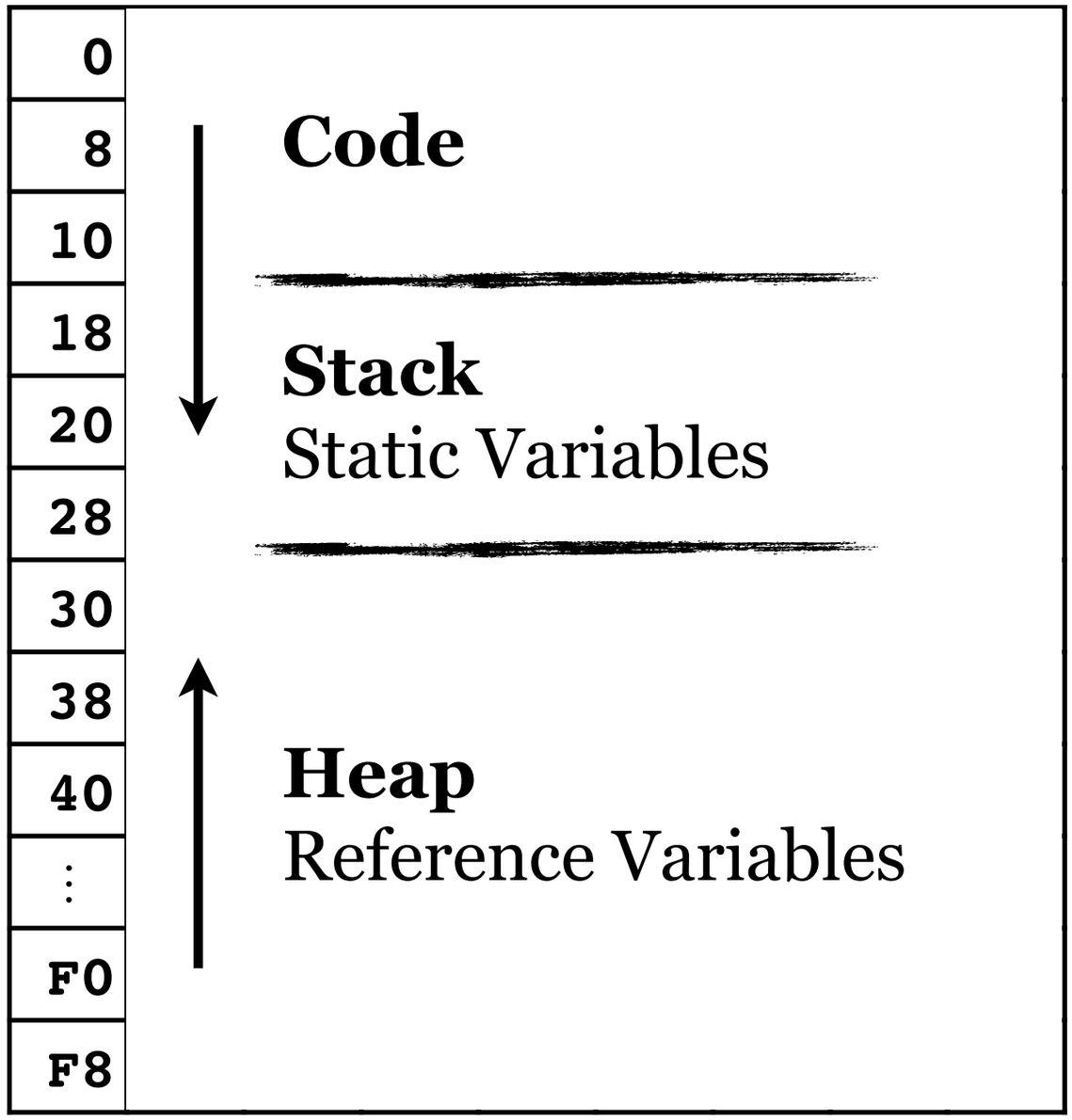
- The machine language op codes.
- Execution begins at location 0x00 and moves *down* towards FF.

Stack

- Storage for static variables: int, bool, string, pointer
- Begins immediately after the code section and moves *down* towards FF.

Heap

- Storage for dynamic/reference variables pointed to by static pointers.
- Begins at 0xFF and moves *up* towards 0x00.



Code Generation

Executable Image
256 bytes of code and data.

It's customary to display as shown here, in rows of 8 bytes at a time.

The **Code** begins a 0x00.
The **Heap** begins at 0xFF.

Q: Where does the **Stack** begin?
A: We don't know until we have generated all the code. We'll use temporary labels while we're generating the code and then go back later and replace them with real values. This is called **backpatching**. We'll use a table to keep track the temps as we go.

Temp	Var	Address
T0XX	a	+0

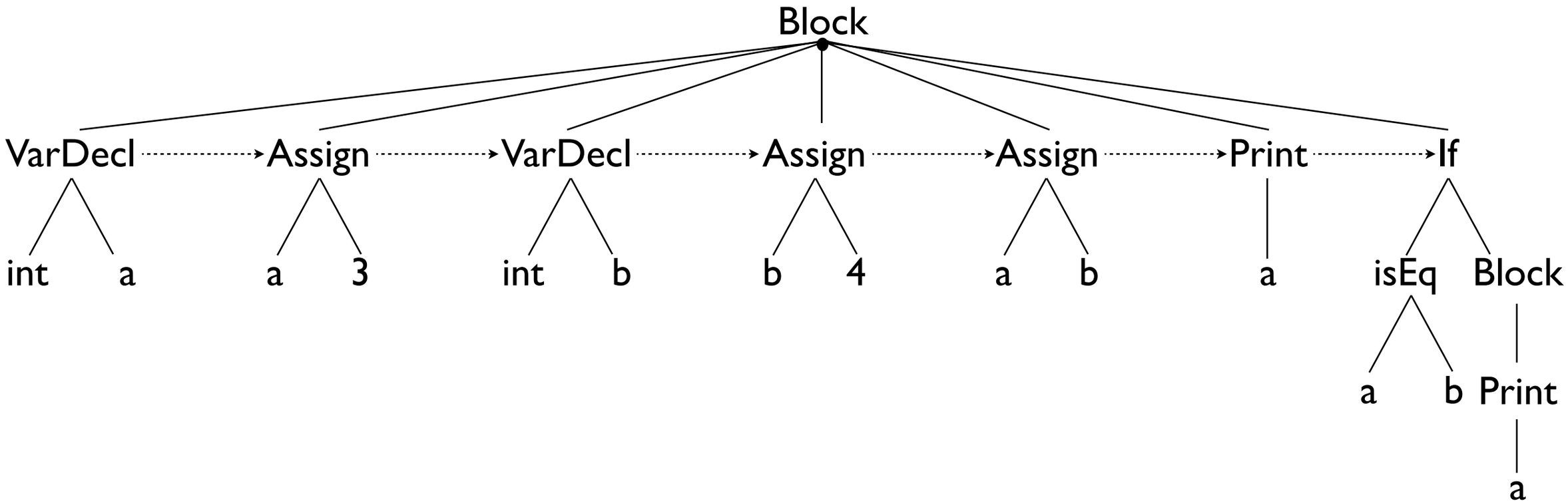
0								
8								
10								
18								
20								
28								
30								
38								
40								
⋮								
F0								
F8								

Code Generation Example

Source Code

```
{
    int a
    a = 3
    int b
    b = 4
    a = b
    print(a)
    if (a == b) {
       print(a)
    }
}
```

AST



Source Code

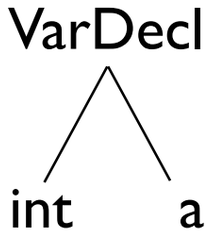
```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```

Our compiler generates code to initialize integers to 0.

Load the accumulator with 0.

Store the accumulator in location Temp0, denoted as **T0XX**. We'll fill this in later, once we have calculated the beginning address of the static area.

Make entry in Static table.



Execution Environment

0	A9	00	8D	T0	XX			
8								
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```

Load the accumulator with 3.

Store the accumulator in location Temp0, denoted as T0XX. We'll fill this in later, once we have calculated the beginning address of the static area.

Opportunity for optimization here.

Execution Environment

0	A9	00	8D	T0	XX	**A9**	**03**	**8D**
8	**T0**	**XX**						
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0

Source Code

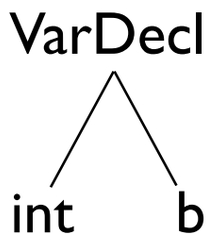
```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```

Our compiler generates code to initialize integers to 0.

Load the accumulator with 0.

Store the accumulator in location Temp1, denoted as **T1XX**. We'll fill this in later, once we have calculated the beginning address of the static area.

Make entry in Static table.



Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	**A9**	**00**	**8D**	**T1**	**XX**	
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```

Load the accumulator with 4.

Store the accumulator in location Temp1, denoted as T1XX. We'll fill this in later, once we have calculated the beginning address of the static area.

Opportunity for optimization here.

Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	A9	00	8D	T1	XX	**A9**
10	**04**	**8D**	**T1**	**XX**				
18								
20								
28								
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```

Load the accumulator with the contents of b, which we look up to find at temp address T1XX.

Store the accumulator in the address for a, which we look up and find is temp address T0XX.

Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	04	8D	T1	XX	**AD**	**T1**	**XX**	**8D**
18	**T0**	**XX**						
20								
28								
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```

Load the Y register with the contents of a, which we look up to find at temp address T0XX.

Load the X register with 1.

System Call.

Print
|
a

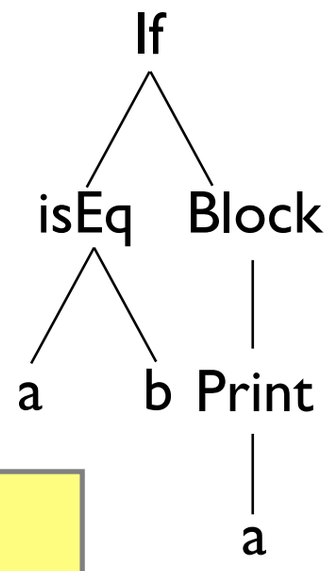
Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	04	8D	T1	XX	AD	T1	XX	8D
18	T0	XX	**AC**	**T0**	**XX**	**A2**	**01**	**FF**
20								
28								
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
}
```



Load the X register with the contents of a.

Compare the X register to the contents of b.

Branch on NOT EQUAL, jumping ahead some number of bytes (**temp J0**) to AFTER the generated code for the “if true” statement list.

Print a (Same op codes as prior statement.)

Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	04	8D	T1	XX	AD	T1	XX	8D
18	T0	XX	AC	T0	XX	A2	01	FF
20	**AE**	**T0**	**XX**	**EC**	**T1**	**XX**	**D0**	**J0**
28	**AC**	**T0**	**XX**	**A2**	**01**	**FF**		
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Temp	Distance
J0	?

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
} (break)
```

This is the end of the program.

Break. (Just to be safe.)

Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	04	8D	T1	XX	AD	T1	XX	8D
18	T0	XX	AC	T0	XX	A2	01	FF
20	AE	T0	XX	EC	T1	XX	D0	J0
28	AC	T0	XX	A2	01	FF	**00**	
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Temp	Distance
J0	?

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
} (break)
```

Calculate the distance to jump for temporary jump value J0.

(Jump over 6 bytes, landing on the 7th.)

Backpatch the code with this value.

Execution Environment

0	A9	00	8D	T0	XX	A9	03	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	04	8D	T1	XX	AD	T1	XX	8D
18	T0	XX	AC	T0	XX	A2	01	FF
20	AE	T0	XX	EC	T1	XX	D0	**06**
28	AC	T0	XX	A2	01	FF	00	
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	+0
T1XX	b	+=len(T0XX)

Temp	Distance
J0	**6**

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
} (break)
```

If were are not concerned with byte alignment, we can begin the Static variable area at location 2F.

Find all T0XX and replace with 2F00 (little endian)

Execution Environment

0	A9	00	8D	**2F**	**00**	A9	03	8D
8	**2F**	**00**	A9	00	8D	T1	XX	A9
10	04	8D	T1	XX	AD	T1	XX	8D
18	**2F**	**00**	AC	**2F**	**00**	A2	01	FF
20	AE	**2F**	**00**	EC	T1	XX	D0	06
28	AC	**2F**	**00**	A2	01	FF	00	used a
30								
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	**2F 00**
T1XX	b	+=len(T0XX)

Temp	Distance
J0	6

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
} (break)
```

Integers are one byte, so the address of T1XX is the base address for Static storage (002F) + the length of T0XX (1), or 0030.

Find all T1XX and replace with 3000 (little endian)

Execution Environment

0	A9	00	8D	2F	00	A9	03	8D
8	2F	00	A9	00	8D	**30**	**00**	A9
10	04	8D	**30**	**00**	AD	**30**	**00**	8D
18	2F	00	AC	2F	00	A2	01	FF
20	AE	2F	00	EC	**30**	**00**	D0	06
28	AC	2F	00	A2	01	FF	00	used a
30	used b							
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	2F 00
T1XX	b	**30 00**

Temp	Distance
J0	6

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
} (break)
```

We can begin dynamic allocation on the HEAP, at location 0031. Or we can start dynamic allocation at 005F (in this example) and work our way back to 0031.

Execution Environment

0	A9	00	8D	2F	00	A9	03	8D
8	2F	00	A9	00	8D	30	00	A9
10	04	8D	30	00	AD	30	00	8D
18	2F	00	AC	2F	00	A2	01	FF
20	AE	2F	00	EC	30	00	D0	06
28	AC	2F	00	A2	01	FF	00	used a
30	used b							
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	2F 00
T1XX	b	30 00

Temp	Distance
J0	6

Source Code

```
int a
a = 3
int b
b = 4
a = b
print(a)
if (a == b) {
  print(a)
} (break)
```

Machine Code

Let’s try the program:

```
A9 00 8D 2F 00 A9 03 8D
2F 00 A9 00 8D 30 00 A9
04 8D 30 00 AD 30 00 8D
2F 00 AC 2F 00 A2 01 FF
AE 2F 00 EC 30 00 D0 06
AC 2F 00 A2 01 FF 00 00
```

Execution Environment

0	A9	00	8D	2F	00	A9	03	8D
8	2F	00	A9	00	8D	30	00	A9
10	04	8D	30	00	AD	30	00	8D
18	2F	00	AC	2F	00	A2	01	FF
20	AE	2F	00	EC	30	00	D0	06
28	AC	2F	00	A2	01	FF	00	used a
30	used b							
38								
40								
48								
50								
58								

Temp	Var	Address
T0XX	a	2F 00
T1XX	b	30 00

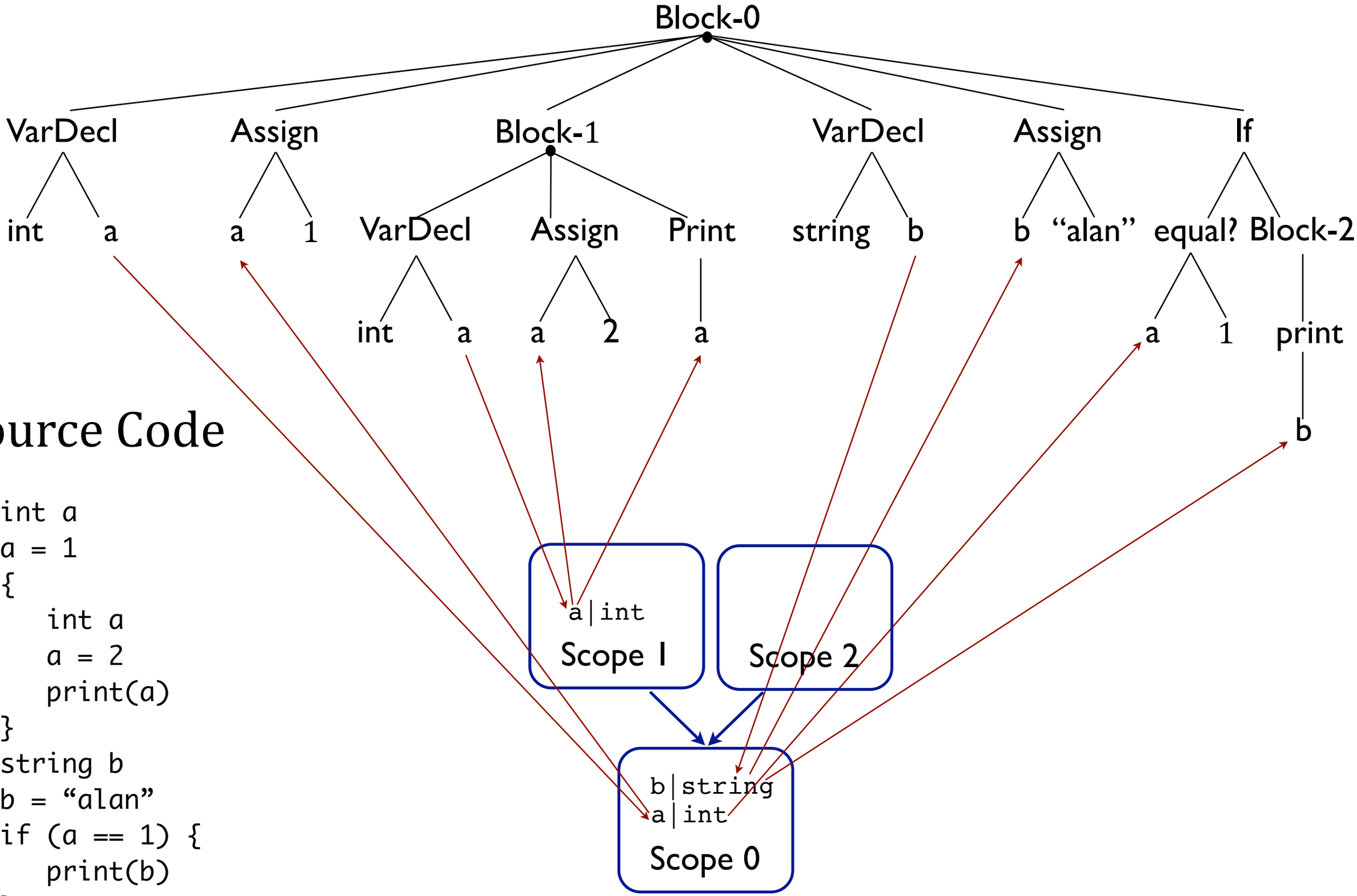
Temp	Distance
J0	6

Another Code Generation Example

Source Code

```
{
   int a
   a = 1
   {
      int a
      a = 2
      print(a)
   }
   string b
   b = “alan”
   if (a == 1) {
      print(b)
   }
}
```

AST

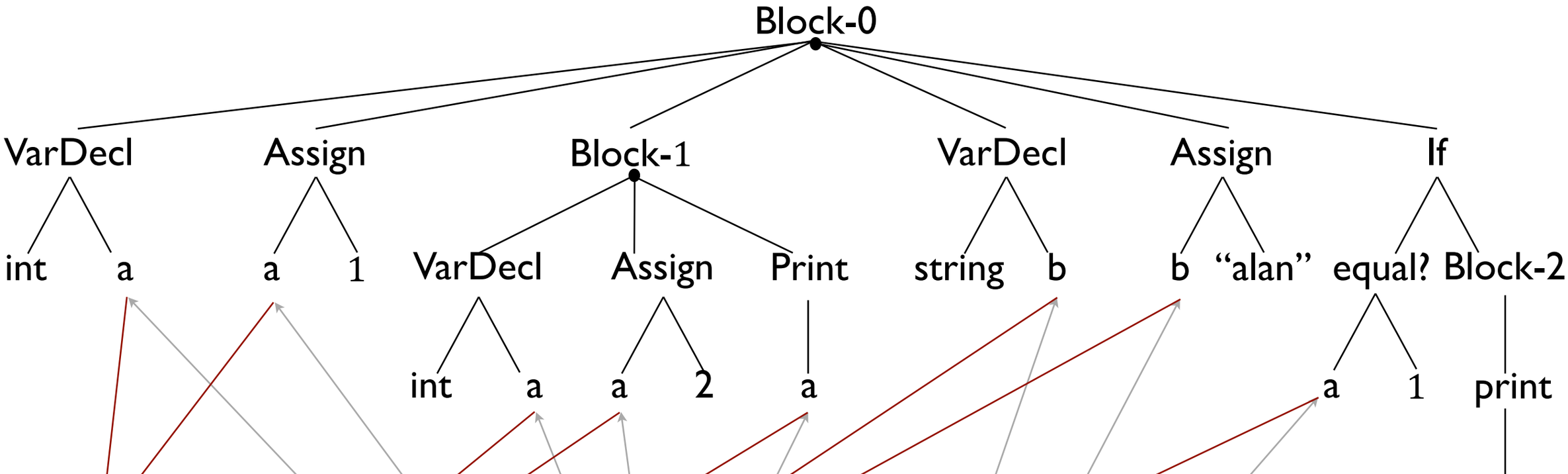


Source Code

```
{
   int a
   a = 1
   {
      int a
      a = 2
      print(a)
   }
   string b
   b = “alan”
   if (a == 1) {
      print(b)
   }
}
```

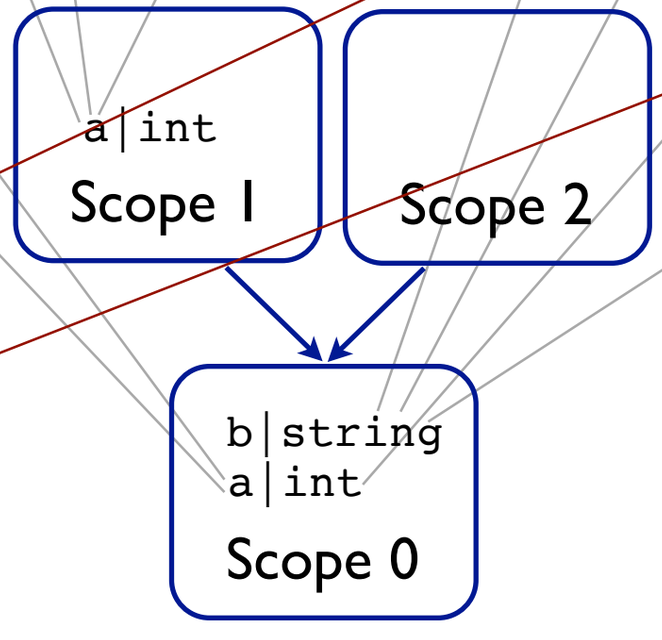
Symbol Table

AST



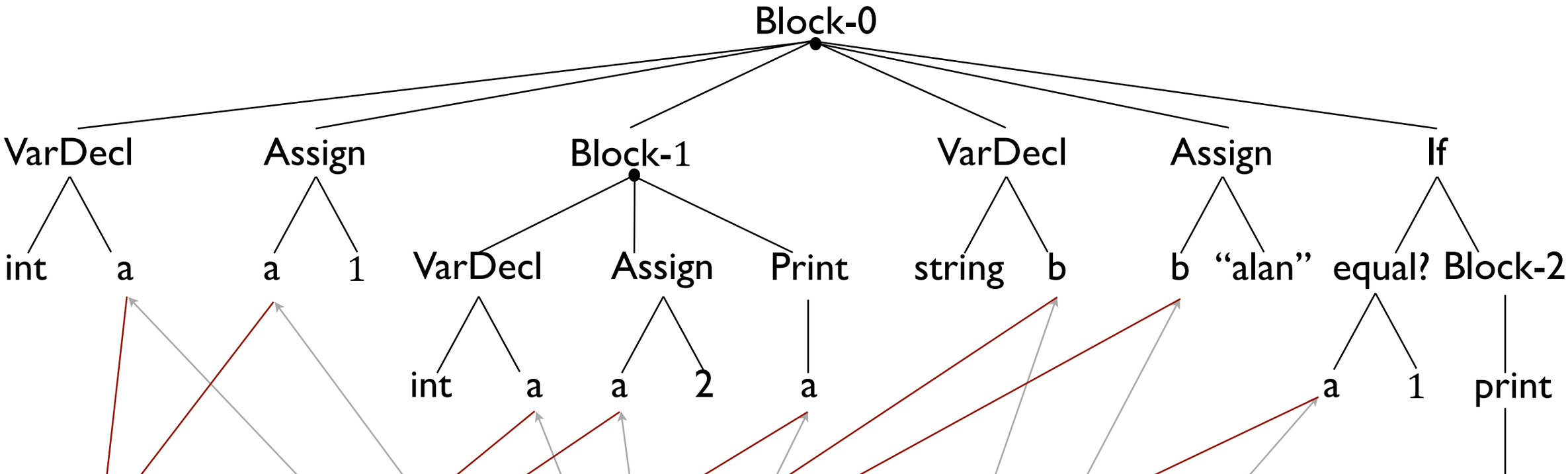
Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



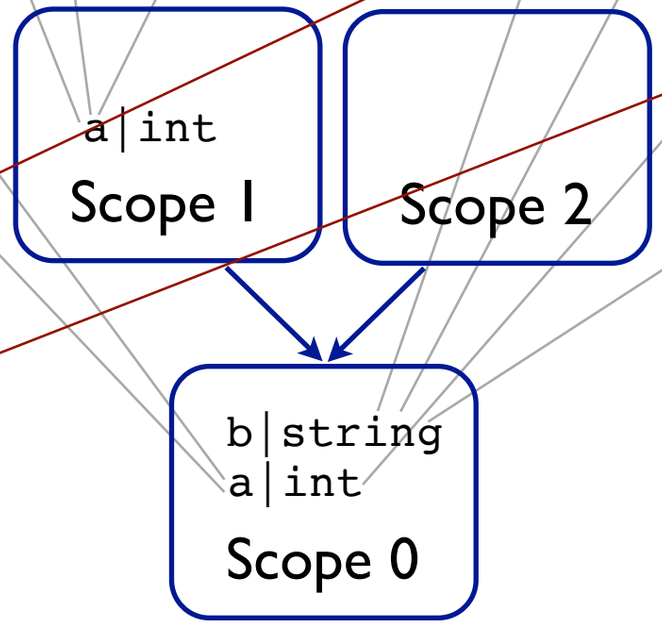
Symbol Table

AST



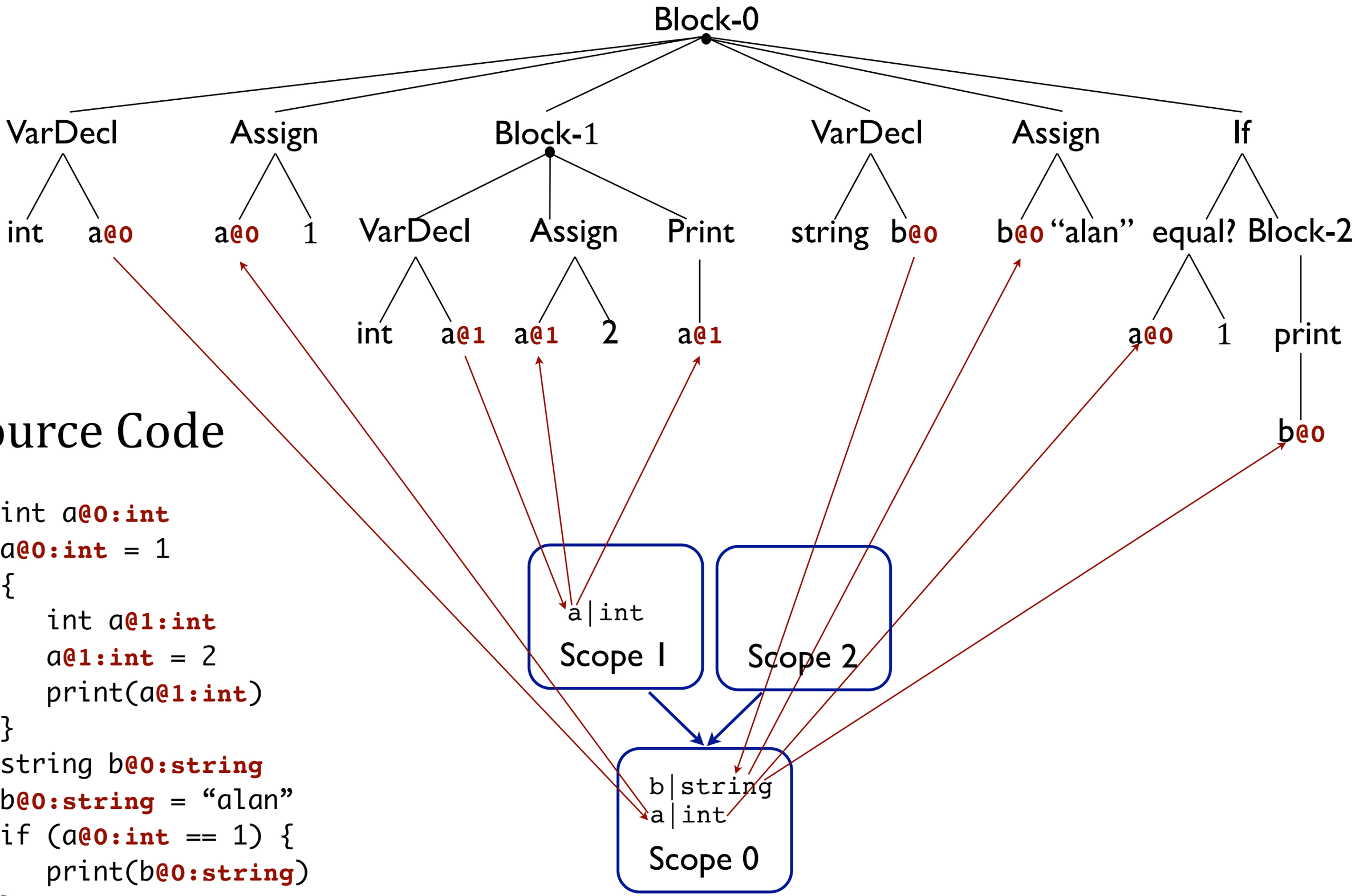
Source Code

```
{
    int a@0:int
    a@0:int = 1
    {
        int a@1:int
        a@1:int = 2
        print(a@1:int)
    }
    string b@0:string
    b@0:string = “alan”
    if (a@0:int == 1) {
        print(b@0:string)
    }
}
```



Symbol Table

AST



Source Code

```
{
   int a@0:int
   a@0:int = 1
   {
      int a@1:int
      a@1:int = 2
      print(a@1:int)
   }
   string b@0:string
   b@0:string = "alan"
   if (a@0:int == 1) {
      print(b@0:string)
   }
}
```


Symbol Table

Source Code

```
{
   int a@0:int
   a@0:int = 1
   {
      int a@1:int
      a@1:int = 2
      print(a@1:int)
   }
   string b@0:string
   b@0:string = “alan”
   if (a@0:int == 1) {
      print(b@0:string)
   }
}
```

Runtime Environment

0								
8								
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Make variable entries in **Static** table.

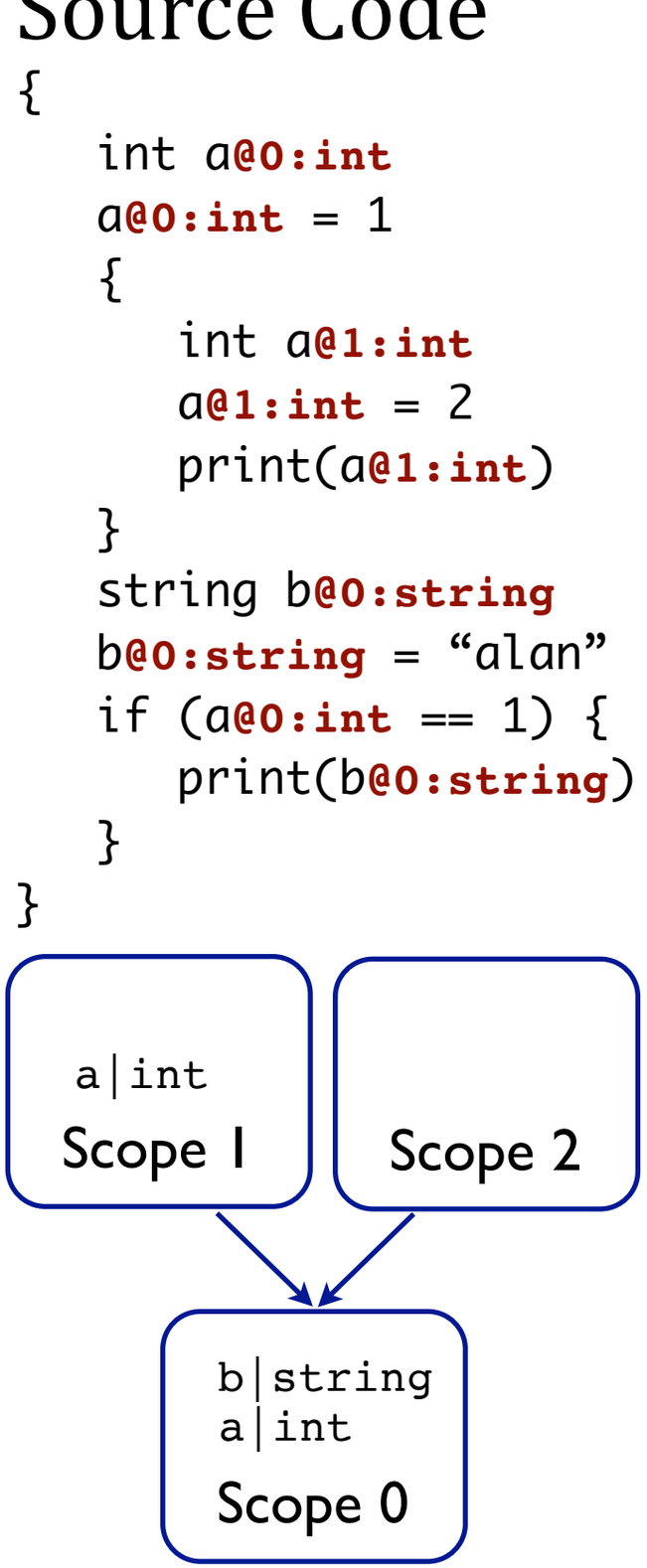
Make entries in **Jump** table for **if** stmts.

Static Data			
Temp	Var	Scope	Offset

Jumps	
Temp	Dist

Source Code

```
{
   int a@0:int
   a@0:int = 1
   {
      int a@1:int
      a@1:int = 2
      print(a@1:int)
   }
   string b@0:string
   b@0:string = “alan”
   if (a@0:int == 1) {
      print(b@0:string)
   }
}
```



Runtime Environment

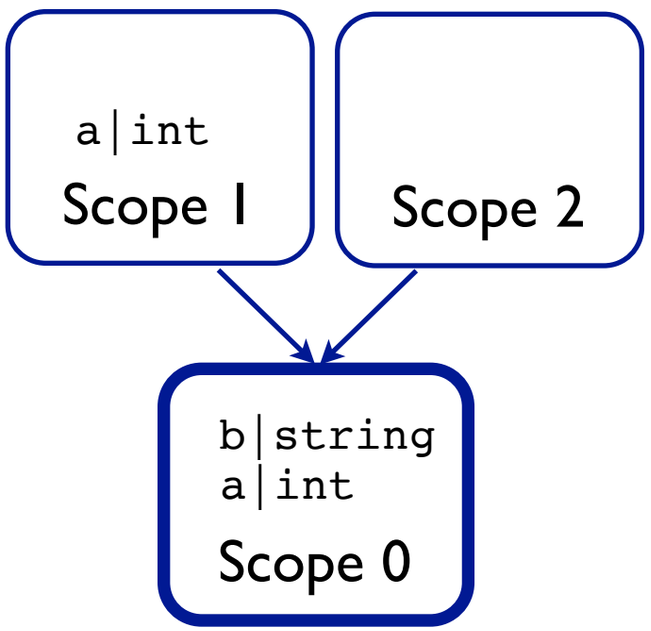
0								
8								
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

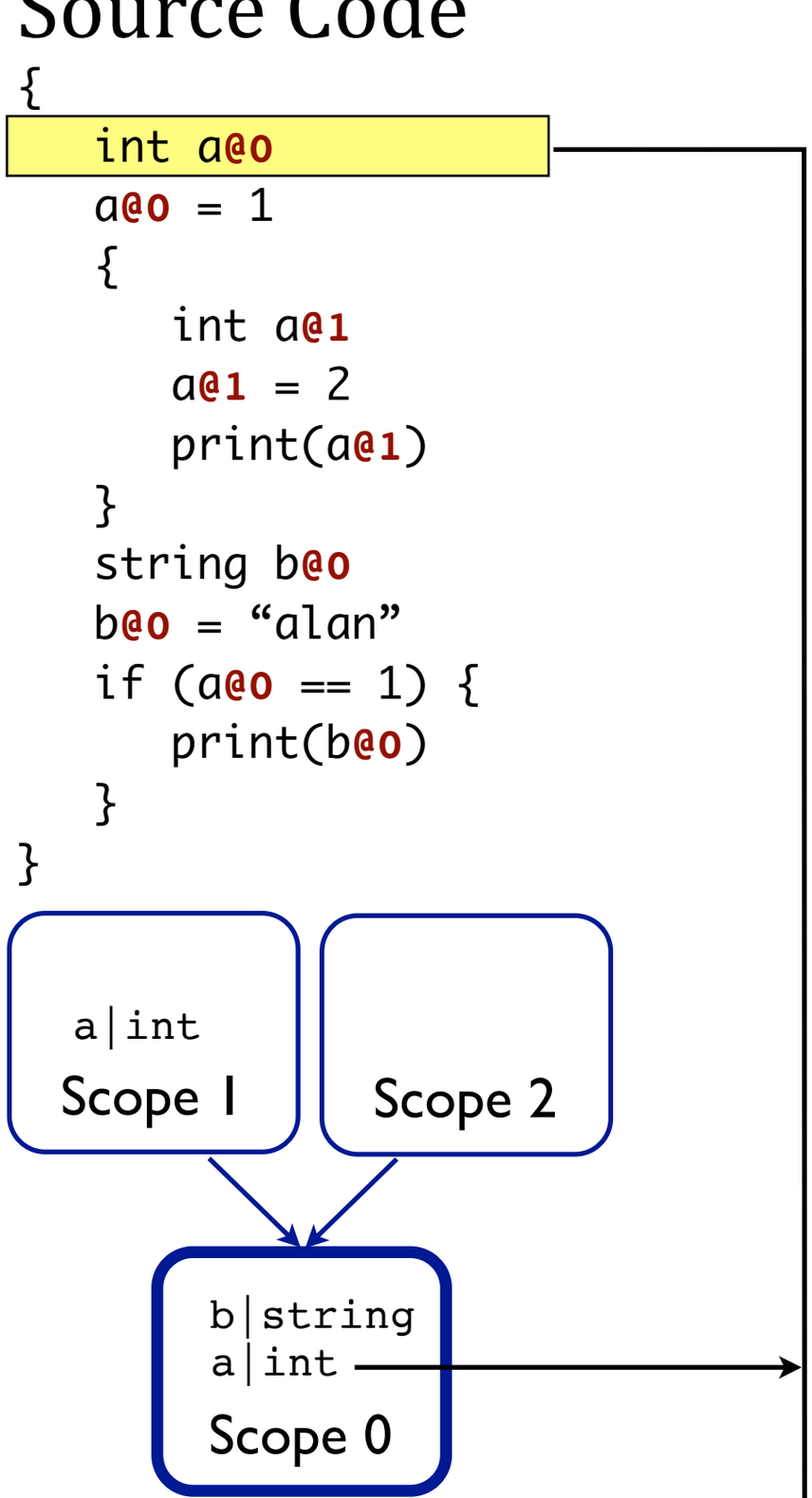
0								
8								
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

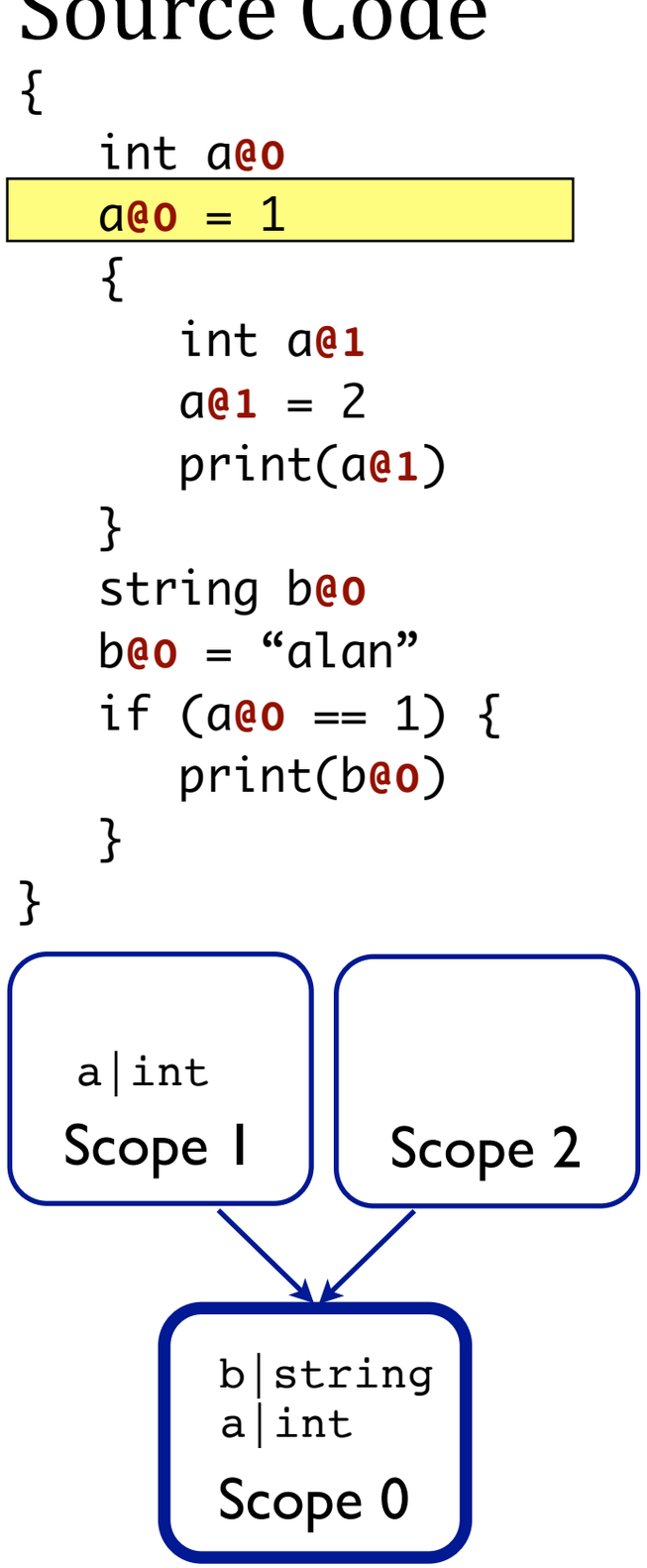
0	A9	00	8D	T0	XX			
8								
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

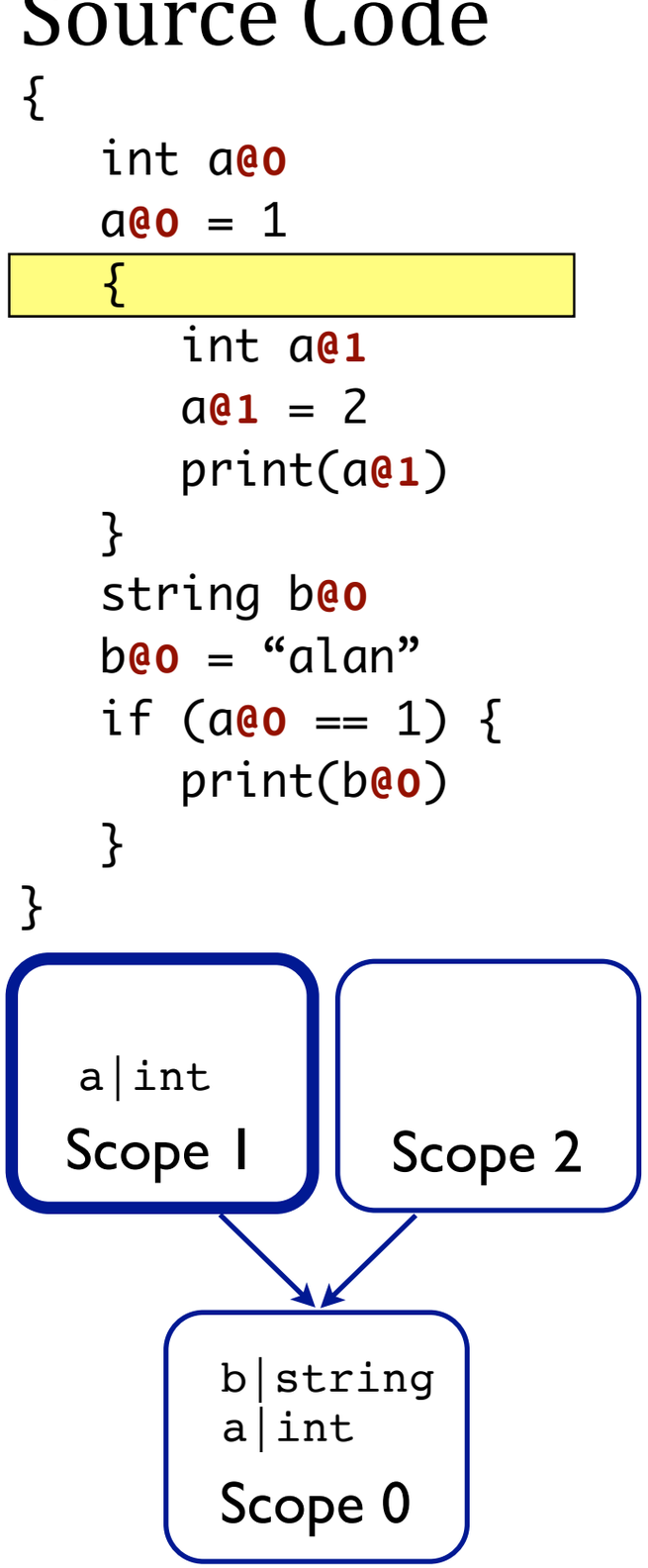
0	A9	00	8D	T0	XX	**A9**	**01**	**8D**
8	**T0**	**XX**						
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

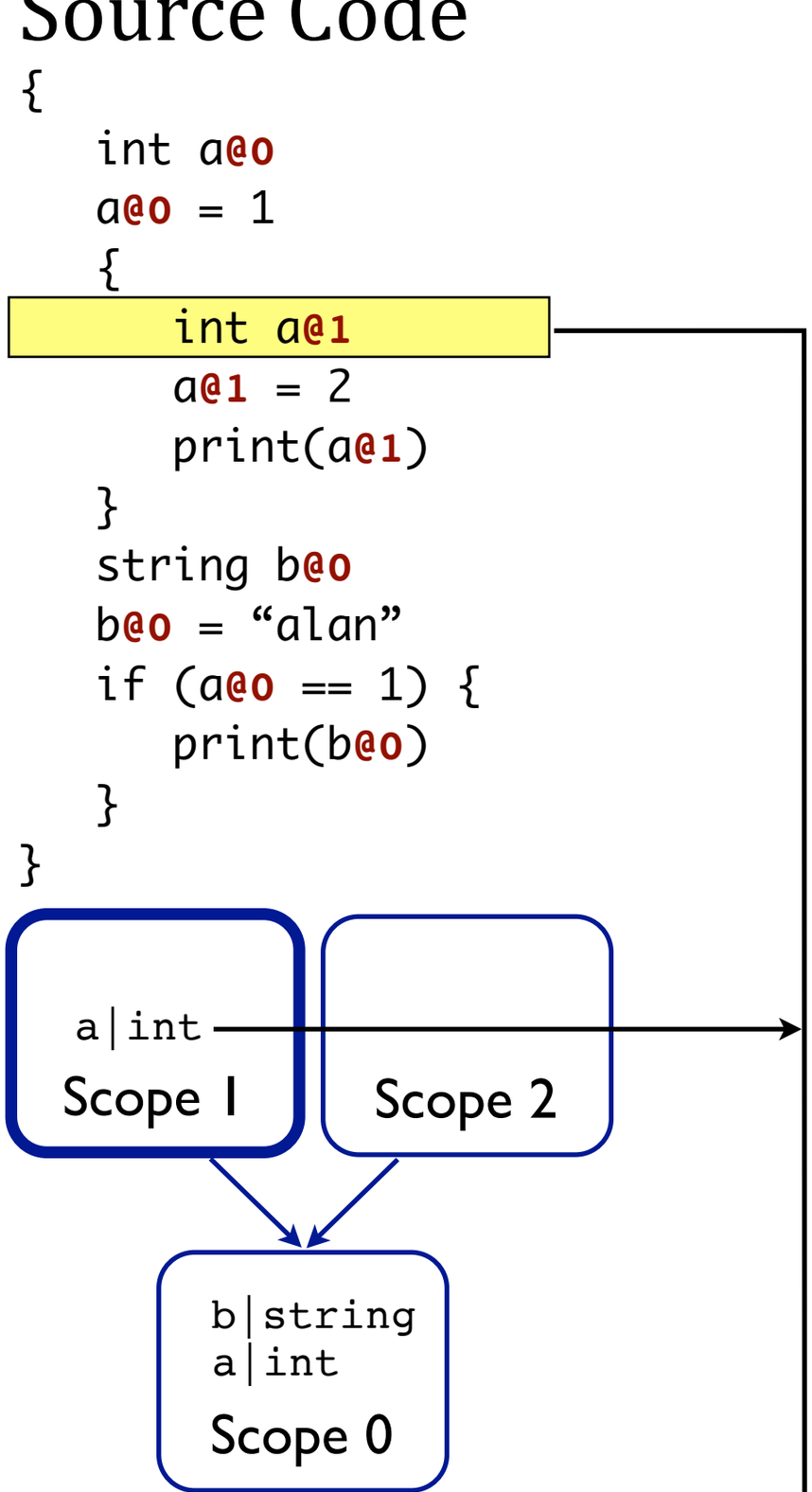
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX						
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

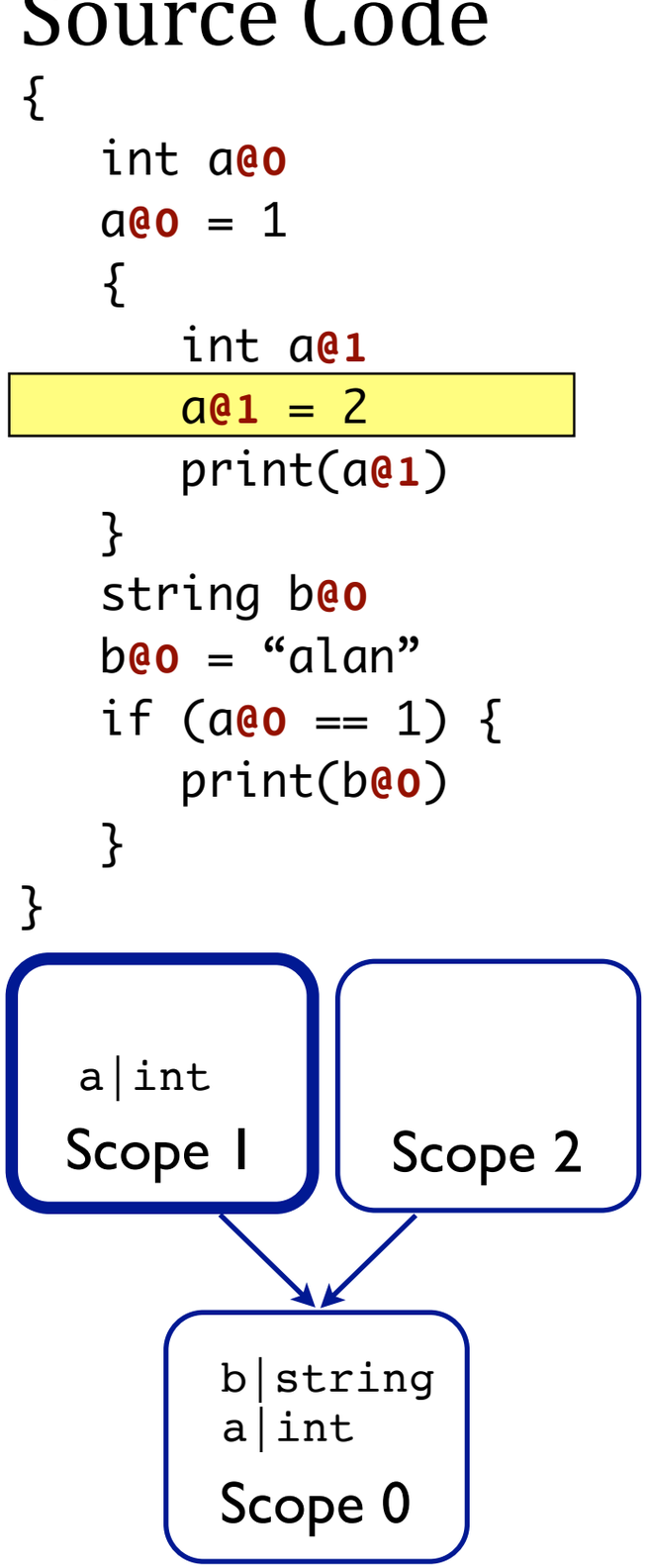
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	**A9**	**00**	**8D**	**T1**	**XX**	
10								
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	**a**	**1**	**+1**

Jumps	
Temp	Dist

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```



Runtime Environment

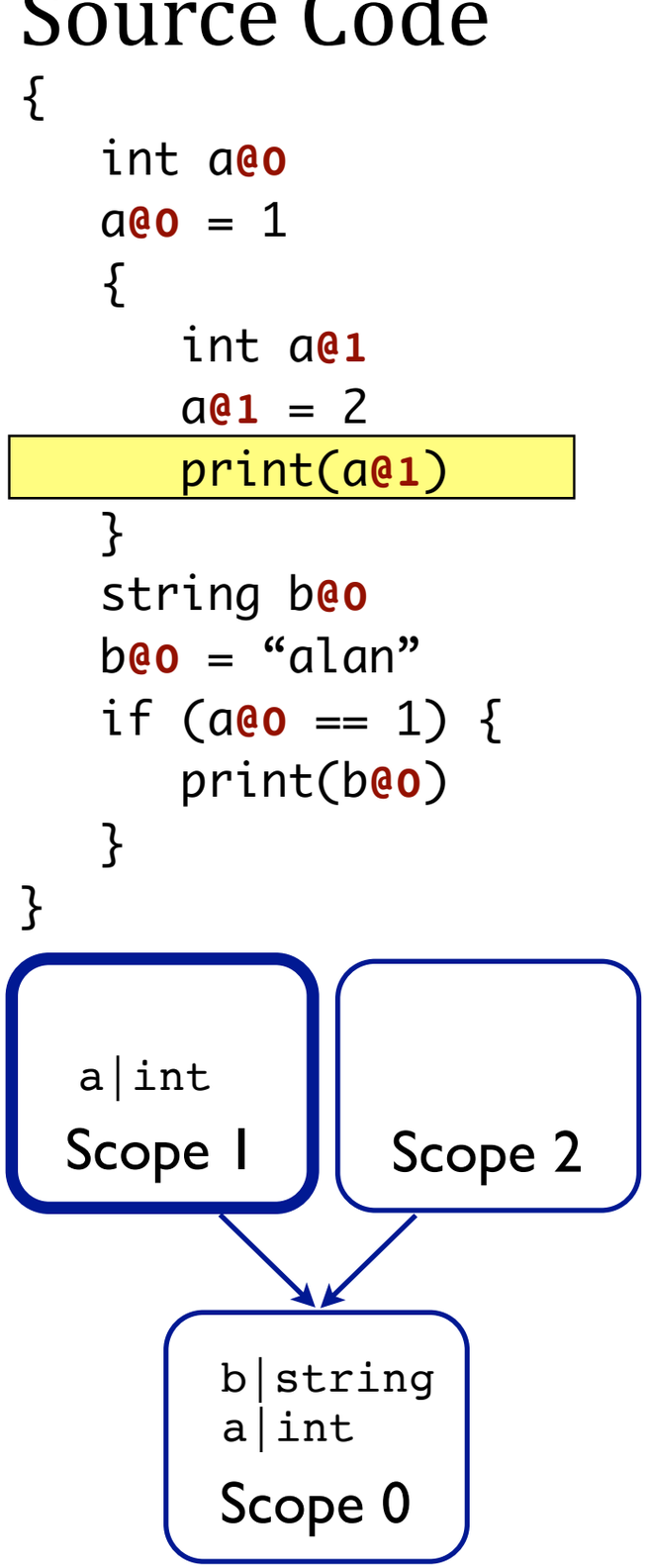
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	**A9**
10	**02**	**8D**	**T1**	**XX**				
18								
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

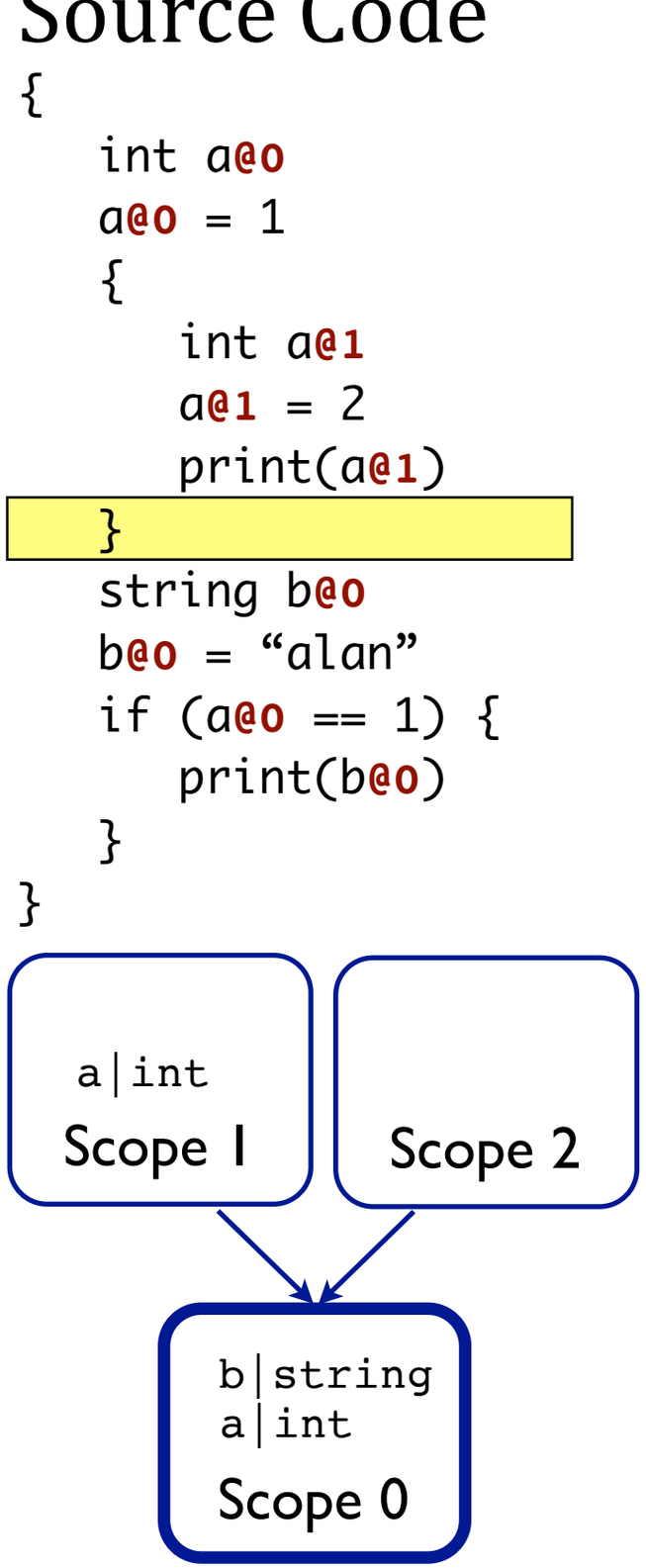
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	**AC**	**T1**	**XX**	**A2**
18	**01**	**FF**						
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

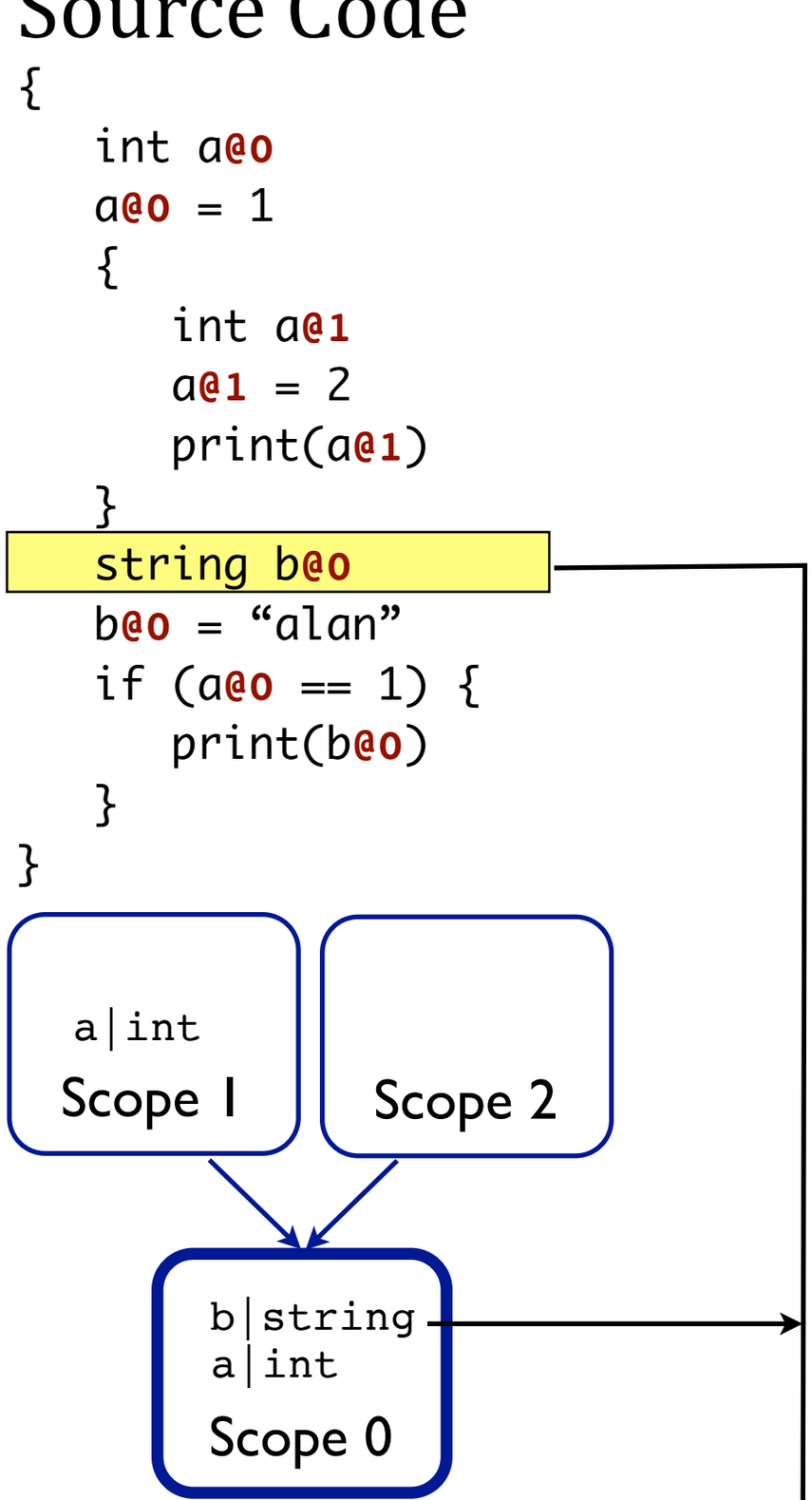
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF						
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1

Jumps	
Temp	Dist

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```



Add a static pointer to the start of the string in the heap.

Runtime Environment

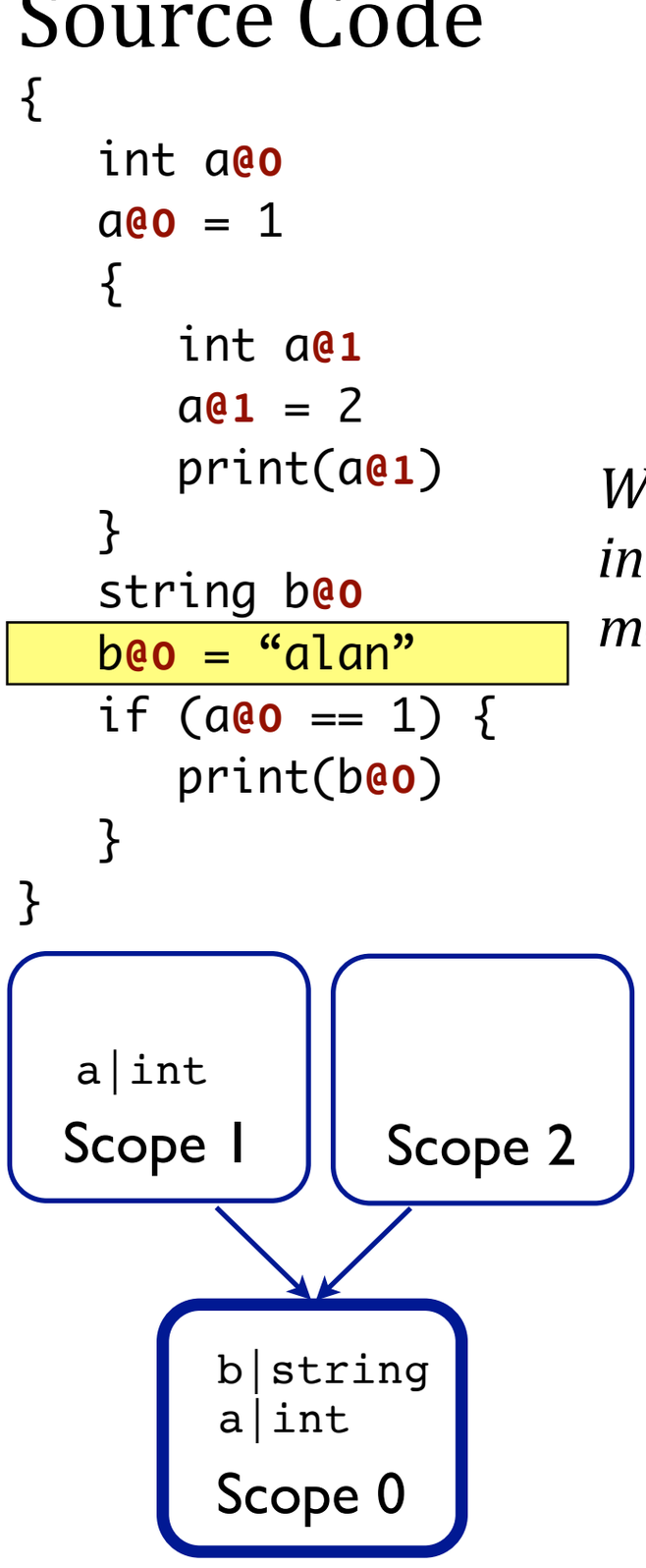
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF						
20								
28								
30								
38								
40								
48								
50								
58								

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	**b**	**0**	**+2**

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Write the data into heap memory . . .

Runtime Environment

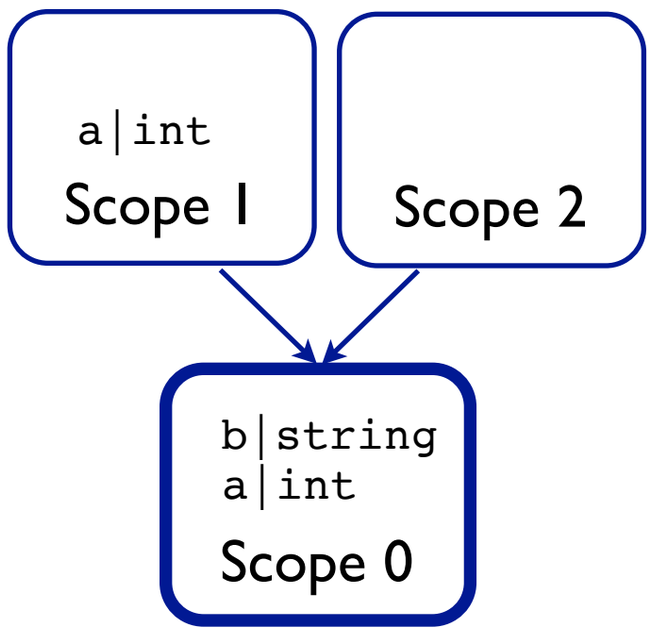
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF						
20								
28								
30								
38								
40								
48								
50								
58				**61**	**6C**	**61**	**6E**	**00**

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	**b**	**0**	**+2**

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Write the data into heap memory . . . then store the static pointer for b@0.

Runtime Environment

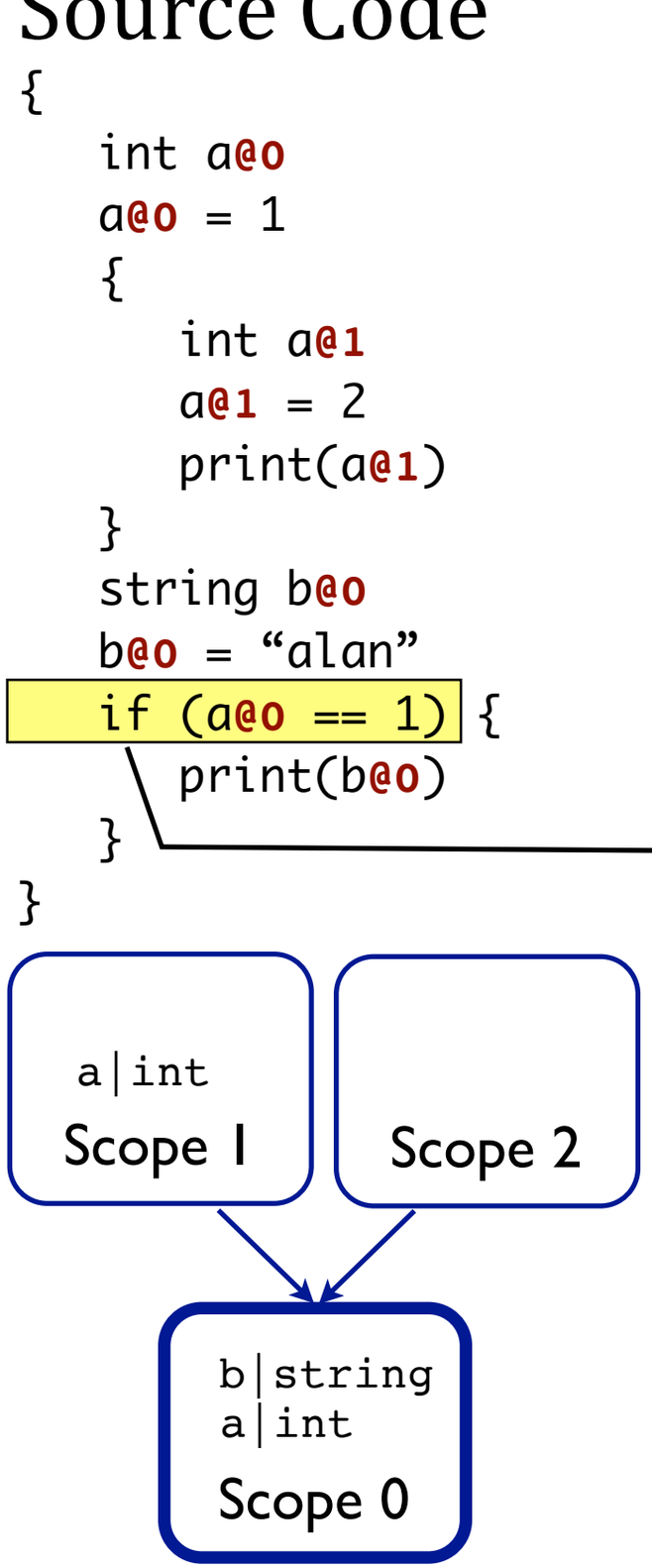
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	**A9**	**5B**	**8D**	**T2**	**XX**	
20								
28								
30								
38								
40								
48								
50								
58				**61**	**6C**	**61**	**6E**	**00**

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	**b**	**0**	**+2**

Jumps	
Temp	Dist

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

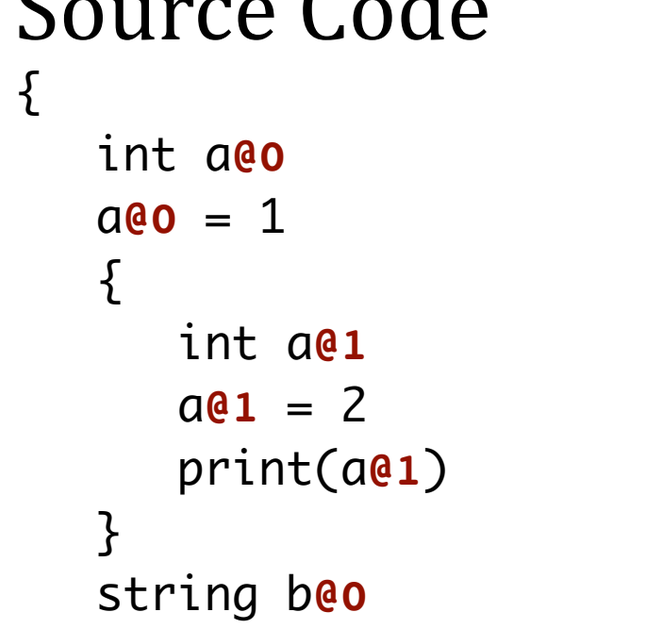
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	**A2**
20	**01**	**EC**	**T0**	**XX**	**D0**	**J0**		
28								
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	?

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```



Runtime Environment

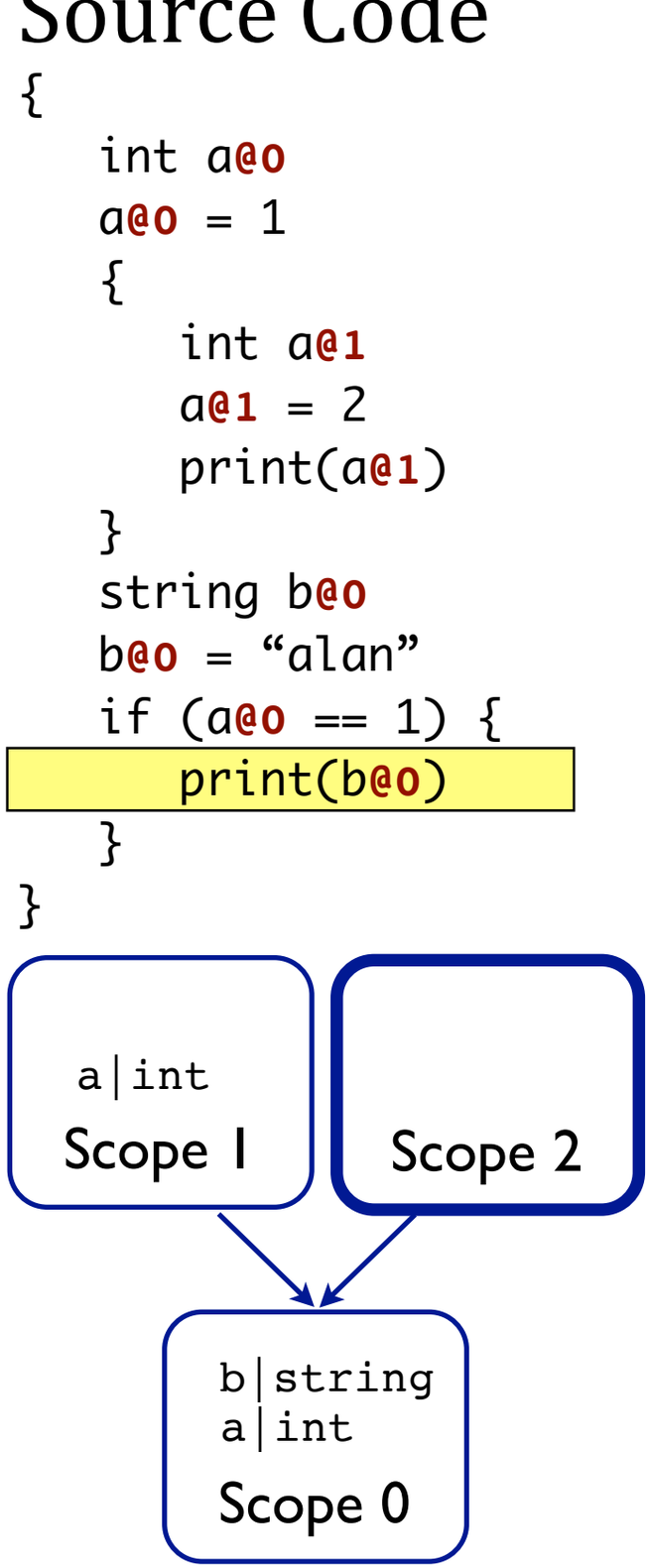
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0		
28								
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	?

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

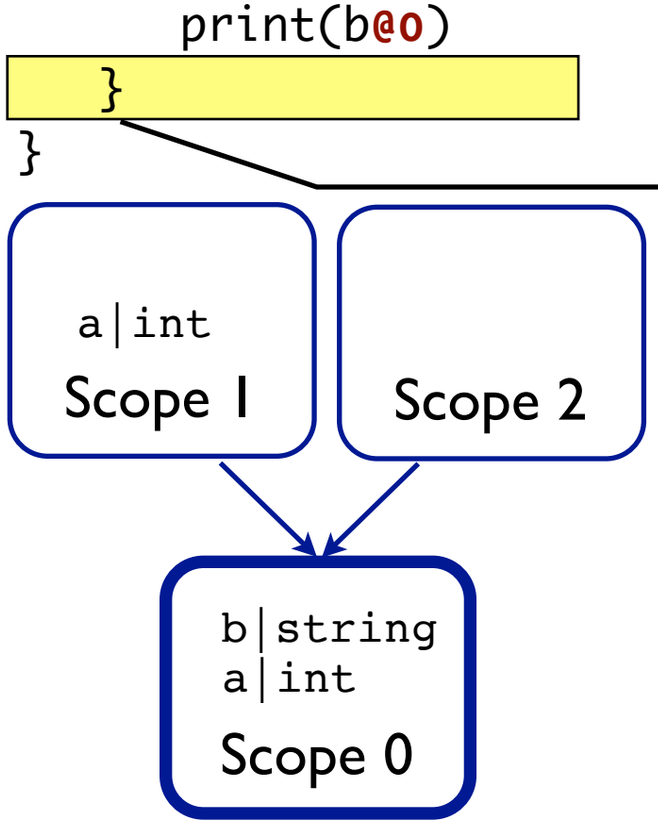
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	**AC**	**T2**
28	**XX**	**A2**	**02**	**FF**				
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	?

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```



Runtime Environment

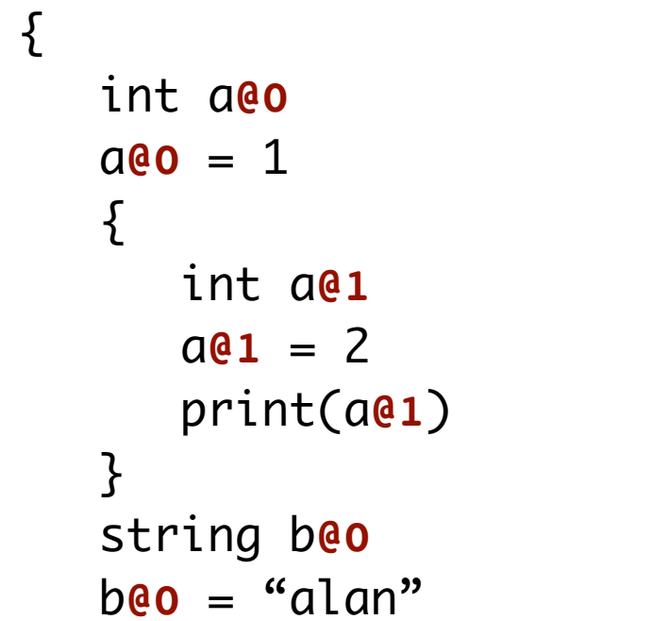
0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF				
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```



Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF	**00**			
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```

Backpatch temporary values:

Jump J0 ➛ 06

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	**06**	AC	T2
28	XX	A2	02	FF	00			
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	**6**

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```

Backpatch temporary values:

Jump J0 ➛ 06

T0 XX ➛ 2D 00

Runtime Environment

0	A9	00	8D	**2D**	**00**	A9	01	8D
8	**2D**	**00**	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	**2D**	**00**	D0	06	AC	T2
28	XX	A2	02	FF	00	used a@0		
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
2D 00	**a**	**0**	**+0**
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```

Backpatch temporary values:

Jump J0 ➛ 06

T0 XX ➛ 2D 00

T1 XX ➛ 2E 00

Runtime Environment

0	A9	00	8D	2D	00	A9	01	8D
8	2D	00	A9	00	8D	**2E**	**00**	A9
10	02	8D	**2E**	**00**	AC	**2E**	**00**	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	2D	00	D0	06	AC	T2
28	XX	A2	02	FF	00	used a@0	**used a@1**	
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
2D 00	a	0	+0
2E 00	**a**	**1**	**+1**
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```

Backpatch temporary values:

Jump J0 ➛ 06

T0 XX ➛ 2D 00

T1 XX ➛ 2E 00

T2 XX ➛ 2F 00

Runtime Environment

0	A9	00	8D	2D	00	A9	01	8D
8	2D	00	A9	00	8D	2E	00	A9
10	02	8D	2E	00	AC	2E	00	A2
18	01	FF	A9	5B	8D	**2F**	**00**	A2
20	01	EC	2D	00	D0	06	AC	**2F**
28	**00**	A2	02	FF	00	used a@0	used a@1	**used b↑@0**
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
2D 00	a	0	+0
2E 00	a	1	+1
2F 00	**b↑**	**0**	**+2**

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```

Backpatch temporary values:
Jump J0 ➛ 06
T0 XX ➛ 2D 00
T1 XX ➛ 2E 00
T2 XX ➛ 2F 00

Fill the space with 00.

Runtime Environment

0	A9	00	8D	2D	00	A9	01	8D
8	2D	00	A9	00	8D	2E	00	A9
10	02	8D	2E	00	AC	2E	00	A2
18	01	FF	A9	5B	8D	2F	00	A2
20	01	EC	2D	00	D0	06	AC	2F
28	00	A2	02	FF	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00
58	00	00	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
2D 00	a	0	+0
2E 00	a	1	+1
2F 00	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```

Try it:

```
A9 00 8D 2D 00 A9 01 8D
2D 00 A9 00 8D 2E 00 A9
02 8D 2E 00 AC 2E 00 A2
01 FF A9 5B 8D 2F 00 A2
01 EC 2D 00 D0 06 AC 2F
00 A2 02 FF 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 61 6C 61 6E 00
```

Runtime Environment

0	A9	00	8D	2D	00	A9	01	8D
8	2D	00	A9	00	8D	2E	00	A9
10	02	8D	2E	00	AC	2E	00	A2
18	01	FF	A9	5B	8D	2F	00	A2
20	01	EC	2D	00	D0	06	AC	2F
28	00	A2	02	FF	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00
58	00	00	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
2D 00	a	0	+0
2E 00	a	1	+1
2F 00	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
}
```

Now let’s make it more complicated by

- adding another string
- changing the value of an existing string.

Rewind to just before the closing brace.

Runtime Environment

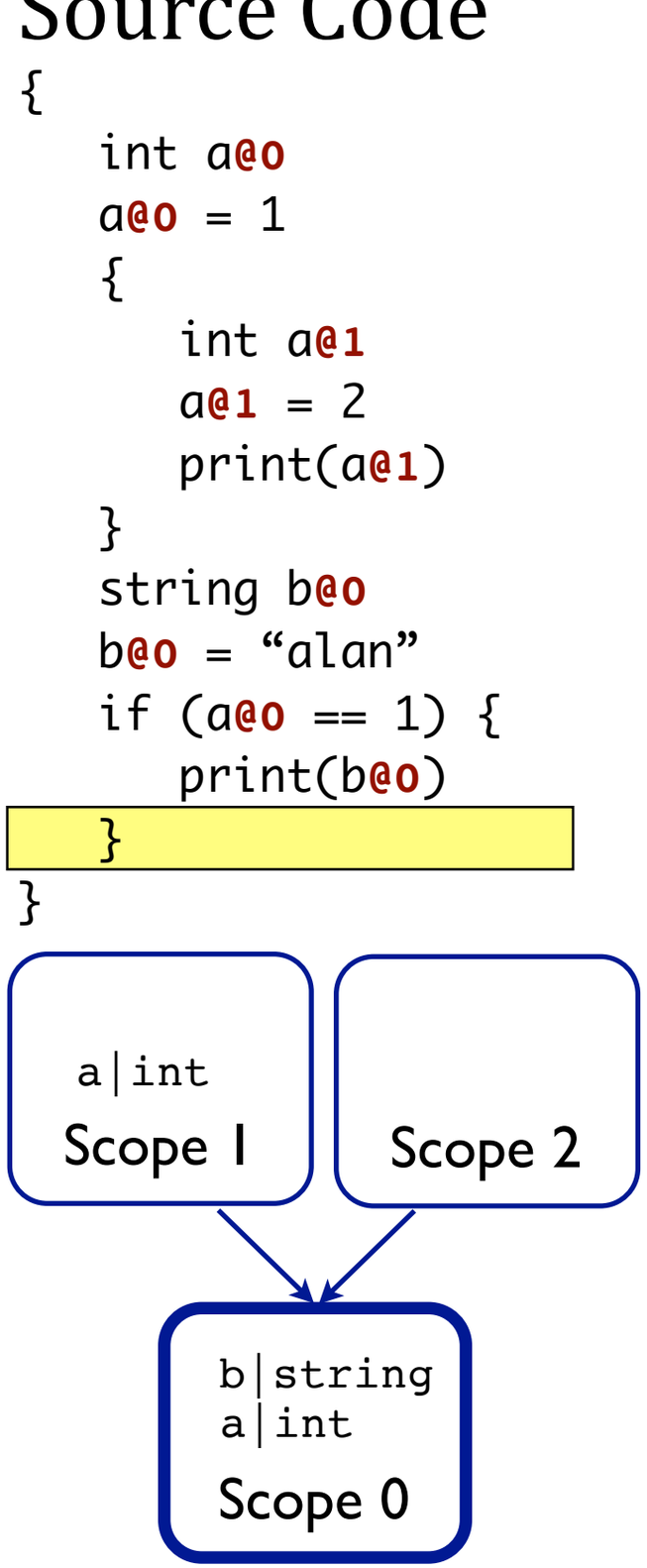
0	A9	00	8D	2D	00	A9	01	8D
8	2D	00	A9	00	8D	2E	00	A9
10	02	8D	2E	00	AC	2E	00	A2
18	01	FF	A9	5B	8D	2F	00	A2
20	01	EC	2D	00	D0	06	AC	2F
28	00	A2	02	FF	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00
58	00	00	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
2D 00	a	0	+0
2E 00	a	1	+1
2F 00	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
}
```



Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF				
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
    int a@0
    a@0 = 1
    {
        int a@1
        a@1 = 2
        print(a@1)
    }
    string b@0
    b@0 = “alan”
    if (a@0 == 1) {
        print(b@0)
    }
    string c@0
    c@0 = “james”
    b@0 = “blackstone”
    print(b@0)
}
```

c | string
b | string
a | int
Scope 0

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF				
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b	0	+2

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

c | string
b | string
a | int
Scope 0

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF				
30								
38								
40								
48								
50								
58				61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
T0XX	a	0	+0
T1XX	a	1	+1
T2XX	b ↑	0	+2

More Static Data			
Temp	Var	Scope	Offset
T3XX	c ↑	0	+3

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

Write the data into heap memory and then store the static pointer for c.

c	string
b	string
a	int

Scope 0

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF	**A9**	**55**	**8D**	**T3**
30	**XX**							
38								
40								
48								
50						**6A**	**61**	**6D**
58	**65**	**73**	**00**	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b ⇡	0	+2

More Static Data			
Temp	Var	Scope	Offset
T3XX	**c ⇡**	**0**	**+3**

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

*Write the data into heap memory and then **update** the static pointer for b.*

c|string
b|string
a|int
Scope 0

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF	A9	55	8D	T3
30	XX	**A9**	**4A**	**8D**	**T2**	**XX**		
38								
40								
48			**62**	**6C**	**61**	**63**	**6B**	**73**
50	**74**	**6F**	**6E**	**65**	**00**	6A	61	6D
58	65	73	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	**b** ↑	**0**	**+2**

More Static Data			
Temp	Var	Scope	Offset
T3XX	c ↑	0	+3

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

c|string
b|string
a|int
Scope 0

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF	A9	55	8D	T3
30	XX	A9	4A	8D	T2	XX	**AC**	**T2**
38	**XX**	**A2**	**02**	**FF**				
40								
48			62	6C	61	63	6B	73
50	74	6F	6E	65	00	6A	61	6D
58	65	73	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
T0XX	a	0	+0
T1XX	a	1	+1
T2XX	b↑	0	+2

More Static Data			
Temp	Var	Scope	Offset
T3XX	c↑	0	+3

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

c|string
b|string
a|int
Scope 0

Runtime Environment

0	A9	00	8D	T0	XX	A9	01	8D
8	T0	XX	A9	00	8D	T1	XX	A9
10	02	8D	T1	XX	AC	T1	XX	A2
18	01	FF	A9	5B	8D	T2	XX	A2
20	01	EC	T0	XX	D0	J0	AC	T2
28	XX	A2	02	FF	A9	55	8D	T3
30	XX	A9	4A	8D	T2	XX	AC	T2
38	XX	A2	02	FF	**00**			
40								
48			62	6C	61	63	6B	73
50	74	6F	6E	65	00	6A	61	6D
58	65	73	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
TOXX	a	0	+0
T1XX	a	1	+1
T2XX	b ↑	0	+2

More Static Data			
Temp	Var	Scope	Offset
T3XX	c ↑	0	+3

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

Backpatch temporary values:
Jump J0 ➛ 06
T0 XX ➛ 3D 00
T1 XX ➛ 3E 00
T2 XX ➛ 3F 00
T3 XX ➛ 40 00

Runtime Environment

0	A9	00	8D	**3D**	**00**	A9	01	8D
8	**3D**	**00**	A9	00	8D	**3E**	**00**	A9
10	02	8D	**3E**	**00**	AC	**3E**	**00**	A2
18	01	FF	A9	5B	8D	**3F**	**00**	A2
20	01	EC	**3D**	**00**	D0	**06**	AC	**3F**
28	**00**	A2	02	FF	A9	55	8D	**40**
30	**00**	A9	4A	8D	**3F**	**00**	AC	**3F**
38	**00**	A2	02	FF	00	used a@0	used a@1	used b↑@0
40	used c↑@0							
48			62	6C	61	63	6B	73
50	74	6F	6E	65	00	6A	61	6D
58	65	73	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
3D 00	a	0	+0
3E 00	a	1	+1
3F 00	b↑	0	+2

More Static Data			
Temp	Var	Scope	Offset
40 00	c↑	0	+3

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

Fill space with 00.

Runtime Environment

0	A9	00	8D	3D	00	A9	01	8D
8	3D	00	A9	00	8D	3E	00	A9
10	02	8D	3E	00	AC	3E	00	A2
18	01	FF	A9	5B	8D	3F	00	A2
20	01	EC	3D	00	D0	06	AC	3F
28	00	A2	02	FF	A9	55	8D	40
30	00	A9	4A	8D	3F	00	AC	3F
38	00	A2	02	FF	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	62	6C	61	63	6B	73
50	74	6F	6E	65	00	6A	61	6D
58	65	73	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
3D 00	a	0	+0
3E 00	a	1	+1
3F 00	b ↑	0	+2

More Static Data			
Temp	Var	Scope	Offset
40 00	c ↑	0	+3

Jumps	
Temp	Dist
J0	6

Source Code

```
{
   int a@0
   a@0 = 1
   {
      int a@1
      a@1 = 2
      print(a@1)
   }
   string b@0
   b@0 = “alan”
   if (a@0 == 1) {
      print(b@0)
   }
   string c@0
   c@0 = “james”
   b@0 = “blackstone”
   print(b@0)
}
```

Try it:

```
A9 00 8D 3D 00 A9 01 8D 3D 00 A9 00
8D 3E 00 A9 02 8D 3E 00 AC 3E 00 A2
01 FF A9 5B 8D 3F 00 A2 01 EC 3D 00
D0 06 AC 3F 00 A2 02 FF A9 55 8D 40
00 A9 4A 8D 3F 00 AC 3F 00 A2 02 FF
00 00 00 00 00 00 00 00 00 00 00 00
00 00 62 6C 61 63 6B 73 74 6F 6E 65
00 6A 61 6D 65 73 00 61 6C 61 6E 00
```

Runtime Environment

0	A9	00	8D	3D	00	A9	01	8D
8	3D	00	A9	00	8D	3E	00	A9
10	02	8D	3E	00	AC	3E	00	A2
18	01	FF	A9	5B	8D	3F	00	A2
20	01	EC	3D	00	D0	06	AC	3F
28	00	A2	02	FF	A9	55	8D	40
30	00	A9	4A	8D	3F	00	AC	3F
38	00	A2	02	FF	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	62	6C	61	63	6B	73
50	74	6F	6E	65	00	6A	61	6D
58	65	73	00	61	6C	61	6E	00

Static Data			
Temp	Var	Scope	Offset
3D 00	a	0	+0
3E 00	a	1	+1
3F 00	b	0	+2

More Static Data			
Temp	Var	Scope	Offset
40 00	c	0	+3

Jumps	
Temp	Dist
J0	6

One More Code Generation Example

While Loops

Source Code

```
{
    int a
    a = 1
    while (a != 5) {
       a = 1 + a
       print(a)
    }
} $
```

Machine Code

```
00: A9 00 8D 52 00 A9 01 8D
08: 52 00 AD 52 00 8D 54 00
10: A9 05 8D 53 00 AE 54 00
18: EC 53 00 A9 00 D0 02 A9
20: 01 A2 00 8D 53 00 EC 53
28: 00 D0 20 AD 52 00 8D 53
30: 00 A9 01 6D 53 00 8D 52
38: 00 AC 52 00 A2 01 FF A9
40: 00 8D 53 00 A2 01 EC 53
48: 00 D0 BF 00
```

6502 Disassembly

0000	A9 00	LDA #$00	**int a** (0x52)
0002	8D 52 00	STA $0052	
0005	A9 01	LDA #$01	**a = 1**
0007	8D 52 00	STA $0052	
000A	AD 52 00	LDA $0052	**while (a != 5) {**
000D	8D 54 00	STA $0054	Copy a to t_2 (0x54).
0010	A9 05	LDA #$05	Copy the compare-to value
0012	8D 53 00	STA $0053	to t_1 (0x53).
0015	AE 54 00	LDX $0054	Compare t_2 and t_1, and
0018	EC 53 00	CPX $0053	assign Z flag.
001B	A9 00	LDA #$00	Acc = 0.
001D	D0 02	BNE $0021	If t_2 != t_1 branch to 0x21.
001F	A9 01	LDA #$01	(if t_2 == t_1) Acc = 1.
0021	A2 00	LDX #$00	X register = 0.
0023	8D 53 00	STA $0053	Store Acc in t_1 (0x53).
0026	EC 53 00	CPX $0053	Compare t_1 and X reg, and
0029	D0 20	BNE $004B	branch if unequal.

One More Code Generation Example

While Loops

Source Code

```
{
    int a
    a = 1
    while (a != 5) {
       a = 1 + a
       print(a)
    }
} $
```

Machine Code

```
00: A9 00 8D 52 00 A9 01 8D
08: 52 00 AD 52 00 8D 54 00
10: A9 05 8D 53 00 AE 54 00
18: EC 53 00 A9 00 D0 02 A9
20: 01 A2 00 8D 53 00 EC 53
28: 00 D0 20 AD 52 00 8D 53
30: 00 A9 01 6D 53 00 8D 52
38: 00 AC 52 00 A2 01 FF A9
40: 00 8D 53 00 A2 01 EC 53
48: 00 D0 BF 00
```

6502 Disassembly

002B	AD 52 00	LDA $0052	
002E	8D 53 00	STA $0053	
0031	A9 01	LDA #$01	**a = 1 + a**
0033	6D 53 00	ADC $0053	
0036	8D 52 00	STA $0052	
0039	AC 52 00	LDY $0052	
003C	A2 01	LDX #$01	**print(a)**
003E	FF	SYS	
003F	A9 00	LDA #$00	**}** unconditional jump to the top of the loop, $000A, which is 0xBF (191) bytes forward, which is actually 64 bytes backward.
0041	8D 53 00	STA $0053	
0044	A2 01	LDX #$01	
0046	EC 53 00	CPX $0053	
0049	D0 BF	BNE $000A	
004B	00	BRK	break

```
   0x4A = 74
74+191 = 265
265-255 = 10 = 0x0A
```