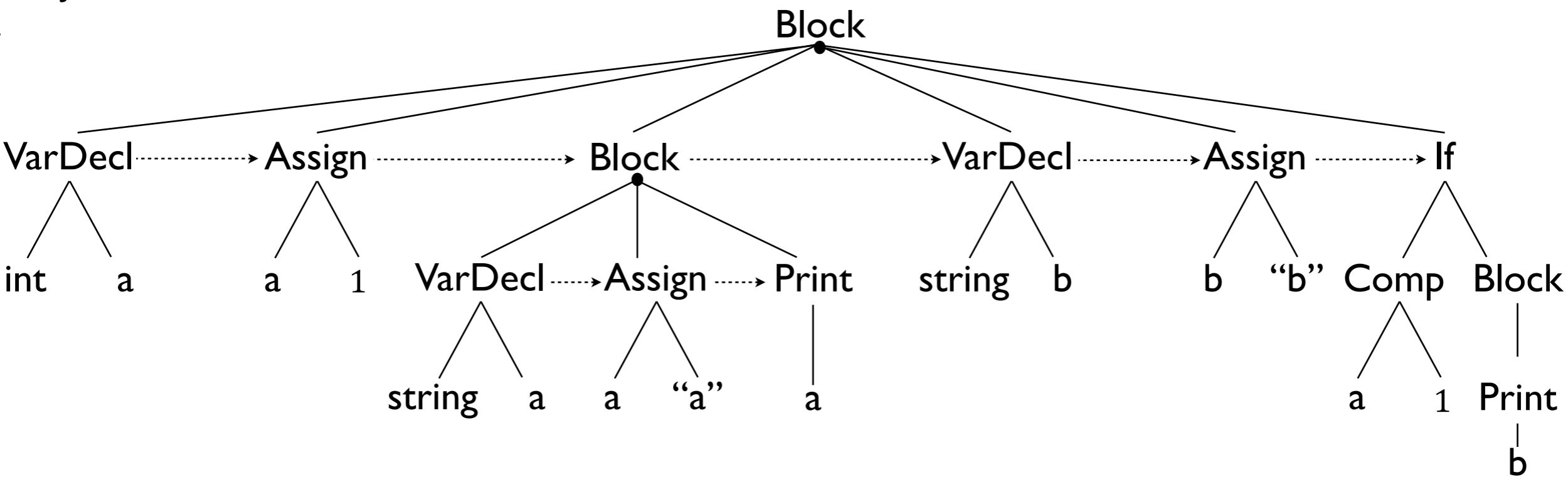


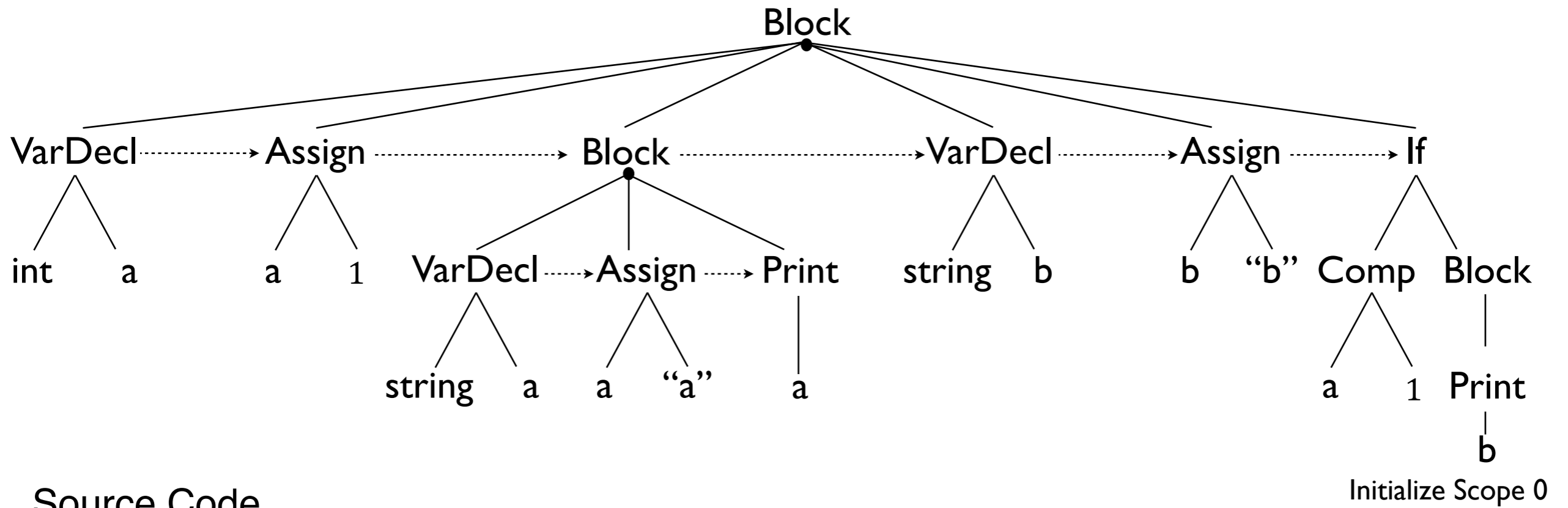
Source Code

```
{  
  int a  
  a = 1  
  {  
    string a  
    a = "a"  
    print(a)  
  }  
  string b  
  b = "b"  
  if (a == 1) {  
    print(b)  
  }  
}
```

AST



AST

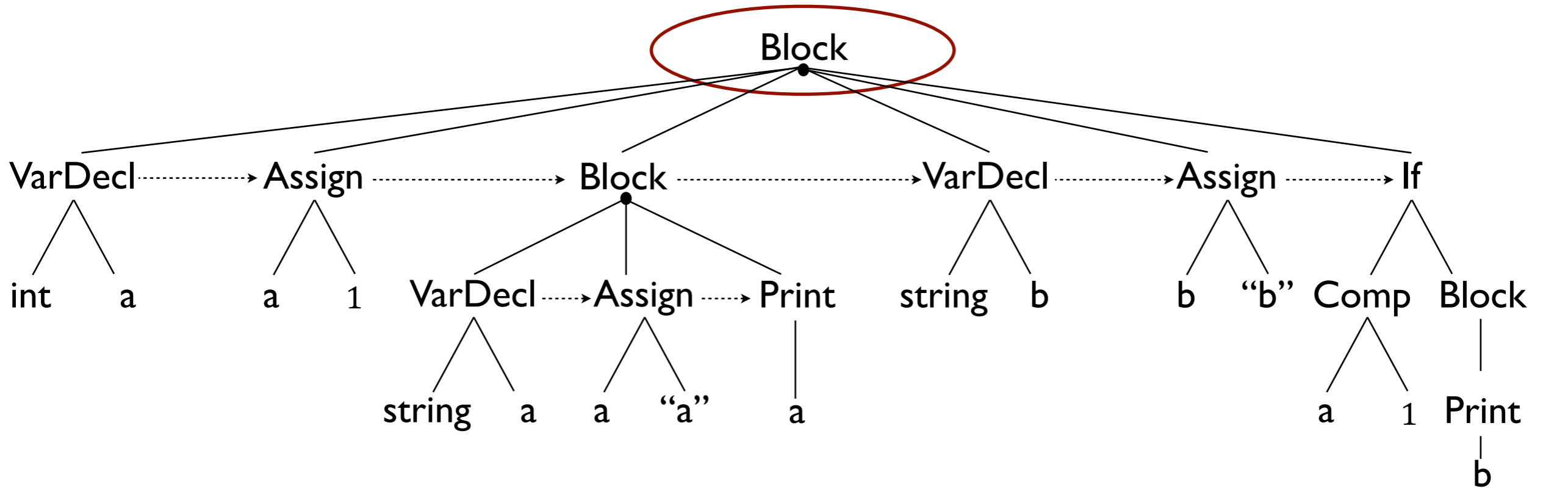


Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

Symbol Table

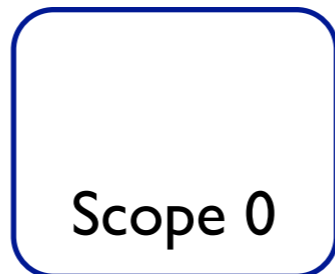
AST



Source Code

```
{
int a
a = 1
{
    string a
    a = "a"
    print(a)
}
string b
b = "b"
if (a == 1) {
    print(b)
}
}
```

Initialize Scope 0
Set the
current scope
pointer.

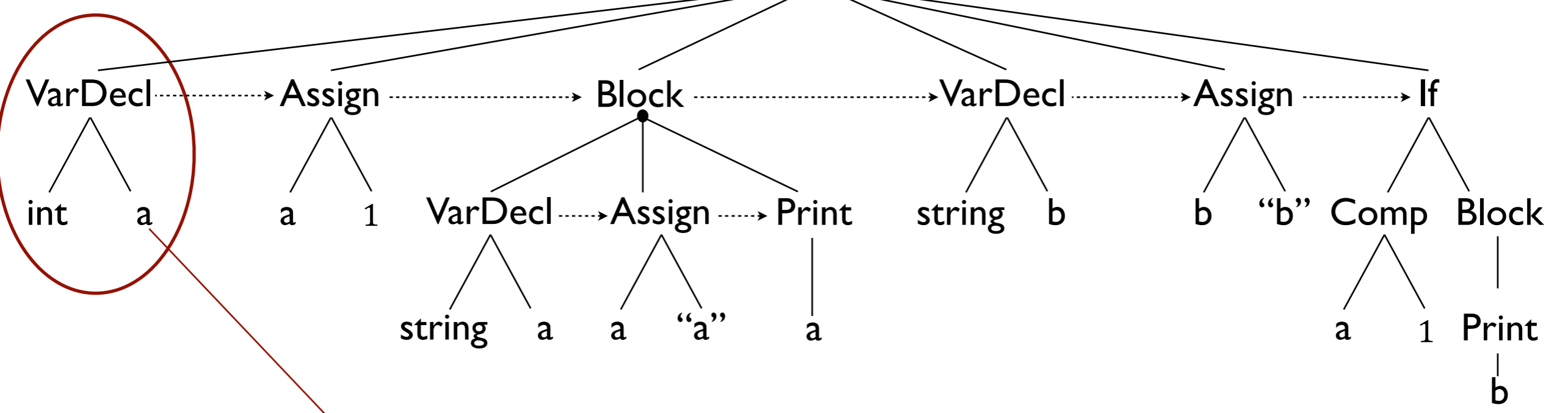


← current scope

Symbol Table

AST

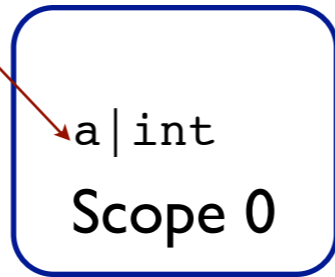
Block



Source Code

```
{  
  int a  
  a = 1  
  {  
    string a  
    a = "a"  
    print(a)  
  }  
  string b  
  b = "b"  
  if (a == 1) {  
    print(b)  
  }  
}
```

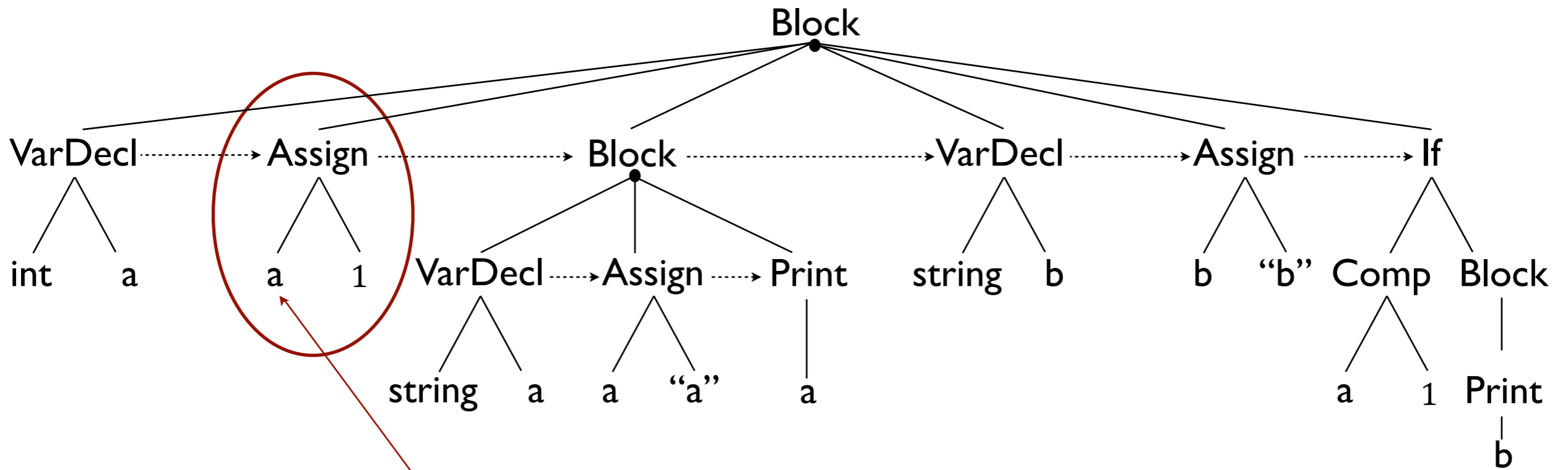
Initialize Scope 0
add symbol A
in the *current scope*



← current scope

Symbol Table

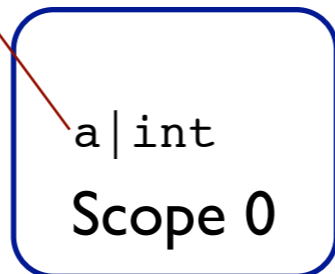
AST



Source Code

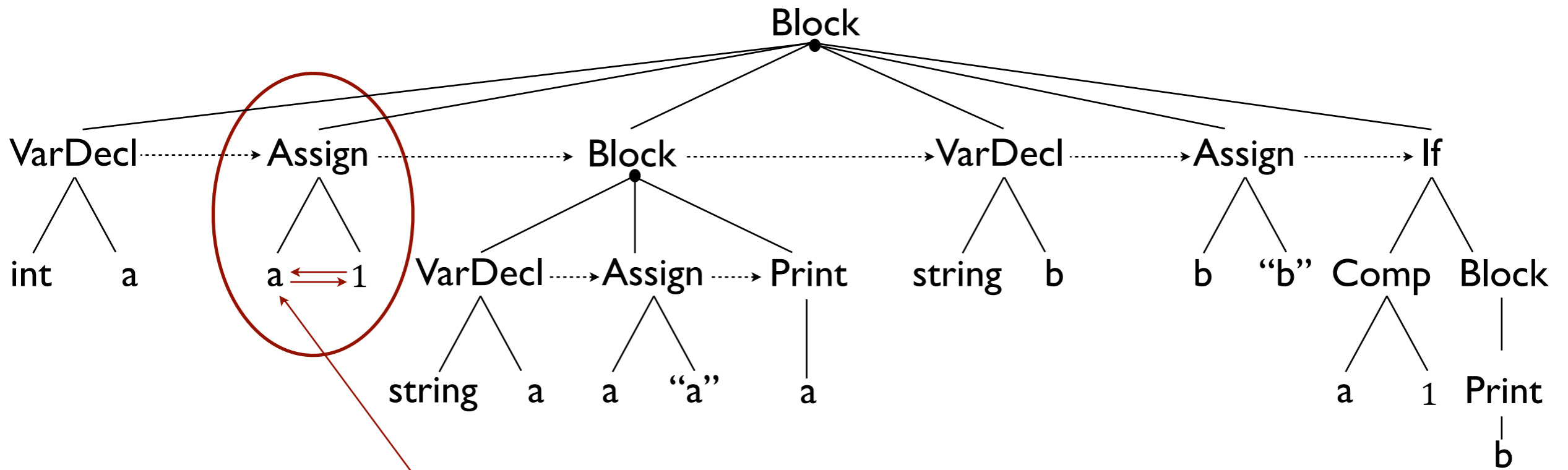
```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

Initialize Scope 0
add symbol A
lookup symbol A
in the *current scope*



Symbol Table

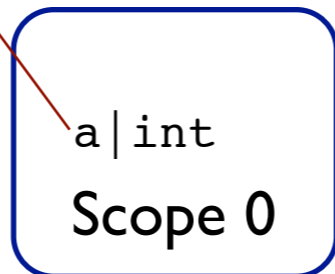
AST



Source Code

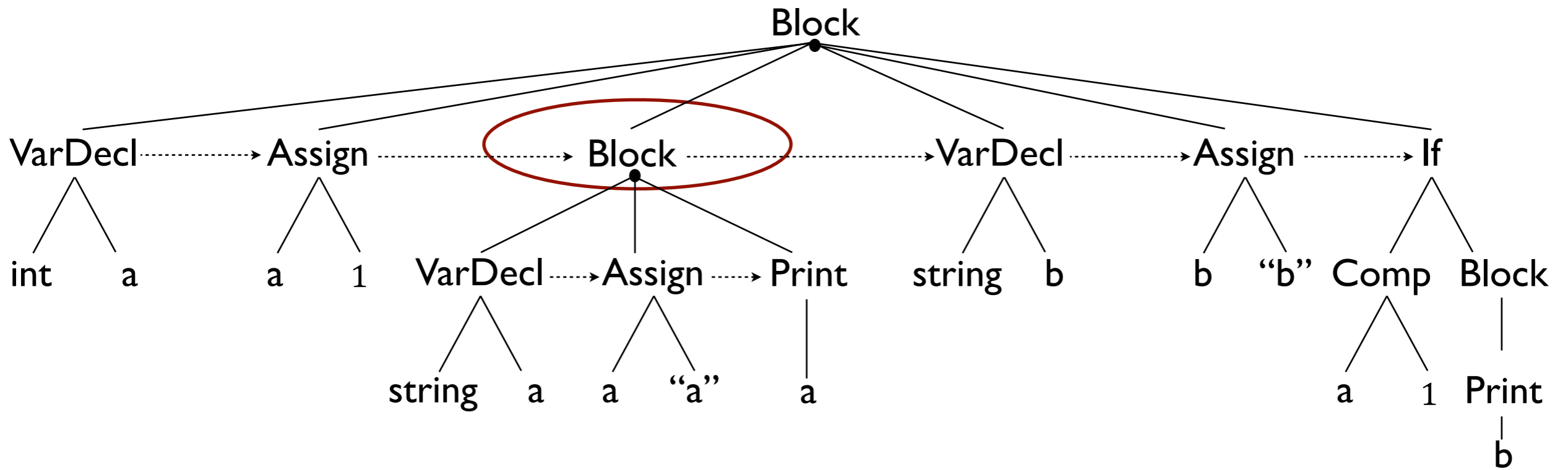
```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

Initialize Scope 0
add symbol A
lookup symbol A
check types
*Verify that the left child
and right child are
type compatible
for assignment.*



Symbol Table

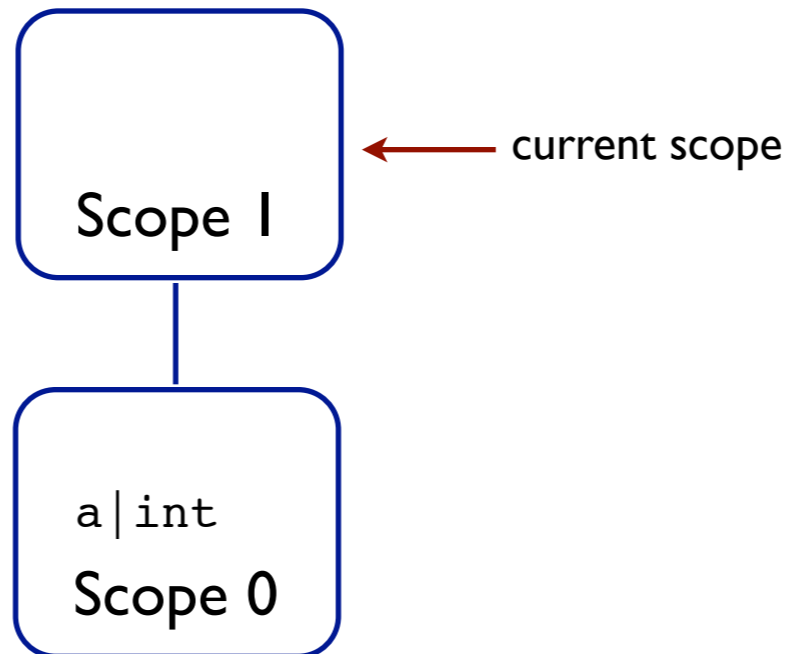
AST



Source Code

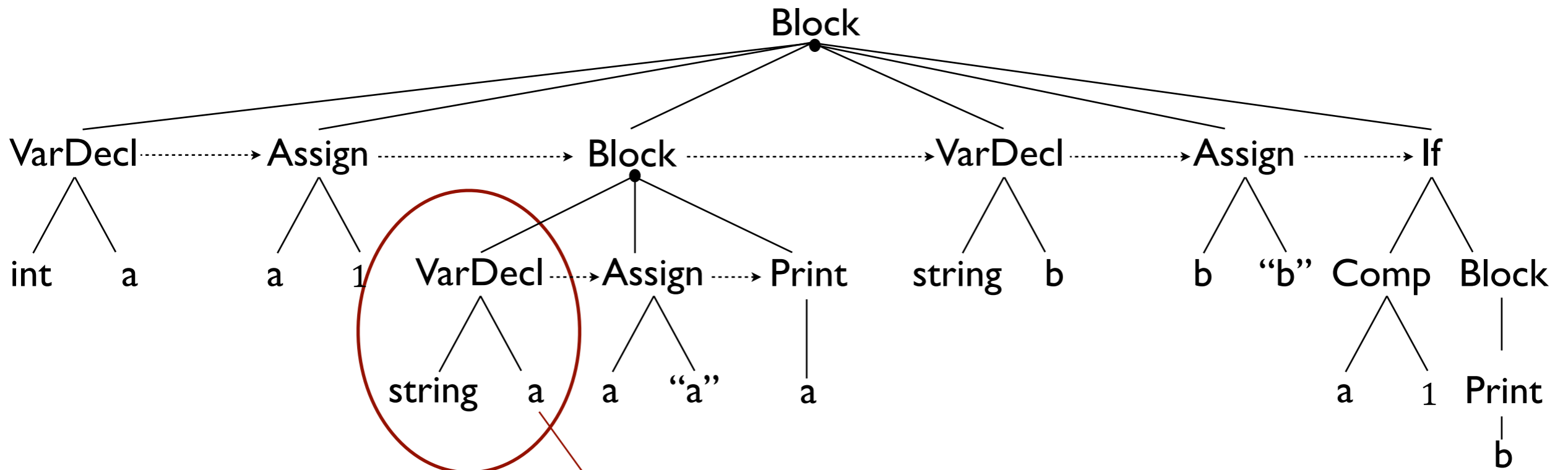
```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
*Move the **current scope** pointer to this child.*



Symbol Table

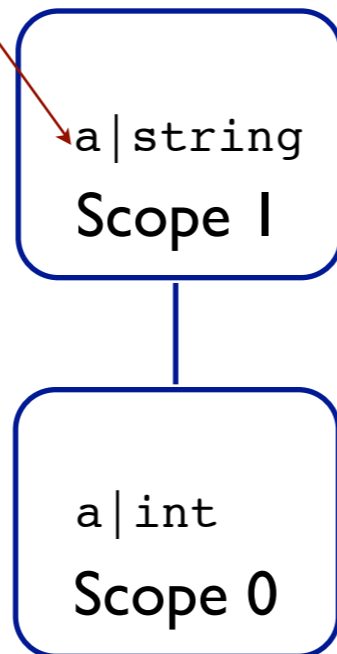
AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

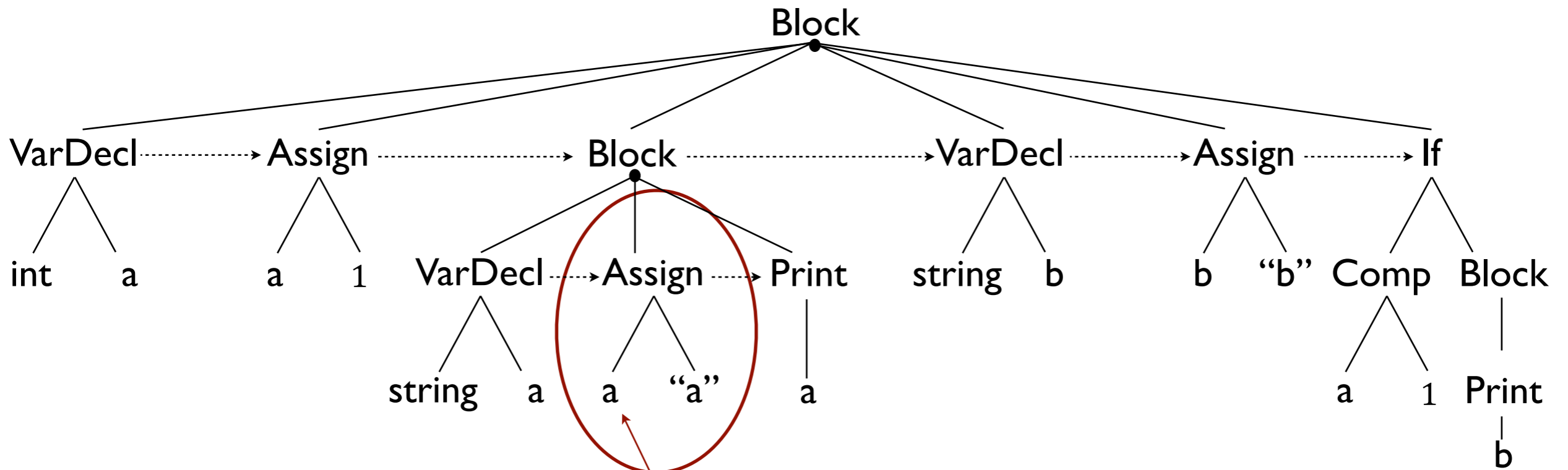
Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
in the current scope



← current scope

Symbol Table

AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
*in the **current scope***

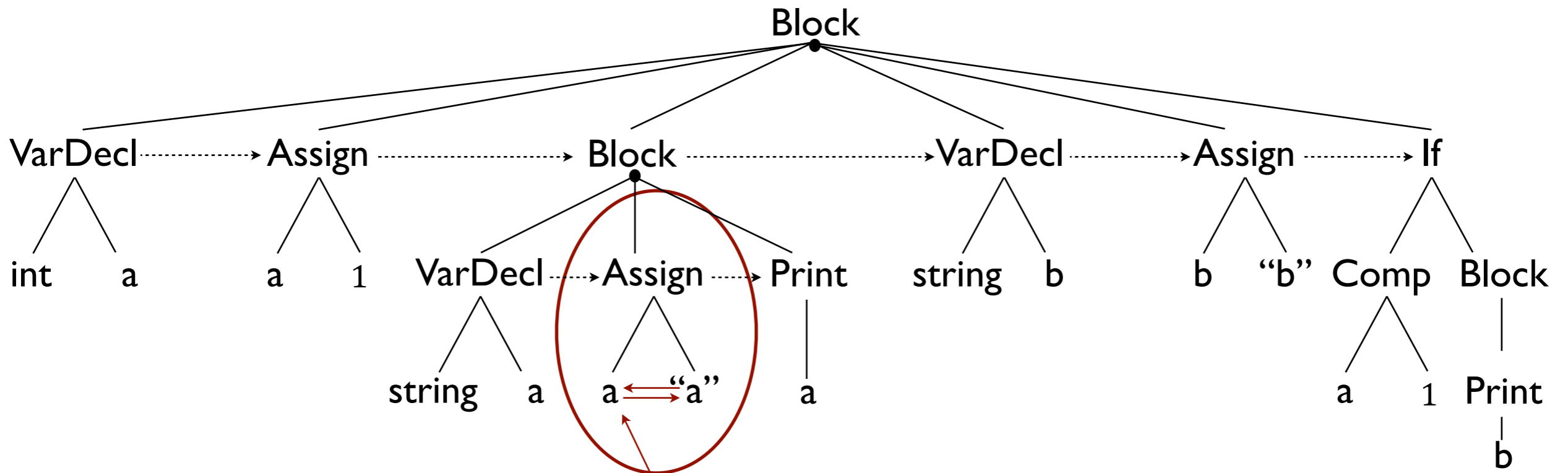
a | string
Scope 1

← current scope

a | int
Scope 0

Symbol Table

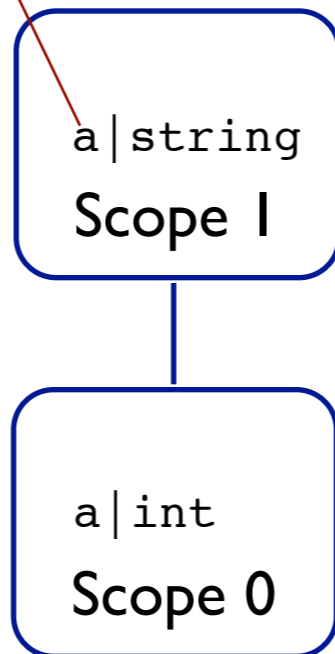
AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

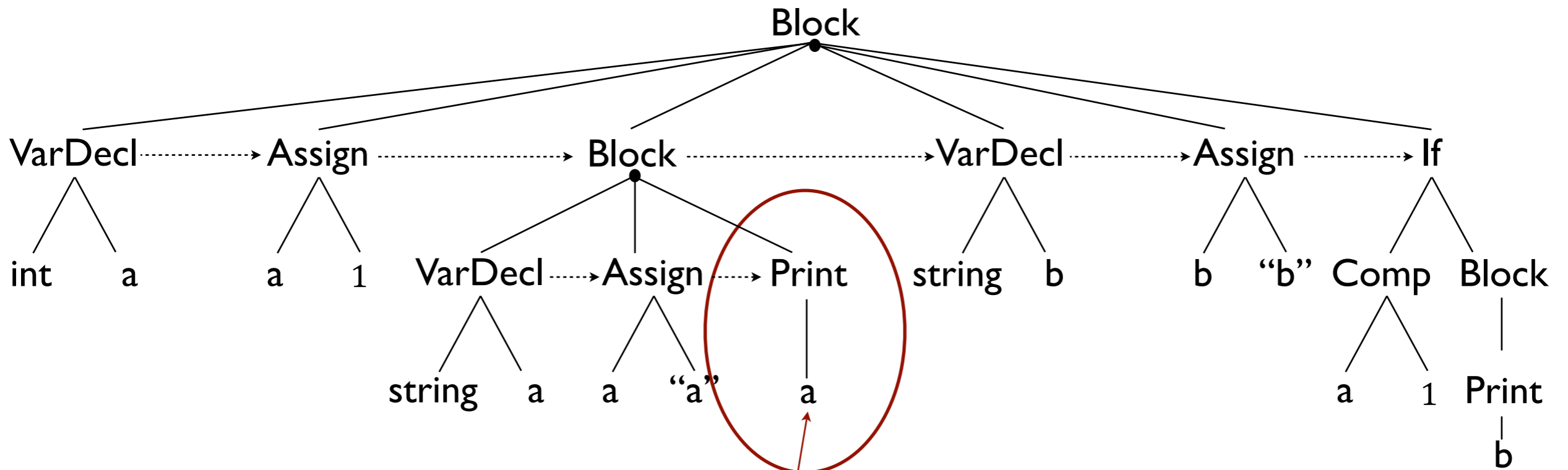
Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
*Verify that the left child
and right child are
type compatible
for assignment.*



← current scope

Symbol Table

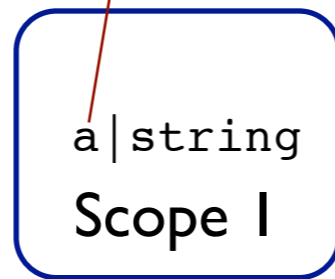
AST



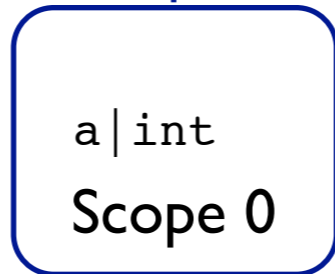
Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
*in the **current scope**.*
Print can take any type,
so there's no need to
type check here.
We must still check the
scope, of course!

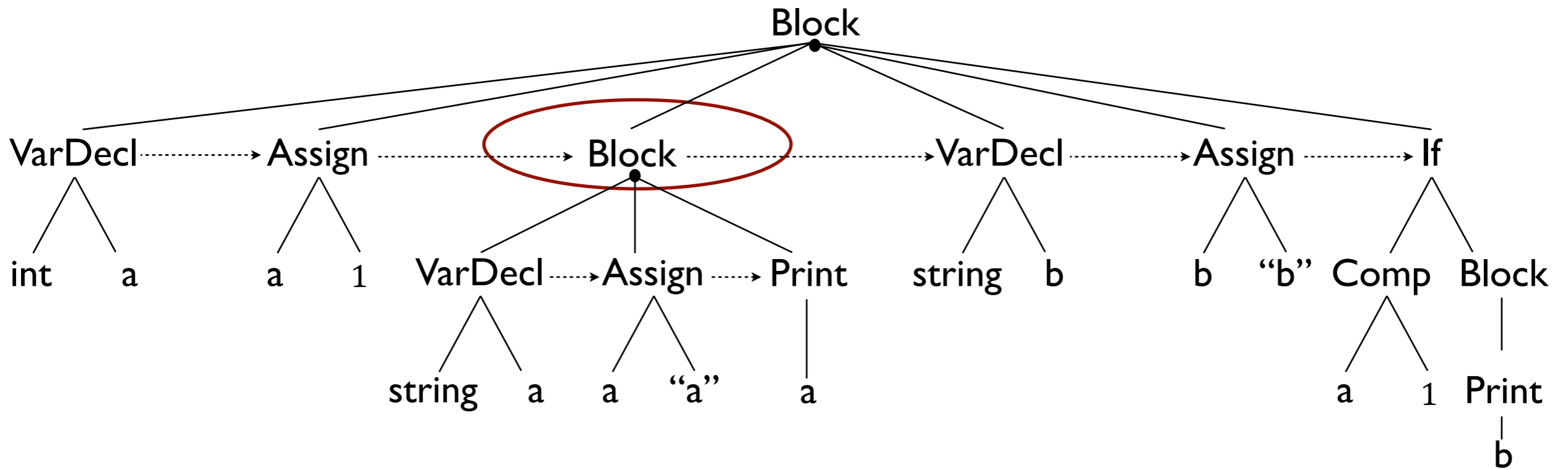


← current scope



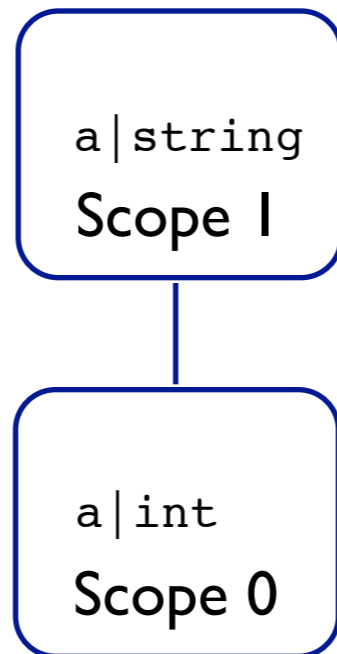
Symbol Table

AST



Source Code

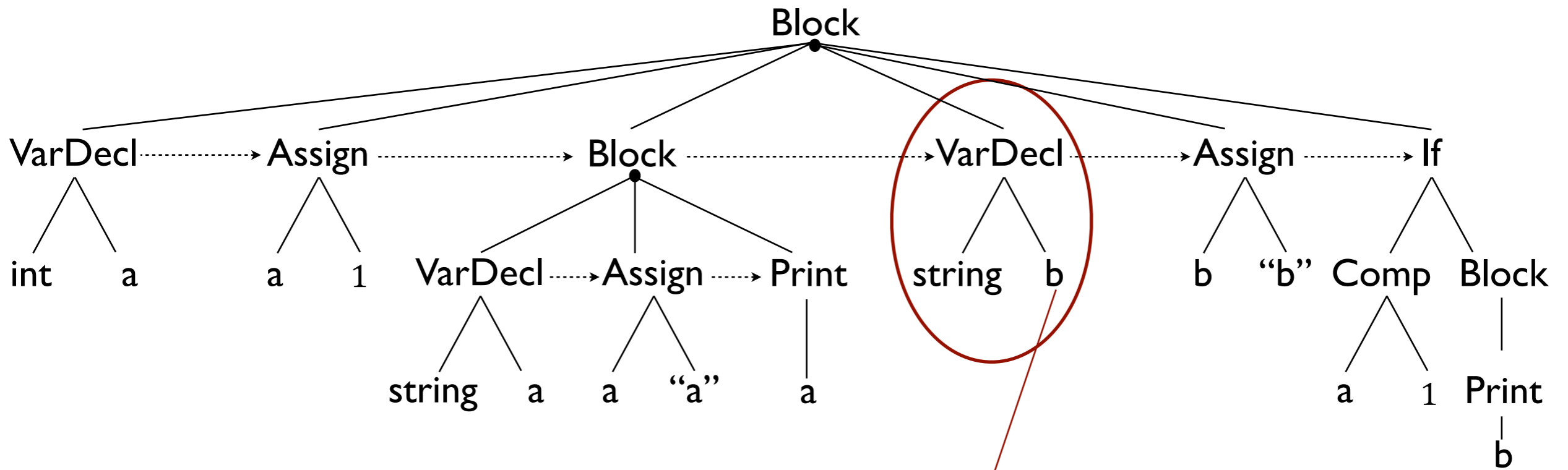
```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```



Symbol Table

Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
Close Scope 1
Move the **current scope** pointer to its parent.

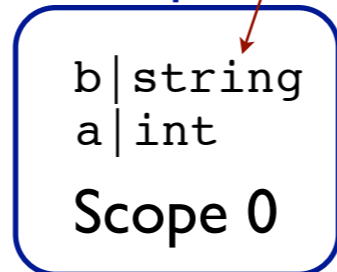
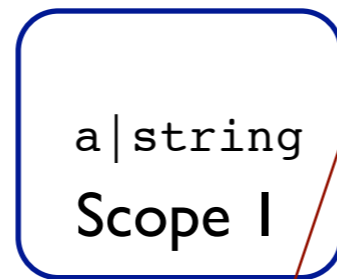
AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

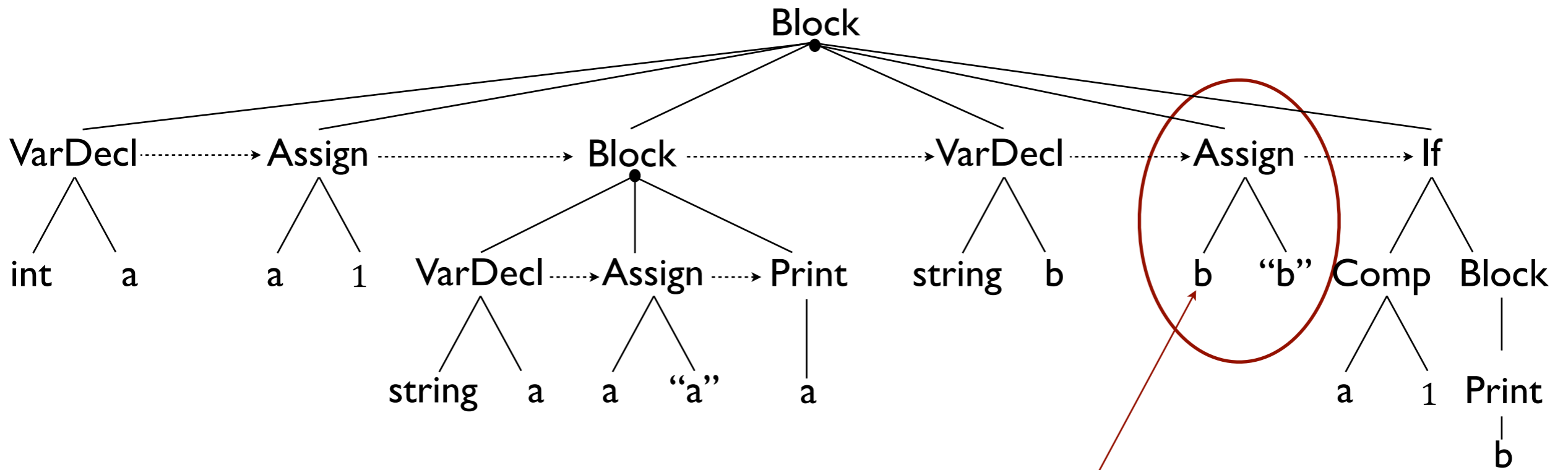
Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
Close Scope 1
add symbol B
*in the **current scope***



← current scope

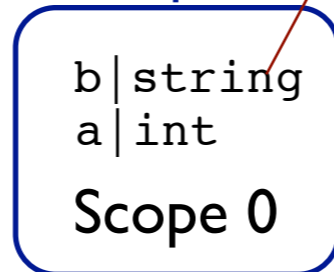
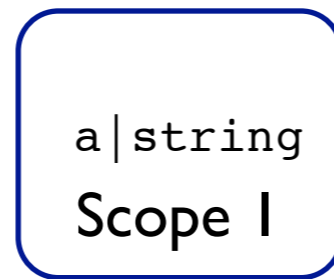
Symbol Table

AST



Source Code

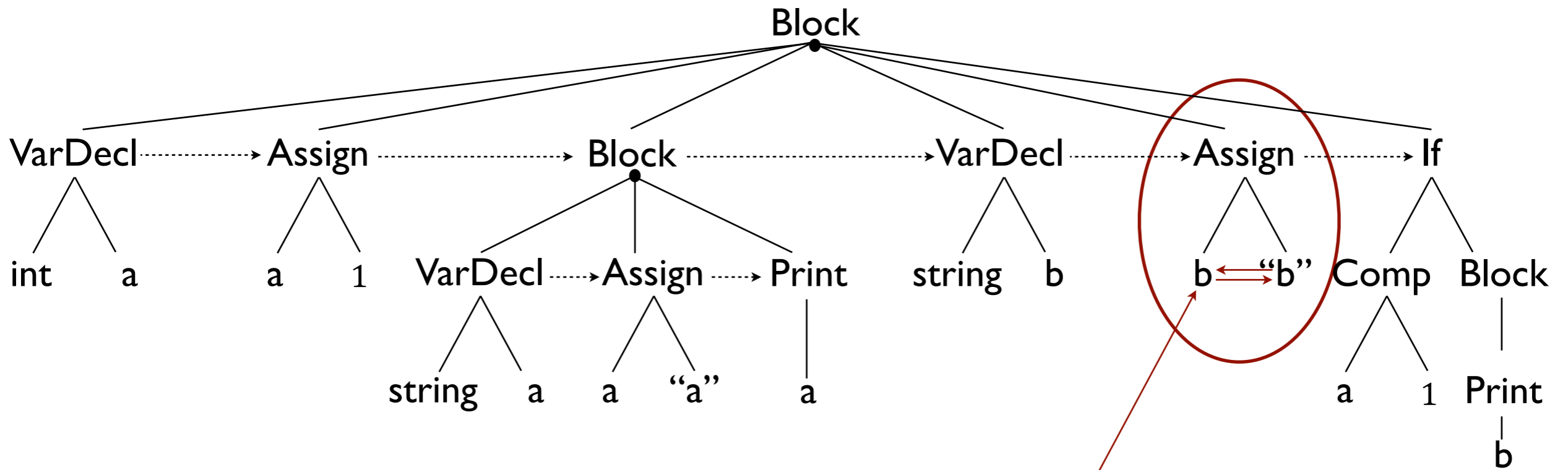
```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```



Symbol Table

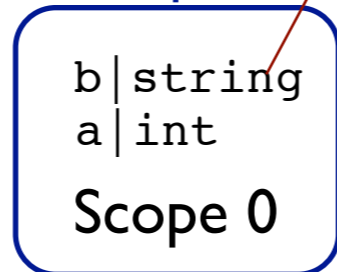
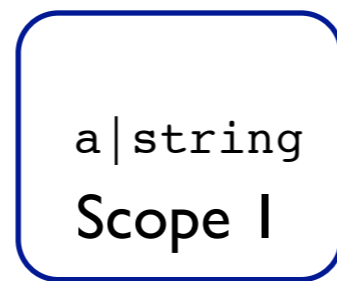
Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
Close Scope 1
add symbol B
lookup symbol B
in the current scope

AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

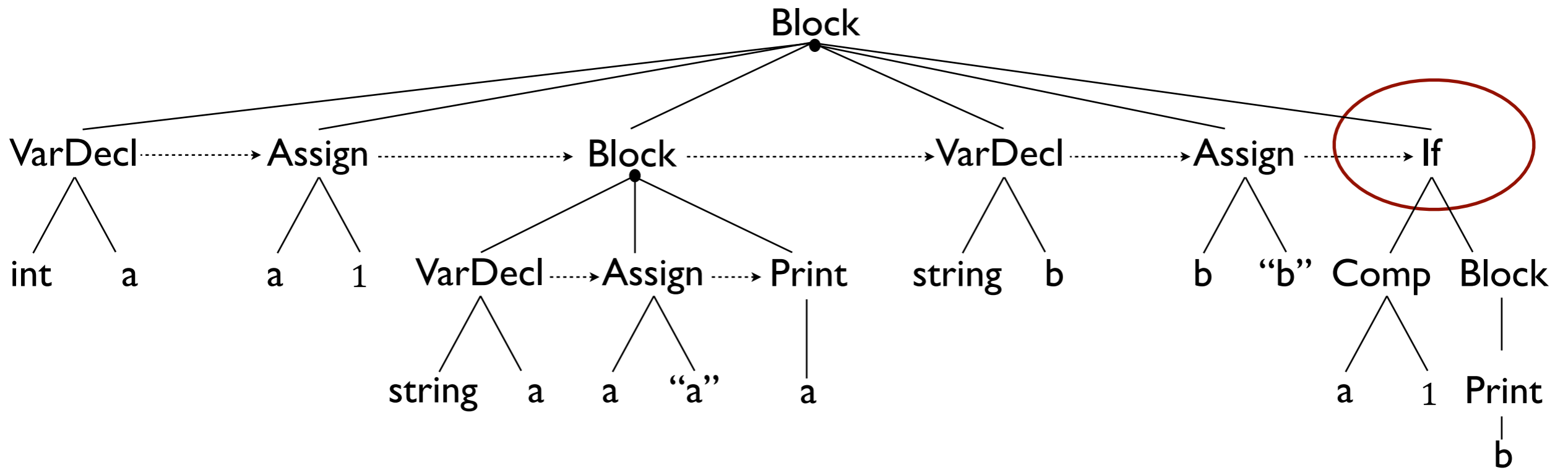


← current scope

Symbol Table

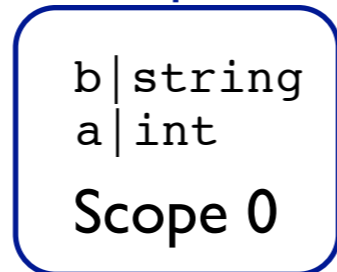
Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
Close Scope 1
add symbol B
lookup symbol B
check types
*Verify that the left child
and right child are
type compatible
for assignment.*

AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

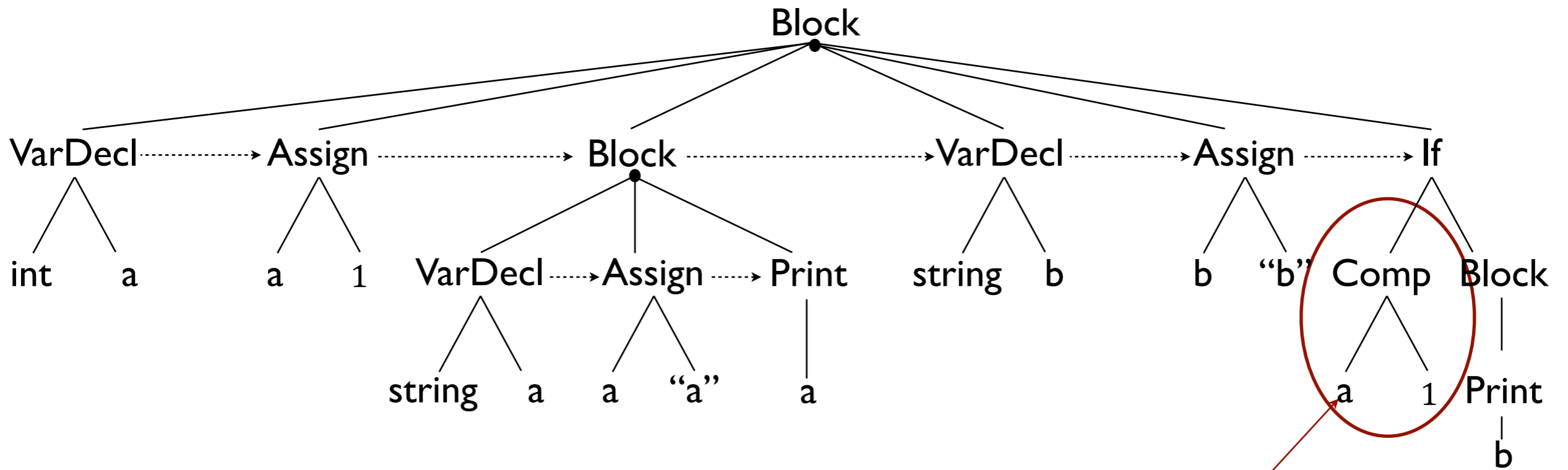


← current scope

Symbol Table

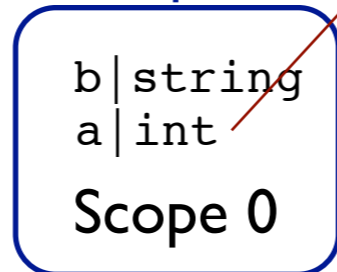
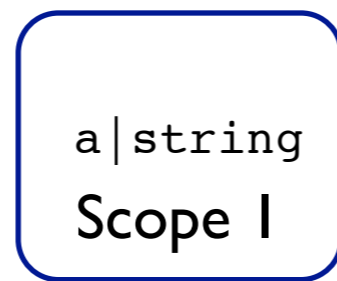
Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
Close Scope 1
add symbol B
lookup symbol B
check types

AST



Source Code

```
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
```

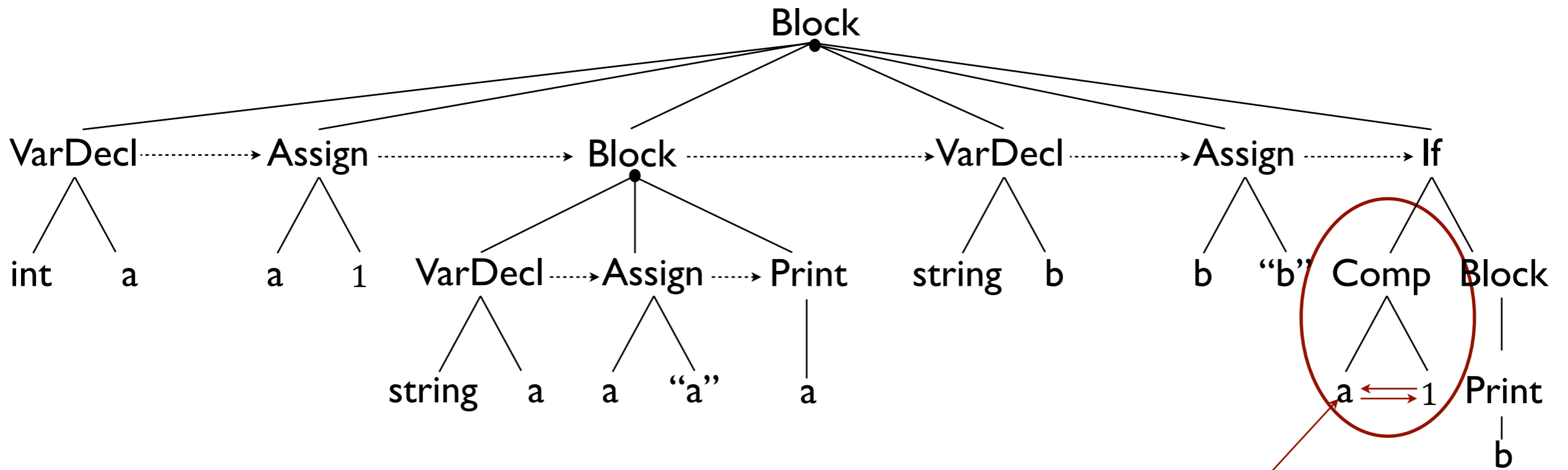


← current scope

Symbol Table

Initialize Scope 0
add symbol A
lookup symbol A
check types
Initialize Scope 1
add symbol A
lookup symbol A
check types
lookup symbol A
Close Scope 1
add symbol B
lookup symbol B
check types
lookup symbol A
*in the **current scope***

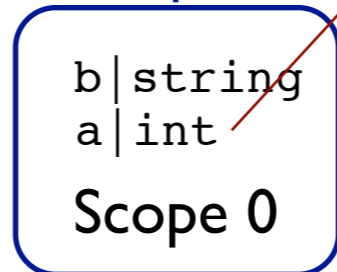
AST



Source Code

```

{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
    
```



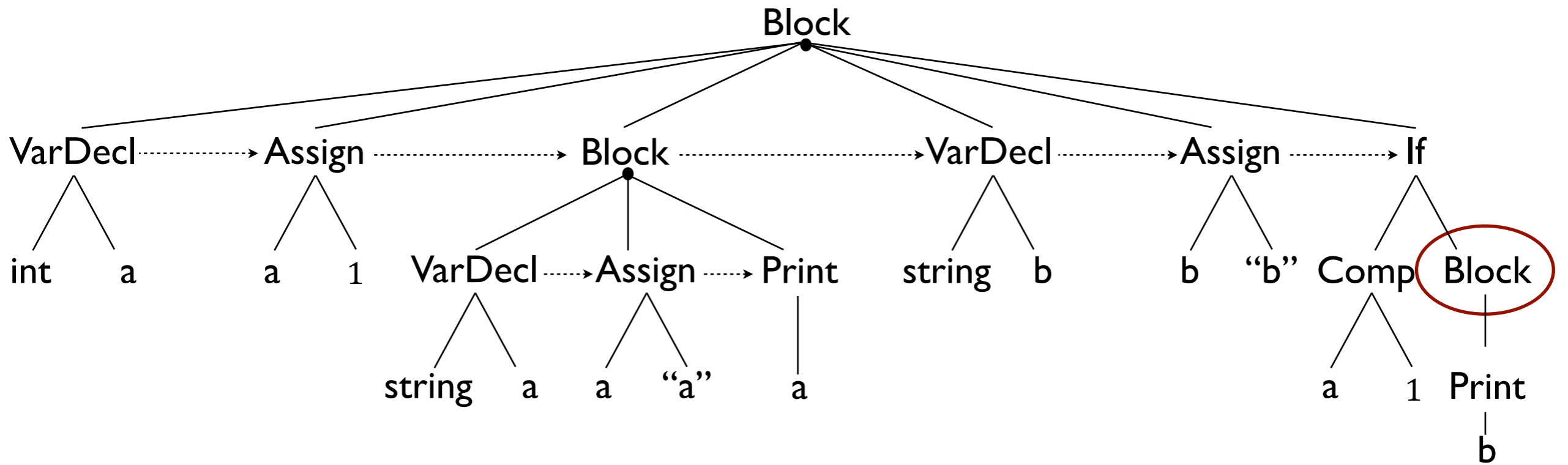
← current scope

Symbol Table

Initialize Scope 0
 add symbol A
 lookup symbol A
 check types
 Initialize Scope 1
 add symbol A
 lookup symbol A
 check types
 lookup symbol A
 Close Scope 1
 add symbol B
 lookup symbol B
 check types
 lookup symbol A
 check types

Verify that the left child and right child are type comparable

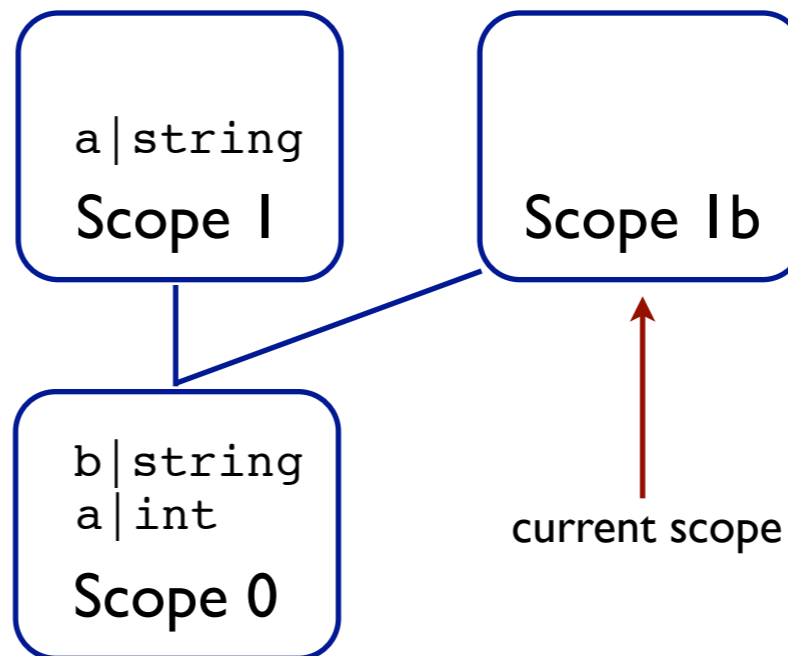
AST



Source Code

```

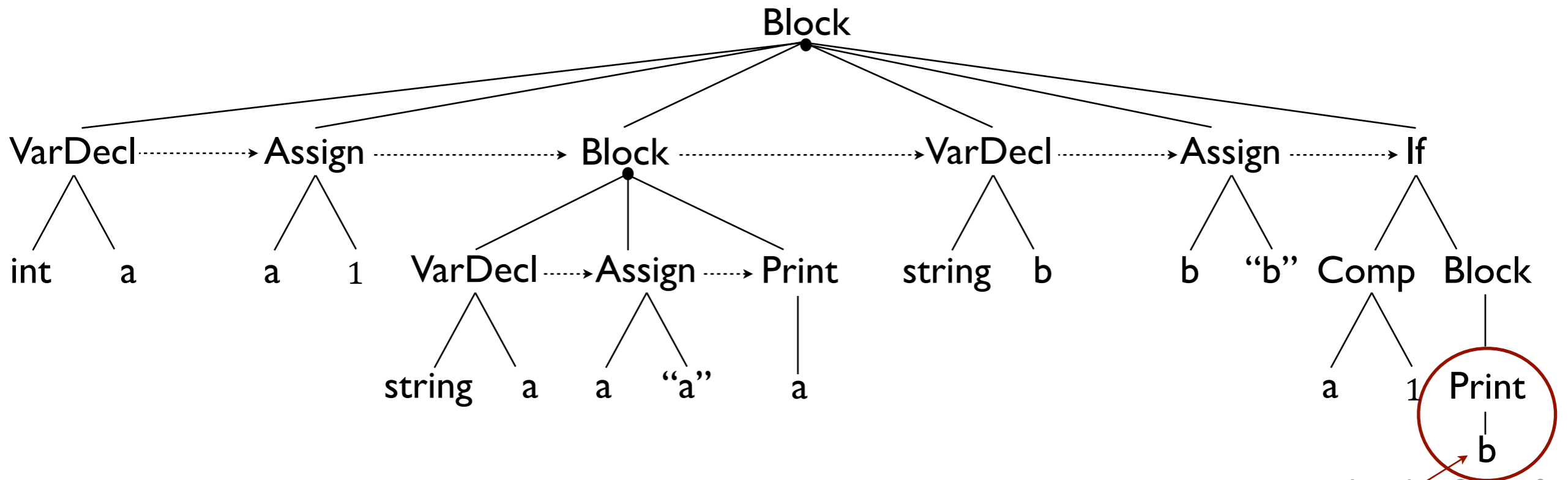
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
  
```



Symbol Table

Initialize Scope 0
 add symbol A
 lookup symbol A
 check types
 Initialize Scope I
 add symbol A
 lookup symbol A
 check types
 lookup symbol A
 Close Scope I
 add symbol B
 lookup symbol B
 check types
 lookup symbol A
 check types
 Initialize Scope Ib
 Move the **current scope** pointer to this child.

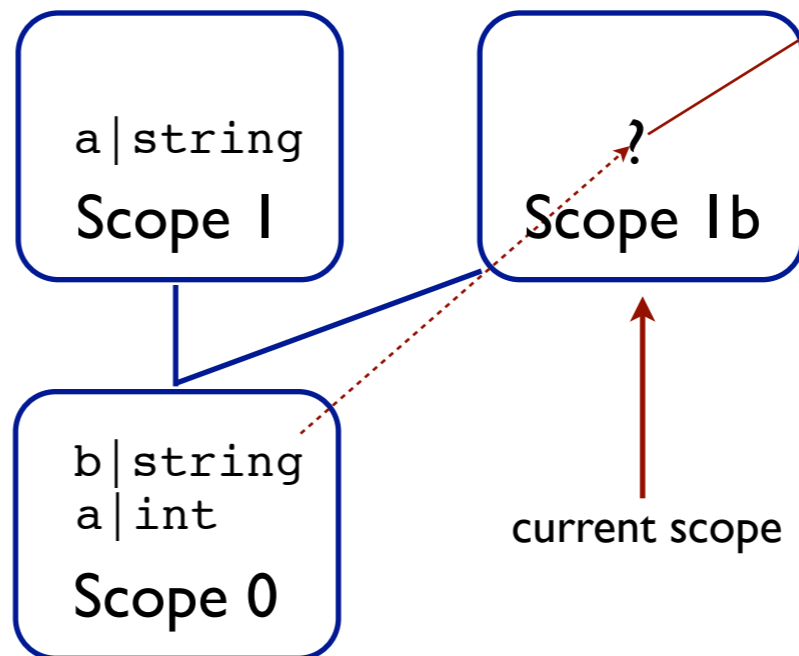
AST



Source Code

```

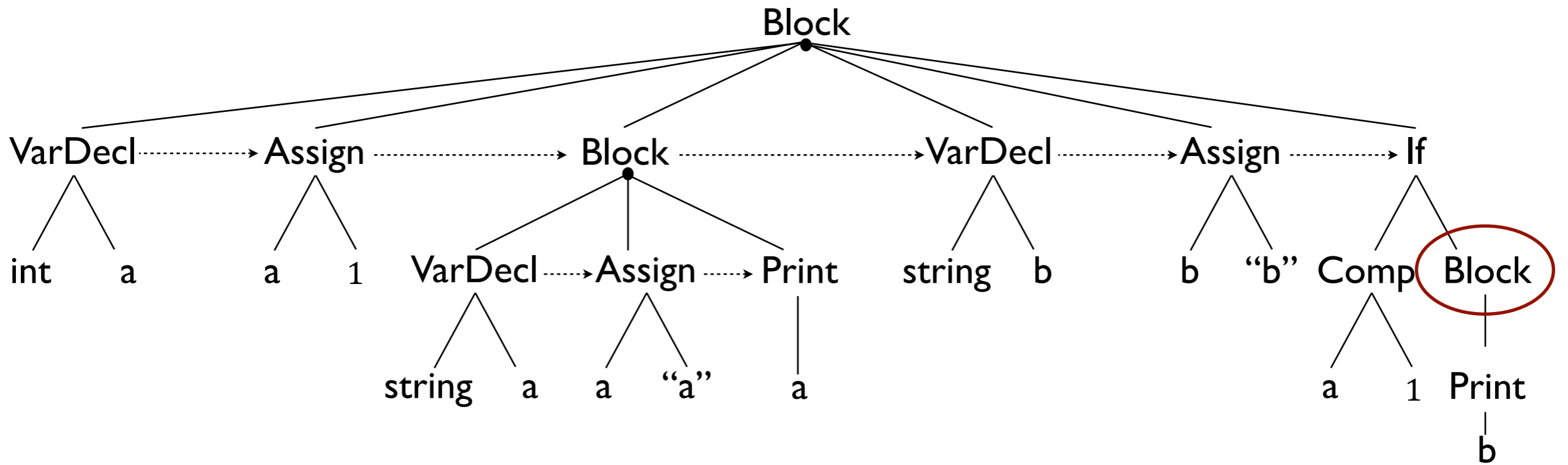
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
  
```



Symbol Table

Initialize Scope 0
 add symbol A
 lookup symbol A
 check types
 Initialize Scope I
 add symbol A
 lookup symbol A
 check types
 lookup symbol A
 Close Scope I
 add symbol B
 lookup symbol B
 check types
 lookup symbol A
 check types
 Initialize Scope Ib
 lookup symbol B
 in the **current scope**.
 Print can take any type.

AST

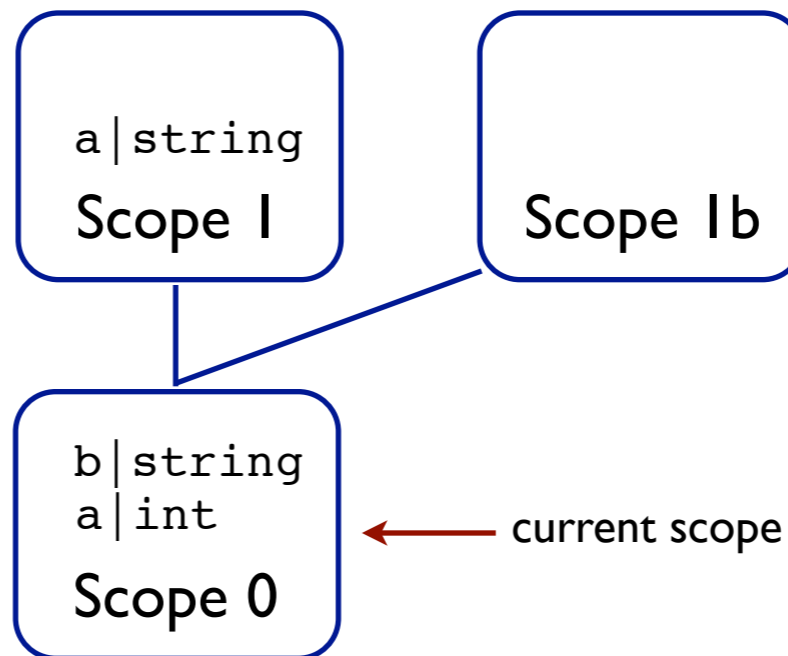


Source Code

```

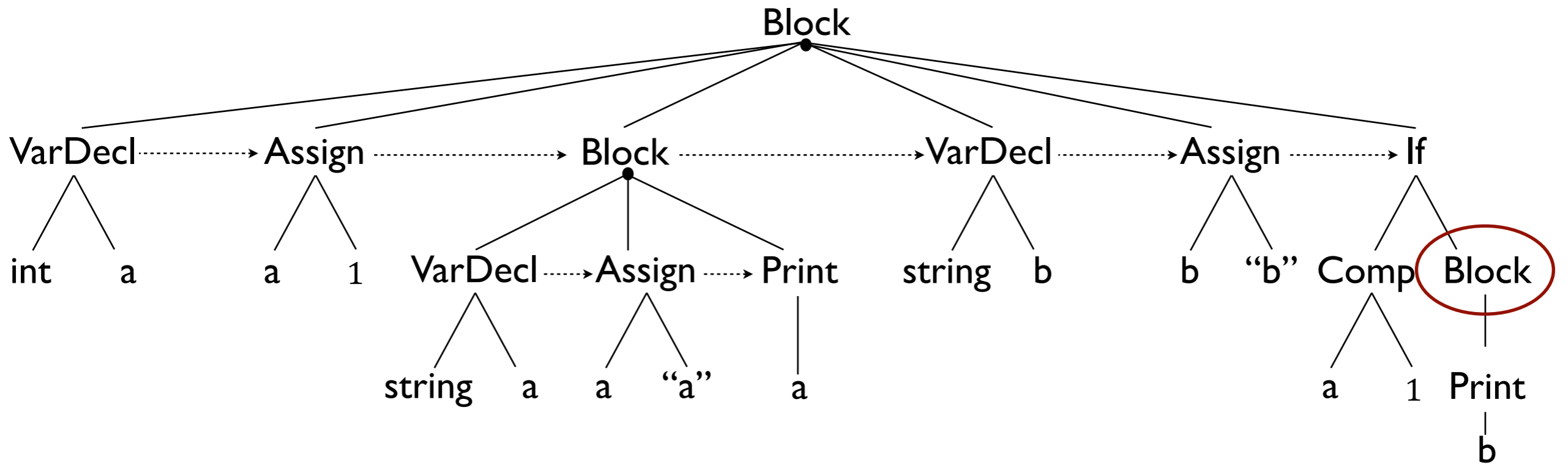
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
  
```

Initialize Scope 0
 add symbol A
 lookup symbol A
 check types
 Initialize Scope 1
 add symbol A
 lookup symbol A
 check types
 lookup symbol A
 Close Scope 1
 add symbol B
 lookup symbol B
 check types
 lookup symbol A
 check types
 Initialize Scope 1b
 lookup symbol B
 Close Scope 1b



Symbol Table

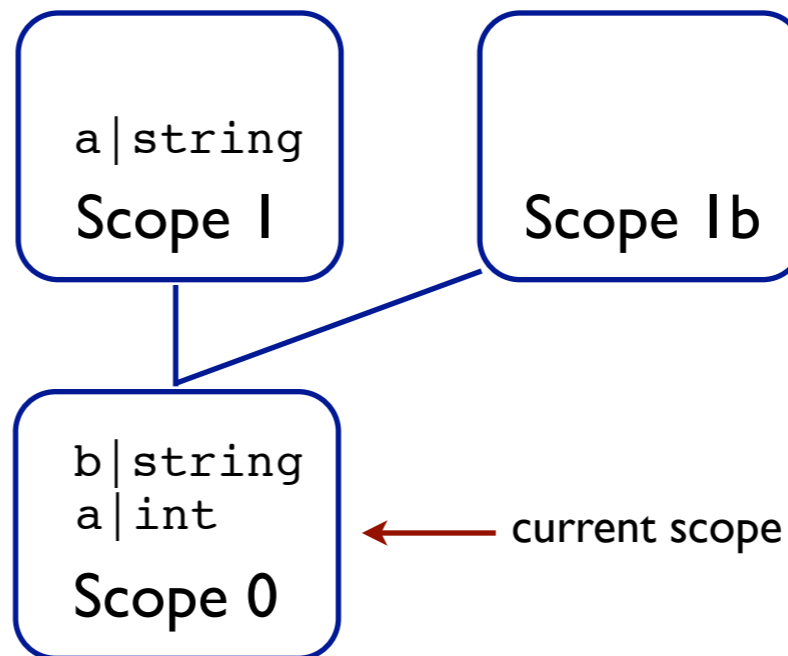
AST



Source Code

```

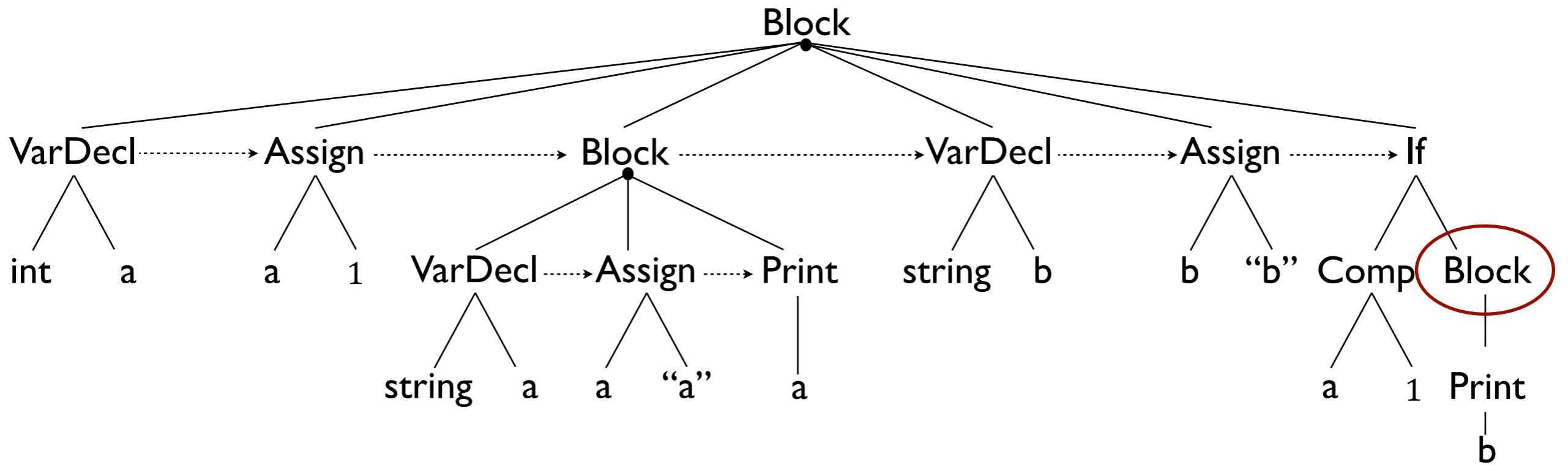
{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
  
```



Symbol Table

Initialize Scope 0
 add symbol A
 lookup symbol A
 check types
 Initialize Scope I
 add symbol A
 lookup symbol A
 check types
 lookup symbol A
 Close Scope I
 add symbol B
 lookup symbol B
 check types
 lookup symbol A
 check types
 Initialize Scope Ib
 lookup symbol B
 Close Scope Ib
 Close Scope 0

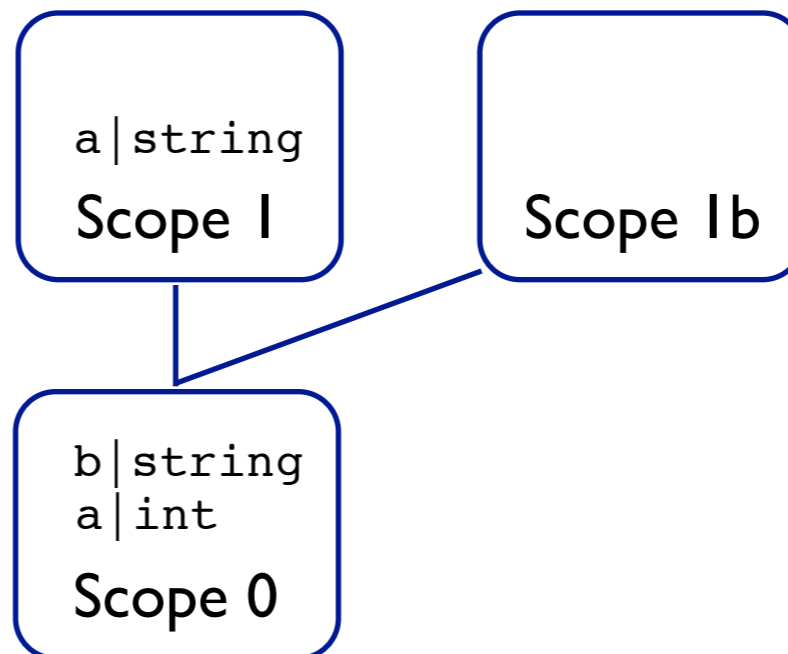
AST



Source Code

```

{
  int a
  a = 1
  {
    string a
    a = "a"
    print(a)
  }
  string b
  b = "b"
  if (a == 1) {
    print(b)
  }
}
  
```



Symbol Table

Initialize Scope 0
 add symbol A
 lookup symbol A
 check types
 Initialize Scope I
 add symbol A
 lookup symbol A
 check types
 lookup symbol A
 Close Scope I
 add symbol B
 lookup symbol B
 check types
 lookup symbol A
 check types
 Initialize Scope Ib
 lookup symbol B
 Close Scope Ib
 Close Scope 0