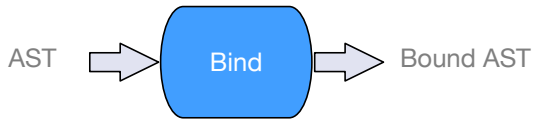


Compiler Construction

~ Binder ~

Goal

Link uses to declarations!

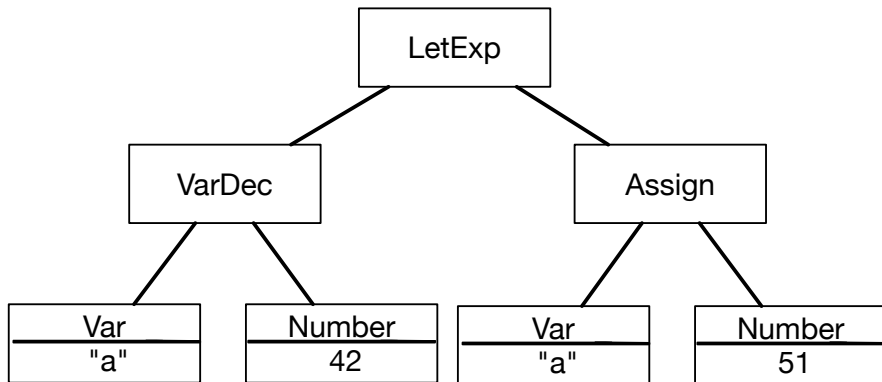


Example: bound/threaded AST

```
let var a := 42 in a := 51 end
```

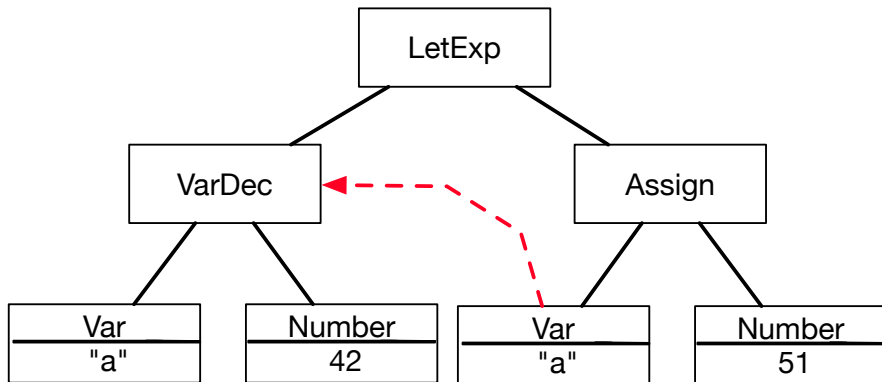
Example: bound/threaded AST

```
let var a := 42 in a := 51 end
```



Example: bound/threaded AST

```
let var a := 42 in a := 51 end
```



Dealing with Scopes

Multiple definitions for an identifier

Which one is the good one?

```
let var a := 42
```

```
in
```

```
    let var a := 51
```

```
    in
```

```
        a
```

```
    end;
```

```
    a
```

```
end
```

1

2

3

4

5

6

7

8

Environments

Definition

An environment is a set of bindings
denoted by \vdash

Example

$\sigma_0 = \{g \vdash \text{string}, a \vdash \text{int}\}$
means that a is an integer variable and
 g is a string variable

Scopes

```
structure M = struct
  structure E = struct
    val a = 5;
  end
  structure N = struct
    val b = 10;
    val a = E.a + b;
  end
  structure D = struct
    val d = E.a + N.a;
  end
end
```

$\sigma_0 = \text{Prelude}$

$\sigma_1 = \{a : \text{int}\}$

$\sigma_2 = \{E : \sigma_1\}$

$\sigma_3 = \{b : \text{int}, a : \text{int}\}$

$\sigma_4 = \{N : \sigma_3\}$

$\sigma_5 = \{d : \text{int}\}$

$\sigma_6 = \{D : \sigma_5\}$

$\sigma_7 = \sigma_2 + \sigma_4 + \sigma_6$

$\sigma_0 + \sigma_2 \vdash N : \sigma_3 \quad (\text{ML})$

$\sigma_0 + \sigma_2 + \sigma_4 \vdash N : \sigma_3 \quad (\text{Java})$

$\sigma_0 + \sigma_2 + \sigma_4 + \sigma_6 \vdash M : \sigma_7$

Binder: main idea

- 1 Start with an empty environment
- 2 For each declaration, add the identifier to the environment
- 3 For each use, link to the correct declaration
 - ⇒ Build here the threaded AST
 - ⇒ The environment handles references to AST nodes rather than types
- 4 At the end of a scope remove identifiers that are no longer visible

Summary

