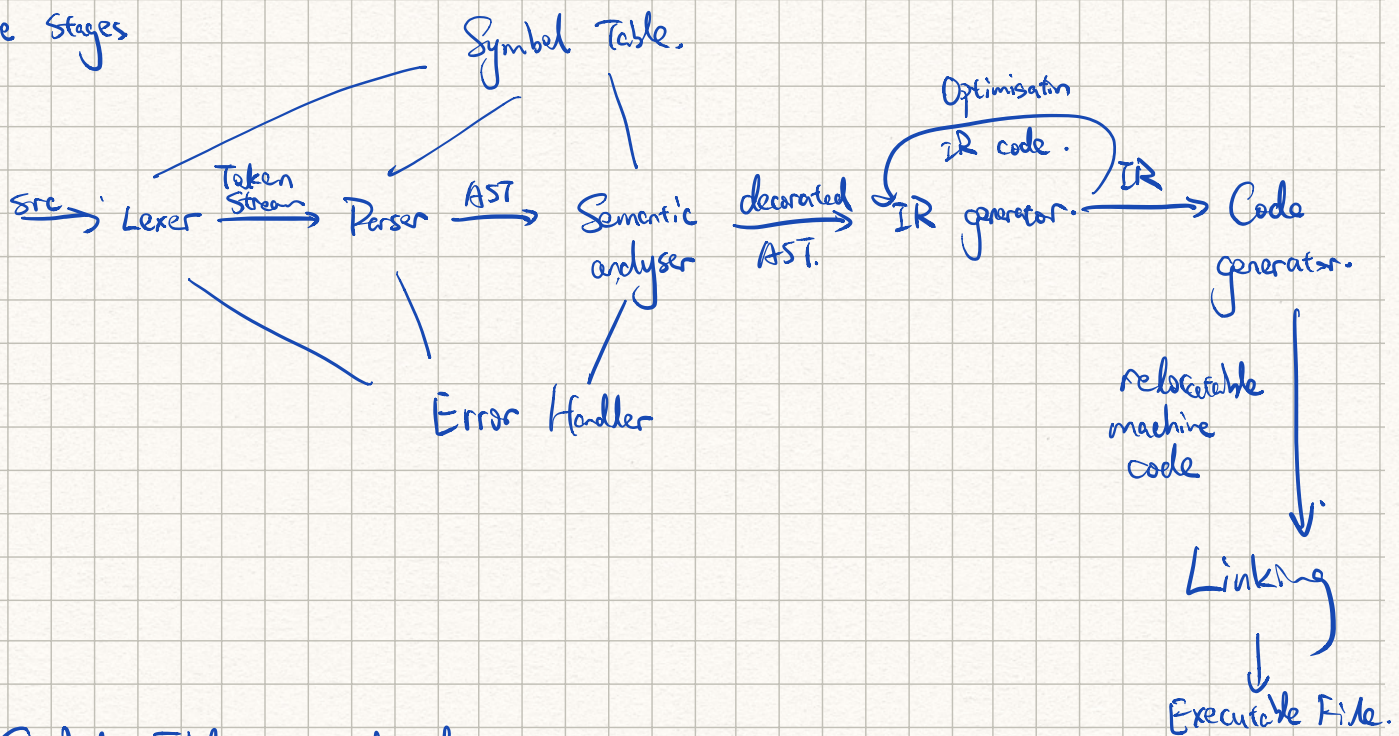Activation record.
- Control link.
- access link
- function parameters
- temp values in evaluation.
- local variables
- return value.
- return address

Five stages

Symbol Table.

$src \rightarrow$ Lexer $\xrightarrow{\text{Token Stream}}$ Parser $\xrightarrow{\text{AST}}$ Semantic analyser $\xrightarrow[\text{AST.}]{\text{decorated}}$ IR generator. $\xrightarrow{\text{IR}}$ Code generator.

Optimisation
IR code.

Error Handler

relocatable machine code

Linking

Executable File.

Symbol Table pseudocode.

```
public class SymbolTable.   {.
    private HashTable SymbolTable.j
    public S.T () {.
        S.t = new HashTable ();
    }.
}.
```

```java
public s.t  add (String id, Object type) {
        s.t. put(id, type);
    }

    public  s.t.  get (String id)      {
        for (var tableID : s.t. entrySet()) {
            if (tableID. equals(id)) return
                            s.t.getKey(tableID );
        }
        return NULL;
    }
}

abstract
public class Node () {
    Node left, right;

    public  abstract Object accept (Visitor V);

}

public class DeclNode extends Node {
    public String id, Object type)

    public Object accept (Visitor v)
        { return v. visit(this); }
}

public class Checker implements Visitor {
    SumbelTable [] SumbelTable Stack.
```

```
     int  stacktop=0.
public      checker () {

        Symbol Table [stacktop] = new Sy. T. ();

    }

    public Boolen visit ( Noden) {

    }
}
```

Eliminate   left recursion

* Eliminate dead rules that can never be reached
* Eliminate obvious ones
* Eliminate substitution recursion
* Eliminate ε-masked recursion