



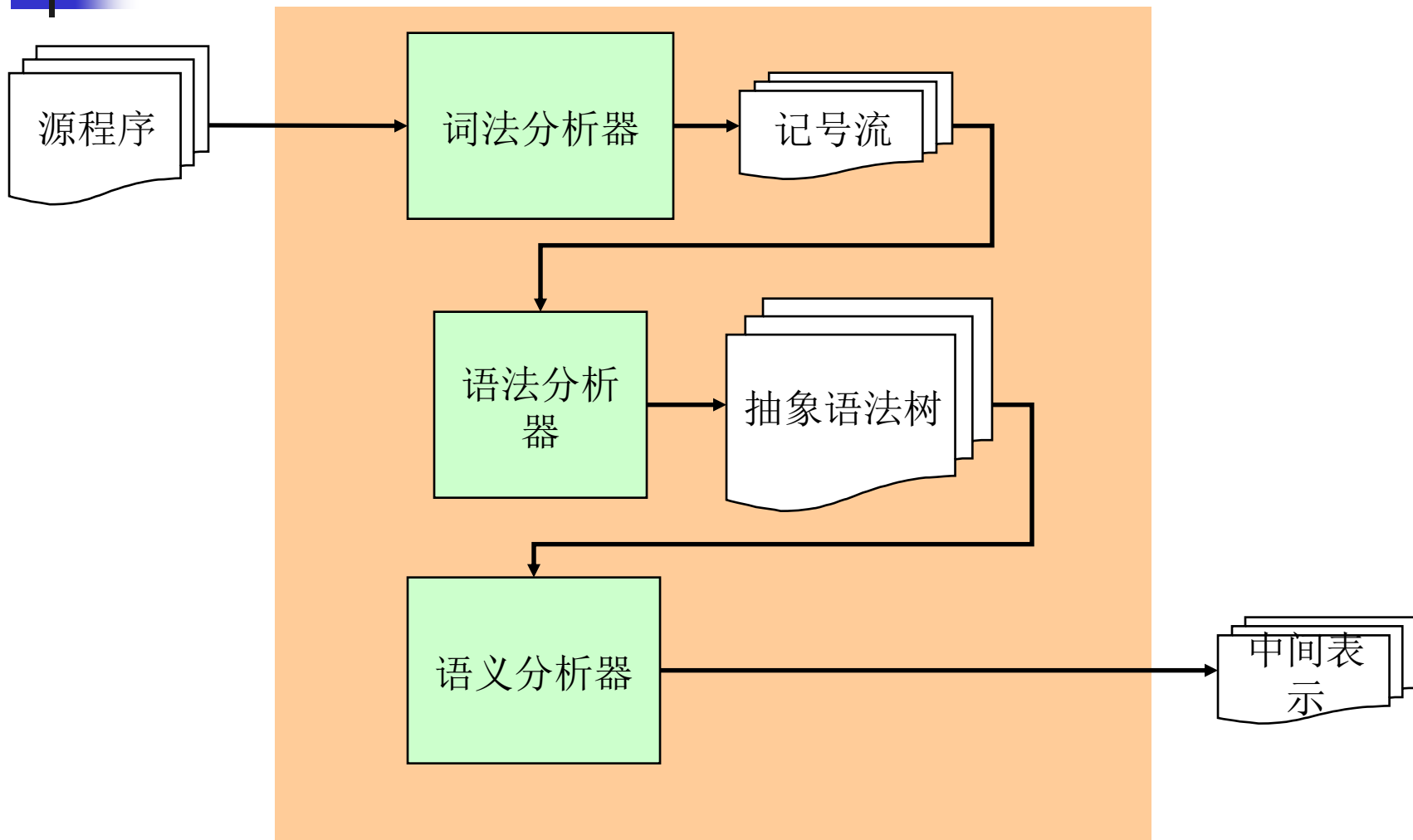
词法分析---手工构造法

编译原理

华保健

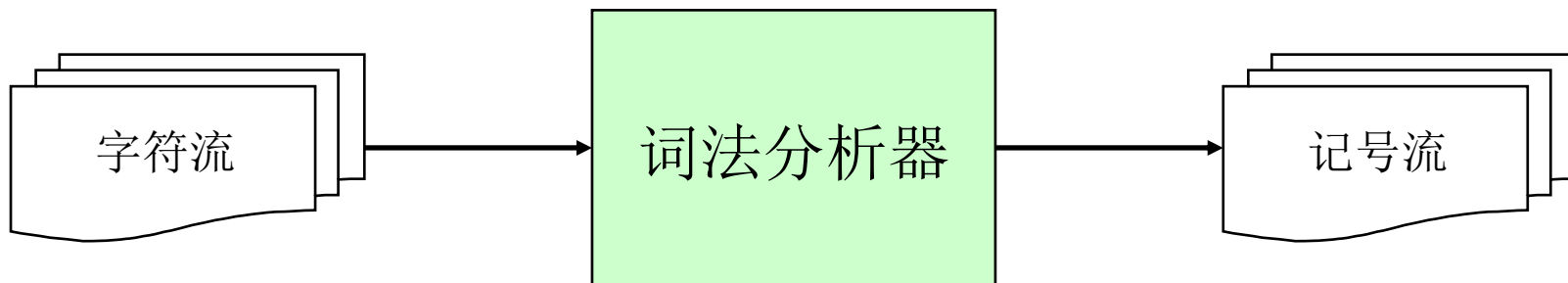
bjhua@ustc.edu.cn

前端





词法分析器的任务

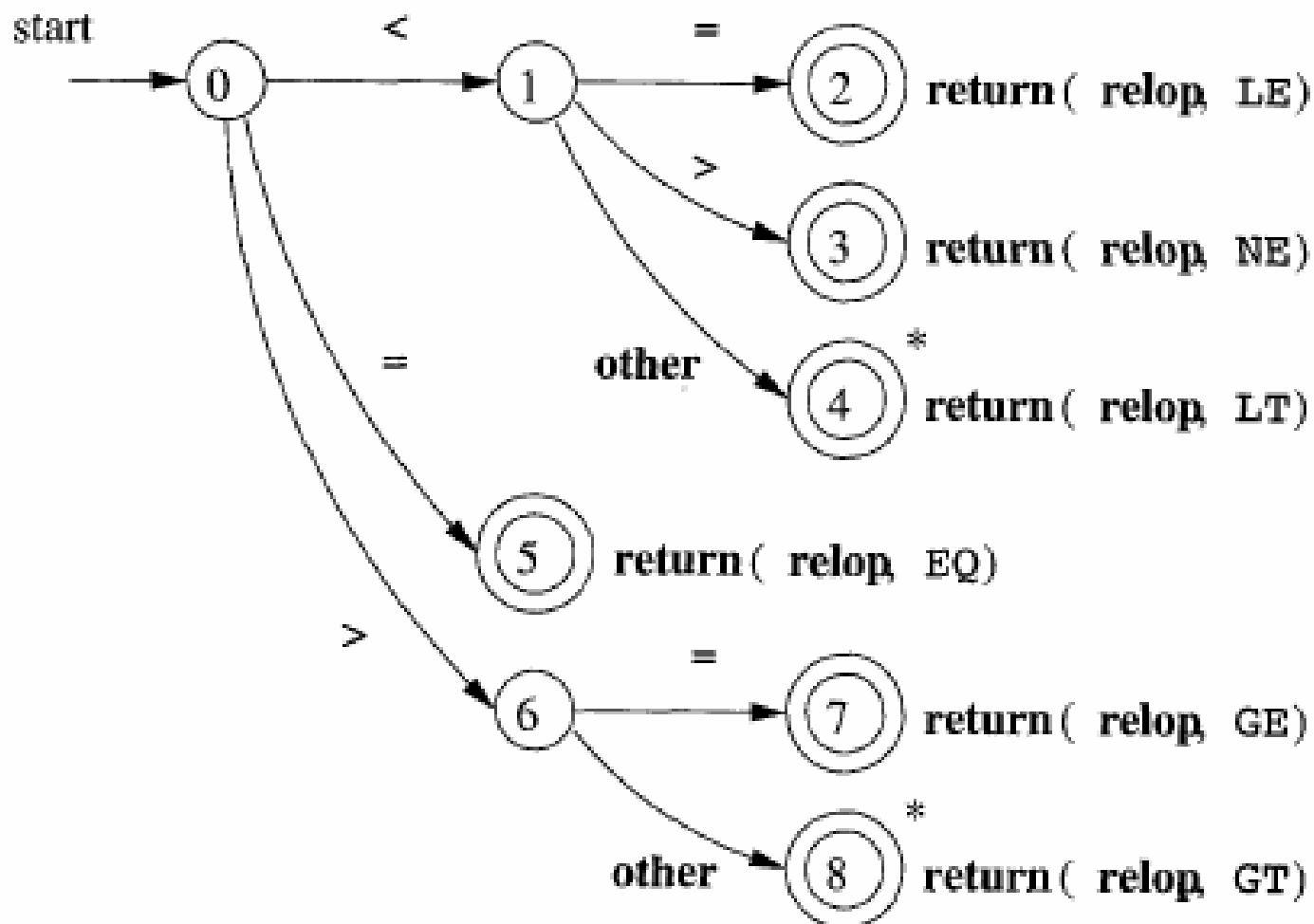




词法分析器的实现方法

- 至少两种实现方案：
 - 手工编码实现法
 - 相对复杂、且容易出错
 - 但是目前非常流行的实现方法
 - GCC, LLVM, ...
 - 词法分析器的生成器
 - 可快速原型、代码量较少
 - 但较难控制细节
- 我们先讨论第一种实现方案
 - 后面几讲会讨论第二种方案

转移图



转移图算法

```
token nextToken ()
```

```
    c = getChar ();
```

```
    switch (c)
```

```
        case '<': c = getChar ();
```

```
            switch (c)
```

```
                case '=': return LE;
```

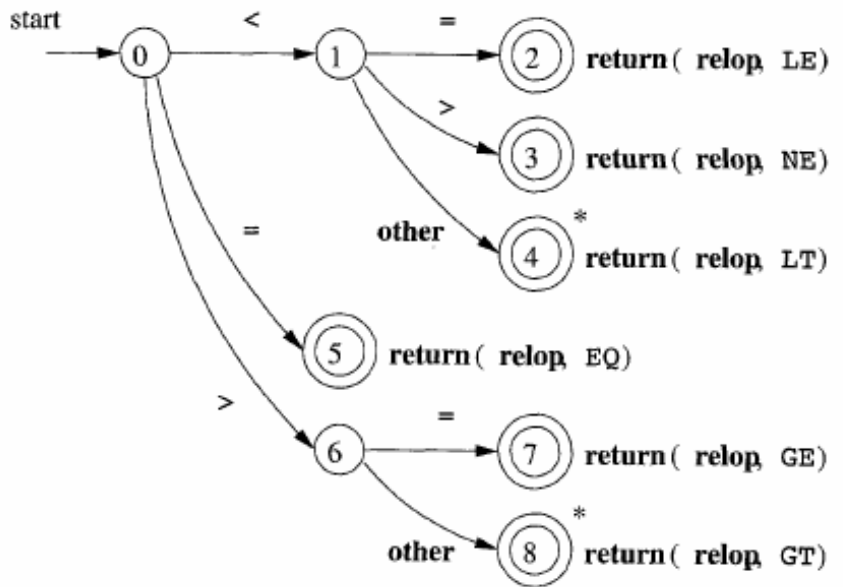
```
                case '>': return NE;
```

```
                default: rollback(); return LT;
```

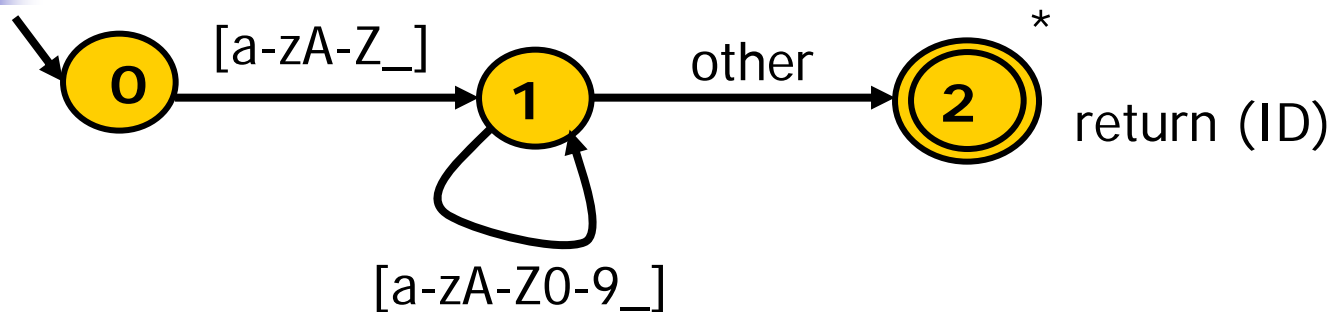
```
        case '=': return EQ;
```

```
        case '>': c = nextChar ();
```

```
            switch (c): // similar
```



标识符的转移图



```
token nextToken ()
```

```
    c = getChar();
```

```
    switch (c)
```

```
    // continued from above cases...
```

```
    case 'a', ..., 'z', 'A', ..., 'Z', '_':
```

```
        c = getChar ();
```

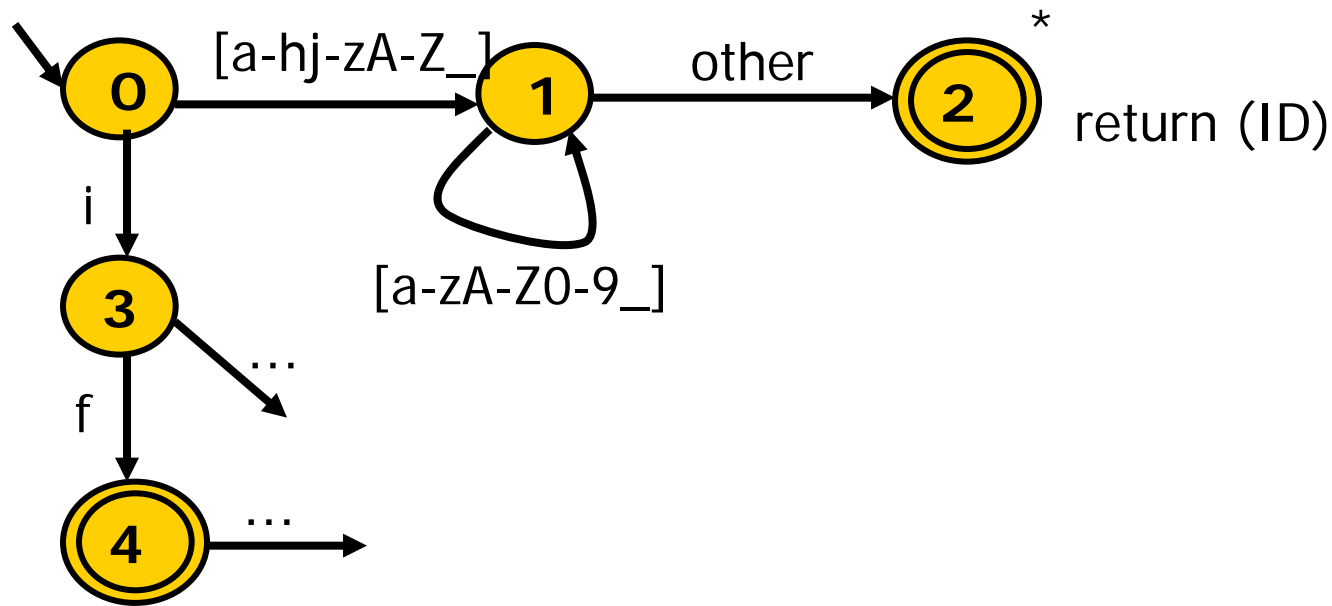
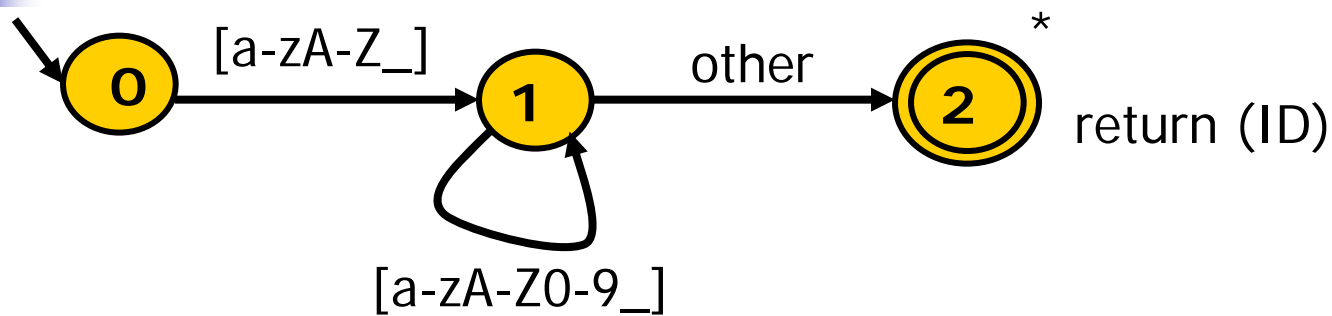
```
        while (c=='a' || c=='b' || ... || c=='_') c=getChar();
```



标识符和关键字

- 很多语言中的标识符和关键字有交集
 - 从词法分析的角度看，关键字是标识符的一部分
- 以C语言为例：
 - 标识符：以字母或下划线开头，后跟零个或多个字母、下划线、或数字
 - 关键字：if, while, else, ...

识别关键字（以if为例）





关键字表算法

- 对给定语言中所有的关键字，构造关键字构成的哈希表H
- 对所有的标识符和关键字，先统一按标识符的转移图进行识别
- 识别完成后，进一步查表H看是否是关键字
- 通过合理的构造哈希表H（完美哈希），可以 $O(1)$ 时间完成