## 极客大学 前端进阶训练营

#### 程劭非 (winter)

前手机淘宝前端负责人



# 重学前端

JavaScript



思考: 如何定义一门语言



## 如何定义中文

主语 谓语

我

吃

蛋糕

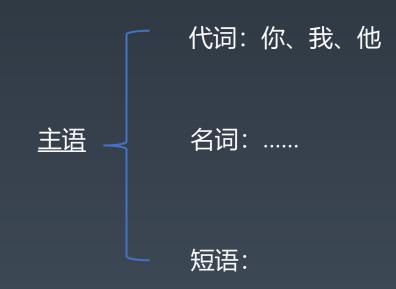
宾语

## 产生式

句子→主语 谓语 宾语

主语→代词|名词|短语

代词→你|我|他





#### 进一步定义产生式

- •符号 (Symbol)
  - 定义的语法结构名称
- 终结符 (Terminal Symbol) 与非终结符 (Non-Terminal Symbol)
  - 终结符: 不是由其它符号定义的符号, 也就是说, 它不会出现在产生式左边
  - 非终结符: 由其它符号经过 "与"、"或" 等逻辑组成的符号
- 语言定义
  - 语言可以由一个非终结符和它的产生式来定义
- 语法树
  - 把一段具体的语言的文本,根据产生式以树形结构来表示出来



#### 产生式的写法

• BNF: 巴科斯-诺尔范式

非终结符用 尖括号包裹

::=表示定义

竖线表示或

<中文>::= <句子> | <中文> <句子>

<句子>::= <主语> <谓语> <宾语> | <主语> <谓语>

<主语>::= <代词>|<名词>|<名词性短语>

<代词>::="你"|"我"|"他"

终结符用引号包裹



### 产生式的写法

• EBNF

不强制加尖 括号

大括号表示可以重复多次

方括号表示可以省略

中文::={句子}

句子::=主语谓语[宾语]

主语 ::= 代词 | 名词 | 名词性短语

代词::="你"|"我"|"他"

#### 产生式的其它写法

• JavaScript标准

```
Syntax

BlockStatement[Yield, Await, Return]:
    Block[?Yield, ?Await, ?Return]

Block[Yield, Await, Return]:
    { StatementList[?Yield, ?Await, ?Return] opt }

StatementList[Yield, Await, Return]:
    StatementListItem[?Yield, ?Await, ?Return]
    StatementListItem[?Yield, ?Await, ?Return]

StatementListItem[Yield, ?Await, ?Return]:
    StatementListItem[Yield, Await, Return]:
    Statement[?Yield, ?Await, ?Return]

Declaration[?Yield, ?Await]
```



#### 产生式的练习

- 外星语言:
  - 某外星人采用二进制交流
  - 它们的语言只有"叽咕"和"咕叽"两种词
  - 外星人每说完一句,会说一个"啪"

<外星语>::= { <外星句> }

<外星句>::= {"叽咕" | "咕叽" } 啪



#### 产生式的练习

• 数学语言四则运算, 只允许10以内整数的加减乘除

```
<四则运算表达式>::= <加法算式>
<加法算式> ::= (<加法算式> ("+"|"-") <乘法算式>) | <乘法算式>
<乘法算式> ::= (<乘法算式> ("*"|"/") <数字>) | <数字>
<数字> ::= {"0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"}
```



#### 产生式的练习

•四则运算,允许整数

```
<四则运算表达式>::= <加法算式>
<加法算式> ::= (<加法算式> ("+"|"-") <乘法算式>) | <乘法算式>
<乘法算式> ::= (<乘法算式> ("*"|"/") <数字>) | <数字>
<数字> ::= {"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"} {"0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"}
```



## 课堂练习

- 四则运算, 允许小数
- •四则运算,允许括号

# 词法和语法



#### 乔姆斯基谱系

- 3型 正则文法(Regular)
  - <A>::=<A>?
  - <A>::=?<A> ×
- 2型 上下文无关文法
  - <A>::=?
- 1型 上下文相关文法
  - ?<A>?::=?<B>?
- O型 无限制文法
  - ?::=?

#### 词法和语法

- 词法:正则文法(3型)
  - 空白
  - 换行
  - 注释
  - Token
- 语法: 上下文无关文法 (2型)
  - 语法树

##