程序=数据结构+逻辑结构

存储结构：线性，链接，哈希

制导翻译

正确判断：语法（合法），语义（合理）

词法lexical，语法，句法syntax，语义semantic

源程序（词法，句法分析）-正确程序（句法制导翻译）-中间代码-

链接：静态（运行前，包含所需库），动态（运行时，链接使用函数）

优化：源程序，数据结构，汇编级

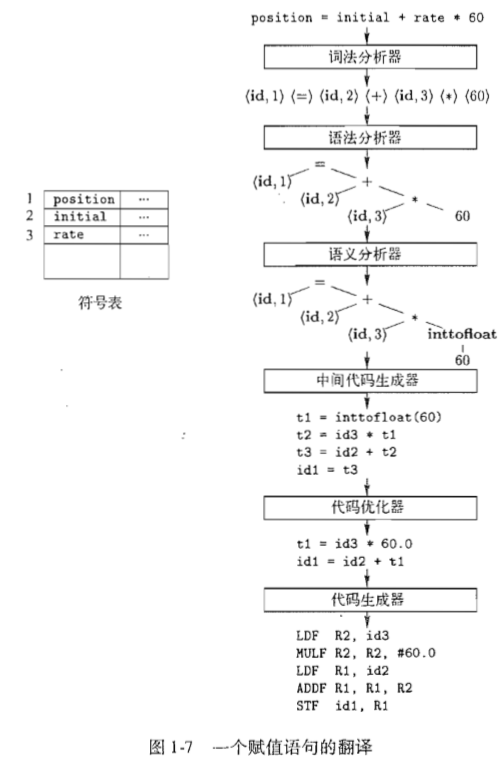
编译器：附带错误信息报告的将源程序变为目标程序的程序（正确性判定，翻译）

Syntax树：算数成分为叶子，运算符为父节点 Parse树：源成分为叶子节点

错误类型：语法，语义（空指针错误）

程序 外部特征：输入、输出 内部特征：数据结构、算法逻辑

词法分析器parse：上下文无关文法（输入）、词法分析程序



语言语法表示

语言结构：自下而上，自上而下

最小成分：普通、转义字符

基本成分：单词、字符序列（空串：零字符字符串）

完整成分：句子、单词序列

笛卡尔：

单词集合：、（以为核心的闭包、正闭包）

句子集合：

文法grammar：

描述语法对象结构的无二义的**产生式**规则

Formal production rules describing the construction of syntax elements.

关联：条件-动作、原因-结果

文法概念：核心、支撑、原子

语法树syntax：语法分析的结果

Parse树：语法分析的细化

开始符：树的根节点，文法核心概念（通过产生式集合P形成产生式）

非终结符：产生式左部，需继续细化（非终结符集合、尖括号/大写字母）

终结符：产生树的叶节点，为原子结构/概念（终结符集合、小写字母）

文法描述且仅描述唯一核心概念（开始符），此概念的支撑概念形成产生树，无法细分的原子概念为终结符

无用产生式：

1. 同义反复（直接/间接环）
2. 与无用非终结符（与开始符无关、使用关联树检测）相关联
3. 无法推出终结符序列

好的文法：删除无用产生式、删除产生式

判定实例合法：

最左、最右、混合

推导Derivation：从**开始符**开始，对非终结符找相关联产生式，使用右部代替左部

预测分析Predictive：对子串预处理，判断其包含终结符情况，提高推导效率

归约Reduction：从**句子实例**开始，对终结符找相关联产生式，使用左替代右

句型：开始符推导的终结/非终结符序列

随机推导/归约：轮盘赌方式（根据概论选择句型）

**最左**（右）归约是最右（左）推导的逆过程（横线推导、双竖线归约）

集合表示：枚举、解析（模板与相应条件）

递归：终止条件、分解方法、推出无限

文法类型：

0型/无限制：产生式左部存在非终结符（图灵机）

1型/长度增加：产生式右部长度左部长度，除、**上下文有关**文法/描述语义（线性自动机）

2型：产生式左部非终结符必推出右部终结/非终结符序列、**上下文无关**/描述语法结构（分层/带栈/下堆有限自动机PDA、函数嵌套形成多层自动机）

嵌入式文法：右部非终结符嵌在两终结符中间

3型：产生式右部为（1终结符+1非终结符）/ 1终结符 （向）右/左线性**正规文法**

给出分析树、寻找模板

**对称法**：找对称轴、对称特性、证实结果（练习）

**逐步求精**（分解部分独立）：自顶向下、自左向右等（将右部视为整体递归）

**等价法**：

**电路状态转换图**：