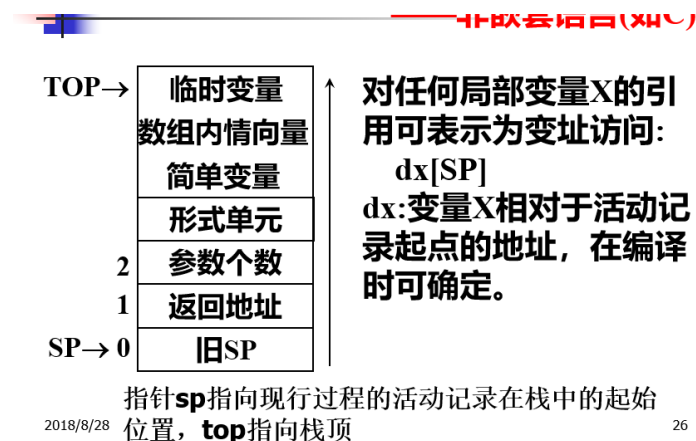


1. **前端后端**：目的：增加可移植性
  - 前端：与源语言有关、与目标机无关的部分。词法分析、语法分析、语义分析与中间代码生成、与机器无关的代码优化，出错处理，符号管理
  - 后端：与目标机有关的部分。与机器有关的代码优化、目标代码生成
2. **素短语**：是一个短语,至少包含一个终结符, 除自身外不再包括其他含终结符的短语  
**短语**:一棵子树的所有叶子自左至右排列起来 形成一个相对于子树根的短语。
3. **C 语言的活动记录格式**：过程的每个活动所需要的信息用一块连续的存储区来管理，这块存储区叫做**活动记录**



4. **代码优化**：公共子表达式的删除，复制传播，无用代码的删除，代码外提，强度削弱(归纳变量删除)
5. **S-属性**:只含综合属性的语法制导定义称为 S-属性定义,通常 在自底向上的分析方法计算属性值;  
**L-属性**:一个语法制导定义被称为 L-属性定义,当且仅当它的每个属性或者是综合属性
6. **五种常见的代码优化方法**？  
 无用代码删除，代码外提，强度削弱，归纳变量删除，复制传播 公共子表达式删除
7. **局部性原理**的概念？  
 程序中的大部分运行时间都花费在较少的一部分代码中，而且只是涉及到一小部分数据。  
**时间局部性**：如果某个程序访问的内存位置有可能在很短的时间内被再次访问  
**空间局部性**：如果被访问过的内存位置的邻近位置有可能在很短的时间内被再次访问
8. **活前缀**：前缀不含相应句型的句柄右部的任何符号，将其称为规范句型的活前缀

9. **遍**：通常称编译程序对源程序或中间结果的完整扫描为遍

10. **词法分析阶段的错误处理**：

- a) 非法字符检查
- b) 单词拼写错误
- c) 不封闭错误检查
- d) 重复声明检查
- e) 错误恢复与续编译

11. **LR(0)项目**：右部某个位置标有圆点的产生式称为相应文法的 LR(0)项目

12. **静态语义分析/静态检查**：

类型检查：操作数和操作符的类型是否相容（ $a + b$ ，例如要求+操作只能作用于整数，那么  $a$  和  $b$  必须都是整数）

- 1) 控制流检查：控制流转向目标地址是否合法
- 2) 唯一性检查：对象是否被重复定义（符号表查重）
- 3) 关联名检查：同一名字多次特定出现是否一致

13. **语法制导翻译**：将**静态检查**和**中间代码生成**结合到**语法分析**中进行的技术

名字等价：使用同样的类型名（完全相同）

结构等价：类型具有相同的内部结构（但可以不是同一种类型名）

类型综合：从子表达式的类型确定表达式的类型，所以必须声明变量类型

类型推断：根据语言结构的使用方式来确定其类型

14.

15. 将“过程、函数和方法”统称为“过程”

- 1) 过程定义是 把一个标识符和一组语句联系起来。该标识符是过程名，这组语句是过程体。
- 2) 当过程名出现在可执行语句中时，称相应的过程在该点被调用。过程调用就是执行被调用过程的过程体
- 3) 过程体的每次执行叫做该过程的一个活动
- 4) 过程的每个活动所需要的信息用一块连续的存储区来管理，这块存储区叫做活动记录
- 5) 过程调用和返回是通过在目标代码中生成调用序列和返回序列来实现的
- 6) 调用序列负责分配活动记录，并将相关信息填入活动记录中
- 7) 返回序列负责恢复机器状态，使调用过程能够继续执行