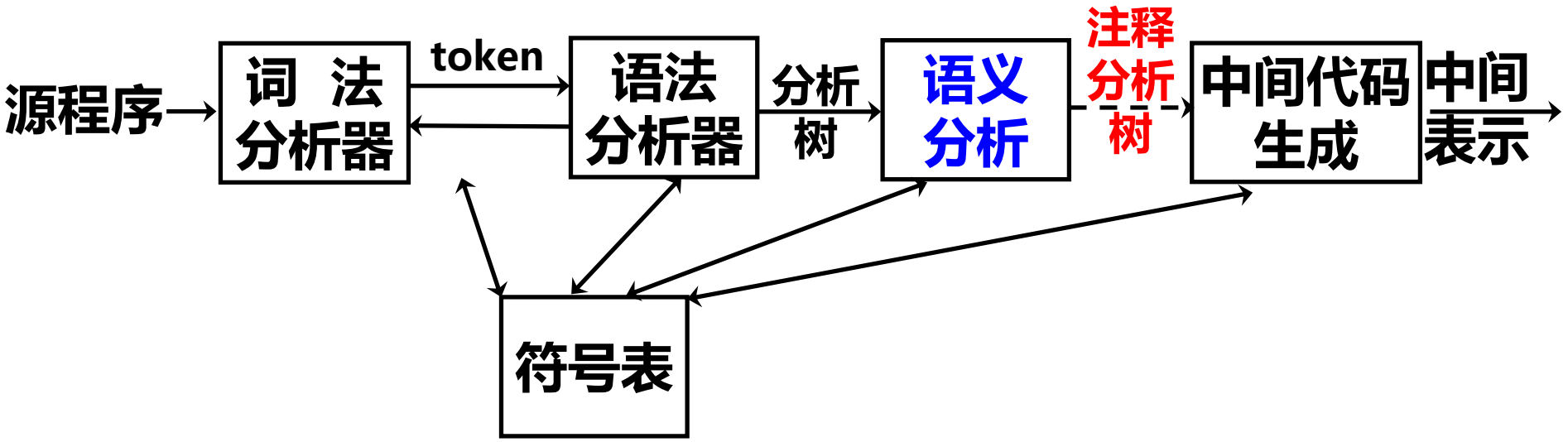
中国科学技术大学
University of Science and Technology of China
USTC, CHINA
ADSLAB
先进数据系统实验室

《编译原理与技术》
语法制导翻译 I

计算机科学与技术学院
李 诚
15/10/2018

主要内容



- ❑语义分析简介
  - ❖语法制导定义、翻译方案
- ❑语法制导定义
  - ❖综合属性、继承属性
  - ❖属性依赖图与属性的计算次序

语义分析简介

- 编译程序的目标：将源程序翻译成为**语义等价**的目标程序。
  - 源程序与目标程序具有不同的语法结构，表达的结果却是相同的。
- 语义分析的主流技术：
  - 语法制导翻译技术

语义分析的功能

- 审查每个语法结构的静态语义
  - 例：类型、运算、维数、越界
- 在验证完静态语义后，才执行真正的翻译
  - 例：变量的存储分配
  - 例：表达式的求值
  - 例：语句的翻译（中间代码的生成）

语义描述的方法

❑语法制导定义(Syntax-directed definition, SDD)

- ❖为每一个产生式写一个语义子程序，当该产生式获得匹配时，调用相应的语义子程序实现语义检查与翻译。

$E \rightarrow E_1 + T$ $\quad\quad$ $E.code = E_1.code \| T.code \| \text{‘+’}$

- ❖可读性好，更适于描述规范

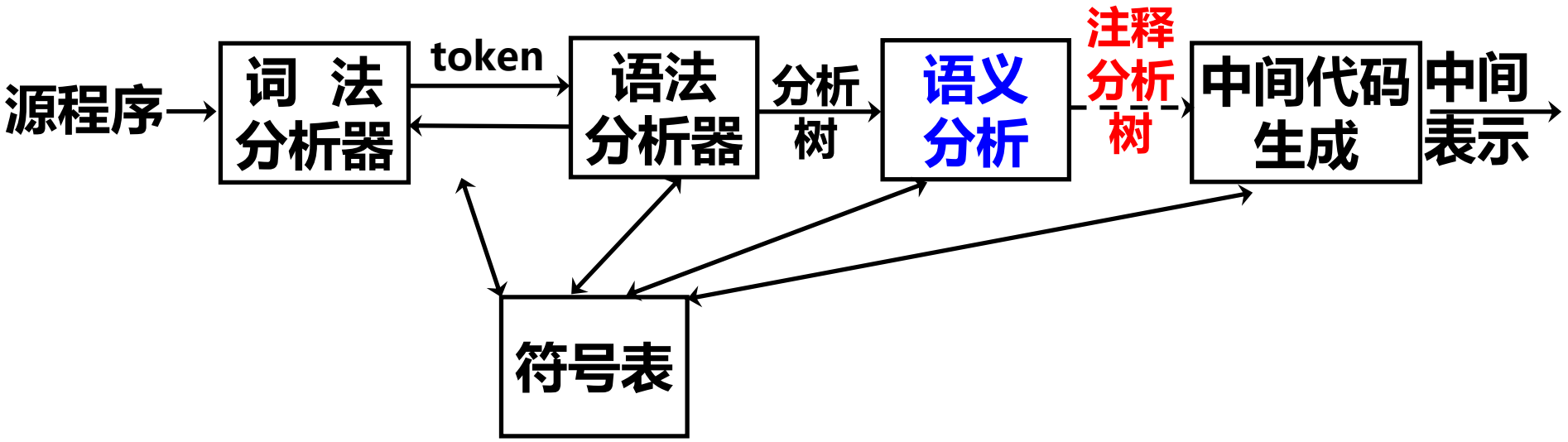
❑翻译方案(Translation scheme, SDT)

- ❖在产生式的右部的适当位置，插入相应的语义动作，按照分析的进程，执行遇到的语义动作。

$E \rightarrow E_1 + T$ $\quad\quad$ $\{ print \text{‘+’} \}$

- ❖陈述了实现细节(如语义规则的计算时机)

主要内容



- ❑语义分析简介
  - ❖语法制导定义、翻译方案
- ❑语法制导定义
  - ❖综合属性、继承属性
  - ❖属性依赖图与属性的计算次序

语法制导的定义

❑语法制导定义的形式

- ❖基础的上下文无关文法
- ❖每个文法符号有一组属性
- ❖每个文法产生式$A \rightarrow \alpha$有一组形式为

  $b=f(c_1, c_2, \ldots, c_k)$的语义规则，其中$f$是函数

  b和$c_1, c_2, \ldots, c_k$是该产生式文法符号的属性
- ❖综合属性(synthesized attribute)：如果b是A的属性，$c_1, c_2, \ldots, c_k$是产生式右部文法符号的属性或A的其它属性
- ❖继承属性(inherited attribute)：如果b是右部某文法符号X的属性

终结符只能有综合属性，属性值无需计算，由词法分析给定

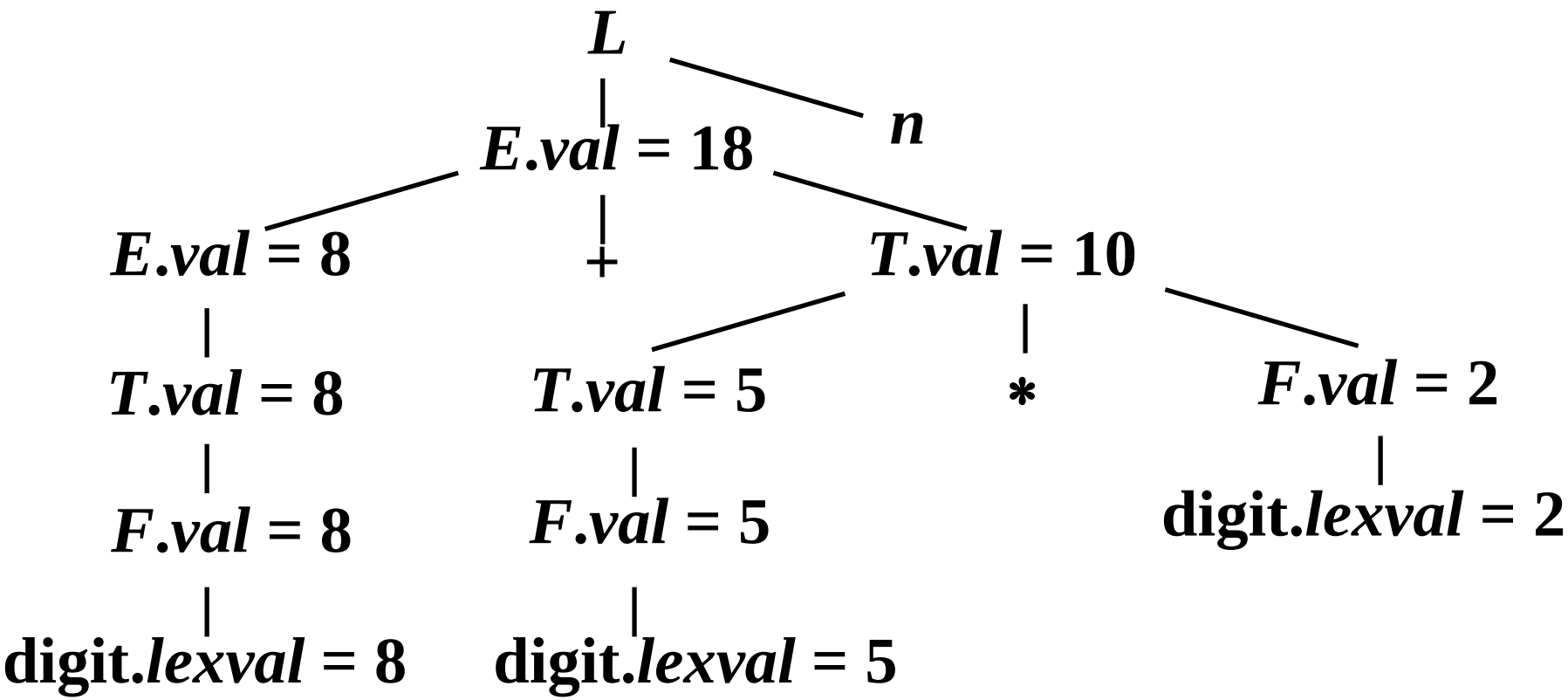
S属性定义：仅使用综合属性的语法制导定义

产 生 式	语 义 规 则
$L \rightarrow E$ n	*print* (*E.val*)
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow (E)$	$F.val = E.val$
$F \rightarrow$ digit	$F.val =$ digit.*lexval*

注释分析树

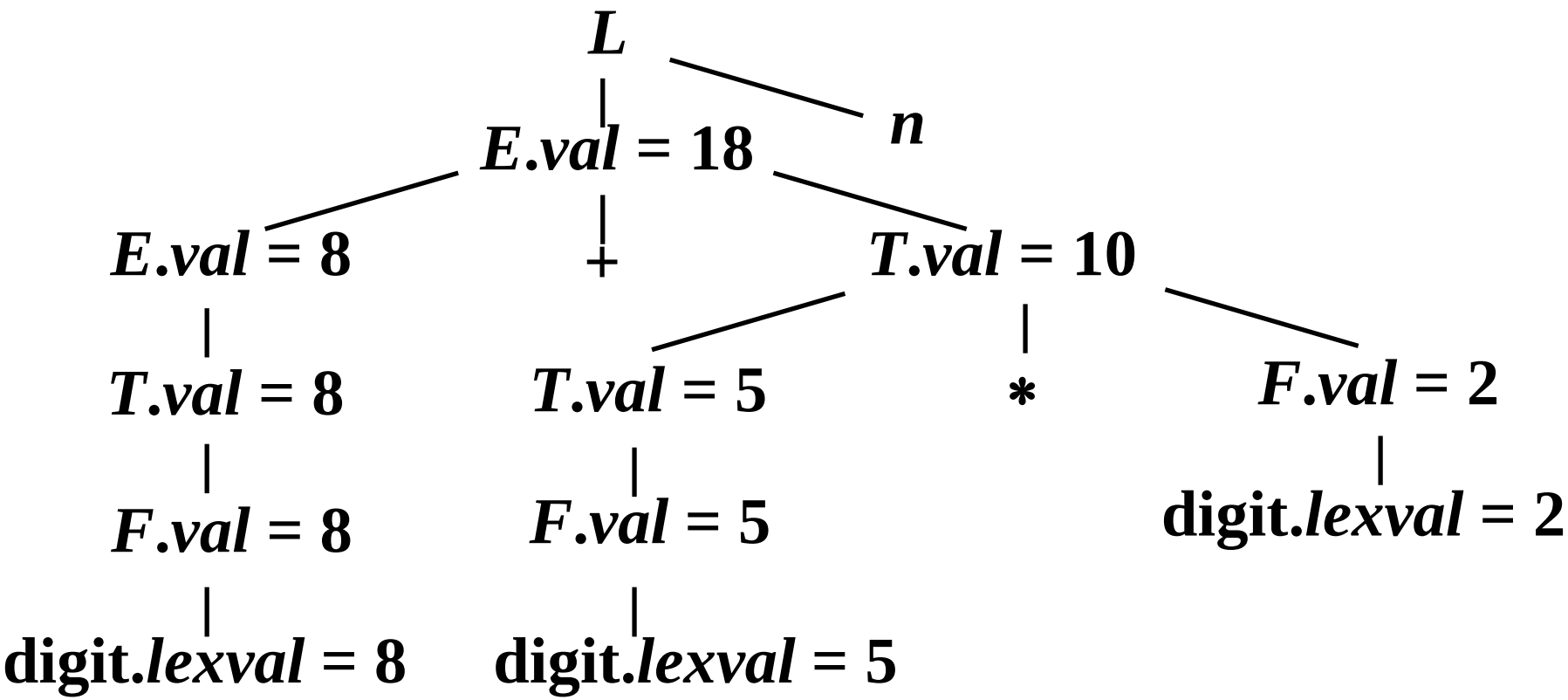
- 定义：结点的属性值都标注出来的分析树

8+5*2 n的注释分析树



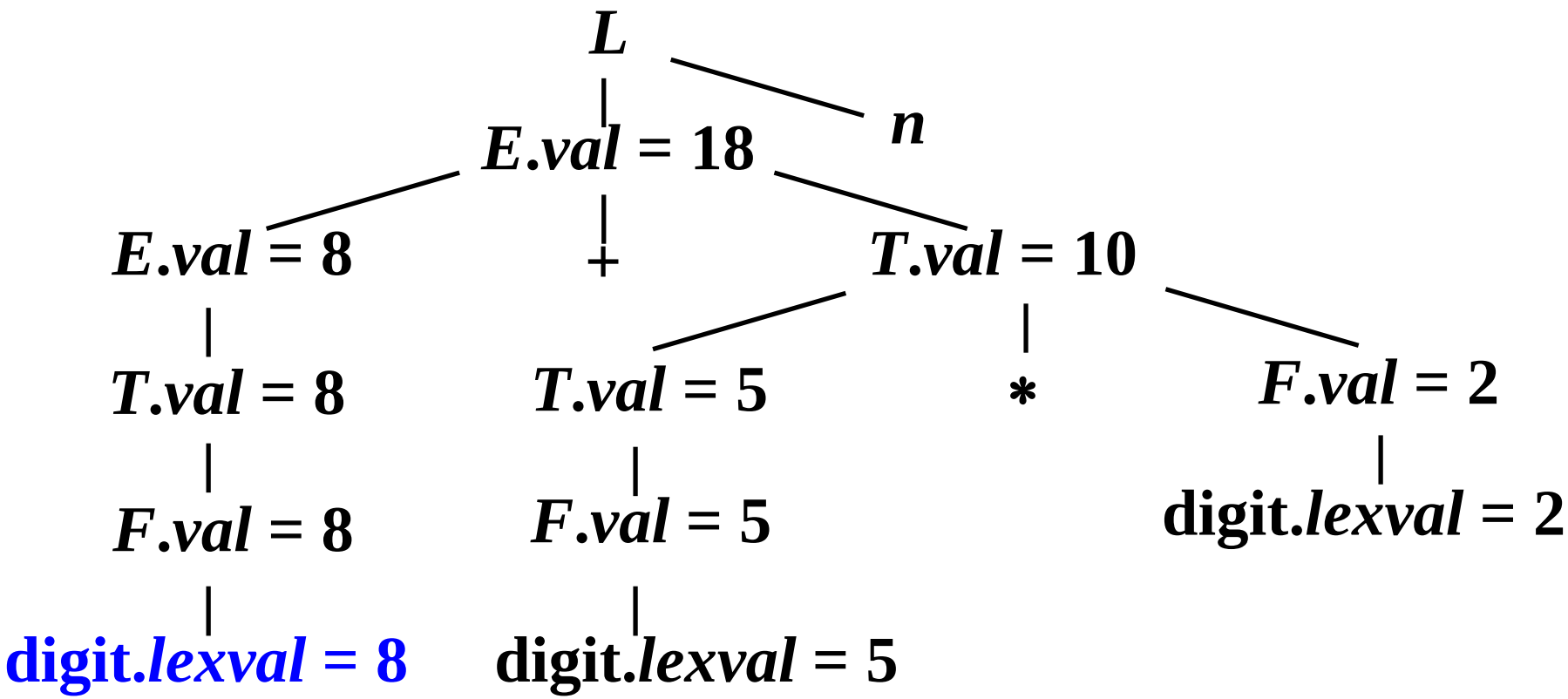
注释分析树+综合属性计算

- 各结点综合属性的计算可以自底向上地完成



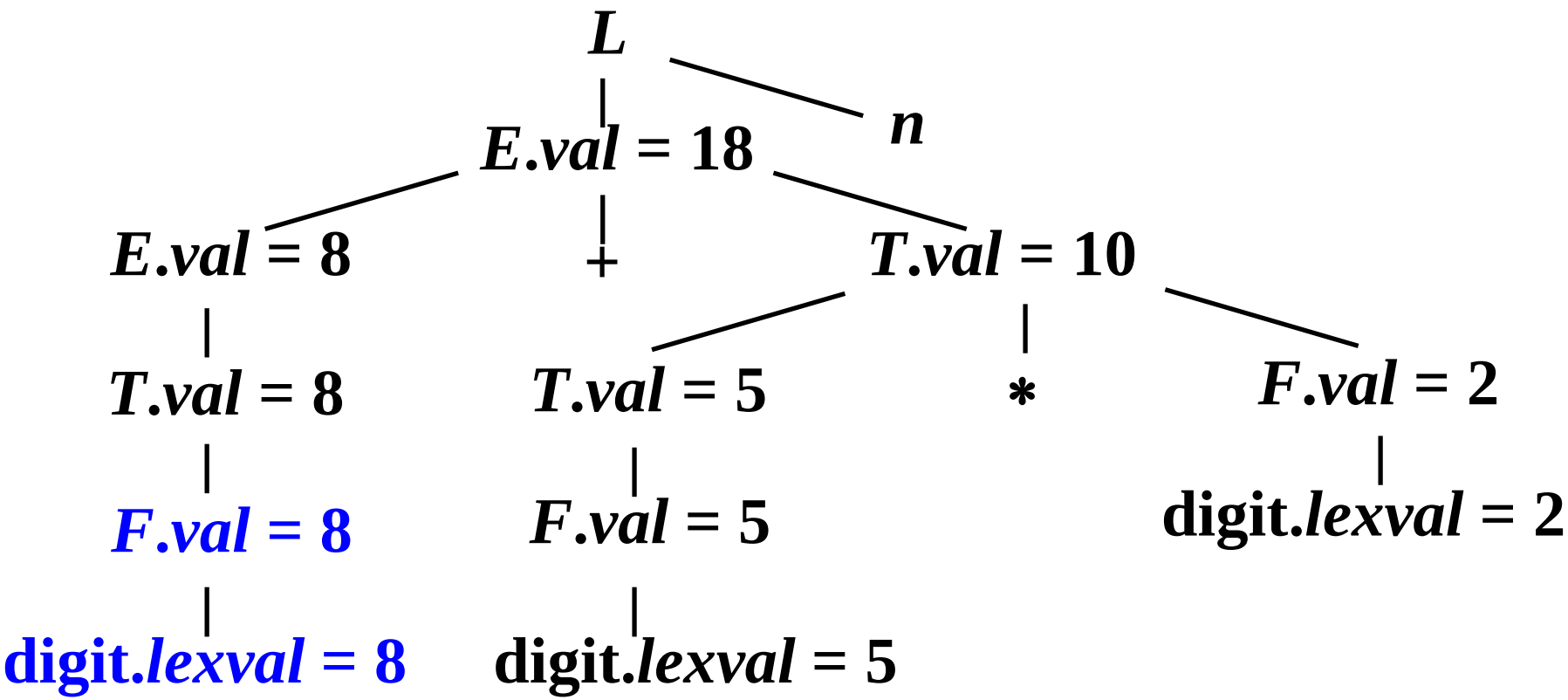
注释分析树+综合属性计算

- 各结点综合属性的计算可以自底向上地完成



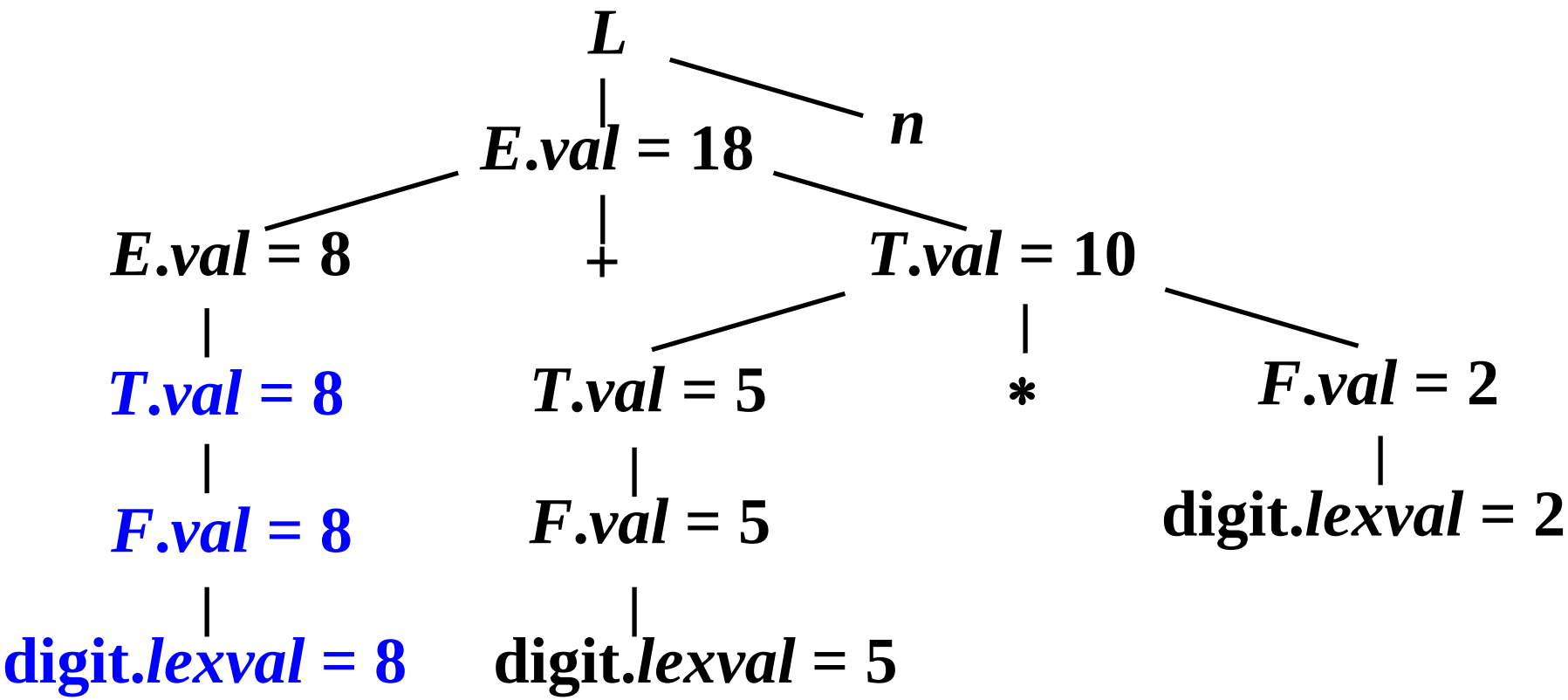
注释分析树+综合属性计算

- 各结点综合属性的计算可以自底向上地完成



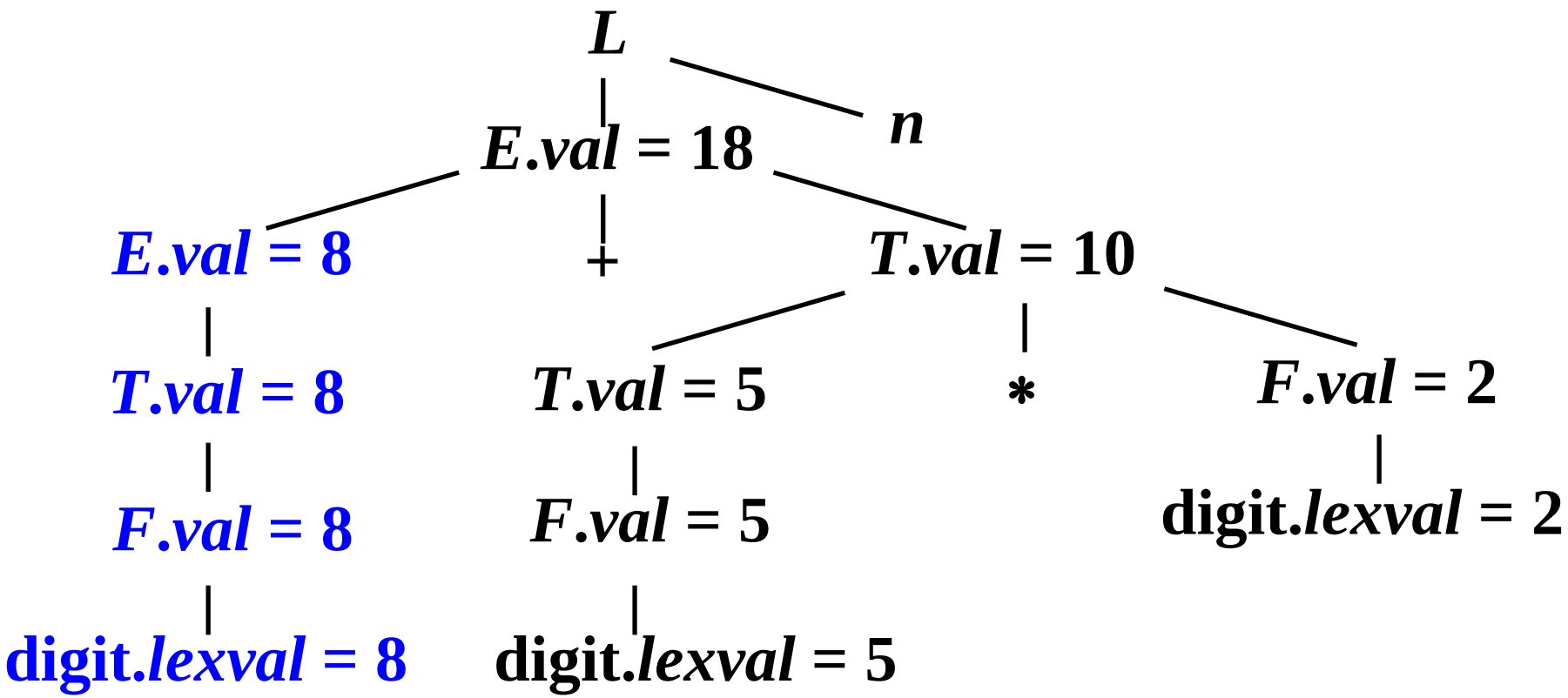
注释分析树+综合属性计算

- 各结点综合属性的计算可以自底向上地完成



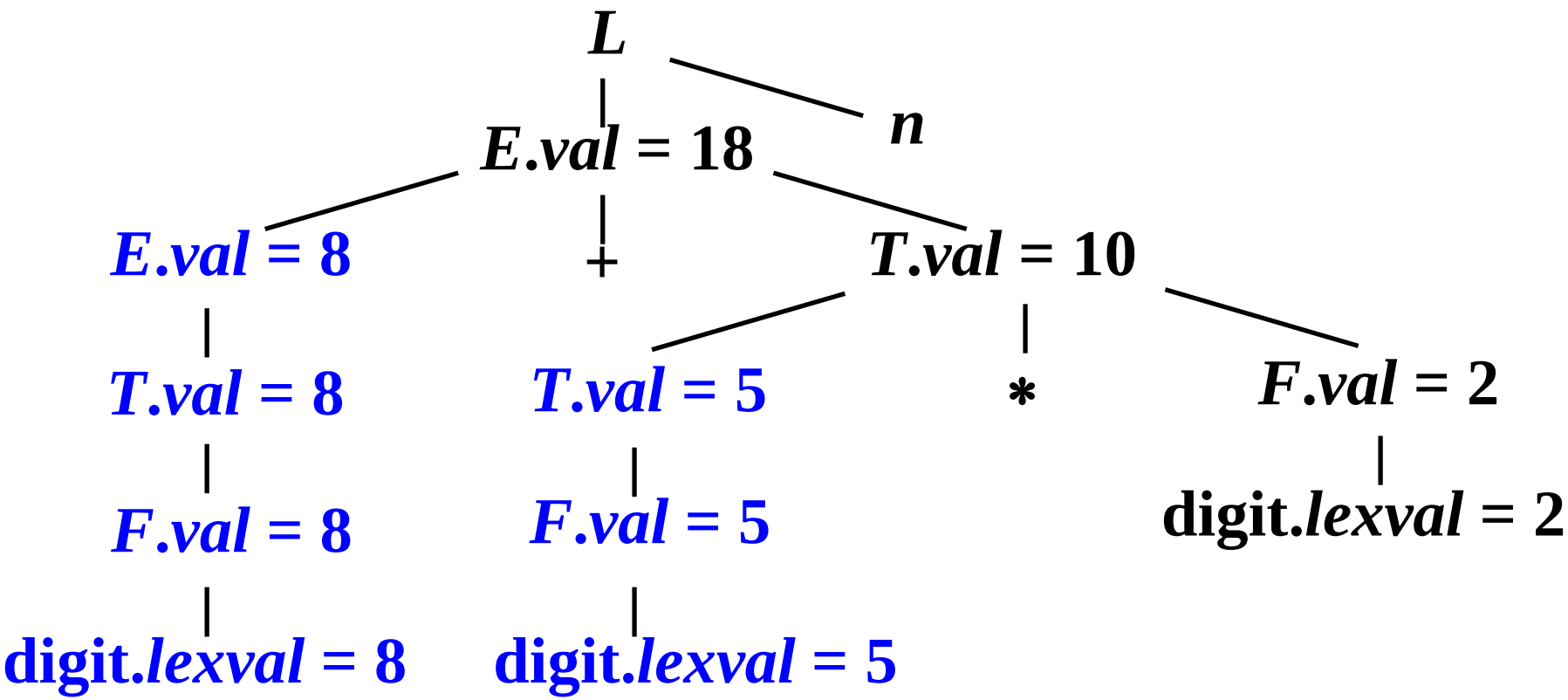
注释分析树+综合属性计算

- 各结点综合属性的计算可以自底向上地完成

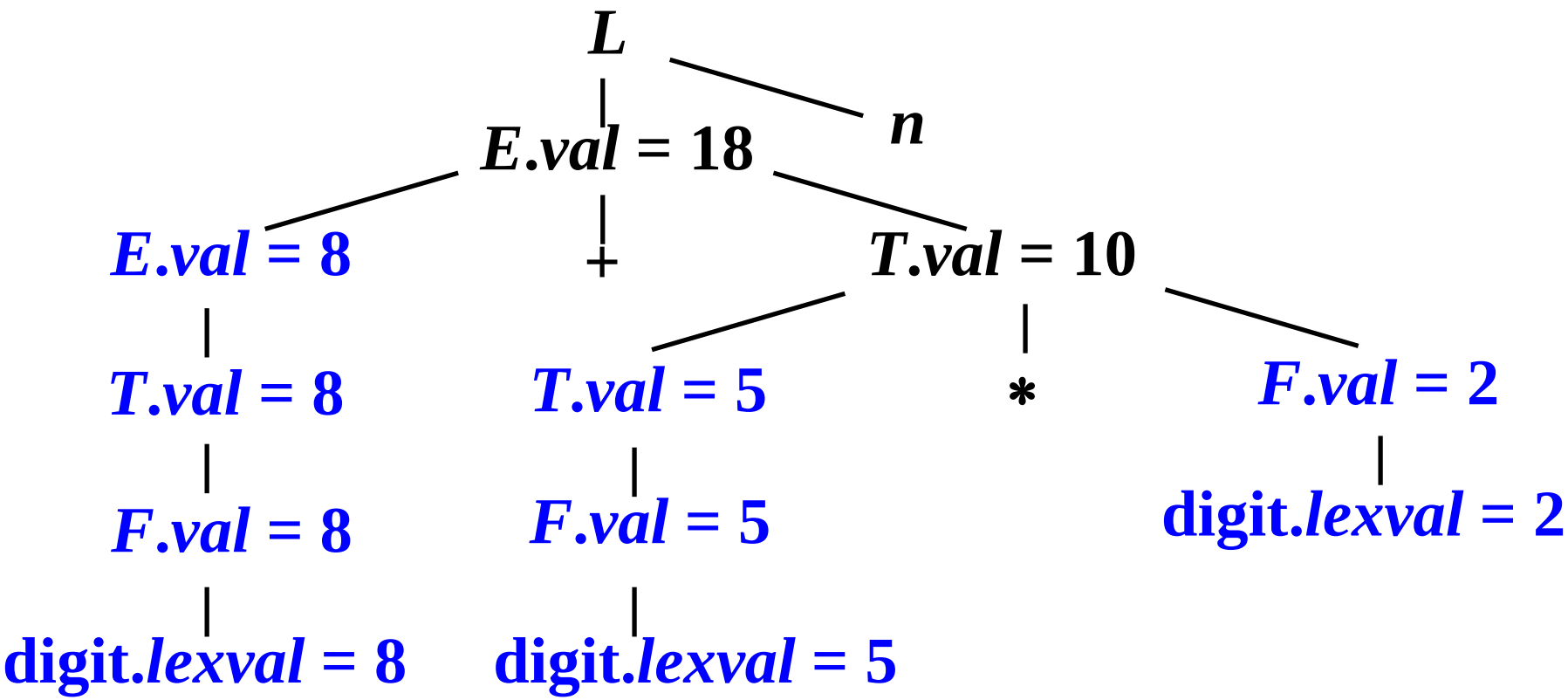


注释分析树+综合属性计算

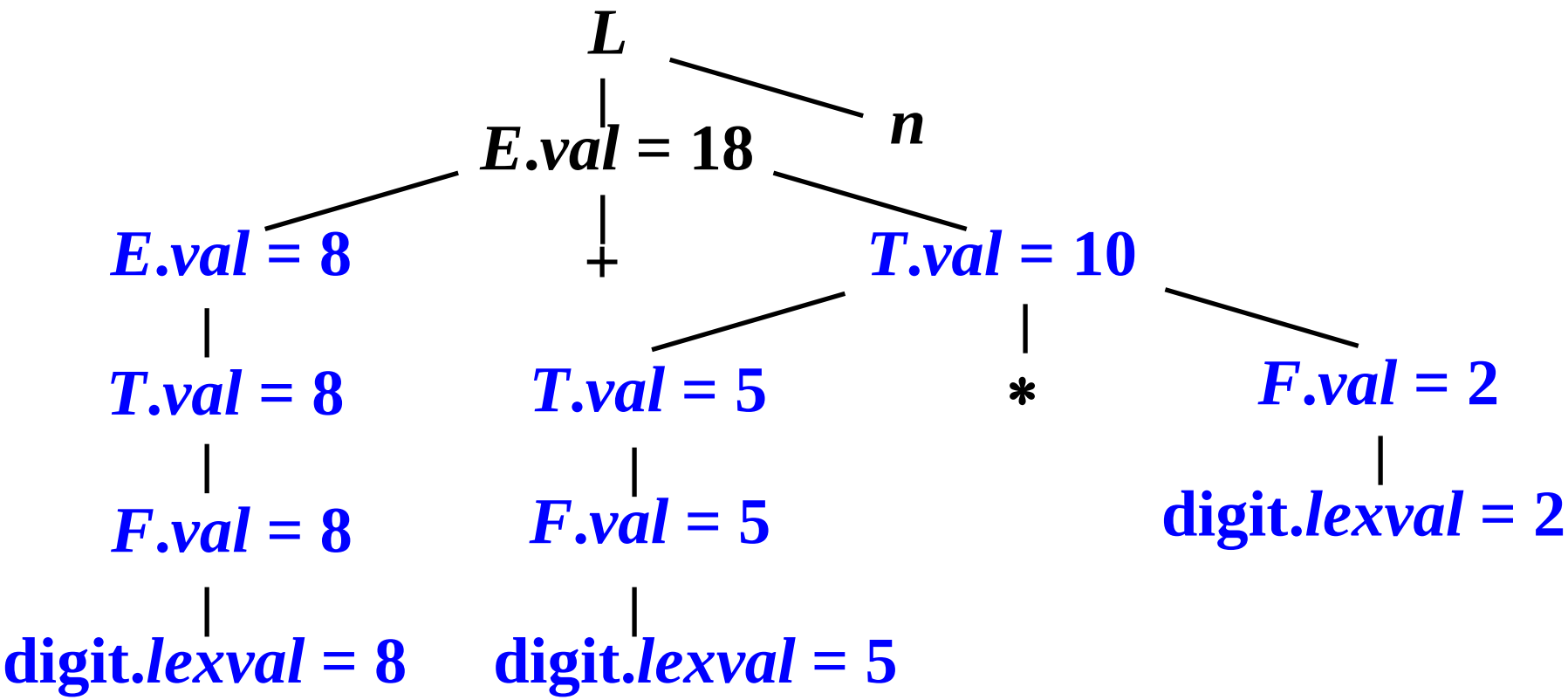
- 各结点综合属性的计算可以自底向上地完成



- 各结点综合属性的计算可以自底向上地完成



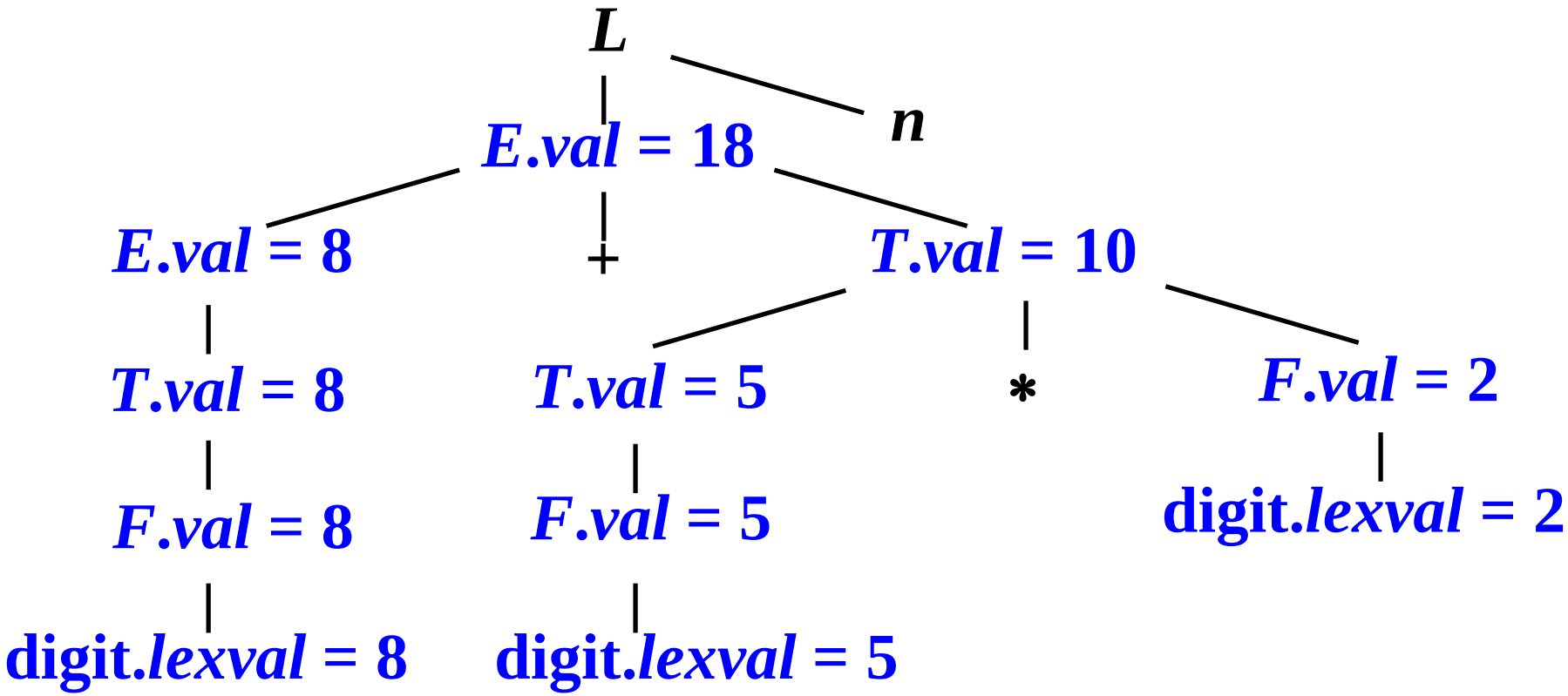
- 各结点综合属性的计算可以自底向上地完成



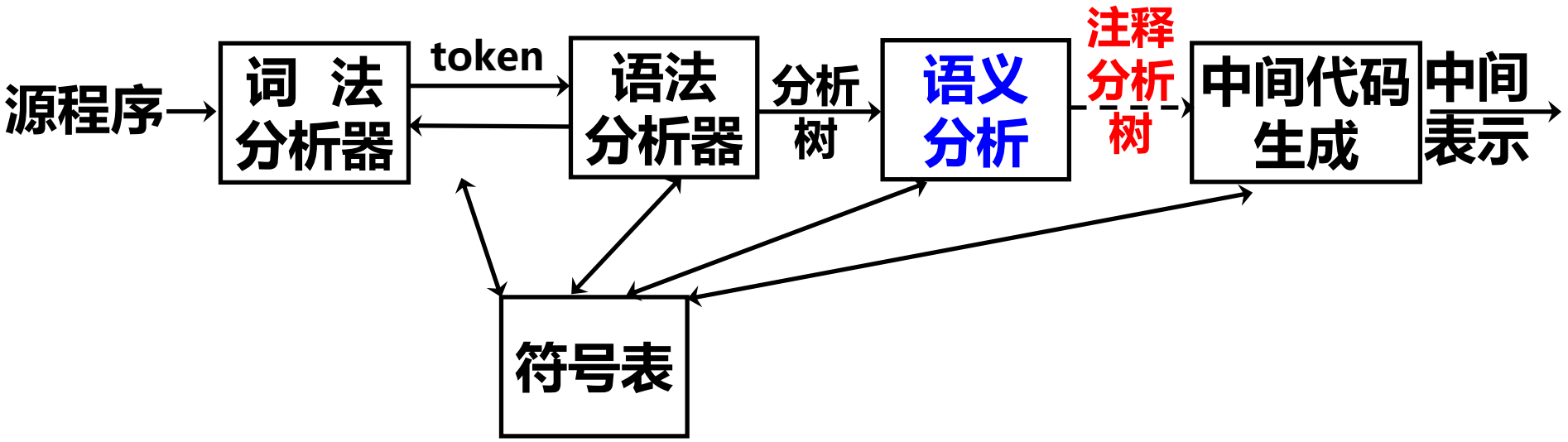
注释分析树+综合属性计算

- 各结点综合属性的计算可以自底向上地完成

S属性的定义可以和LR分析器一起自然地实现。



主要内容



- ❑语义分析简介
  - ❖语法制导定义、翻译方案
- ❑语法制导定义
  - ❖综合属性、继承属性
  - ❖属性依赖图与属性的计算次序

继承属性：举例

int id, id, id

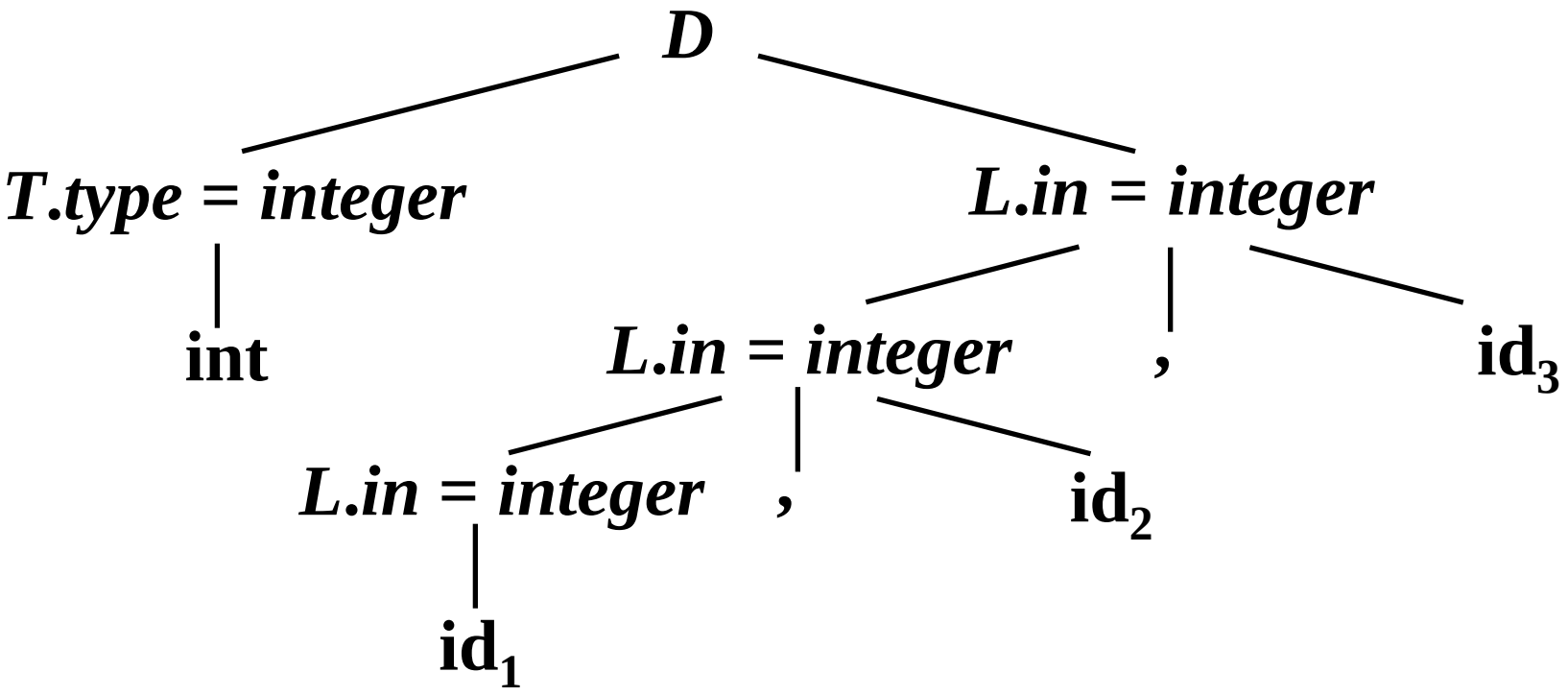
产　生　式	语 义 规 则
$D \rightarrow TL$	$L.in = T.type$
$T \rightarrow$ int	$T.type = integer$
$T \rightarrow$ real	$T.type = real$
$L \rightarrow L_1$, id	$L_1.in = L.in$; $addType$(id.$entry$, $L.in$)
$L \rightarrow$ id	$addType$(id.$entry$, $L.in$)

- type – T的综合属性
- in – L的继承属性，把声明的类型传递给标识符列表
- addType– 把类型信息加到符号表中的标识符条目里

继承属性：举例

- 例 int id_1, id_2, id_3的标注了部分属性的分析树

  不可能像像综合属性那样自底向上标注属性



《编译原理与技术》
语法制导翻译 I

People think that computer science is the art of geniuses but the actual reality is the opposite, just many people doing things that build on each other, like a wall of mini stones.

—— Donald Knuth