

Árboles de análisis

- Representa derivaciones
- Nodo interno = LHS
- Hijos = RHS
- Hojas = yield
- Parsear = verificar pertenencia
- Derivación \leftrightarrow Reducción
- Resultado: aceptar o error

7

Ambigüedad

- Gramática ambigua: varios árboles mismos terminales
- Ejemplo correcto: $1+2+3$ (única forma)
- Ejemplo incorrecto: $S \rightarrow AB, A \rightarrow x, B \rightarrow x$
- Dos derivaciones, un solo árbol
- Ambigüedad debe evitarse
- Regla no depende de aritmética

8

Asociatividad y Precedencia

- Gramática original: izquierda
- Ejemplo: $8-8-8 = -8$
- Asociatividad derecha: cambiar producciones
 - $\text{lista} \rightarrow \text{digito} + \text{lista}$
 - $\text{lista} \rightarrow \text{digito} - \text{lista}$
- Precedencia: expr , term , factor
- Operadores $+$, $-$, $\backslash*$, $/$ con orden normal
- Paréntesis opcionales
- Igual precedencia: izquierda

9

Ejemplo

```

expr  → expr + term | expr - term | term
term  → term factor | term / factor | factor
factor → digito | ( expr )
digito → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

10

Sentencias

- STMT: asignación, if, while, begin-end
- OPT-STMTS: opcional o vacío
- STMT-LIST: secuencia con ;
- Uso de mayúsculas solo ilustrativo
- Epsilon = cadena vacía
- Algunos lenguajes no aceptan bloques vacíos
- Ambigüedad: dangling else

11

Ejemplo

```

STMT → id := EXPR
      | if EXPR then STMT
      | if EXPR then STMT else STMT
      | while EXPR do STMT
      | begin OPT-STMTS end
OPT-STMTS → STMT-LIST | ε
STMT-LIST → STMT-LIST ; STMT | STMT
  
```

12