

Análisis sintáctico

El análisis sintáctico toma una cadena de *tokens* y una gramática para construir un árbol de análisis o verificar si este existe. Existen dos enfoques: **descendente**, que inicia desde el símbolo de inicio (raíz), y **ascendente**, que parte de las hojas. En este tema se estudiará solo el método descendente, ya que es más sencillo de implementar manualmente, aunque menos general. En cambio, las herramientas automáticas suelen usar técnicas ascendentes. Se asume que el analizador léxico ya transformó la entrada en una secuencia de *tokens*.

Análisis descendente

El análisis descendente parte del símbolo inicial y aplica reglas para construir el árbol de análisis de arriba hacia abajo. Funciona mediante dos pasos básicos: seleccionar una producción cuyo lado izquierdo coincida con el no terminal actual y cuyo lado derecho corresponda a la entrada en ese punto, y luego expandir el siguiente nodo que requiere un subárbol.

En la práctica, se implementa como un procedimiento recursivo para cada no terminal, que elige una producción y llama a los procedimientos correspondientes a los no terminales del lado derecho. De ahí el nombre de *analizadores sintácticos recursivos descendentes*.

El principal problema ocurre cuando un no terminal tiene varias producciones posibles. La solución es restringirse al **análisis predictivo**, en el cual se examina el siguiente token de la entrada y se selecciona únicamente la producción que puede derivar a partir de él. Esto evita ambigüedades y resuelve tanto la elección entre múltiples producciones como la determinación de qué nodo expandir en cada paso.

Análisis predictivo

En el análisis predictivo, cada producción con el mismo símbolo no terminal en el lado izquierdo se asocia a un conjunto **FIRST**, que contiene los terminales que pueden aparecer como primer símbolo en una cadena derivada del lado derecho de la producción.

Formalmente, para una cadena α de terminales y no terminales, **FIRST(α)** es el conjunto de terminales que pueden iniciar una derivación de α ; si α es ϵ o puede derivar ϵ , entonces ϵ también pertenece a **FIRST(α)**. Así, para una producción P con lado derecho r , se define **FIRST(P) = FIRST(r)**.

El análisis predictivo se basa en la condición de que, para dos producciones con el mismo no terminal en el lado izquierdo, sus conjuntos FIRST sean disjuntos. De este modo, conociendo el no terminal y el token siguiente en la entrada, se garantiza la elección de una única producción a aplicar.

Cuándo usar ϵ -producciones

Cuando una producción tiene ϵ en su lado derecho, o puede derivar ϵ , este símbolo se incluye en el conjunto **FIRST**. El problema es que ϵ siempre podría coincidir con la posición actual de la entrada.

La regla es: si el *lookahead* no pertenece al FIRST de ninguna producción con un mismo lado izquierdo, se aplica la producción única cuyo FIRST contiene a ϵ .

Diseño de un analizador predictivo

El analizador predictivo es un analizador descendente recursivo que construye el árbol de análisis de arriba hacia abajo mediante un procedimiento para cada no terminal. Para su correcto funcionamiento, es necesario que las producciones con el mismo lado izquierdo tengan conjuntos FIRST disjuntos.

El diseño se basa en las siguientes reglas:

1. Cada no terminal tiene un procedimiento que selecciona la producción cuyo conjunto FIRST contiene el *lookahead*; si no coincide ninguna, se usa la producción con ϵ , y si esta tampoco existe, se genera un error.
 2. Una vez elegida la producción, el procedimiento replica su lado derecho, llamando a otros procedimientos para no terminales y a *match* para los terminales.
 3. El procedimiento **match(terminal)** avanza el *lookahead* al siguiente token después de verificar que coincide con el esperado.
 4. Un programa principal inicializa el *lookahead* con el primer token de la entrada e invoca el procedimiento del símbolo inicial.
-

Recursividad izquierda

La recursividad izquierda ocurre cuando una producción tiene como primer símbolo del lado derecho el mismo no terminal del lado izquierdo. En analizadores descendentes recursivos, esto provoca bucles infinitos y conjuntos FIRST no disjuntos, lo que limita el análisis predictivo.

Para evitarlo, la recursividad izquierda se reemplaza por **recursividad por la derecha**, mediante un nuevo no terminal R, transformando:

$$A \rightarrow A, \alpha | \beta$$

en

$$A \rightarrow \beta R,$$

$$R \rightarrow \alpha, R \mid \varepsilon$$

Esta técnica mantiene las mismas cadenas derivables, pero permite que el analizador predictivo funcione correctamente. La recursividad por la derecha hace que los árboles de análisis crezcan hacia la derecha, evitando bucles infinitos, aunque cambia la forma de asociatividad de las operaciones en algunos casos.