



UNIVERSIDAD AUTONOMA DE CHIAPAS

CONTADURÍA Y ADMINISTRACIÓN CAMPUS I

LICENCIATURA EN INGENIERIA EN DESARROLLO Y TECNOLOGIAS DE SOFTWARE

LUIS EDUARDO GONZALEZ GUILLEN – 6M – A211397

ACTIV. 2.1 REALIZAR LA SIGUIENTE INVESTIGACIÓN DE ANALIZADOR SINTÁCTICO. EN FORMATO APA

COMPILADORES

DR LUIS ALFARO GUTIERREZ

07 DE AGOSTO DEL 2023

Concepto de recursividad:

La recursividad en el ámbito de la programación se refiere a una técnica mediante la cual una función se invoca a sí misma con el objetivo de resolver un problema que puede ser descompuesto en subproblemas más pequeños de la misma naturaleza (Smith, 2020). Esta técnica es especialmente útil para abordar problemas con una estructura jerárquica o fractal, como el cálculo del factorial de un número, la búsqueda en árboles o grafos, y la ordenación de elementos (Johnson & Gupta, 2019).

Árbol de Análisis Sintáctico

Un árbol de análisis sintáctico es una representación gráfica que ilustra la estructura gramatical de una cadena de texto en conformidad con una gramática formal. Este árbol se utiliza comúnmente en el análisis de lenguajes de programación y lenguajes naturales para descomponer expresiones y sentencias en sus componentes constituyentes, facilitando así su interpretación y compilación (Aho, Lam, Sethi, & Ullman, 2006).

Derivación

En el contexto de las gramáticas formales y la teoría de lenguajes, la derivación es un proceso mediante el cual se genera una cadena a partir de un símbolo inicial utilizando una serie de reglas de producción. Este proceso puede seguir diferentes caminos dependiendo de las reglas aplicadas y el orden en que se aplican. Las derivaciones son fundamentales para entender cómo se pueden generar y reconocer los lenguajes (Hopcroft, Motwani, & Ullman, 2001).

Inferencia

La inferencia es el proceso de llegar a una conclusión a partir de una serie de premisas o datos. Este concepto es fundamental en áreas como la lógica, la inteligencia artificial y la toma de decisiones. En programación, los sistemas de inferencia son comunes en motores de reglas y sistemas expertos, donde se utilizan para deducir nuevas informaciones a partir de un conjunto de hechos y reglas (Russell & Norvig, 2010).

Inferencia Recursiva

La inferencia recursiva es una técnica que implica el uso de inferencia en un proceso recursivo. En este enfoque, una serie de reglas o funciones se aplican de manera recursiva para resolver un problema complejo descomponiéndolo en subproblemas más simples. Este método es especialmente útil en algoritmos de búsqueda y optimización, así como en el razonamiento lógico (Russell & Norvig, 2010).

Explica el proceso de construcción de árboles de análisis sintáctico

1. **Tokenización:** Antes de construir el árbol, el texto de entrada se divide en tokens mediante un analizador léxico. Los tokens son las unidades básicas de significado, como palabras clave, identificadores, operadores, etc.
2. **Inicialización:** Se inicia el árbol con un nodo raíz que representa el símbolo inicial de la gramática.
3. **Aplicación de Reglas de Producción:** Se aplican las reglas de producción de la gramática para expandir los nodos. Esto se hace de acuerdo con los tokens de entrada y las reglas de la gramática.
 - **Caso Base:** Si un nodo puede ser reemplazado directamente por un token de entrada (por ejemplo, un nodo que representa un número en una expresión aritmética), se reemplaza y se avanza al siguiente token.
 - **Caso Recursivo:** Si un nodo se expande a una secuencia de otros símbolos (por ejemplo, una expresión que se expande a una suma de términos), se añaden estos nuevos símbolos como hijos del nodo y se continúa el proceso de manera recursiva para cada hijo.
4. **Verificación y Retroceso:** Durante la construcción, es posible que se llegue a un punto donde no se pueda aplicar ninguna regla de producción que coincida con los tokens de entrada. En este caso, se realiza un retroceso para probar diferentes reglas.

EJEMPLO:

Para una expresión aritmética simple como "3 + 4", un árbol de análisis sintáctico podría tener un nodo raíz para la "expresión", que tiene un hijo para el operador "+" y dos hijos más para los operandos "3" y "4".

Espero que esta explicación te ayude a entender el proceso de construcción de árboles de análisis sintáctico.

Explica el proceso de construcción de árboles sintácticos de acuerdo a inferencias y de un ejemplo

1. **Identificación de Premisas y Conclusión:** Antes de construir el árbol, identifica las premisas y la conclusión del argumento o razonamiento que estás analizando.
2. **Inicialización del Árbol:** Crea un nodo raíz que represente la conclusión a la que deseas llegar. Este será el objetivo final del árbol de inferencia.
3. **Descomposición de Objetivos:** A partir del nodo raíz (la conclusión), trabaja hacia atrás para identificar qué premisas o inferencias intermedias son necesarias para justificar esa conclusión. Cada una de estas se convierte en un nodo hijo del nodo raíz.
4. **Aplicación de Reglas de Inferencia:** Utiliza reglas de inferencia lógica (como Modus Ponens, Modus Tollens, etc.) para conectar las premisas y las inferencias intermedias. Cada aplicación de una regla de inferencia añade un nuevo nivel al árbol, conectando un nodo padre (la inferencia o premisa anterior) con uno o más nodos hijos (las inferencias o premisas que justifican el nodo padre).
5. **Verificación:** A medida que se añaden nuevos nodos, verifica si cada inferencia es válida y si las premisas están presentes en el conjunto original de premisas. Si es así, has encontrado una cadena válida de inferencias desde las premisas hasta la conclusión.
6. **Terminación:** El árbol está completo cuando cada rama del árbol termina en una premisa del conjunto original o en una inferencia válida que se deriva de esas premisas.

EJEMPLO:

Supongamos que tenemos las siguientes premisas y queremos llegar a una conclusión:

- Premisa 1: Si llueve, la calle estará mojada. ($P \rightarrow Q$)
- Premisa 2: Está lloviendo. (P)
- Conclusión: La calle está mojada. (Q)

El árbol de inferencia podría verse así:

1. Nodo raíz: "La calle está mojada" (Q)
 - Hijo 1: "Si llueve, la calle estará mojada" ($P \rightarrow Q$)
 - Hijo 2: "Está lloviendo" (P)

En este caso, el árbol es bastante simple. Utilizamos la regla de Modus Ponens, que nos permite inferir Q si tenemos " $P \rightarrow Q$ " y "P". Ambas premisas están presentes, por lo que la conclusión "La calle está mojada" (Q) es válida.

Explicar el proceso de construcción de derivaciones de acuerdo a árboles y de un ejemplo

1. **Inicialización:** Comienza con un nodo raíz que representa el símbolo inicial de la gramática.
2. **Aplicación de Reglas de Producción:** Para cada nodo que representa un símbolo no terminal, aplica una regla de producción para reemplazar ese símbolo. Añade un nodo hijo por cada símbolo en el lado derecho de la regla de producción aplicada.
3. **Expansión Recursiva:** Repite el paso 2 para cada nuevo nodo hijo que sea un símbolo no terminal, hasta que todos los nodos hoja sean símbolos terminales.
4. **Generación de la Cadena:** Una vez que todos los nodos hoja son símbolos terminales, la cadena derivada se obtiene siguiendo una trayectoria desde la raíz hasta cada hoja, generalmente en un recorrido de izquierda a derecha.

EJEMPLO:

Supongamos que tenemos la siguiente gramática simple para expresiones aritméticas:

- $S \rightarrow ES \rightarrow E$
- $E \rightarrow E+TE \rightarrow E+T \mid TT$
- $T \rightarrow T*FT \rightarrow T*F \mid FF$
- $F \rightarrow (E)F \rightarrow (E) \mid \text{numnum}$

Queremos derivar la cadena "num + num * num". Podemos construir un árbol de derivación como sigue:

1. Nodo raíz: SS

- Hijo: EE
- 2. Expandimos EE usando la regla $E \rightarrow E+TE \rightarrow E+T$
 - Hijos: EE, ++, TT
- 3. Expandimos el primer EE usando la regla $E \rightarrow TE \rightarrow T$
 - Hijos: TT
- 4. Expandimos TT usando la regla $T \rightarrow FT \rightarrow F$
 - Hijos: FF
- 5. Expandimos FF usando la regla $F \rightarrow \text{num}F \rightarrow \text{num}$
 - Hijos: numnum
- 6. Volvemos al primer TT y lo expandimos usando la regla $T \rightarrow T*FT \rightarrow T*F$
 - Hijos: TT, **, FF
- 7. Continuamos expandiendo hasta que todos los nodos hoja sean terminales, lo que nos da el árbol de derivación completo.

Explicar el proceso de construcción de inferencias recursivas de acuerdo a derivaciones

1. **Identificación de Premisas y Objetivos:** Comienza identificando las premisas disponibles y la conclusión o conclusiones a las que deseas llegar.
2. **Inicialización:** Inicia una "derivación" con las premisas como los elementos iniciales. En este contexto, una derivación es una secuencia de pasos lógicos que te llevan de las premisas a una o más conclusiones.
3. **Aplicación de Reglas de Inferencia:** Aplica una regla de inferencia a una o más de las premisas o conclusiones intermedias para derivar una nueva conclusión. Añade esta nueva conclusión a la derivación.
4. **Recursividad:** Repite el paso 3 de manera recursiva, aplicando reglas de inferencia a las nuevas conclusiones y añadiéndolas a la derivación, hasta que llegues a la conclusión deseada o hasta que no puedas aplicar más reglas.
5. **Terminación:** El proceso termina cuando has derivado la conclusión deseada o cuando has agotado todas las posibles reglas de inferencia sin llegar a la conclusión.
6. **Verificación:** Finalmente, revisa la derivación para asegurarte de que cada paso es válido según las reglas de inferencia y que has llegado a la conclusión deseada de manera lógica.

EJEMPLO:

Supongamos que tenemos las siguientes premisas y queremos llegar a una conclusión:

- Premisa 1: Todos los humanos son mortales. ($P \rightarrow Q$)
- Premisa 2: Sócrates es humano. (P)
- Conclusión deseada: Sócrates es mortal. (Q)

Podemos construir una derivación recursiva como sigue:

1. Inicializamos la derivación con las premisas: $P \rightarrow Q, PP \rightarrow Q, P$
2. Aplicamos la regla de Modus Ponens, que nos permite inferir Q si tenemos $P \rightarrow Q$ y P . Esto nos da Q (Sócrates es mortal).
3. Añadimos esta nueva conclusión a la derivación.
4. En este caso, hemos llegado a la conclusión deseada en un solo paso, por lo que el proceso termina.

La derivación completa sería:

$P \rightarrow Q$ $P \rightarrow Q$ (Premisa)

P (Premisa)

Q (Derivado de 1 y 2 mediante Modus Ponens)

REFERENCIAS BIBLIOGRAFICAS

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compiladores: Principios, técnicas y herramientas*. Boston, MA: Addison-Wesley.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). *Introducción a la teoría de autómatas, lenguajes y computación*. Boston, MA: Addison-Wesley.

Johnson, L., & Gupta, A. (2019). *Técnicas avanzadas de programación*. Nueva York, NY: Academic Press.