Pregunta:

La notación polaca inversa esta siempre asociada a una pila. Considere una generación de código que se baso en los siguientes pasos: Lista de reglas->Polaca Inversa->Assembler

Indicar si es realmente necesaria una pila en cada una de las siguientes actividades y porque. En caso afirmativo explique que contiene la misma

1. Durante la traducción de la lista de reglas a polaca inversa.
2. Durante la traducción de polaca inversa a assembler
3. Durante el uso del programa ejecutable.

Respuestas (SUPUESTAS).

1. Si es necesario para resolver los loop y los saltos.
2. Operando y operador.
3. No es necesaria.

Pregunta

Cuáles de las siguientes características deben escribirse mediante las reglas de BNF y cuáles no. Justificar:

a) Una variable no puede utilizarse sin haber estado presente previamente en el lado izquierdo de

una asignación.

b) Las sentencias REPEAT UNTIL solo permiten 3 niveles de anidamiento.

c) No pueden existir funciones declaradas dentro de funciones.

d) El lenguaje no permite coerciones (conversiones implícitas)

e) El lado izquierdo y de derecho de una asignación deber ser del mismo tipo.

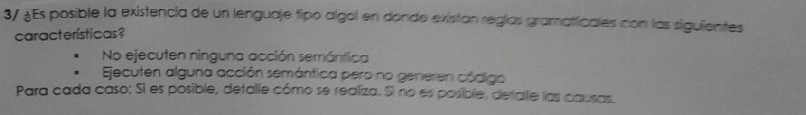
Pregunta

Parsing Descendente Predictivo, Parsing Ascendente Predictivo, que limitaciones y características tienen, considerando solo LL(1) y SLR. Ejemplifique brevemente con reglas.

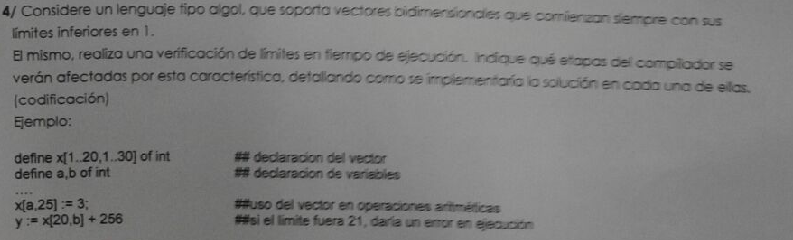
Pregunta

Suponga un compilador que genera assembler. Se sabe que este archivo se genera en la última etapa del compilador. Se sabe también que las variables y sus tipos están presentes en la tabla de símbolos. ¿Cómo toman lugar los tipos de las variables en el programa assembler que entrega el compilador? Ejemplifique.

Pregunta



Pregunta



Pregunta



Pregunta

Un compilador maneja constantes string. El automata lexico (o sea, el flex) llega a su estado final y al devolver el token STRING arroja: "CHAR invalido en la string".

?Es esto posible?

RESPUESTA

No es posible el token no se llega armar (no llega al estado final).

Pregunta

Dado:

integer a

ALIAS x%a

ALIAS b%c // Esto da error, porque c no esta definido

Decir que etapas se ven afectadas durante la compilación y describir que pasa en cada una

a) AL

b) AS

c) TS

d) GCI

e) GCASM

RESPUESTA

AL) Palabra reservada alias y %

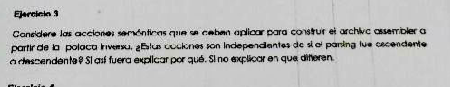
AS) Regla <alias> -> id % id

TS) Columna alias

GCI) Se remplaza var alias por var arg

GCASM) NADA

Pregunta



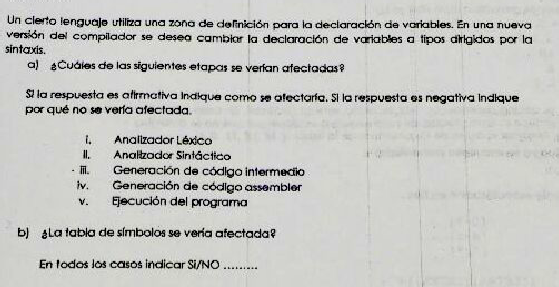
(Rta: No, si se pudo generar codigo intermedio tanto con ascendente y descendente

(a veces parsing descente no puede, si no esta factorizada o es recursiva por izquierda)

la lista de reglas generada sera distinta pero disparara las mismas acciones semanticas,

por lo que el ASM generado a partir del GCI sera igual tambien.)

PREGUNTA



a) AL: SI -> Nuevos caracteres a reconocer ( %, &, etc.)

y enviar tokens de IDs ya con tipo aclarado

b) AS: SI -> Nuevas reglas para que la sintaxis acepte esos tokens.

c) GCI: NO -> El codigo intermedio que se genera no se ve afectado

por el momento donde defino los tipos de variable.

d) GCASM: NO -> Si el GCI no esta afectado, menos el ASM.

e) Ejecucion: NO -> El cambio concierne solo la parte de compilacion.

f) ?Cambia la tabla de simbolos? : Ni, no cambia la estructura pero cambia el

momento donde se asignan los tipos, ya desde el lexico podes saber el tipo de

la variable y no en analisis sintactico como suele ser.

PREGUNTA

Decir a que parte de la compilacion/ejecucion pertenecen estos mensajes y justificar por que.

1. No se reconoce el simbolo recibido

RTA: Lexico (Recibo un simbolo para el cual no defini ningun TOKEN)

1. El indice del vector esta fuera de rango

RTA: Ejecucion o semantico (depende si valido o no limites de vectores)

1. Indice de matriz no es del tipo entero

RTA: Semantico (Busco en TS y valido si el tipo es INT)

1. El TOKEN recibido no se esperaba

RTA: Sintactico (Llego a un estado donde espero un TOKEN distinto al que recibo)

1. Se esta indexando un id que no es un vector

RTA: Semantico (Busco en TS si el id es vector o no)

1. Puntero apunta a una direccion de memoria prohibida

RTA: Ejecucion (No hay direccionamiento de memoria durante compilacion)

1. Variable fuera de rango, dentro de una unión

RTA: Ejecucion (no se sabe valor de variable hasta ejecutar)

PREGUNTA

Polaca inversa: Se hace una compilacion con polaca

inversa siguiendo las etapas Lista de Reglas -> Polaca Inversa -> Codigo ASM.

Sabiendo que la polaca inversa hace uso de una pila, es esta necesaria siempre?

Decir si es necesaria o no en cada una de las siguientes etapas:

1. Lista de Reglas -> Polaca Inversa

RTA: Depende del codigo, la pila guarda los comparadores principalmente, que seran usados por la tira de polaca inversa. Puede que no haya que meter nada.

1. Polaca Inversa -> Codigo Assembler

RTA: Lo mismo, si esta vacia no pasa una. Si tiene algo, seguira el algoritmo de polaca inversa y desapilara comparadores a medida que sea necesario.

1. Ejecucion

RTA: Cuando el codigo ASM esta listo, la pila ya no existe. El codigo ASM tiene todo lo necesario para ejecutarse.

PREGUNTA

Durante la etapa de creacion del compilador:

a)Donde se guardan los valores de las variables de usuario y quien se encarga de administrarlas?

RTA: En compilacion no se saben los valores de las variables. MAL NO JUSTIFICA 2 (DOS) segun dani

b) Decir cual es la regla que dispara la generacion de codigo intermedio y explicar.

RTA: La regla 0, la que detecta el start symbol. Si se llega al start y no hay mas codigo para analizar, la sintaxis es correcta y ahi comienza la generacion de codigo intermedio.

PREGUNTA

Considere una generacion de codigo intermedio de arbol sintactico. Quiero que cuando se este por armar un nodo que divida b/1, en vez de crearlo optimice el codigo y cree solo una hoja "b". Explicar que necesito cambiar para lograr esto.

RTA: Una manera posible seria en la accion semantica que llame a crearnodo(), detectar que si el operador es "/" y la hoja derecha es "1", en vez de continuar con la ejecucion normal llame a la funcion crearhoja() para lo que iba a ser la hoja izquierda, osea cree solo la hoja "b" en este caso.