

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

INTEGRANTES:

RUEDA IMÁN SOFIA VIRIDIANA N° CONTROL:171080043

CELISEO GOMEZ ADAN GAMALIEL N° CONTROL:171080044

LUIS ENRIQUE MEJIA RAMOS N° CONTROL:161080183

Actividades semana 4 (Oct 12-16, 2020)

1) Actividades teóricas:

Ver con atención las clases "Introducción a Compiladores" y "Análisis Léxico (Parte 1)".

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONCLUSIÓN (Viridiana Imán).

Introducción a Compiladores.

El compilador es el programa Procedimientos escritos de traducción
Generado en el idioma de origen utilizando Lengua de llegada.

Hemos dicho que en informática la compilación es un proceso que
consiste en convertir un programa desarrollado en un determinado
lenguaje de programación en otro programa escrito en otro lenguaje.
Por lo general, el segundo programa está escrito en lenguaje de
máquina, que es un código que puede ser interpretado directamente
por el procesador

Análisis Léxico.

El analizador sintáctico aísla el analizador sintáctico de la
representación del morfema del componente léxico. El analizador
léxico funciona de acuerdo con la solicitud del analizador, y cuando
el analizador lo necesita para avanzar la gramática, devuelve un
componente léxico. Suele implementarse como una subrutina del
analizador. Cuando recibe el comando "get next token", el analizador

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

léxico lee los caracteres de entrada hasta que identifica el siguiente token.

CONCLUSIÓN (Adan Gomez).

Introducción a Compiladores.

El compilador es solo un programa que traduce otros programas. Los compiladores clásicos convierten el código fuente en código de máquina ejecutable que la computadora puede entender. (Algunos compiladores convierten el código fuente en otro lenguaje de programación. Estos se denominan compiladores o convertidores de fuente a fuente). LLVM es un proyecto de compilación ampliamente utilizado que consta de muchas herramientas de compilación modular.

Análisis Léxico.

En este tema, continuaremos Definir y comprender todo Tareas realizadas por el analizador Diccionario, esta es la clave Funcionamiento correcto traductor. Como se muestra, tiene Código fuente como entrada Lenguaje de programación Acepta el compilador Salida, proporcionar analizador Marca de sintaxis.

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

El analizador léxico funciona de acuerdo con la solicitud del analizador. Devuelve un componente léxico porque el analizador lo necesita para avanzar en la gramática. El componente léxico es el símbolo final de la gramática. Suele implementarse como una subrutina del analizador. Cuando reciba el pedido, obtendrán la siguiente parte de vocabulario, El analizador léxico lee los caracteres de entrada hasta que reconoce los siguientes componentes de la palabra.

CONCLUSIÓN (Luis Enrique).

Introducción a compiladores.

Servirá mucho en el momento de la creación de un compilador y que se detallan todas y cada una de las partes que involucran a este, saber que existen tantos tipos de compiladores, me gustaría crear un compilador de optimización ya que pienso que es muy útil al crear un algoritmo o programa. La función de los compiladores es leer un programa escrito en un lenguaje en este caso el lenguaje fuente y lo traduce a un programa equivalente en otro lenguaje, Me parece fascinante que nosotros podamos crear un compilador en 6 meses

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

(en un curso) cuando en los años 50 ya que en aquellos tiempos se tardaron hasta 18 años trabajando en un compilador.

Analisis Lexico

Podemos Concluir que el Análisis Léxico es la fase más importante de un compilador. No solo por ser su primera fase, sino que también su función es revisar errores gramaticales en el código fuente evitando así la mayoría de errores que pueden seguir durante la compilación y convierte este en código entendible para los demás analizadores y poder continuar el proceso.

CONCLUSIÓN (Oscar).

Introducción a compiladores.

Los principios y técnicas que se usan en la escritura de compiladores se pueden emplear en muchas otras áreas. Se basan en los conceptos de teoría de autómatas y lenguajes formales que se están

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

exponiendo en la parte teórica, y constituyen un campo de aplicación práctica bastante directo.

El objetivo es, básicamente es traducir un programa (o texto) escrito en un lenguaje "fuente", que llamaremos programa fuente, en un equivalente en otro lenguaje denominado "objeto", al que llamaremos programa o código objeto. Si el programa fuente es correcto (pertenece al lenguaje formado por los programas correctos), podrá hacerse tal traducción; si no, se deseara obtener un mensaje de error (ovarios) que permita determinar lo más claramente posible los orígenes de la incorrección. La traducción podrá hacerse en dos formas:

interpretación: la traducción se hace "frase a frase"

compilación: la traducción se hace del texto completo

Concentramos nuestra atención en el segundo de los métodos

Mientras tanto "LLVM" es un proyecto de código abierto que busca la creación de compiladores para cualquier lenguaje de programación, proporcionando la infraestructura necesaria para su desarrollo.

Análisis léxico

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

La cadena de entrada se recibe como una sucesión de caracteres. El análisis léxico agrupa los caracteres en secuencias con significado colectivo y mínimo en el lenguaje, llamadas componentes léxicos (palabras o "token"), conciertos atributos léxicos. En el ejemplo, se detectarían 7:

- 1.el identificador posición
- 2.el operador de asignación: =
- 3.el identificador inicial
- 4.el operador +
- 5.el identificador velocidad
- 6.el operador *
- 7.la constante numérica 60

Cada vez que se detecta un nuevo identificador, se anota una entrada en la tabla de símbolos. El lenguaje de entrada tendrá un catálogo de componentes posibles, qué se podrán codificar (por ejemplo, mediante constantes enteras).

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

Actividades semana 5 (Oct 19-23, 2020)

2) Actividades prácticas:

Identificar las secciones del Front-End (Análisis Léxico, Sintáctico y Semántico) de LLVM

Opinión: Mejia Ramos Luis Enrique.

ANÁLISIS DE SINTAXIS: GRAMÁTICAS LIBRES DE CONTEXTO,
AUTÓMATAS PUSHDOWN

El término libre de contexto se refiere al hecho de que el no terminal V puede siempre ser sustituido por w sin tener en cuenta el contexto en el que ocurra. Un lenguaje formal es libre de contexto si hay una gramática libre de contexto que lo genera.

Las gramáticas libres de contexto permiten describir la mayoría de los lenguajes de programación, de hecho, la sintaxis de la mayoría de lenguajes de programación está definida mediante gramáticas libres de contexto. Por otro lado, estas gramáticas son suficientemente simples como para permitir el diseño de eficientes algoritmos de

TECNOLÓGICO NACIONAL DE MEXICO
INSTITUTO TECNOLÓGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

análisis sintáctico que, para una cadena de caracteres dada, determinen cómo puede ser generada desde la gramática. Los analizadores LL y LR tratan restringidos subconjuntos de gramáticas libres de contexto.

Opinión: Rueda Iman Sofia Viridianan.

**ANÁLISIS DE SINTAXIS: GRAMÁTICAS LIBRES DE CONTEXTO,
AUTÓMATAS PUSHDOWN**

En la teoría del lenguaje formal, una gramática libre de contexto (CFG) es una gramática formal en la que cada regla de producción tiene la forma donde α es un solo símbolo no terminal y β es una cadena de terminales y / o no terminales (puede estar vacío). Una gramática formal se considera "libre de contexto" cuando sus reglas de producción se pueden aplicar independientemente del contexto de un no terminal. Independientemente de los símbolos que lo rodeen, el no terminal único en el lado izquierdo siempre se puede reemplazar por el lado derecho. Esto es lo que lo distingue de una gramática sensible al contexto. Una gramática formal es esencialmente un conjunto de reglas de producción que describen todas las cadenas posibles en un lenguaje formal dado. Las reglas de producción son reemplazos simples. Por ejemplo, la primera regla en la imagen, reemplaza S con AB . Puede haber varias reglas de reemplazo

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

para un símbolo no terminal dado. El lenguaje generado por una gramática es el conjunto de todas las cadenas de símbolos terminales que pueden derivarse, mediante aplicaciones de reglas repetidas, de algún símbolo no terminal particular ("símbolo de inicio"). Los símbolos no terminales se utilizan durante el proceso de derivación, pero es posible que no aparezcan en su cadena de resultado final. Los lenguajes generados por gramáticas libres de contexto se conocen como lenguajes libres de contexto (CFL). Diferentes gramáticas libres de contexto pueden generar el mismo lenguaje libre de contexto. Es importante distinguir las propiedades del lenguaje (propiedades intrínsecas) de las propiedades de una gramática particular (propiedades extrínsecas).

Opinión: Celiseo Gomez Adan Gamaliel.

**ANÁLISIS DE SINTAXIS: GRAMÁTICAS LIBRES DE CONTEXTO,
AUTÓMATAS PUSHDOWN**

El análisis sintáctico o "parseo" es la segunda fase de un compilador. En este post aprenderemos los conceptos básicos que se usan en la construcción de un parser.

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Hemos visto que un analizador léxico puede identificar tokens con la ayuda de expresiones regulares y reglas de patrones. Pero un analizador léxico no puede revisar la sintaxis para una sentencia dadas las limitaciones que las expresiones regulares presentan. Las expresiones regulares no pueden revisar el balanceo de tokens, como paréntesis llaves y corchetes, entonces esta fase utiliza una gramática libre de contexto la cual es reconocida por un automata "push-down".

Un analizador sintáctico o "parser" toma como entrada la salida de un analizador léxico en la forma de streams de tokens. El parser compara el código fuente (stream de tokens) contra las reglas de producción de la gramática para detectar cualquier error en el código. La salida de un analizador sintáctico es entonces un "árbol de parseo".

Actividades semana 6 (Oct 26-30, 2020)

Para complementar la comprensión de los conceptos presentados en los videos consulten los libros de referencia de la asignatura... de manera personal y en equipo deben actualizar sus respectivos apuntes de los conceptos de esta semana. Cada representante de equipo deberá enviarme por correo la liga a sus apuntes actualizados en Google Docs a más tardar el lunes 2 de Noviembre 2020 a la media noche.

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONCLUSIÓN RUEDA IMAN SOFIA VIRIDIANA.

Análisis sintáctico La fase de análisis sintáctico tiene por objetivo solicitar tokens al analizador léxico y construir una representación arborescente de todo el código fuente. Este proceso se encuentra dirigido por el conocimiento gramatical que del lenguaje tiene el analizador sintáctico

Lenguajes libres de contexto Desde una perspectiva sintáctica un lenguaje es una colección de construcciones sintácticas bien formadas desde un alfabeto de entrada y correctamente combinadas entre sí de acuerdo a una colección de reglas sintácticas.

De manera similar a como ocurre en los lenguajes regulares, los lenguajes libres de contexto se pueden expresar mediante el uso de gramáticas libres de contexto. Su definición es una extensión de las gramáticas regulares con reglas de producción potencialmente más complejas.

CONCLUSIÓN CELISEO GOMEZ ADAN GAMALIEL.

Todo lenguaje posee una serie de reglas para describir los programas fuentes (sintaxis).

- Un analizador sintáctico implementa estas reglas haciendo uso de GICs
- Gramáticas

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

- Son un formalismo matemático que permite decidir si una cadena pertenece a un lenguaje dado.
- Se define como la cuarteta $G = (N, \Sigma, S, P)$, en donde N es el conjunto de símbolos terminales, Σ es conjunto de símbolos terminales, S es el símbolo inicial (S pertenece a N) y P es un conjunto de reglas de producción. Gramáticas • Los símbolos no terminales (N) son aquellos que pueden seguir derivando en otros; mientras que los terminales el proceso finaliza allí.
- Las reglas de producción siguen el formato: $\alpha \rightarrow \beta$ donde α y β pertenecen a N y Σ en cualquier forma.

CONCLUSIÓN LUIS ENRIQUE

Habla sobre autómatas de empuje hacia abajo M tiene un conjunto finito de afirma de Q , tiene un alfabeto de pila γ , así como una vez nos explico el profesor en autómatas siempre tendrá un estado inicial y aquí lo que vemos que es esta mas detallado ya con un método de pila por así decirlo un símbolo de pila inicial en Z , y también encontramos estados finales de F . Este video nos lleva más allá de saber las funciones que existen para los autómatas, muy entendible y de hecho muy explicativo lo que explican, solo que en mi opinión para entender mejor deberían estar en español.

Mod-03 Lec-07 Análisis de sintaxis: Gramáticas libres de contexto, Autómatas pushdown y Parte de análisis 3

Opinión Mejia Ramos Luis Enrique:

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Hemos visto que un analizador léxico puede identificar tokens con la ayuda de expresiones regulares y reglas de patrones. Pero un analizador léxico no puede revisar la sintaxis para una sentencia dadas las limitaciones que las expresiones regulares presentan. Las expresiones regulares no pueden revisar el balanceo de tokens, como paréntesis llaves y corchetes, entonces esta fase utiliza una gramática libre de contexto la cual es reconocida por un automata "push-down"

Las gramáticas libres de contexto por otra parte, pueden ser un super-conjunto de una gramática regular

Mod-03 Lec-08 Análisis de sintaxis: Gramáticas libres de contexto, Autómatas pushdown y Parte de análisis – 4

Opinión:

Sobre este video sigue hablando de mas reglas, siguiendo reglas de derivación, pilas, solo un detalle que el problema es que las gramaticales recursivas izquierdas no satisfacen la condición LL1. Entonces si una gramática no tiene una terminal a tal que A, en más de una aplicación de las producciones de A produce A Alfa. Esta un poco enredado el tema pero pues es de aplicarnos y meternos de lleno.

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Actividades semana 7 (Nov 2-6, 2020)

Para complementar la comprensión de los conceptos presentados en los videos consulten los libros de referencia de la asignatura... de manera personal y en equipo deben actualizar sus respectivos apuntes de los conceptos de esta semana. Cada representante de equipo deberá enviarme por correo la liga a sus apuntes actualizados en Google Docs a más tardar el lunes 9 de Noviembre 2020 a la media noche.

CONCLUSIÓN MEJIA RAMOS LUIS ENRIQUE

Mod-03 Lec-09 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 5

Buen tutorial de este profesor con los conceptos del análisis y técnicas. Tenemos en cuenta que hoy en día podemos escribir gramáticas demasiado fáciles. Y también en parte tener en cuenta los generadores de analizador sintáctico la famosa LR.

Mod-03 Lec-10 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 6

El problema con el análisis de SLR (1)", ignore por completo la explicación del profesor. El problema en el ejemplo anterior fue que la gramática tiene una regla de tipo "L =". Por lo tanto, si el símbolo de anticipación es "=" y aplicamos la reducción R-> L, es decir, reducimos L a R, entonces aterrizamos con un prefijo de tipo "R =". No existe una forma de oración correcta para la gramática dada que

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

comienza con "R =" (solo tenemos la producción " $S \rightarrow L = R$ "), por lo tanto, la aplicación de la reducción $R \rightarrow L$ resultará en un callejón sin salida. Por tanto, debe evitarse la aplicación anterior de $R \rightarrow L$.

Mod-03 Lec-11 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 7

Todos y cada uno de los conceptos, especialmente el análisis de abajo hacia arriba, se enseña de una manera muy fácil, son interesantes los conceptos y todo si es cansado mucha teoría e información en un solo video.

CONCLUSIÓN RUEDA IMAN SOFIA VIRIDIANA

Mod-03 Lec-09 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 5

Hemos visto que un analizador léxico puede identificar tokens con la ayuda de expresiones regulares y reglas de patrones. Pero un analizador léxico no puede verificar la sintaxis de una oración dada debido a las limitaciones de las expresiones regulares. Las expresiones regulares no pueden verificar los tokens de equilibrio, como los paréntesis. Por lo tanto, esta fase usa gramática libre de contexto (CFG), que es reconocida por autómatas pushdown

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

Mod-03 Lec-10 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 6

El objetivo del analizador es reconocer si la frase pertenece a la gramática y para ello utiliza una pila para almacenar los distintos símbolos gramaticales (terminales y no terminales) donde incorpora el estado 0 (S_0) en lugar de \$ en el inicio de la pila. En la entrada nos encontramos al final de la cadena. Es ahora la tabla la que indica si para un estado en concreto la acción es ACEPTAR la entrada y terminar el análisis con éxito. La parte de acción de la tabla de análisis indica una acción del analizador, mientras que la parte de ir_a indica transiciones entre los estados (contendrá los números de estos estados)

CONCLUSIÓN CELISEO GOMEZ ADAN GAMALIEL

Mod-03 Lec-09 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 5

De esta manera, el analizador realiza dos tareas, es decir, analizar el código, buscar errores y generar un árbol de análisis como resultado de la fase.

Se espera que los analizadores analicen todo el código incluso si existen algunos errores en el programa.

Mod-03 Lec-10 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 6

Entenderemos los conceptos y operaciones básicos de un analizador SLR y aprenderemos a construir la tabla de análisis SLR, la cual revisaremos mediante

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

un ejemplo. Además conoceremos qué problemas o límites presentan los analizadores SLR.

El analizador sintáctico ascendente SLR, también denominado LR sencillo (del inglés simple LR), se basa en el AFD LR(0). Sin embargo, incrementa de manera importante la potencia del análisis sintáctico LR(0) al utilizar el token siguiente en la cadena de entrada para dirigir sus acciones (Louden, 2004).

Conclusión Quintero Bolio Erik Eduardo

Mod-03 Lec-09 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 5

Como se ha visto en los videos anteriores el LR es un método de análisis de abajo a arriba (de reversa) por lo que es fácil escribir en gramáticas en LR. En el analizador se tienen como tal dos partes la pila y la entrada no utilizada por lo que su tabla se vuelve más compleja (acción y goto), no pueden haber bucles infinitos ya que de ser el caso se llama a la rutina de recuperación de errores.

Pueden existir varios elementos válidos para un único prefijo viable.

En la gramática el símbolo E es el inicio más sin embargo ya con las validaciones, en la gramática aumentada, el símbolo de inicio cambia a S

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Mod-03 Lec-10 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 6

Cada estado de DFA contiene sólo los elementos que son válidos para un solo prefijo viable. Cuando un nuevo elemento entra, el elemento anterior es empujado hacia abajo en la pila.

Existen elementos de reducción y de cambio , el elemento cambio es el que se pasa de estado a estado mientras que el que reduce es que no molesta en los siguientes estados. Cada vez que hay una reducción, se calcula el seguimiento

Existe una manera de eliminar el conflicto entre el cambio y la reducción y es por medio de la LRSL.

Mod-03 Lec-11 Syntax Analysis: Context-free Grammars, Pushdown Automata and Parsing Part - 7

Los analizadores LALR son poco conflictivos y menos si el analizador LR1 no tiene tampoco uno de sus usos los encontramos en la web con el cifrado WEP. En este tipo de analizadores si un símbolo provoca un error habrá reducciones más nunca se desplazará por la pila, así mismo elimina los conflictos de reducción de turnos.

Tiene un error de token especial que hace parecer que el error fue en la entrada.

El analizador YACC no define expresiones regulares que conozca, define las reglas gramaticales libres de contexto. Este asigna automáticamente los números a los tokens después del 257. Cualquier valor de la expresión debe ser

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

asignado al nombrar la extensión, su sistema de seguridad consta en traspasar tres tokens a la pila, si esto no pasa su sistema de error sigue activo

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

Actividades semana 8 (Nov 9-13, 2020)

Para complementar la comprensión de los conceptos presentados en los videos consulten los libros de referencia de la asignatura... de manera personal y en equipo deben actualizar sus respectivos apuntes de los conceptos de esta semana. Cada representante de equipo deberá enviarme por correo la liga a sus apuntes actualizados en Google Docs a más tardar el lunes 16 de Noviembre 2020 a la media noche.

Conclusión Quintero Bolio Erik Eduardo

Mod-04 Lec-12 Semantic Analysis with Attribute Grammars Part 1

El árbol de sintaxis se ingresa en el módulo del analizador semántico, que genera estos árboles de sintaxis después de validar la semántica del programa.

Debido a que el árbol de sintaxis ocupa mucho más espacio que la tabla de símbolos, generalmente lo almacenamos en una tabla de símbolos y luego colocamos un punto en el apropiado en la tabla de símbolo en uno de los campos en los nodos del árbol de sintaxis.

Las gramáticas de atributos no son más que extensiones de gramáticas libres de contexto.

Hay dos tipos de atributos:

- Heredados que se denotan como AI de X
- Atributos sintetizados que se denotan por AS de X

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Mod-04 Lec-13 Semantic Analysis with Attribute Grammars Part 2

Los atributos de los símbolos se evalúan sobre un árbol de análisis haciendo pasadas sobre el análisis de Árbol. La gramática de atributos diferente para el cálculo del número real de una representación de cadena de bits.

Un solo análisis sobre el árbol es suficiente y sucede, por lo que sucede que el atributo. La evaluación también se puede combinar con el análisis L R.

YACC sólo permite atributos sintetizados y las reglas que escribimos para YACC.

Mod-04 Lec-14 Semantic Analysis with Attribute Grammars Part 3

Las gramáticas de atributos son extensiones de contexto gramáticas libres. La evaluación de atributos de LAG s es muy simple. Sólo hacemos esa primera búsqueda en él, la salida es un árbol de análisis con valores de atributo coherentes.

Es el dfVisit secuencia que se utiliza para evaluar los atributos, los valores siempre fluyen de izquierda a derecha.

Cada identificador tiene varias piezas de información unido a ella.

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Conclusión RUEDA IMAN SOFIA VIRIDIANA

Mod-04 Lec-12 Semantic Analysis with Attribute Grammars Part 1

Durante la generación de código máquina de nuevo, necesitaremos conocer los tipos de variables para asignarles espacio, etcétera. Entonces, la optimización nuevamente requiere la información sobre variables para el momento del código, etcétera. Entonces, todos estos son usos de la tabla de símbolos durante otras fases de compilaciones. Entonces, si las declaraciones que conoce no necesitan aparecer antes de su uso como en el caso del lenguaje C plus plus, entonces el análisis semántico requiere más de una pasada. El punto es cuando escribes declaraciones de clase en C plus plus, pones las funciones miembro y luego sabes que normalmente escribes todas las declaraciones de las variables que se utilizan dentro de los métodos. Entonces, en tal caso cuando intentamos pasar los métodos o funciones miembro no sabremos el tipo de variables, porque la declaración aparece más tarde. La única forma de manejarlo es pasar por el programa una vez crea la tabla de símbolos y luego pasa por el programa por segunda vez y realiza semántica análisis. Entonces, veremos más de esto hacia el final. de estas conferencias tiene una semántica estática de los lenguajes de programación que se puede especificar utilizando lo que se conoce como gramáticas de atributos. Entonces, veremos cómo verificar la semántica estática de lenguajes de programación y cómo especificarlos usando gramáticas de atributos. El análisis semántico se puede generar semi automáticamente a partir de gramáticas de atributos gramaticales. Entonces, escribimos el semántica, conoce las especificaciones del programa para el lenguaje de programación y

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

luego podríamos utilizar un generador que los genere a partir de dicha gramática de atributos. La razón por la que decimos semi automáticamente es que las semánticas se suelen especificar en un lenguaje de programación como la notación, no es una notación pura libre de efectos secundarios, por lo que esa es la razón por la que se llama semiautomático.

Mod-04 Lec-13 Semantic Analysis with Attribute Grammars Part 2

Hay dos tipos de atributos heredados y sintetizados, por supuesto, el mismo atributo no puede ser ambos heredados y sintetizados, pero podría haber varios heredados y varios atributos sintetizados asociados con cada símbolo. Cada atributo toma un valor de un dominio especificado, como un entero o un producto real o cruzado de estos, etcétera, etcétera. Y asociamos reglas de cálculo de atributos con cada producción, así que asociamos específicamente reglas para el cálculo de atributos sintetizados del lado izquierdo no terminal, y reglas asociadas para el cálculo de atributos heredados de los no terminales del lado derecho de la producción. Por supuesto, el aspecto más importante de la gramática de un atributo es que es estrictamente las reglas estrictamente locales para cada producción P no hay efectos secundarios en las normas. Los atributos de los símbolos se evalúan sobre un árbol de análisis sintáctico haciendo pases sobre el análisis árbol, es posible tener más de una pasada y en cada pasada habría al menos un atributo calculado en los nodos. los atributos sintetizados se calculan en un fondo de las hojas hacia arriba, por lo que básicamente se sintetizan a partir de los valores de los atributos de los hijos del nodo en el árbol de análisis. Los nodos hoja que son los terminales, tienen solo

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

sintetizan atributos y estos son inicializados por el analizador léxico y, por supuesto, no se puede modificar. En cuanto a los atributos heredados, estos en realidad fluyen desde el padre o los hermanos hasta el nodo en cuestión y luego, por supuesto, fluir hacia abajo las palabras de nuevo desde ese nodo hacia abajo. Una breve descripción de la estrategia de evaluación de atributos, debemos construir el árbol de análisis, luego construimos el gráfico de dependencia de los atributos. Realizar el ordenamiento topológico de la dependencia gráfico y obtenemos un orden de evaluación, por supuesto, luego se lleva a cabo la evaluación de atributos utilizando este orden. y la evaluación de atributos adjunta a las respectivas producciones se utilizan en el atributo proceso de evaluación. Entonces, miramos el ejemplo, continuamos mirando el ejemplo que les mostré la última vez, por lo que esta es la gramática del atributo para la evaluación de un número real a partir de su representación de cadena de bits. Por ejemplo, si tiene una cadena de bits 110.101, su valor decimal es 6.625, aquí hay un contexto la gramática libre N va a L punto R, L genera varios bits en el lado izquierdo del punto. Y si la gramática L va a BL o B, de manera similar, R genera los bits en el lado derecho de el punto y la gramática es R que va a BR o BB es, por supuesto, un bit 0 o 1. Ahora, los atributos de los varios, así que si miras esta cadena genera esta gramática genera cadenas binarias con un punto en el medio. Pero, no tenemos ninguna indicación de qué es el valor decimal es, se supone que la gramática del atributo nos da el valor decimal de esta cualquier cadena de bits en particular generada por esta gramática. Los no terminales NR y B tener solo un valor de atributo sintetizado, que es del dominio de los reales, donde ya que la L no terminal tiene una longitud de atributo sintetizada y tiene otro atributo sintetizado valor.

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

Mod-04 Lec-14 Semantic Analysis with Attribute Grammars Part 3

Todos estos son similar. Entonces, veamos la importante producción E que va a dejar id igual a E en E . Entonces, aquí el atributo heredado que es la tabla de símbolos de E uno es en realidad, ya sabes, enviado a E dos. Entonces, este es un cálculo perfectamente válido; porque depende de; el, ya sabes, el símbolo de dos puntos E depende del símbolo del padre. Entonces, no hay nada de malo en eso. Y luego, la tabla de símbolos de E tres ahora es la tabla de símbolos de E uno anulado con la asociación de id a E 2. Entonces, dado que id , E dos son todos, saber a la izquierda de E tres y el símbolo de un punto E es un atributo heredado. Ahí Tampoco hay nada de malo en esta regla de cálculo. Entonces, esto también es satisfactorio ya que en lo que respecta a la propiedad LAG. Ninguno de los otros, sabes que en realidad viola la propiedades de un LAG. Y por lo tanto, la gramática es LAG. Ahora, veamos la modificación de la regla let id igual a E dos en E tres. Modificar como retorno E tres con id igual a E dos. Digamos la semántica de esta declaración. es lo mismo que esto, sea id igual a E dos en E tres. Entonces, la semántica sería evaluar E tres con las apariciones de id reemplazadas por el valor de E dos. Entonces, la gramática del atributo en realidad no cambia en absoluto; porque es muy similar a esta regla. Entonces, vamos utilizar la asociación de id a E dos dentro de E tres. Pero, desafortunadamente, cuando hacemos eso, el símbolo de tres puntos E ahora depende de los atributos de símbolos a la derecha. Entonces, depende del nombre de id y depende de E two dot val, que están a la derecha de E tres. Entonces, esta es una clara violación de las reglas de LAG. Por lo tanto, esta gramática con la nueva producción devuelve E tres con id igual a E dos no es una L gramática atribuida. Entonces, prometo mostrar la secuencia de evaluación usando el procedimiento $dfVisit$. Entonces, es lo mismo ejemplo de evaluación

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

de esta expresión. Sea igual a un más cuatro en un más tres. Entonces, la secuencia de la visita está escrita aquí. Entonces, comenzamos aquí y luego vamos a dos, el La tabla de símbolos se inicializa en phi, luego visitamos tres, luego cuatro y luego cinco y luego visitamos seis. Entonces, esta es la estrategia dfVisit. Entonces, este atributo se evalúa aquí; la tabla de símbolos de atributos heredados, que se pasa a siete y luego a ocho también. En este punto, el número cuatro se pasa hacia arriba como un atributo sintetizado. Entonces, estas son las visitas; diez, once y doce. Entonces, una vez que se completen estas visitas, de hecho, vamos al número trece. Entonces, esto es diez, once, doce, etcétera. Y luego, una vez que hacemos eso, ya sabes, el trece y el catorce son dos símbolos terminales aquí. Por lo tanto, realmente no tienen ninguna acción semántica asociada. Pasamos al quince. Nosotros ahora están listos para evaluar la tabla de símbolos aquí utilizando el atributo heredado de la padre y hermano. Entonces, eso se hace aquí. Y luego, se pasa a. La visita continúa; la tabla de símbolos se pasa a los dieciocho. Y aquí el valor de a se busca en el tabla de símbolos que produce cuatro. Esto se convierte en un atributo sintetizado, que ahora se pasa hasta veinte, veintiuno y veintidós. Finalmente, pasados los veintidós visitamos plus, que tiene veintitrés. Ahora, esto visita; veinticuatro veinticinco producen el número tres, que se pasa de nuevo hacia arriba como atributo sintetizado. Y aquí, el valor cuatro de este y el valor tres de esto se agrega para producir siete, que nuevamente se pasa como un atributo sintetizado a este número de visita; durante la visita número treinta y uno. Y entonces, finalmente el valor se produce en S en la visita número treinta y dos. Entonces, esta es la dfVisit secuencia que se utiliza para evaluar los atributos. Y si lo miras, sabes, los valores siempre fluye de izquierda a derecha. Realmente nunca fluyen de derecha a izquierda o fluyen de arriba hacia abajo o de abajo

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

hacia arriba, pero nunca de derecha a izquierda. Ahora, veamos una clasificación de, ya sabes, gramáticas de atributos en atributos puros gramáticas y gramáticas de traducción atribuidas.

Conclusión CELISEO GOMEZ ADAN GAMALIEL

Mod-04 Lec-12 Semantic Analysis with Attribute Grammars Part 1

El programador tiene que secuenciar las comprobaciones semánticas de forma adecuada y luego enviarlo al generador. De hecho, yacc también es un generador de analizador semántico si se proporciona la semántica verifica apropiadamente en C. Las gramáticas de atributos son extensiones de gramáticas libres de contexto. Entonces, ahora sigamos adelante y estudiemos las gramáticas de atributos con cierto detalle. Entonces, sucede que las gramáticas de atributos requieren que se comprenda alguna terminología antes de tomar un ejemplo y estudiarlo. Entonces, veamos algo de terminología para entenderlo y luego proceda con un ejemplo. Entonces, dejemos que G igual a NTPS sea una gramática libre de contexto, como dije, las gramáticas de atributos no son más que extensiones de gramáticas libres de contexto. Entonces la base es definitivamente una gramática libre de contexto y deja que el conjunto de variables V de la gramática sea N unión T . Para cada símbolo de X de V podemos asociar un conjunto de atributos denotados como X punto a , X punto b , etcétera, por eso el nombre atribuye gramática. Hay dos tipos de atributos atributos heredados que se denotan como AI de X , por lo que atributos heredados de X y atributos sintetizados que se denotan por AS de X , estos son realmente conjuntos de atributos. Cada atributo toma valores de un determinado dominio, podría ser un dominio finito o podría ser un dominio infinito y llamamos al

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

dominio que conoce tal dominio como su tipo. Por ejemplo, incluso en lenguajes de programación decimos int de x, entonces x dominio entero es el dominio a partir del cual el valor de x, proporcionamos a valores para x en nuestro programa. Entonces, podríamos decir que entero es el tipo de variable X, la misma noción se traslada también a los atributos.

Mod-04 Lec-13 Semantic Analysis with Attribute Grammars Part 2

veamos el atributo gramática N que va a L punto R, por lo que el valor de N es; obviamente, el valor de L más el valor de R, asumiendo que asignamos pesos a la bits en L y R apropiadamente. Entonces, L genera un poco por la producción L yendo a B, por lo que el valor del punto L será solo el valor del punto B o 0 y la longitud del punto L es 1 porque, estamos generando solo 1 bit. L 1 va a B L 2, entonces L 2 tiene cierta longitud y B es un bit extra, que ahora se ha generado. Entonces, la longitud del punto L 1 es; obviamente, L 2 longitud de punto más 1, ¿qué pasa con el valor L 1 valor de punto? obviamente, tomamos la L 2 valor de punto al igual que tomamos 2 a la potencia 2 más el valor de 1 0 que es 2 que es el total será 6. De manera similar, tomamos el valor de L 2, por lo que el valor del punto L 2 y tomamos el valor del valor del punto del bit B 0 o 1 multiplicado por el peso apropiado 2 a la longitud de punto de potencia L 2. Entonces, aquí la longitud del punto L 2 habría sido 2, así que esto habría sido asignar el valor 2 a la potencia 2. Entonces, así es como el valor de L 1 obtiene, sabes que R a B calculado es muy simple El valor del punto R es igual a la barra del valor del punto B 2 y R 1 van a BR 2, por lo que hay una razón por la que tenemos el valor del punto B barra 2 es que generamos el, sabes el primer bit después de que el punto se aleja 2 elevado a la potencia menos 1, por lo que el valor del bit dividido por 2 es el valor del punto R. Ahora, R que va a BR 2 R 1 valor de punto será tome

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

el valor del punto B más el valor del punto R 2 y todo se dividirá por 2 que es bastante fácil de ver. Si miras la producción porque, R 2 tenía algún valor, ahora debido a el bit se ha desplazado a la derecha en 1 posición. Entonces, el valor dividido por 2 es el correcto y valor para este R 2 en particular y se ha introducido un nuevo bit, por lo que su valor es B valor de punto y como dije siempre debemos dividir el valor de bit por 2. Porque, los pesos comienzan con menos 1 aquí, B a 0 es el valor del punto b igual a 0 y B a 1 será el valor del punto B igual a 1. Entonces, aquí está, digamos, una cadena de muestra 110.101, es un árbol de análisis, las rojas son todas las hojas y este es el gráfico de dependencia basado en este árbol de análisis y la gramática de atributos. Entonces, el valor de N punto depende del valor de L punto y el valor de R, luego la longitud de N punto en este nodo depende de la longitud del punto L a continuación, mientras que el valor del punto L aquí depende del valor del punto B Longitud del punto L y valor del punto L en el nivel inferior. Entonces, aquí en este caso estamos buscando en este nodo, por lo que el valor del punto L aquí dependerá del valor del punto B, luego la longitud del punto L y Valor del punto L en este punto. Entonces, de manera similar para los otros nodos del árbol también y similar para la parte R El valor del punto R depende del valor del punto B y del valor del punto R, por lo que esto recursivamente hacia abajo. Entonces, veamos cómo se lleva a cabo la evaluación del atributo dado este gráfico de dependencia particular, un tipo topológico del gráfico de dependencia hace que estas hojas para ser evaluado primero y luego el siguiente nivel y luego el siguiente nivel y así sucesivamente. Aquí está el mismo gráfico, devuelva lo que sabe de una manera ligeramente diferente con el número de nodos según la secuencia de evaluación. Entonces, 1, 2 luego 3 estos son los 3 nodos que son evaluado primero porque, no requieren ningún otro nodo para ser evaluado, por lo que es

TECNOLOGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

posible para evaluar la longitud del punto L también junto con estos 2 o más bien estos 3 porque, no requiere cualquier otra cosa en este momento, pero no podemos evaluar el valor del punto L porque, el punto L el valor requiere un valor de punto b. Por lo tanto, preferimos esperar hasta que los valores de los puntos B sean todo completado y luego vaya a este nodo 4, que tiene estos 2 atributos y evalúe ambos al mismo tiempo. Entonces, como dije, el nodo 1 se evalúa primero, luego el nodo 2 y luego el nodo 3, solo para guardar space He mostrado todas estas 3 evaluaciones en la misma diapositiva. En los nodos 1 y 2, aplicamos la producción B va a 1 y el valor obtenido es el valor del punto B igual a 1, en el nodo 3 aplicamos la producción B a 0 con un valor de punto B igual a 0. Y luego, entonces estos 3 ya fuimos evaluados, ahora es el turno de estos nodos de ser evaluados. Entonces, ahora L longitud de punto igual a 1, L valor de punto igual a 0, entonces 4 y luego 5 y ya sabes estos son los nodos, que serán evaluados, en el nodo 4 la producción aunque todos esos valores como dije, el nodo 4 se evalúa primero y luego el nodo 5 porque, los valores del nodo 4 son necesarios para el nodo 5.

Mod-04 Lec-14 Semantic Analysis with Attribute Grammars Part 3

Dos tipos de los atributos son posibles; heredado y sintetizado. Y hay reglas asociadas con cada uno de las producciones para calcular estos atributos. Entonces, atributos sintetizados de la mano izquierda. se proporcionan atributos heredados y no terminales del lado derecho de los no terminales con reglas para calcular los atributos. Las reglas son, por supuesto, estrictamente locales a la producción. y no tienen efectos secundarios. Entonces, ahora la clasificación de gramáticas atribuidas a L y atribuidas a S. Los SAG son muy simples. Son solo atributos sintetizados. Y podemos utilizar cualquier estrategia de evaluación

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

ascendente en una sola pasada para evaluar todos los atributos. Entonces, debido a esta propiedad se pueden combinar con LR- análisis sintáctico. Y por lo tanto, YACC es muy útil con estos SAG. En L-atribuido gramáticas, las dependencias van de izquierda a derecha. Entonces, muy específicamente los atributos puede sintetizarse. Y si se heredan, tienen la siguiente limitación. Entonces en una producción p , $A \rightarrow X_1 X_2 \dots X_n$. Supongamos que consideramos un atributo X_i punto A que se hereda, y luego X_i punto a puede depender sólo de los elementos de A de A ; ese son los atributos heredados del no terminal del lado izquierdo. O los atributos de cualquier de los símbolos a la izquierda de X_i , que es k igual a 1 a i menos 1 . Entonces, nos concentraremos en SAG y LAG de 1 paso, en el que podemos hacer evaluación de atributos con LR, LL o RD análisis. Por tanto, la evaluación de atributos de los LAG es muy sencilla. Solo hacemos esa primera búsqueda en el, ya sabes, árbol de análisis. Entonces, dado un árbol de análisis con atributos no evaluados, la salida es un árbol de análisis con valores de atributo consistentes. Entonces, el método es muy simple para cada niño. m de n , de izquierda a derecha. Evalúe los atributos heredados de m porque estamos visitando m siguiente; entonces $dfVisit(m)$. Esa es la llamada recursiva. Y después de visitar a todos los niños, evaluar los atributos sintetizados de n . Entonces, este es el algoritmo para la evaluación de atributos de las gramáticas atribuidas a L. Entonces, este es el ejemplo que vimos la última vez. De hecho, es un LAG. Entonces, no hay dependencias que van de derecha a izquierda. Entonces, por ejemplo, en $D \rightarrow TL$, el tipo de punto L es igual a un punto T tipo. Entonces, el tipo de punto T se sintetiza a la izquierda de L y luego se hereda en el L . Entonces, veamos qué hace una gramática; una gramática no atribuida a L o no LAG como dicen. Entonces, en realidad es una gramática

TECNOLÓGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

similar; la información de tipo de asociación con nombres en declaraciones de variables. Pero, veamos la sintaxis.

Conclusión MEJIA RAMOS LUIS ENRIQUE

Mod-04 Lec-12 Semantic Analysis with Attribute Grammars Part 1

De manera similar, R también genera cualquier número de B aquí con cada B yendo a 0 o 1. Entonces, ahora, tenemos una gramática libre de contexto, que genera cadenas en este lado y cadenas en este lado y las une con un punto aquí, ¿Correcto? Entonces, dada esta gramática libre de contexto, aquí hay una gramática de atributo que evalúa la cadena y obtiene este valor en particular, nuevamente este atributo gramatical requiere solo atributos sintetizados. Entonces, A del N no terminal es lo mismo que los atributos sintetizados de R y que a su vez es el mismo que los atributos sintetizados de B . Por lo tanto, estos tres tienen el mismo atributo de nombre llamado valor que es del dominio de los números reales. Mientras, el no terminal L tiene dos atributos, uno es la longitud que es un número entero y otro valor llamado que es un número real. Entonces, la idea básica es que, a medida que generamos estos B , entonces este L habría acumulado cierto valor y B se coloca a la izquierda. Así que, por lo tanto, la B debe obtener un valor que sea igual a la longitud de la cadena de bits aquí, que es la longitud de L . Entonces, es por eso que requerimos tanto la longitud como el valor para este no terminal L . De manera similar, R genera BR , por lo que este B se genera fresco, R tendría algún valor de propio, pero como está en el lado derecho del punto, implica que el valor de R ahora se reduce en 2 y el nuevo valor se asigna como la suma de los valores de B y R , así que vayamos a través

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

de él y veamos. Entonces, N yendo a L punto R, el valor de N no es nada pero el valor del punto L más el valor del punto R es muy simple. Si L genera un solo bit B, el valor del punto L es el valor del punto B y la longitud del punto L es 1, de modo que es muy simple de nuevo, porque cualquier b es 0 o 1 es el valor de L. Si L 1 va a BL 2, entonces la longitud del punto L 1 es la longitud del punto L 2 más 1, lo cual es muy simple, porque estamos generando un nuevo bit aquí y el valor del punto L 1 es el valor del punto B, esta es una nueva posición de bit en 2 a la longitud del punto de potencia L 2, esto es lo que le estaba diciendo hace un momento L tiene algo de número de bits ya generados y más valor de punto L 2, por lo que este es el nuevo valor de L 1 aquí. De manera similar, R a B es el valor del punto R igual a Valor del punto B que es simple 0 o 1 y R 1 va a BR 2, por lo que se convierte en R 1 valor de punto igual al valor de punto B más valor de punto R 2 por 2. Entonces, aquí esto habría generado usted conoce cierto número de bits aquí y en realidad ahora hemos dicho que un nuevo bit está siendo generado, R tiene un cierto valor que ya es antiguo. Entonces, tome el valor de R, agregue el valor de B y divida todo por 2, porque B ahora tiene un valor que siempre es 0.5, el valor máximo del bit después del punto es 0.5, entonces esa es la razón por la que decimos el valor del punto B más el valor del punto R por 2. Entonces, B a 0 y B a 1 tiene B igual a 0 y el valor del punto B es igual a 1. Entonces, este es el atributo gramatical como en lo que respecta a la evaluación de números reales de la picadura de bits. Entonces, pararemos aquí y en la siguiente clase consideraremos los árboles de análisis para este atributo en particular gramática.

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Mod-04 Lec-13 Semantic Analysis with Attribute Grammars Part 2

Entonces, la longitud es del dominio de los enteros y real es del dominio de los números reales, ¿por qué requerimos dos atributos para L, donde como uno nos suministra para NLR y B. Entonces, para convencernos de que esto es necesario, echemos un vistazo a las cadenas del lado izquierdo del punto, ya sabes, suponga consideramos este uno de la cadena 110. El ya sabes el poder que se le da más bien al valor de exponenciación dado a este en particular, que es 2 elevado a 2. Entonces, el primero es 0 2 elevado a 0, el segundo es 2 elevado a 1 y el tercero uno es 2 elevado a 2, esto en realidad se basa en el número de bits correcto lado hasta el punto. Entonces, hay 2 aquí, si tuviéramos más, ese sería el poder del base 2, esto no se puede determinar y si mira la gramática para L, esta b indica esto uno que estamos considerando. Y L a la derecha indica el resto de los bits a la derecha de este en particular en este caso. Entonces, sin conocer la longitud de la cadena que genera L, no es posible para asignar el peso apropiado a este bit en particular. En el caso de la fracción el peso en realidad comienza con menos 1 justo después del punto, por lo que es 2 elevado a menos 1, esto es 2 elevado a menos 2 y el tercer bit es 2 elevado a menos 3. Y la gramática es también tal que, la B, que es la más cercana al punto, se genera primero y R es la resto de los puntos después de B.

Mod-04 Lec-14 Semantic Analysis with Attribute Grammars Part 3

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Los ejemplos son a, b, c: int X, Y: flotar. En la gramática anterior, teníamos int a, b, c y float x, y. Aquí, el tipo la información viene después de la lista de variables. Entonces, esta es una declaración de estilo Pascal. Los el efecto es similar. a, b, c están etiquetados con el tipo integer. Y x, y, por supuesto que hay no z aquí. Este es un error menor. x, y están etiquetados con el tipo real. Entonces, en este caso si miras la producción D yendo a L: T, todas las demás son iguales. Ahora, la información de tipo T es el tipo de punto T y se sintetiza. Y hay un no terminal L, que tiene un atributo heredado L tipo de punto. Por lo tanto, nos gustaría que el tipo de punto L se convierta en Tipo de punto T; es decir, el atributo sintetizado fluye realmente hacia la izquierda y entra en L tipo de punto. Entonces, esta es una dependencia de derecha a izquierda y no una dependencia de izquierda a derecha. Y esto es exactamente lo que hace que la gramática de atributos no sea LAG. No podemos evaluarlo en un pase de izquierda a derecha porque cuando llegamos a L, la información de tipo aún no está lista. Y después de sintetizar la información de tipo, ya hemos pasado L. Entonces, como estamos mirando la evaluación de 1 paso, este atributo gramatical no es LAG. otro ejemplo de gramáticas atribuidas a L. Entonces, esta es la evaluación de la expresión con, ya sabes, una buena construcción como L. Sea id igual a expresión en expresión. Entonces, este atributo gramatical es de hecho atribuido a L. Entonces, si observa alguna regla de atributo, atributo reglas de cálculo, por ejemplo, aquí se inicializa el atributo heredado E dot symtab de this. Entonces, no hay violación de ninguna regla aquí y S dot val se sintetiza y eso se convierte en E dot val. Entonces, no hay violación de ninguna regla aquí, con respecto a los LAG. En este caso, el atributo heredado de E uno entra en E dos y T.

TECNOLÓGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

Actividades semana 10 (Nov 23-27, 2020)

CONCLUSIÓN LUIS ENRIQUE MEJIA RAMOS

Es un buen comienzo para saber realizar o gestionar un proyecto, tener un enfoque optimista, Me atraen las razones de este fenómeno son muchas, la mayoría relacionada con malas prácticas de estimación, comunicación y administración del tiempo.. En este webinar dan buenos consejos para aplicarlos afuera , así como las técnicas que me parecieron indicadas en una parte que estoy trabajando hacia el marketing, con la productividad, saber los tiempos, tener algo bien planeado.

SG campus está super bien explicado ya que ayudan a llegar a buen fin ya sea con proyectos ágiles. Aunque es importante identificar las complejidades técnicas.

TECNOLÓGICO NACIONAL DE MEXICO
INSTITUTO TECNOLOGICO DE IZTAPALAPA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONCLUSIÓN RUEDA IMAN SOFIA VIRIDIANA

Si tenemos un equipo comprometido y si estamos todos juntos y comprendimos el problema y si podemos tener un compromiso juntos que quieras el compromiso del equipo entonces la otra pregunta con su proyecto que tiene en mente hoy el quién de quién es el compromiso de ustedes como equipo .

NO olvidar que nosotros siendo parte de la organización somos parte de ese ADN y de esa cultura organizacional y que el cambio lo empieza desde una sola persona si una sola persona que implementen nuevas prácticas que sus proyectos le salgan bien y entonces el vecino pregunta qué fue lo que hiciste y se conviertan en casos de éxito y de repente eso que hiciste se pueda replicar entonces aunque hay muchas cosas que vamos a ver el día de hoy que son factores externos en los cuales podemos involucrarnos para mejorar quiero que dediquemos esta hora larga alerón a reflexionar sobre qué estamos haciendo que estamos dejando de hacer la parte interna en estos tres pilares estimación administración de tiempo y comunicación y bueno lo que vamos a ver es bueno esta relación porque a fin de cuentas tanto la comunicación como estimación o estas debilidades vamos tener sacrifican esta parte de la administración del tiempo.

TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

CONCLUSIÓN CELISEO GOMEZ ADAN GAMALEIL

Generalmente cuando empezamos ver qué le pasó un proyecto porque ese está atorando cuando empezamos a ver problemáticas muchas recaen en la parte de productividad vamos a ver otros factores que también se meten por ahí en el camino pero digamos que a fin de cuentas la productividad es una preocupación no solamente del cliente o de la empresa donde trabajamos sino es algo que también nos preocupa a nosotros porque mientras más productivos seamos más no rendirá el tiempo y mejor saber nuestra calidad de vida en todos los sentidos entonces estamos en una industria que tiene varias implicaciones una de las complicaciones del desarrollo de software es que de por sí el desarrollo del software tiene muchas variables que controlar a nivel gestión y a nivel técnico.

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Conclusión Quintero Bolio Erik Eduardo

Se suelen tener ciertas debilidades al momento de establecer un proyecto, sin embargo, estas son fácilmente identificables por los expertos ya que se realiza una comparativa con los tres pilares del desarrollo de un proyecto, con esto podemos definir mejor un proyecto más adecuado.

Un proyecto debe de contar con un objetivo claro al cual se llegará por medio de la realización del proyecto con las debidas herramientas para una correcta gestión, estas herramientas son estrategia, plan de trabajo, métricas, matriz de rastreabilidad, riesgos, administración de calidad).

Con eso será suficiente para poder tener un proyecto exitoso.

TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INGENIERÍA EN SISTEMAS COMPUTACIONALES