



Construcción de Herramientas Didácticas para la Enseñanza y Ejercitación Práctica en Laboratorio de Informática Teórica en las Carreras con Informática

Gramáticas Formales

Una gramática formal G , es una 4-tupla, que queda definida de la siguiente manera:

$$G = (\Sigma_T, \Sigma_N, S, P)$$

En donde:

Σ_T : Conjunto de Símbolos que representan el Alfabeto de Símbolos Terminales, en donde toda palabra del lenguaje generado por esta gramática, estará formada por símbolos o caracteres definidos en este conjunto.

Σ_N : Conjunto de Símbolos que representan el Alfabeto de Símbolos No Terminales. Este conjunto de símbolos será utilizado como símbolos auxiliares en la derivación de cadenas, pero no formarán parte de las cadenas del lenguaje.

S : Símbolo No Terminal especial, que pertenece al conjunto de Símbolos No Terminales (Σ_N), denominado símbolo inicial o Axioma de la Gramática.

P : Conjunto finito de reglas o producciones que tienen como única restricción que en la parte izquierda debe haber al menos un símbolo no terminal.

OBJETIVO GENERAL

El objetivo del proyecto es lograr que los estudiantes fijen los contenidos curriculares de teoría de autómatas, gramáticas formales, traducción de lenguajes y compiladores impartidos en la asignatura SSL, realizando práctica efectiva de los mismos en simuladores de máquinas abstractas, estudiando el funcionamiento y los programas fuente de los simuladores, de un generador de analizadores léxicos y de un generador de analizadores sintácticos, herramientas éstas construidas utilizando lenguajes de programación avanzados.

Además, se alentará la conformación de grupos de trabajo para la experimentación en laboratorio con los programas fuentes de las herramientas, permitiendo modificaciones que agreguen nuevas funcionalidades y para optimizar su funcionamiento.

Como beneficios adicionales, las herramientas y su documentación estarán desarrolladas en idioma español y los estudiantes y docentes de SSL tomarán un primer contacto con las últimas tecnologías de desarrollo de productos informáticos.

Objetivos Específicos

Cognitivos: lograr profundo conocimiento teórico y práctico, de las técnicas y herramientas conceptuales de la informática teórica, relacionadas con la construcción de compiladores, con el reconocimiento general de patrones, la programación automática desde una especificación formal, la traducción de códigos y los modelos de computación lineal e introducirnos a los modelos paralelos.

Académicos: actualizar la currícula de las materias relacionadas con esta temática de la carrera de ingeniería en sistemas de información, propender a la actualización y al perfeccionamiento docente e involucrar a alumnos en el estudio serio y la investigación de temas fundamentales de su carrera.

PRODUCTOS A CONSTRUIR

Página Web, con los contenidos teóricos de la asignatura, y para la ejercitación práctica de los contenidos, las siguientes herramientas:

Simuladores de autómatas finitos, con pila, linealmente acotados y máquina de Turing, para que los alumnos evalúen sus soluciones a los problemas planteados en las clases prácticas.

Generador de analizadores léxicos a partir de una gramática regular que muestre la generación automática de código desde una especificación formal y las distintas formas de tratar con el no determinismo del autómata finito obtenido: conversión a autómata finito determinista, búsqueda en árboles generados por estados posibles y procesamiento en paralelo de los distintos caminos deterministas generados por el no determinismo del autómata.

Generador de analizadores sintácticos LL(k) y LR(k) desde la especificación de un lenguaje según su gramática libre de contexto. Aquí, la técnica para atacar el no determinismo del autómata con pila, isomorfismo con la gramática del lenguaje, será la de pre-análisis de profundidad k-ésima de la cadena de entrada.

RESULTADOS ESPERADOS

Los contenidos Teóricos, junto con las herramientas desarrolladas, estarán a disposición de los alumnos para que efectúen simulaciones de manera que logren un aprendizaje significativo de los conceptos teóricos.

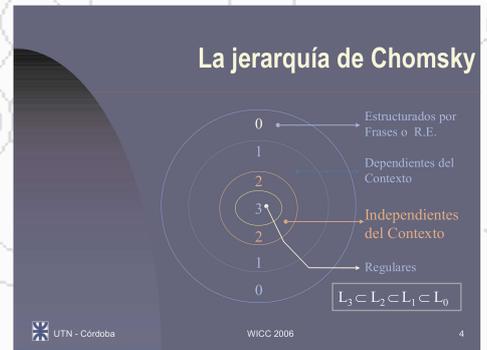
Estas herramientas estarán presentadas en la web, y se ejemplificará su utilización a través de un ejemplo real de aplicación con es la construcción de un compilador académico para la máquina abstracta RAM (Máquina de acceso aleatorio).

Al poder los alumnos visualizar una aplicación práctica de los contenidos, y poder simular la ejecución de autómatas finitos, con pila, linealmente acotados y máquina de Turing, de manera que ellos mismos evalúen las soluciones a los problemas planteados en las clases prácticas, es de esperar que los conocimientos y habilidades adquiridas, sean realmente significativas y eleven el rendimiento en el cursado y aprobación de la asignatura.

IMPORTANCIA DEL PROYECTO

Con la ejecución de este proyecto, no solo se pretende organizar y unificar la nomenclatura del material para el dictado de la asignatura, presentarlo en forma didáctica, posibilitando a los alumnos la práctica efectiva de los mismos, de manera de lograr mayor eficiencia en el cursado de la asignatura.

También se persigue la actualización de todos los docentes y auxiliares integrantes de la cátedra Sintaxis y Semántica del Lenguaje, provocando además, una actualización en el desarrollo de aplicaciones con lenguajes y técnicas de modelado actualizadas. Una vez construidas estas herramientas, y logrado un acabado conocimiento teórico y práctico, de las técnicas y herramientas de la informática teórica, quedarán sentadas las bases para la aplicación efectiva en el reconocimiento de patrones, la traducción automática a partir de una especificación formal, estudio de los modelos de los modelos de computación lineal, e introducirnos en los modelos paralelos.



Formas Normal de Chomsky (FNC)

Las reglas de producción pueden adoptar las siguientes formas:

$$A \rightarrow BC$$

$$A \rightarrow a$$

$$S \rightarrow \lambda$$

Siendo S - Axioma Inicial

Donde: $A, B, C \in \Sigma_N$ y $a \in \Sigma_T$

Formas Normal de Chomsky (FNC)

Ventaja: Todos los árboles de derivación son binarios y favorecen la etapa de generación de Código Intermedio. Siempre es posible pasar a un código de 3 direcciones.

Desventaja: Deja la posibilidad de permitir recursiones por la izquierda en uno o más pasos, que es un efecto no deseado en una gramática para la implementación de algoritmos.

Formas Normal de Greibach (FNG)

Las reglas de producción pueden adoptar las siguientes formas:

$$A \rightarrow a\eta$$

$$S \rightarrow \lambda$$

Siendo S - Axioma Inicial

Donde: $A \in \Sigma_N$ y $a \in \Sigma_T$, $\eta \in \Sigma_N^+$

Formas Normal de Greibach (FNG)

Ventaja: Resulta un representación sumamente útil para realizar el proceso de conversión de una GFL a un autómata a Pila, que se derive en un analizador sintáctico LL.

Desventaja: El árbol sintáctico resultante, puede no ser un árbol binario lo que en la Generación del Código Intermedio necesario de un proceso más laborioso.

Gramáticas Independientes del Contexto

Tipo 2, Gramáticas Independientes del Contexto

$$A \rightarrow \alpha$$

donde: $A \in \Sigma_N$
 $\alpha \in (\Sigma_T \cup \Sigma_N)^+$

Para Izquierda: Solo un Símbolo No Terminal
 Para Derecha: Sin Restricciones

Con una derivación es posible generar cadenas vacías

También es común definir con reglas 1 o permitidos sólo desde el axioma inicial

Definición de Gramáticas Gramática Independiente del Contexto

Dada la siguiente CGL = (E, Σ_N , S, P)

$$E = \{E\}$$

$$S = \{S\}$$

$$\Sigma_N = \{A, B, C, D, E\}$$

$$S \rightarrow AxB$$

$$P = \{S \rightarrow aBb, B \rightarrow aB, B \rightarrow \lambda, C \rightarrow aB, D \rightarrow \lambda, E \rightarrow \lambda\}$$

Clasificación de Gramáticas

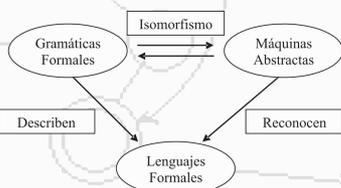
TIPOS DE GRAMATICAS	Formato de las Reglas de producción	Significado de Símbolos
Tipo 3 - Regulares Lineal por Derecha	$A \rightarrow aB$ $A \rightarrow \lambda$ $S \rightarrow \lambda$	$A, B \in ? N$ $a, \epsilon \in ? T$ $S \rightarrow$ Axioma Inicial
Tipo 3 - Regulares Lineal por Izquierda	$A \rightarrow Ba$ $A \rightarrow \lambda$ $S \rightarrow \lambda$	$A, B \in ? N$ $a, \epsilon \in ? T$ $S \rightarrow$ Axioma Inicial
Tipo 2 - Independientes de Contexto	$A \rightarrow \alpha$ $S \rightarrow \lambda$	$A \in ? N$ $\alpha \in (? T \cup ? N)^+$ $S \rightarrow$ Axioma Inicial
Tipo 2 - Independientes de Contexto Forma Normal de Chomsky	$A \rightarrow BC$ $A \rightarrow a$ $S \rightarrow \lambda$	$A, B, C \in \Sigma_N$ $a \in \Sigma_T$ $S \rightarrow$ Axioma Inicial
Tipo 2 - Independientes de Contexto Forma Normal de Greibach	$A \rightarrow a\eta$ $S \rightarrow \lambda$	$A, S \in \Sigma_N$ $a \in \Sigma_T$ $\eta \in \Sigma_N^+$ $S \rightarrow$ Axioma Inicial
Tipo 1 - Dependientes del Contexto	$\eta A \beta \rightarrow \eta B \beta$ $S \rightarrow \lambda$	$A, B \in ? N$ $\eta, \beta \in (? T \cup ? N)^*$ $S \rightarrow$ Axioma Inicial
Tipo 0 - Estructuradas por frase	$\eta \rightarrow \beta$	$\eta \in (? T \cup ? N)^*$ $\beta \in (? T \cup ? N)^*$

Formato de reglas de producción

TIPOS DE GRAMATICAS	Reglas de Producción	
	Parte Izquierda	Parte derecha
Tipo 3	Solo un Símbolo No Terminal	Podrá contener un terminal seguido de un no terminal o No terminal seguido de un terminal ya sea por derecha o por izquierda, de un terminal solamente, o producir λ , sólo si el no terminal de la izquierda es el axioma inicial de la gramática.
Regulares		Sin Restricciones, excepto que sólo podrá producir λ , si a la izquierda está el axioma inicial.
Tipo 2	Solo un Símbolo No Terminal	Sin Restricciones, excepto que sólo podrá producir λ , si a la izquierda está el axioma inicial.
Independientes del contexto		Sin Restricciones, excepto que sólo podrá producir λ , si a la izquierda está el axioma inicial.
Tipo 1	Sin Restricciones, solo debe contener algún símbolo No Terminal a ser reemplazado.	Sin Restricciones, excepto que sólo podrá producir λ , si a la izquierda está el axioma inicial.
Dependientes del Contexto		Sin Restricciones, y admite reglas compuestas
Tipo 0	Sin Restricciones, solo debe contener algún símbolo No Terminal a ser reemplazado.	Sin Restricciones, y admite reglas compuestas
Estructuradas por frase		Sin Restricciones, y admite reglas compuestas

RELACIONES

Lenguajes Gramáticas Maquinas Abstractas



Integrantes:

- Ing. Marciszack, Marcelo Martín
- Ing. Vazquez, Juan Carlos Jesús
- Ing. Meloni, Brenda Elizabeth
- Ing. Paz Menvielle, María Alejandra
- Constable, Leticia Edith Serrano, Diego Javier Pérez, Ramiro

AGRADECIMIENTOS

Agradecemos a la Agencia Córdoba Ciencia el subsidio al proyecto y a las autoridades de la Facultad Córdoba de la UTN por su apoyo en el mismo.

VIII - WICC 2006
Workshop de Investigadores en Ciencias de la Computación