

Universidad Simón Bolívar
Dpto. de Computación y Tecnología de la Información
CI3721 - Traductores e Interpretadores
Abril-Julio 2006

Proyecto Unico - Parte 3

Esta tercera etapa del proyecto corresponde al módulo de análisis de contexto e interpretador de Máquinas de Turing [1, Sección 7.2] descrito en el enunciado de la Parte 1 y continuado en la Parte 2. La implementación debe separar el análisis de contexto del interpretador en módulos, completando el módulo de análisis sintáctico con las acciones apropiadas, e implementando un nuevo módulo con la funcionalidad necesaria para interpretar las Máquinas de Turing.

La implantación se hará asociando acciones a las producciones de la gramática para:

- Construir una lista de definiciones de Máquinas de Turing presentes en el archivo de entrada.
- Construir una lista de definiciones de cintas de entrada presentes en el archivo de entrada.
- Construir un árbol abstracto del “programa principal” presente en el archivo de entrada, compuesto por las especificaciones de ejecución.

Estas estructuras luego serán utilizadas por un conjunto de funciones auxiliares para realizar la interpretación de las Máquinas de Turing, produciendo en pantalla las salidas descritas en el enunciado de la Parte 1.

Implantación del Análisis de Contexto

Ud. debe completar el analizador sintáctico de MTD construido durante la Parte 2, agregando acciones a las reglas gramaticales con el propósito de construir una estructura de datos que represente las máquinas, cintas y ejecuciones, de manera que puedan ser consultadas y recorridas por el módulo de interpretación de Máquinas de Turing¹. Esta estructura de datos, también llamada Arbol Abstracto, ha de ser definida por Ud. con varios TADs Haskell asociados a los símbolos no-terminales de interés, y aprovechando las acciones asociadas a las reglas gramaticales para su construcción durante el análisis sintáctico.

El módulo de interpretación de Máquinas de Turing contendrá funciones que empleen el Arbol Abstracto para exhibir el comportamiento apropiado para las máquinas. Note que la implantación de este módulo puede hacerse de manera independiente al analizador sintáctico, simplemente con definir un TAD para la Máquina de Turing e implementar la semántica de operación de las mismas. Luego se integra el tipo de datos con el analizador sintáctico.

Durante la construcción de la estructura de datos, tanto como en su posterior interpretación, Ud. debe agregar verificaciones de consistencia para detectar errores de contexto que impidan la interpretación. Estos errores deben ser reportados de forma clara al usuario, posiblemente indicando la posición en el archivo de entrada. Las condiciones erróneas a detectar son aquellas derivadas de la definición formal de la Máquina de Turing, por ejemplo:

¹Pista: diseñe una estructura de datos “principal” que pueda ser consultada con la función `lookup` de Haskell.

- Errores estructurales en la definición de una máquina.
 - Los conjuntos **deben** ser conjuntos, i.e. no puede haber elementos repetidos.
 - El símbolo para el blanco **debe** pertenecer al conjunto lambda.
 - Sigma **debe** ser subconjunto de lambda y **no** puede contener al símbolo del blanco.
 - El estado inicial debe pertenecer al conjunto de estados.
 - El conjunto de estados finales debe ser un subconjunto del conjunto de estados.
 - El dominio y el rango de delta debe estar contenido en la definición de la máquina.
 - Las máquinas deben ser determinísticas².
- Errores de contexto.
 - No puede haber varias máquinas con el mismo nombre. Recuerde que el lenguaje **es** sensible a mayúsculas y minúsculas.
 - No puede haber varias cintas con el mismo nombre. Recuerde que el lenguaje **es** sensible a mayúsculas y minúsculas.
- Errores de ejecución.
 - No se puede ejecutar una máquina que no ha sido definida.
 - No se puede utilizar una cinta que no ha sido definida.
 - Debe interrumpirse la ejecución si la cinta contiene un caracter que no está en el alfabeto de entrada de la máquina.
 - Debe interrumpirse la ejecución si la máquina “se cae por la izquierda”.

El analizador sintáctico a ser completado en esta etapa se mantiene en el módulo Haskell de nombre **Parser**, sólo que ahora debe exportar **únicamente** la función **parser** generada por Happy y las definiciones de TADs diseñadas por Ud.. Utilizará el analizador lexicográfico desarrollado en la Parte 1, el cual deberá estar encapsulado en un módulo **Lexer** separado que exporte **únicamente** el tipo de datos de los *tokens* y la función **lexer**. En un nuevo módulo Haskell de nombre **Turing**, encapsulará las funciones y TADs que considere necesario implementar para interpretar las Máquinas de Turing; es válido importar los TADs de Máquinas de Turing dentro del Parser si considera que esto facilita el trabajo. La función **mtd** que recibirá como argumento una cadena alfanumérica indicando el nombre del archivo de texto a procesar, y que producirá una cadena con las producciones utilizadas, debe encapsularse en un módulo Haskell de nombre **Main** que importa los módulos **Lexer**, **Parser** y **Turing** para hacer su trabajo.

Revisión Teórico-Práctica

Los detalles de la Revisión Teórico-Práctica serán publicados el Jueves de la Semana 8.

²Pista: ¿Qué condición deben cumplir las transiciones?

Detalles de la Entrega

- **Implantación.** Ud. debe entregar en el formato electrónico acordado, en clase personalmente o por correo electrónico, los archivos `Lexer.x` con el código para Alex y la definición del TAD para Tokens, `Parser.y` con el código para Happy, `Turing.hs` y `Main.hs` con el código para el programa principal. Los módulos deben incluir documentación Haddock para describir los detalles de implementación suficientes.
- **Revisión Teórico-Práctica.** Ud. debe entregar en papel o en PDF el desarrollo de las preguntas.
- **Fecha de Entrega.** Jueves 13 de Julio (Semana 12) hasta las 9:30am
- **Valor de Evaluación.** Catorce (14) puntos.

References

- [1] J. HOPCROFT, J. ULLMAN
Introduction to Automata Theory, Languages and Computation
Addison-Wesley Publishing Company, 1979
ISBN 0-201-02988-X QA267.H56