Taller 2: Abstracción de datos y sintáxis abstracta

Fundamentos de Lenguajes de Programación / 750095M / Grupo 01 / Prof. Robinson Duque / Monitora Monitora Sara Jazmín Maradiago Calderón / 2023-I

Los indicadores de logro que se abordarán en este taller están determinados por:

- Implementa Tipos Abstractos de Datos utilizando dos componentes: una interfaz y una implementacón
- Utiliza estrategias con listas y datatypes para representar tipos de dato definidos por una gramática
- Construye un árbol de sintaxis abstracta a partir de una representación concreta en una gramática BNF
- Implementa procedimientos Parse y Unparse para derivar entre representación concreta y abstracta de una gramática BNF
- Utiliza herramientas para especificar analizadores léxicos y sintacticos de una gramática BNF

1 Instancias SAT

El Problema de Satisfactibilidad Booleana (SAT) consiste en un conjunto V de n variables booleanas $v_1, v_2, v_3 ... v_n$ y un conjunto C de m clausulas $c_1, c_2, c_3 ... c_m$ en **forma normal conjuntiva (FNC)**. Por ejemplo:

$$C = (x \vee \neg y \vee z \vee w) \wedge (\neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (z \vee w) \wedge y$$

C en este caso es una instanica del problema y la idea es responder lo siguiente: ¿Existe una asignación de valores para (x, y, z, w), donde C sea verdadera?

En la actualidad existen competencias `http://www.satcompetition.org/` donde se desarrollan solvers especializados que intentan resolver este tipo de problemas. Sin embargo, usualmente estos solvers requieren de un manejo avanzado de técnicas basadas en inferencia lógica para su solución.

En nuestro caso, para el problema C realizaremos una exploración ingenua del las posbiles combinaciones de valores de verdad para las variables (x, y, z, w) hasta lograr encontrar una solución o hasta explorar todas las posibles combinaciones y demostrar que el problema no tiene solución.

Por ejemplo, podriamos intentar primero con $(\#t, \#t, \#t, \#t)$ y nos daremos cuenta que C no se puede satisfacer debido a la clausula $(\neg x \vee \neg y \vee \neg z)$ resulta ser falsa. Podriamos intentar luego con $(\#t, \#t, \#t, \#f)$, y luego con $(\#t, \#t, \#f, \#t)$, y luego con $(\#t, \#t, \#f, \#f)$ hasta llegar a una posible solución. Por ejemplo, una de las soluciones se logra con $(\#f, \#t, \#t, \#t)$ y se puede parar la búsqueda porque se ha encontrado una solución (sin embargo, es posible que existan otras). Así pues, se dice que la instancia C es SATISFACTIBLE para (x, y, z, w) con valores $(\#f, \#t, \#t, \#t)$

Veamos otro ejemplo:

$$C = (x \vee y) \wedge (\neg x) \wedge (\neg y)$$

Observe que después de explorar todas las posibles combinaciones para (x, y) con valores $(\#t, \#t)$, $(\#t, \#f)$, $(\#f, \#t)$, $(\#f, \#f)$ no se logra satisfacer C. Así pues, se dice que C es INSATISFACTIBLE.

1.1 (30pts) Gramática BNF

- Implemente una gramática que permita escribir expresiones en FNC. Una forma usual de escribir estas instancias es utilizando un formato numérico que represente el problema, pero si usted lo desea, **puede personalizar la forma como desea escribir las instancias SAT**. Entonces la instanica:

$$C = (x \vee \neg y \vee z \vee w) \wedge (\neg y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (z \vee w) \wedge y$$

puede ser representada así:

```
FNC 4 ( (1 or −2 or 3 or 4) and
        (−2 or 3) and
        (−1 or −2 or −3) and
        (3 or 4) and
        (2) )
```

Donde FNC indica que la expresión es una FNC, el 4 que le sigue indica que hay 4 variables y luego hay una construcción de clausulas con números de 1 a 4 donde: $x = 1, \neg x = -1, y = 2, \neg y = -2, z = 3, \neg z = -3, w = 4, \neg w = -4$.

  - (15 Pts) Proponga una implementación de la gramática basada en listas: Esta implementación deberá contener los respectivos constructores (fnc, and, or) y extractores ($fnc- > var, fnc- > clausulas, or- > varlist$), o los que considere en su gramática. Incluya ejemplos donde se evidencie su utilización y la creación de por lo menos 3 instancias SAT.
  - (15 Pts) Proponga una implementación basada en datatypes. Incluya ejemplos donde se evidencie su utilización y la creación de por lo menos 3 instancias SAT.

2 (40 pts) Funciones Parse y Unparse

  - (20pts) Para la representación basada en listas, construya una función PARSEBNF donde dada una lista con la representación concreta de una instancia SAT, construya el árbol de sintáxis abstracta basado en listas.

– (20pts) Para la representación basada en listas, construya una función UNPARSEBNF donde dado un árbol de sintáxis abstracta de una instancia SAT, entregue la representación concreta basada en listas.

3 (30 pts) Evaluación de Instancias SAT

- Implemente una función llamada EVALUARSAT que reciba una instancia FNC (basada en listas o datatypes), y la evalúe así:

  Ejemplo 1:

  FNC 4 ((1 or −2 or 3 or 4) and (−2 or 3) and
  (−1 or −2 or −3) and (3 or 4) and (2))

  % Respuesta:
  (satisfactible (#f #t #t #t))

  Ejemplo 2:

  FNC 2 ((1 or 2) and (−1) and (−2))

  % Respuesta:
  (insatisfactible '())

Aclaraciones

1. El taller es en grupos de mínimo dos (2) alumnos, máximo tres (3).

2. Se debe subir al campus virtual en el enlace correspondiente a este taller un archivo comprimido **.zip** que siga la convención *CódigodeEstudiante1-CódigodeEstudiante2-CódigodeEstudiante3...-Taller2FLP.zip*.

   Este archivo debe contener los archivos: **ejercicio1.rkt, ejercicio2.rkt, ejercicio3.rkt** que contengan el desarrollo de los ejercicios.

3. Para todos los tipos de datos se debe **incluir la gramática que utilizó**.

4. En las primeras líneas de cada archivo deben estar comentados los nombres y los códigos de los estudiantes participantes.

5. Se debe incluir para cada procedimiento un comentario que explique lo que realiza cada función y para qué es empleada. También deben documentar los procedimientos que hayan implementado como solución a los problemas y de igual manera las funciones auxiliares que se utilicen, con ejemplos de prueba (mínimo 2 pruebas).

Entregas Tardías o por Otros Medios

1. Este taller **sólo se recibirá a través del campus virtual**. Adicionalmente, sólo se evaluarán los documentos solicitados en el punto 2 de la sección anterior. Cualquier otro tipo de correo o nota aclaratoria será descartado.

2. Las entregas tarde serán penalizadas así: (-1pt de la nota final obtenida) por cada hora de retraso o fracción. Por ejemplo, si usted realiza su entrega y el campus registra las 24:00 (i.e., 1min después de la hora de entrega), usted está incurriendo en la primer hora de retraso. Asegurese con mínimo dos horas de anticipación que el link de carga funciona correctamente toda vez que es posible incurrir en una entrega tardía debido a los tiempos de respuesta.