

Programación Funcional. Tarea 1

Instituto de Computación

Abril 2006

Índice

1. Presentación	1
2. Fórmulas	1
2.1. Ejemplos de fórmulas	2
3. Asignaciones	2
4. Comandos	3
4.1. Comandos <code>fnc</code> y <code>fnd</code>	3
4.2. Comando <code>evaluar</code>	3
4.3. Comando <code>tabla</code>	3
5. Se pide	4
6. Sugerencias	4
7. Apéndice	5
7.1. Definición de formas normales	5

1. Presentación

Se desea implementar un sistema para operar con **fórmulas** de la *lógica proposicional*. El mismo debe disponer de las siguientes facilidades:

- Operar con fórmulas con variables proposicionales y los conectivos de conjunción, disyunción, implicancia y negación.
- Operar con asignaciones de verdad
- Formas normales: Transformación de una fórmula a forma normal disyuntiva y conjuntiva

2. Fórmulas

Las fórmulas atómicas son las *variables proposicionales* que serán representadas por letras minúsculas.

Una fórmula no atómica se construye a partir de las variables proposicionales utilizando los siguientes conectivos:

- Negación: \sim
- Conjunción: $\&$
- Disyunción: $|$
- Implicancia: \rightarrow
- Doble implicancia: \leftrightarrow

La precedencia de los conectivos es la siguiente:

1. \sim
2. $\&$, $|$
3. \rightarrow , \leftrightarrow

Los conectivos $\&$ y $|$ asocian a la izquierda. Los conectivos \rightarrow y \leftrightarrow asocian a la derecha.

2.1. Ejemplos de fórmulas

```
a -> ~~ a
b
(~(a | b) ) <-> ( b -> c &~d | a)
```

Pueden aparecer 0 o más espacios separando los diferentes componentes sintácticos de la expresión.

3. Asignaciones

Una asignación de valores de verdad (*asignación* en lo que sigue) es una función que asocia Verdadero o Falso a las variables proposicionales.

En este sistema una asignación se puede representar por alguna de las siguientes maneras:

- a. Dos lista de variables entre llaves, separadas por una barra vertical:

```
{a, b, c, z | f, g}
```

- b. Una variable de asignación representada con una letra mayúscula.
- c. Una variable “corregida” con una lista de asignaciones:

```
A[b=V,d=F]
```

En el caso a) la primera lista son las variables que tienen asociado el valor verdadero, y la segunda las variables que tienen asociado el valor falso. Puede haber variables repetidas. Si una variable aparece en las dos listas se considera que su valor es verdadero.

En el caso b) se utiliza un nombre simbólico para representar una asignación. Este nombre debe haber sido definido mediante un comando `asignar` (ver comandos)

En el caso c) la asignación coincide con `A` salvo en los valores especificados entre corchetes. Si se repiten variables se toma el último valor (el que aparece más a la derecha)

4. Comandos

El sistema es un intérprete interactivo de comandos: lee un comando de la entrada estándar, ejecuta el comando y despliega el resultado en la salida estándar.

Los comandos que acepta el sistema son los siguientes:

- `fnc` *fórmula*
- `fnd` *fórmula*
- `asignar` *variable asignación*
- `evaluar` *asignación fórmula*
- `tabla` *fomula*
- `salir`

4.1. Comandos `fnc` y `fnd`

Los comandos `fnc` (`fnd`) despliegan una fórmula en forma normal conjuntiva (disyuntiva) equivalente a la fórmula dada.

Ejemplo:

```
> fnc ~ (~ (a <-> b) | ~ (a & ~ c))
( ~ a | b) & ( ~ b | a) & a & ~ c
```

El comando `asignar` asocia un nombre de variable que debe ser una letra mayúscula con una asignación de valores de verdad. Este comando despliega en la salida el resultado de la asignación:

```
> asignar K {a,b,c| d, f}
V = { a=V, b=V, c=V, d=F, f=F}
> asignar U K
V = { a=V, b=V, c=V, d=F, f=F}
> asignar S K[b=F,c=V,f=V]
S = { a=V, b=F, c=V, d=F, f=V}
```

4.2. Comando `evaluar`

El comando `evaluar` despliega el valor de verdad que toma una fórmula bajo la asignación dada. Si alguna de las variables de la fórmula no aparece en la asignación, el resultado es indefinido.

```
> evaluar K (a -> (~ c | a -> b))
Verdadero
> evaluar {a,c|} (a -> (~ c | a -> b))
Indefinido
```

4.3. Comando `tabla`

El comando `tabla` despliega la tabla de verdad de la fórmula dada

```

> tabla  a | (~b -> ~c) & (a | ~c)
V,V,V   V
V,V,F   V
V,F,V   V
V,F,F   V
F,V,V   F
F,V,F   V
F,F,V   F
F,F,F   V

```

Los valores de las variables en cada asignación se ordenan alfabéticamente. Por ejemplo la asignación V, F, V corresponde a $\{a=V, b=F, C=V\}$

El comando **salir** termina la ejecución del programa.

Todos los comandos se ingresan en una sola línea. Si la sintaxis de un comando no es correcta o la ejecución del comando produce un error, el sistema deberá desplegar un mensaje que comience con la palabra `ERROR`.

Cuando el sistema espera el ingreso de un comando debe desplegar al comienzo de la línea el carácter `'>'` seguido de un espacio.

Nota: Es importante que su tarea siga **exactamente el mismo formato** que se presenta en estos ejemplos; de lo contrario puede tener problemas con la corrección automática.

5. Se pide

Escribir un programa *haskell* que implemente un sistema como se describe más arriba.

El programa deberá estar formado por un conjunto de módulos, uno de los cuales se debe llamar `Main.hs`. La primera línea de este módulo debe ser de esta forma:

```
module Main(main) where
```

Este módulo debe contener una función de nombre `main` con el siguiente tipo:

```
main :: IO ()
```

Este función es la que implementa el intérprete de comandos.

6. Sugerencias

- Definir tipos para representar las fórmulas y las asignaciones.
- Implementar *parsers* para las fórmulas y asignaciones.
- Definir un tipo para los comandos. Implementar un parser y un intérprete para los comandos.
- Utilizar para implementar las funciones de *parsing*, los combinadores definidos en la **clase 10** y en la sección 17.5 del libro.
- Para repasar algunos conceptos sobre lógica proposicional recomendamos consultar:
 - *Logic and Structure*. Dirk van Dalen. Ed. Springer-Verlag.
 - Lógica proposicional. http://en.wikipedia.org/wiki/Propositional_calculus
 - Tablas de verdad. http://en.wikipedia.org/wiki/Truth_table
 - Forma normal conjuntiva. http://en.wikipedia.org/wiki/Conjunctive_normal_form

7. Apéndice

7.1. Definición de formas normales

La forma normal conjuntiva se define por la siguiente gramática (EBNF)

```
<literal> ::= [~] <variable> .  
<disyuncion> ::= '(' <literal> { '|' <literal> } ')'  
<fnc> ::= <disyuncion> { '&' <disyuncion> } .  
  
<conjuncion> ::= '(' <literal> { '&' <literal> } ')'  
<fnc> ::= <conjuncion> { '|' <conjuncion> } .
```

La forma normal disyuntiva se define análogamente intercambiando conjunción y disyunción.

Notación:

- '...' literal
- [...] opcional
- { ... } repetición (0 o más veces)
- < ... > no terminal