Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ciencias Exactas y Naturales
Departamento de Computación
Estructuras de Datos y Algoritmos I

Trabajo Práctico - Mesas Julio/Agosto

1. Motivación

El objetivo de este trabajo práctico es la implementación de un intérprete para operar con conjuntos numéricos. Entre las operaciones que se deberán soportar se encuentran: unión, intersección, resta y complemento.
El usuario podrá definir conjuntos de números enteros, ya sea por extensión o por compresión, y asociarles alias. Por ejemplo, si un usuario ingresa el comando

$$A = \{1, 2, 3\}$$

el intérprete guardará el conjunto $\{1, 2, 3\}$ en memoria (utilizando alguna estructura de datos conveniente) y lo asociará con el alias A. A partir de ese momento y cada vez que el usuario utilice dicho alias en algún comando, el intérprete deberá reemplazarlo por su respectiva definición.
A continuación presentamos con mayor detalle los comandos que deberán ser soportados por el intérprete.

2. Intérprete

Utilizaremos la palabra `alias` para denotar una cadena arbitraria y k para denotar un número entero arbitrario.

- **Creación de un conjunto definido por extensión:**

  `alias = {`k_0`,`k_1`,...,`k_n`}`

  Este comando crea el conjunto de enteros $\{k_0, k_1, \ldots, k_n\}$ y lo asocia con `alias`. Ejemplos de uso:

  ```
  A = {}
  A1 = {0}
  A2 = {-10,4,6,-7}
  ```

- **Creación de un conjunto definido por compresión:**

  `alias = {x:` k_0 `<= x <=` k_1`}`

  Este comando crea el conjunto $\{x : k_0 \leq x \leq k_1\}$ y lo asocia con `alias`. La sintaxis debe por lo menos aceptar el carácter `x` para describir al conjunto, no obstante también se aceptarán sintaxis más amplias donde adicionalmente se permitan otros nombres de variables. Ejemplos de uso:

  ```
  A = {x : -2 <= x <= 5}
  A1 = {x : 10 <= x <= 2}
  ```

En este caso, `A1` denotará un conjunto vacío.

- **Unión de dos conjuntos:**

  `alias1 = alias2 | alias3`

  Este comando computa la unión de los conjuntos asociados con `alias2` y `alias3`, y asocia el conjunto resultante con `alias1`. Ejemplos de uso:

  ```
  A = {x : -2 <= x <= 5}
  A1 = {2}
  A = A | A1
  ```

  ```
  B = {x : -2 <= x <= 5}
  B1 = {-10,4,6,-7}
  B2 = B | B1
  ```

  Notar que los alias se pueden sobrescribir como se muestra en el primer ejemplo de uso.

- **Intersección de dos conjuntos:**

  `alias1 = alias2 & alias3`

  Este comando computa la intersección de los conjuntos asociados con `alias2` y `alias3`, y asocia el conjunto resultante con `alias1`. Ejemplos de uso:

  ```
  A = {x : -2 <= x <= 5}
  A1 = {2}
  A = A & A1
  ```

  ```
  B = {x : -2 <= x <= 5}
  B1 = {-10,4,6,-7}
  B2 = B & B1
  ```

- **Resta de dos conjuntos:**

  `alias1 = alias2 - alias3`

  Este comando computa la resta de los conjuntos asociados con `alias2` y `alias3`, y asocia el conjunto resultante con `alias1`. Ejemplos de uso:

  ```
  A = {x : -2 <= x <= 5}
  A1 = {2}
  A = A - A1
  ```

  ```
  B = {x : -2 <= x <= 5}
  B1 = {-10,4,6,-7}
  B2 = B - B1
  ```

- **Complemento de un conjunto**

  `alias1 = ~alias2`

Este comando computa el complemento del conjunto asociado con `alias2`, y asocia el conjunto resultante con `alias1`. Ejemplos de uso:

```
A = {x : -2 <= x <= 5}
A = ∼A
```

```
B = {-10,4,6,-7}
B1 = ∼B1
```

El universo estará delimitado por el mínimo y el máximo entero representable por el tipo de datos `int`. En el primer ejemplo de uso, `A` deberá quedar asociado con el conjunto $\{x : INT\_MIN \leq x \leq -1\} \cup \{x : 6 \leq x \leq INT\_MAX\}$.

- **Imprimir los elementos de un conjunto:**

  ```
  imprimir alias
  ```

  Este comando imprime los elementos del conjunto asociado con `alias`, respetando el orden de los números enteros. Ejemplo de uso:

  ```
  A = {x : -2 <= x <= 5}
  imprimir A
  -2:5
  B = {-10,4,6,-7}
  imprimir A
  -10,-7,4,6
  C = A | B
  imprimir C
  -10,-7,-2:5,6
  ```

  Observar que se utiliza la notación "k_0:k_1" para denotar a los elementos del intervalo cerrado $[k_0, k_1]$.

  El resultado de imprimir que se muestra en los ejemplos es meramente ilustrativo y dependerá de la implementación de las operaciones. En este sentido, resaltamos que pueden existir distintas formas de imprimir un mismo conjunto, también sería válido:

  ```
  imprimir C
  -10,-7,-2:6
  ```

- **Cerrar el intérprete:**

  ```
  salir
  ```

  Este comando cierra el intérprete, liberando adecuadamente los recursos solicitados.

3. Consigna

a) Implementar el programa intérprete según lo consignado en la sección anterior. Para ello será necesario definir estructuras de datos adecuadas para representar los conjuntos y la asociación de los alias con sus respectivas definiciones.

Requisito: Los conjuntos definidos por compresión no deben ser expandidos innecesariamente. Es decir, no se deben mantener en memoria todos sus elementos, sino solamente los extremos del intervalo cerrado que lo define.

b) Cumplir con las **convenciones de código** y las **convenciones de proyecto** elaboradas por la cátedra en el sitio de Comunidades.

c) Confeccionar un informe detallando:

- Estructuras de datos utilizadas.
- Formatos aceptados para los alias.
- Mensajes de error devueltos por el intérprete.
- Cómo se compila y usa el programa.
- Dificultades encontradas y cómo fueron resueltas.
- Bibliografía utilizada.
- Otra información relevante para la comprensión del trabajo.