Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC1253 - Matemáticas Discretas

Tarea 2

26 de agosto de 2024

2^{o} semestre 2024 - Profesores P. Bahamondes - D. Bustamante - M. Romero

Requisitos

- La tarea es individual. Los casos de copia serán sancionados con la reprobación del curso con nota 1,1.
- **Entrega**: Hasta las 23:59 del 04 de septiembre a través del buzón habilitado en el sitio del curso (Canvas).
  - Esta tarea debe ser hecha completamente en LaTeX. Tareas hechas a mano o en otro procesador de texto **no serán corregidas**.
  - Debe usar el template LaTeX publicado en la página del curso.
  - Cada solución de cada problema debe comenzar en una nueva hoja. ***Hint:*** Utilice `\newpage`
  - Los archivos que debe entregar son el archivo `PDF` correspondiente a su solución con nombre `numalumno.pdf`, junto con un `zip` con nombre `numalumno.zip`, conteniendo el archivo `numalumno.tex` que compila su tarea. Si su código hace referencia a otros archivos, debe incluirlos también. En caso de entregar el bonus de código, este debe incluirse por separado en único archivo con nombre `numalumno.py`.
- El no cumplimiento de alguna de las reglas se penalizará con un descuento de 0.5 en la nota final (acumulables).
- La nota final de la tarea es la nota obtenida sin el bonus opcional (e) de la Pregunta 1, más 1 punto, en caso de haber entregado el bonus y tenerlo correcto. En caso de obtener una nota mayor a 7.0, **no** se acumularán décimas.
- No se aceptarán tareas atrasadas (salvo que utilice su cupón `#problemaexcepcional`).
- Si tiene alguna duda, el foro de Github (issues) es el lugar oficial para realizarla.

Pregunta 1

Sea Σ un conjunto finito no vacío de símbolos. Se define el conjunto Σ^* de las palabras sobre Σ, como el menor conjunto que cumple las siguientes reglas:

- $\varepsilon \in \Sigma^*$, donde ε corresponde a la palabra vacía que no contiene símbolos.
- Si $w \in \Sigma^*$ y $a \in \Sigma$, entonces $wa \in \Sigma^*$.

A modo de ejemplo, si $\Sigma = \{a, b, c\}$, algunas posibles palabras en Σ^* serían:

$$\varepsilon \qquad b \qquad acc \qquad cbabb$$

Notar que por comodidad, omitimos la palabra vacía ε cuando no es necesaria, es decir, escribimos a, acc y $cbabb$, en vez de εa, εacc y $\varepsilon cbabb$, respectivamente.

(a) Sea $a \in \Sigma$. Defina inductivamente la función $\#_a : \Sigma^* \to \mathbb{N}$, que a cada palabra le asigna la cantidad de ocurrencias del símbolo a en la palabra. Algunos ejemplos:

$$\#_a(b) = 0 \qquad \#_b(b) = 1 \qquad \#_c(acc) = 2 \qquad \#_b(cbabb) = 3$$

(b) Sean $a, b \in \Sigma$. Defina inductivamente la función $r_{a\to b} : \Sigma^* \to \Sigma^*$, que a cada palabra le asigna la palabra resultante de reemplazar cada símbolo a por el símbolo b. Algunos ejemplos:

$$r_{c\to b}(acc) = abb \qquad r_{b\to a}(acc) = acc \qquad r_{b\to a}(cbabb) = caaaa \qquad r_{c\to c}(cbabb) = cbabb$$

(c) Sean $a, b \in \Sigma$ tal que $a \neq b$. Demuestre usando inducción estructural que para toda palabra $w \in \Sigma^*$ se cumple:

$$\#_b(r_{a\to b}(w)) = \#_a(w) + \#_b(w)$$

(d) Sean $a, b \in \Sigma$ tal que $a \neq b$. Demuestre usando inducción estructural que para toda palabra $w \in \Sigma^*$ se cumple:

$$r_{b\to a}(r_{a\to b}(w)) = w \implies \#_b(w) = 0$$

La siguiente pregunta es opcional y ofrece un bonus de 1 punto sobre la nota final de la tarea (ver instrucciones en la primera página).

(e) Utilizando su definición inductiva de la función $r_{a \to b}$, escriba en `Python` un programa recursivo `reemplazar_simbolo(w, a, b)`, que dada una palabra w, y dos símbolos a y b, retorna la palabra correspondiente a $r_{a \to b}(w)$.

Importante: No puede usar ninguna función de `Python` que haga el reemplazo directamente. Debe respetar la definición inductiva de Σ^* y la definición inductiva de $r_{a \to b}$.

Condiciones de entrega: Un único archivo de nombre `numero_alumno.py` tal que su solución se pruebe haciendo un llamado a la función `reemplazar_simbolo(w, a, b)` en dicho archivo.

Ejemplos de ejecución:

`reemplazar_simbolo('acc', 'c', 'b')`	retorna `'abb'`
`reemplazar_simbolo('acc', 'b', 'a')`	retorna `'acc'`
`reemplazar_simbolo('cbabb', 'b', 'a')`	retorna `'caaaa'`
`reemplazar_simbolo('cbabb', 'c', 'c')`	retorna `'cbabb'`

Pregunta 2

Suponga que tenemos un conjunto $V = \{1, \dots, n\}$ de n personas. Algunas parejas de personas son compatibles y otras no. Estas relaciones de compatibilidad están descritas por un conjunto E de parejas de personas compatibles. Es decir, $(i, j) \in E$ si y sólo si las personas i y j son compatibles. Un *núcleo* para V y E es un subconjunto C de personas de V que son compatible entre ellas, es decir, tal que para todo par (i, j) de personas en C se cumple que $(i, j) \in E$.

A modo de ejemplo, suponga que nuestro conjunto de personas es $V = \{1, 2, 3, 4\}$ y nuestro conjunto de compatibilidades es $E = \{(1, 2), (2, 3), (1, 3), (1, 4)\}$. Tenemos que $C = \{1, 2, 3\}$ es un núcleo para V y E de 3 personas, ya que tenemos las compatibilidades $(1, 2)$, $(2, 3)$ y $(1, 3)$. Por otra parte, si escogemos $C = \{1, 3, 4\}$, no obtenemos un núcleo ya que 3 y 4 no son compatibles.

Dado un conjunto V de n personas, un conjunto de compatibilidades E y un parámetro $k \leq n$, nuestra misión es encontrar un núcleo con k personas para V y E. Por supuesto, queremos utilizar la lógica proposicional para resolver este problema. Escriba una fórmula en lógica proposicional φ tal que:

Existe un núcleo con k personas para V y E si y sólo si φ es satisfacible.

Debe demostrar que su fórmula φ es correcta, es decir, que cumple la propiedad enunciada arriba.

Para definir φ, **debe** utilizar las siguientes variables proposicionales:

- Variables p_{ij}, donde $1 \leq i, j \leq n$, que expresan que i y j son compatibles.
- Variables x_{hi} donde $1 \leq h \leq k$ e $1 \leq i \leq n$, que expresan que la persona i es la h-ésima persona del núcleo.