Álvarez Aldea, Alberto - NIP: 681687

• Ejercicios 12 de Diciembre de 2014

(1) Demostrar que el siguiente lenguaje es semidecidible:

$A = \{ p\#x \mid p$ es un programa y x es una cadena que cumplen que p con entrada x para en tiempo impar$\}$

Un lenguaje es semidecidible si es aceptado por un programa que no tiene por que parar siempre.
Así pues, construimos un programa para A, en el que si dados p y w, si el programa p con entrada w termina en tiempo impar devuelve "acepta" y en caso contrario devuelve rechaza o se cuelga.

Entrada p, w

```
boolean ha_parado := false;
cadena resultado := "";
entero T := 1;
mientrasQue NOT ha_parado hacer
        simular(p, w, T, ha_parado, resultado);
        T := T + 1;
fmQ
si (T-1) mod 2 = 1 entonces
        devuelve acepta;
sino devuelve rechaza.
fsi
```

2) Demostrar que el siguiente lenguaje es semidecidible:

$A = \{ p\#x \mid$ p es un programa y x es una cadena que cumple que p con entrada x para y devuelve como resultado un número múltiplo de 3 $\}$.

Un lenguaje es semidecidible si es aceptado por un lenguaje que no tiene porque parar siempre.

Así pues, construimos un programa para A, en el que si dados p y w, si el programa p con entrada w para y da un resultado múltiplo de 3 devuelve acepta, en caso contrario devuelve rechaza o se cuelga.

Entrada p, w

```
entero resultado := 1;
resultado = p(*)(w)
si resultado mod 3 = 0 entonces
        devuelve acepta;
sino devuelve rechaza;
fsi
```

(*) función p (ent c:cadena) devuelve entero.
→ Esta función simula el programa p.

③ Demostrar que si A y B son decidibles, entonces $A \cap B$ es decidible

- Si A es decidible tengo un programa, P_A, tal que dada una cadena x, si $x \in A$ devuelve "acepta" y si $x \notin A$ devuelve "rechaza".

- Si B es decidible tengo un programa P_B, tal que dada una cadena x, si $x \in B$ devuelve "acepta" y si $x \notin A$ devuelve "rechaza"

Para demostrar que la intersección de A y B es decidible podemos construir un programa que dada una cadena x, si $x \in A$ y $x \in B$ devuelve "acepta" y si $x \notin A$ o $x \notin B$ devuelve "rechaza".

Entrada x
```
cadena respuestaA := PA(x)
cadena respuestaB := PB(x)
si respuestaA = acepta AND respuestaB = acepta entonces
        devuelve acepta;
sino devuelve rechaza;
fsi
```

4 Demostrar que si A y B son semidecidibles entonces $A \cup B$ es semidecidible.

Si A es semidecidible tengo un programa, pA, tal que dada una cadena x, si $x \in A$ devuelve "acepta" y si $x \notin A$ devuelve "rechaza" o se cuelga.

Si B es semidecidible tengo un programa, PB, tal que dada una cadena x, si $x \in B$ devuelve "acepta" y si $x \notin B$ devuelve "rechaza" o se cuelga.

Para demostrar que la unión de A y B es semidecidible, construimos un programa que dada una cadena x, si $x \in A$ o $x \in B$ devuelve "acepta", y si $x \notin A$ y $x \notin B$ se cuelga.

<u>Entrada x</u>

```
entero T := 1;
booleano Fin := false, TerminA := false, TerminB := false;
cadena resA := "", resB := "";
mientrasQue NOT Fin
        Simular(pA, x, T, TerminA, resA);
        Simular(pA, x, T, TerminB, resB);
        si (TerminA AND resA = acepta) OR (TerminB AND resB = acepta) entonces
                Fin := true;
        sino T := T + 1;
        fsi
fmQ
devuelve acepta;
```

⑤ Demostrar que si A y B son semidecidibles entonces $A \cap B$ es semidecidible

Si A es semidecidible tengo un programa, p_A, tal que dada una cadena x, si $x \in A$ devuelve "acepta" y si $x \notin A$ devuelve "rechaza" o se cuelga.

Si B es semidecidible tengo un programa, p_B, tal que dada una cadena x, si $x \in B$ devuelve "acepta" y si $x \notin B$ devuelve "rechaza" o se cuelga

Para demostrar que la intersección de A y B es semidecidible, construimos un programa que dada una cadena x, si $x \in A$ y $x \in B$ devuelve "acepta" y si $x \notin A$ o $x \notin B$ devuelve "rechaza" o se cuelga

Entrada x

```
cadena resA := pA(x);
cadena resB := pB(x);
si resA = acepta AND resB = acepta entonces
    devuelve acepta;
sino devuelve rechaza;
fsi
```

Adrián Alcolea Moreno 15-12-14

(5) Como el anterior pero aceptando únicamente cuando acepten ambos. ¿cómo?

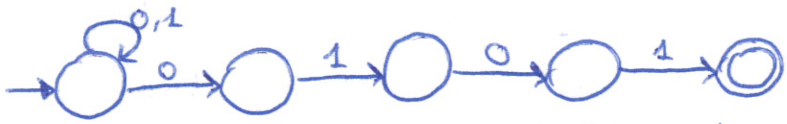
(6)

√ 1/ V → Un automata se puede simular con una máquina de Turing, y además siempre termina.

√ 2/ F → Por ejemplo $a^n b^n c^n$

√ 3/ V → El propio AFD es un caso particular de un AFnD

√ 4/ F → El AFnD que reconoce $\{w \mid w \text{ termina en } 0101\}$



se complicaría bastante al pasarlo a Automata Determinista.

¿seguro? ¿cómo sería el mínimo?

(7)

√ 1/ $\{(p,w) \mid p \text{ es un programa que para con } w\}$

√ 2/ Ya he puesto antes $a^n b^n c^n$ como ejemplo. Es fácil plantear una máquina de Turing que lo reconozca y siempre sepa si pertenece o no al lenguaje, sin colgarse en ninguna palabra, ya que recorrerá la entrada "contando" → un poco más de detalle

√ 3/ Un ejemplo típico es $a^n b^n$, y basta con apilar las aes y desapilarlas con las bes.

√

⑧ Mal justificado

1/ F → Si lo fuera, querría decir que existe un programa que siempre de respuesta para las palabras del complementario, por lo que existiría un programa que siempre diera respuesta a la pertenencia o no para el lenguaje inicial. Luego esto implicaría que el primero es decidible

¿cómo?

✓ 2/ V → Basta con hacer un programa que invoque al programa del lenguaje inicial y cambiar la salida (invertirla).

✓ 3/ V → Es fácil de ver transformando el autómata para que los estados de no aceptación pasen a ser de aceptación y viceversa.

✓ 4/ F → $\{uv \mid u \neq v, |u| = |v|\} \in LIC$, mientras que su complementario $\{w \mid |w| \text{ impar}\} \cup \{uu \mid u \in \{a,b\}^*\} \notin LIC$.

⑨

✓ a) Sí lo es. Basta concatenar xy (x es conocido, y es la palabra de entrada) y pasarla como entrada al programa que reconoce L. más detalle

b) ~~Sí lo es, ya que si hubiera alguna cadena xy para la que no recibiéramos respuesta,~~ → ¿con qué programa? ~~tampoco podríamos asegurar que $y \in P_x(L)$. Luego si $P_x(L)$ decidible $\Rightarrow L$ decidible~~

Mal, la x es fija en cada caso

La respuesta es coger $x = \varepsilon$

✓ ④

3. Dar un ejemplo de un lenguaje incontextual que no sea regular. Demostrar que dicho ejemplo es incontextual.

$L = \{a^n b^n \mid n \in \mathbb{N}\}$ es un lenguaje incontextual que no es regular.

Existe un autómata de pila que represente L, por lo que L es incontextual. ¿cuál?

(8)

Para cada una de las siguientes afirmaciones, decir si es cierta o falsa justificando la respuesta.

1. El complementario de un lenguaje semidecidible es semidecidible

$\boxed{\text{Falso}}$ → $H = \{(p, w) : p$ es un programa que para con entrada $w\}$ es un lenguaje semidecidible

~~$\bar{H}$ es un lenguaje no decidible (indecidible).~~ y también esto es lo importante

$\bar{H}$ no es semidecidible ←

2. El complementario de un lenguaje decidible es decidible

$\boxed{\text{Verdadero}}$ → Sea A decidible: hay un algoritmo que acepta A y para siempre.

Un algoritmo para $\bar{A}$ es como el algoritmo para A pero cambiando "devuelve acepta" por "devuelve rechaza" y viceversa.

3. El complementario de un lenguaje regular es regular

$\boxed{\text{Verdadero}}$ → Sea A regular hay un autómata que representa A.

El autómata para $\bar{A}$ es igual que para A pero cambiando los estados finales por no finales y los no finales por finales.

8

4. El complementario de un lenguaje incontextual, es incontextual.

$\boxed{\text{Falso}}$ → El lenguaje $L = \{uv \mid |u| = |v|\}$ es incontextual

$L^c = \{uv \mid |uv| = \text{impar}\} \cup \{ww \mid w \in \{a,b\}^*\}$ no es

un lenguaje incontextual.

Mal

te refieres a $\{uv \mid |u| = |v| \land u \neq v\}$

9

Dado un alfabeto Σ, un lenguaje $L \subseteq \Sigma^*$ y una palabra $x \in \Sigma^*$, se define $P_x(L) = \{y \in \Sigma^* \mid xy \in L\}$

a) Si L es un lenguaje recursivo o decidible, determinar si $P_x(L)$ es un lenguaje decidible o recursivo para cada $x \in \Sigma^*$. $\boxed{\text{VERDADERO}}$

Q. es un programa que acepta L y contesta siempre.

Un programa para $P_x(L)$ será:

Entrada y

cadena $z = x \cdot y$

cadena res = Q(z)

devuelve res;

b) Si para cada $x \in \Sigma^*$ $P_x(L)$ es un lenguaje recursivo o decidible, determinar si L es un lenguaje decidible o recursivo.

$\boxed{\text{Verdadero}}$ → Si cogemos $x = \varepsilon$ entonces $P_x(L) = L$ ⇒

⇒ L es decidible.

10

1. Describe razonadamente un contraejemplo para cada una de estas afirmaciones

a) Si L es un lenguaje semidecidible, entonces todo lenguaje $A \subseteq L$ también es semidecidible.

$\boxed{\text{Falso}} \to L = (0+1)^*$ es un lenguaje semidecidible

$\overline{H}$ no es semidecidible y $\overline{H} \subseteq L$

2. Demuestre que el lenguaje $L = \{p, x, q \mid$ los programas p y q con entrada x devuelven el mismo valor $\}$

Sean p y/o q programas que paren o no siempre, podemos construir un programa que represente L, es decir, dada la entrada x, si p y q devuelven el mismo valor devuelve "acepta" en caso contrario devuelve "rechaza" o se cuelga.

Entrada x

```
cadena resP := p(x);
cadena resQ := q(x);
si resP = resQ entonces
    devuelve acepta.
Sino devuelve rechaza
fsi
```

1.b) La intersección de un lenguaje semidecidible y un decidible es decidible.

$\boxed{\text{Falso}} \to$ Deberíamos reconocer ~~el~~ lenguaje semidecidible de forma decidible, lo cual es imposible.

Mal razonado, hay que dar un contraejemplo

$H \cap (0+1)^* = H$

(11) Determinar si es decidible: determinar si un lenguaje regular L cualquiera tiene una o más palabras en común con el lenguaje:

$E = (00 + 11 + (01 + 10)(00 + 11)^* (01 + 10))^*$

Podemos demostrar que es decidible construyendo un programa que si L y E tienen una o más palabras en común devuelve "acepta" y en caso contrario devuelve "rechaza".

Entrada A (autómata que reconoce L)

Construir un autómata B que acepte $L(A) \cap L(E)$ ← ¿cómo? algo más de justificación
$B' =$ minimizar (B)
si B' tiene almenos un estado final entonces
devuelve "acepta"
sino devuelve rechaza.

(12) 1. Demostrar que si A y B son decidibles entonces $A \cup B$ es decidible

- Si A es decidible tengo un programa, p_A, tal que dada una cadena x, si $x \in A$ devuelve "acepta" y si $x \notin A$ devuelve "rechaza".
- Si B es decidible tengo un programa, p_B, tal que dada una cadena x, si $x \in B$ devuelve "acepta" y si $x \notin B$ devuelve "rechaza".

Para demostrar que la unión de A y B es decidible podemos construir un programa que dada una cadena x, si $x \in A$ o $x \in B$ devuelve "acepta" y si $x \notin A$ y $x \notin B$ devuelve "rechaza"

Entrada x
cadena resA := pA(x);
cadena resB := pB(x);
si resA = acepta OR resB = acepta entonces
devuelve acepta;
sino devuelve rechaza
fsi

2. Demostrar que si $A \subseteq \Sigma^*$ es un lenguaje decidible y $w \in \Sigma^*$ entonces $A_w = \{u \mid wu \in A\}$ es decidible.

Verdadero

Q es un programa que ~~acepta~~ A y para siempre.

Un programa para A_w sería:

Entrada u

cadena $z = w \cdot u$;

cadena res := Q(z);

devolver res;

3. Dar un ejemplo de un lenguaje semidecidible que no sea decidible.

El problema de parada. Dado un programa P y una cadena w. ¿P para con entrada w?

$H: \{(p, w) : P$ es un programa que para en entrada $w\}$