COMPLEJIDAD Y COMPUTABILIDAD

Preguntas más frecuentes

SUMARIO

Sobre Co−P,Co−NP y Co−NP−Completo

¿Cuál es la definición de Co−P, Co−NP y Co−NP−Completo?
Un problema P_1 se dice que está en la clase Co−P si su complementario está en la clase P, se dice que P_1 está en la clase Co−NP si su complemetario está en la clase NP , y se dice que P_1 está en la clase Co−NP−Completo si su complementario está en la clase NP−Completo.

¿P=Co−P?
Sí, P=Co−P ya que se da el doble contenido:

- $P \subset$ Co−P, ya que si $P_1 \in P$, entonces existe una M que decide a P_1, por lo que $\overline{M}$ decide a $\overline{P}_1$, con lo que $\overline{P}_1 \in P$ y por tanto $P_1 \in$ Co−P.

- Co−P $\subset P$, ya que si $P_1 \in$ Co−P, entonces $\overline{P}_1 \in P$ y existe una M que decide a $\overline{P}_1$ por lo que M decide a $\overline{P}_1 = P_1$ y por tanto $P_1 \in P$.

¿Si NP=Co−NP, entonces existe algún problema NP−Completo cuyo complementario pertenece a NP?

Sí. En realidad no es que exista uno, es que son todos. Es decir, es cierto que:

Si NP=Co−NP, entonces todo problema NP−Completo verifica que su complementario está en NP.

Para ver esto, hay que tomar $P_1 \in$ NP−Completo y ver que $P_1 \in$ NP.

La cadena de razonamientos es:

- Al ser $P_1 \in$ NP−Completo, entonces $P_1 \in$ NP.
- Como por hipótesis NP=Co−NP, $P_1 \in$ Co−NP y por tanto $\overline{P}_1 \in$ NP.

¿Si existiera algún problema NP−Completo cuyo complementario perteneciera a NP, entonces NP⊂Co−NP?
Sí. El enunciado pide demostrar que si existe $P_1 \in$ NP−Completo con $P_1 \in$ NP, entonces NP⊂Co−NP.

Para ver que NP⊂Co−NP, hay que tomar un $P_2 \in$ NP y ver que P_2 está en Co−NP.

La cadena de razonamientos es:

- Si $P_2 \in$ NP, al ser $P_1 \in$ NP−Completo, $P_2 \prec_P P_1$.
- Al ser $P_2 \prec_P P_1$, también se tiene que $\overline{P}_2 \prec_P \overline{P}_1$.
- Lo anterior combinado con que $\overline{P}_1 \in$ NP lleva a que $\overline{P}_2 \in$ NP, con lo que $P_2 \in$ Co−NP.

¿Si existiera algún problema NP−Completo cuyo complementario perteneciera a NP, entonces Co−NP⊂NP?

Sí. El enunciado pide demostrar que si existe $P_1 \in$ NP−Completo con $P_1 \in$ NP, entonces Co−NP⊂NP.

Para ver que Co−NP⊂NP, hay que tomar un $P_2 \in$ Co−NP y ver que P_2 está en NP.

La cadena de razonamientos es:

- Si $P_2 \in$ Co−NP entonces $\overline{P}_2 \in$ NP.
- Al ser $P_1 \in$ NP−Completo, $\overline{P}_2 \prec_P P_1$.

- Al ser $\overline{P}_2 \prec_P P_1$, también se tiene que $P_2 \prec_P \overline{P}_1$.
- Lo anterior combinado con que $\overline{P}_1 \in$ NP lleva a que $P_2 \in$ NP.

¿NP=Co−NP ⇔ existe algún problema NP−Completo cuyo complementario pertenece a NP?

Sí. Dado que es una doble implicación hay que demostrar las dos:

- La implicación $\Rightarrow$se demuestra por la pregunta 3.
- La implicación $\Leftarrow$se demuestra por doble contenido. El contenido NP$\subset$Co−NP por la pregunta 4 y el contenido Co−NP$\subset$NP por la pregunta 5.

Sobre P, NP y NP−Completo

¿Cuál es la definición de P, NP y NP−Completo?

Un problema P_1 se dice que está en la clase P sii es resoluble en tiempo polinómico mediante una MTD , se dice que P_1 está en la clase NP si es resoluble en tiempo polinómico mediante una MTND, y se dice que P_1 está en la clase NP−Completo sii $P_1 \in$ NP y $\forall P_2 \in$ NP, $P_2 \prec_P P_1$.

¿Si P1 ∈ NP−Completo, entonces no existe una MTD que lo resuelva?

No es cierto. Si $P_1 \in$ NP−Completo, $P_1 \in$ NP y por tanto P_1 es decidible y $\exists$MTD que lo resuelve (aunque posiblemente no en tiempo polinomial).

¿Bastaría encontrar un problema P_1 que sea NP−Completo y P para poder ya decir que P=NP?

Sí. Por un lado P $\subset$ NP de forma trivial. Por otro, NP $\subset$ P, ya que si se toma $P_2 \in$ NP, entonces $P_2 \prec_P P_1$ y por tanto $P_2 \in$ P.

¿Bastaría encontrar un problema P1 que sea NP y que no sea P para poder ya decir que ningún NP−Completo está en P?

Sí. Si suponemos (por reducción al absurdo) que existe un $P_2 \in$ NP−Completo y que también $P_2 \in$ P, se tendría que por ser NP−Completo, $P_1 \prec_P P_2$, y al ser $P_2 \in$ P, también $P_1 \in$ P con lo que se llegaría a una contradicción.

¿Cuál es el esquema de demostración de que un problema P_2 es NP−Completo?

El esquema es:

Demostrar que P_2 es NP.

Tomar un P_1 que se sepa que está en NP−Completo y demostrar que $P_1 \prec_P P_2$.

¿El problema de la parada es NP−Completo?

No. El problema H es indecidible y no puede estar en NP, por lo tanto no puede ser NP−Completo.

Sobre PCP y PCPM

¿Cuál es la definición de PCP y PCPM?

Sean A y B dos listas de cadenas de un alfabeto Σ, con al menos dos símbolos, dadas por $A=\{w_1,w_2,\ldots,w_k\}$ y $B=\{x_1,x_2,\ldots,x_k\}$ con $k \in N$.

El PCP consiste en determinar si existe $\{i_1,\ldots,i_m\}$ de forma que $\{w_{i1},w_{i2},\ldots,w_{im}\}=\{x_{i1},x_{i2},\ldots,x_{im}\}$.

El PCPM o PCP modificado es una restricción del PCP en el que se obliga a que $i_1=1$.

¿Existe un algoritmo para resolver el PCP?

El PCP es indecidible, es decir no hay un algoritmo que resuelva todos las posibles instancias. Sin embargo, algunos casos particulares como el PCPT (PCP "Tonto") y el PCPU (PCP "Unario") sí son decidibles, es decir sí hay un algoritmo que resuelve todas sus instancias.

¿Cuál es el esquema de demostración de que el PCP es indecidible?

El esquema de demostración es $L_u \prec PCPM \prec PCP$ y dado que L_u es indecidible, el PCPM es indecidible y el PCP es indecidible.

¿A veces es posible afirmar que el PCP tiene solución positiva o solución negativa?

Aunque el PCP sea indecidible, a veces sí se puede afirmar que una instancia tenga solución positiva o solución negativa. Por ejemplo, el PCP planteado sobre (10,1) y (110,01) tiene solución negativa porque $|w_1|>|x_1|$ y $|w_2|>|x_2|$.

Es posible que para una instancia en concreto del PCP se tenga que el PCPM asociado tenga solución negativa y, sin embargo, el PCP tenga solución positiva?

Sí, por ejemplo, para (1,111),(10111,10) y (10,0) el PCP tiene solución positiva con los índices {2,1,1,3} y el PCPM tiene solución negativa ya que los índices {1}, {1,1}, {1,2} y {1,3} dan negativo.

¿El PCP "Unario" puede tener solución positiva para alguna instancia?

Sí, por ejemplo, el PCP planteado sobre (1,1111), (11,1) y (11,111) tiene solución positiva para los índices {1,2,2,2}.

Sobre SAT, CSAT y 3SAT

¿Cuál es el esquema de demostración de que 3SAT es NP−Completo?

Los pasos esquemáticos para demostrar que 3SAT es NP−Completo son:

- Demostrar que 3SAT es NP.
  - Es evidente ya que 3SAT $\subset$ CSAT $\in$ NP.
- Demostrar que CSAT $\prec$ P3SAT.
  - En primer lugar se describe un algoritmo que a partir de una instancia E de CSAT construya una instancia F de 3SAT. Es decir que a partir de $E=e_1 \wedge \ldots \wedge e_k$ donde cada e_i tiene un número distinto de literales, construya $F=f_1 \wedge \ldots \wedge f_k$ donde f_i tiene exactamente 3 literales. Este algoritmo distingue cuatro casos: que e_i tenga un único literal, que tenga dos, que tenga tres y que tenga cuatro o más.
    - Si $e_i=x$, se considera $f_i=f_{1i} \wedge f_{2i} \wedge f_{3i} \wedge f_{4i}$ con $f_{1i}=(x \vee u \vee v)$, $f2i=(x \vee u \vee \bar{v})$, $f_{3i}=(x \vee \bar{u} \vee v)$, $f_{4i}=(x \vee \bar{u} \vee \bar{v})$.
    - Si $e_i=x \vee y$, se considera $f_i=g_{1i} \wedge g_{2i}$ con $g_{1i}=(x \vee y \vee z)$, $g_{2i}=(x \vee y \vee \bar{z})$.
    - Si $e_i=x \vee y \vee z$, se considera $f_i=e_i$.
    - Si $e_i=x_1 \vee x_2 \ldots \vee x_m$, se considera $f_i=h_{1i} \wedge h_{2i} \wedge \ldots \wedge h_{m-3,i} \wedge h_{m-3+1,i}$ con $h_{1i}=(x_1 \vee x_2 \vee y_1)$, $h_{2i}=(x_3 \vee \bar{y}_1 \vee y_2)$, …, $h_{m-3,i}=(x_{m-2} \vee \bar{y}_{m-4} \vee y_{m-3})$, $h_{m-3+1,i}=(x_m-1 \vee x_m \vee \bar{y}_{m-3})$.
  - En segundo lugar se comprueba que dicho algoritmo es polinomial.
  - En tercer lugar se comprueba que dicho algoritmo transforma instancias positivas de CSAT en instancias positivas de 3SAT e instancias negativas de CSAT en instancias negativas de 3SAT. Es decir que E es satisfacible $\Leftrightarrow$ F es satisfacible (obsérvese que E y F no son equivalentes ya que se han introducido nuevos literales).

¿Cuál es el esquema de demostración de que 4SAT es NP−Completo?

Los pasos esquemáticos para demostrar que 4SAT es NP−Completo son:

- Demostrar que 4SAT es NP.
  - Es evidente ya que 4SAT $\subset$ CSAT $\in$ NP.
- Demostrar que 3SAT $\prec_p$ P4SAT.
  - En primer lugar se describe un algoritmo que a partir de una instancia E de 3SAT construya una instancia F de 4SAT. Es decir que a partir de $E=e_1 \wedge \ldots \wedge e_k$ donde ei tiene exactamente 3 literales, construya $F=f_1 \wedge \ldots \wedge f_k$ donde fi tiene exactamente 4 literales. Este algoritmo es el siguiente: a partir de $e_i= x \vee y \vee z$ construye $f_i=f_{1i} \wedge f_{2i}$ con $f_{1i}=(x \vee y \vee z \vee h)$ y $f_{2i}=(x \vee y \vee z \vee \bar{h})$.
  - En segundo lugar se comprueba que dicho algoritmo es polinomial (en este caso es trivial).
  - En tercer lugar se comprueba que dicho algoritmo transforma instancias positivas de 3SAT en instancias positivas de 4SAT e instancias negativas de 3SAT en instancias negativas de 4SAT.

Es decir que E es satisfacible $\Leftrightarrow$ F es satisfacible (obsérvese que E y F no son equivalentes ya que se han introducido nuevos literales).

¿Cuál es el esquema de demostración de que 2SAT es P?

El esquema de demostración de que 2SAT es P es considerar el problema CFC (El problema de encontrar las componentes fuertemente conexas en un grafo dirigido) y definir una reducción polinómica 2SAT $\prec_p$ PCFC y dado que el CFC es P, se tiene que 2SAT también es P.

¿Cómo se demuestra que 1SAT es P?

Para demostrar que 1SAT es P se recorre la expresión buscando si en la misma aparece la misma variable proposicional negada y sin negar.

Si se da esta situación, al estar multiplicadas, la expresión es no satisfacible. Si no se da esa situación, es satisfacible.

Sobre codificación de MT

¿Cuál es el contexto de codificación de MT?

En el contexto de codificación de MT se considera $M=(Q=\{q_1, q_2\ldots, q_e\}, \Sigma=\{0,1\}, \Gamma=\{X_1=0,X_2=1,X_3=B, \ldots,X_s\}, \delta, q_1, B, \{q_2\})$

¿Cómo se codifica una MT?

La codificación de una máquina de Turing M se realiza de la siguiente forma:

- El estado inicial q_1 se codifica con $0^1=0$, el estado final q_2 con $0^2=00$, q_3 (si existe) con $0^3=000$, y así sucesivamente en el caso de que haya más estados.
- El símbolo de la cinta $X_1=0$ se codifica con $0^1=0$, el $X_2=1$ con $0^2=00$, el $X_3=B$ con $0^3=000$, el X_4 (si existe) con $0^4=0000$, y así sucesivamente en el caso de que haya más símbolos de cinta.
- La dirección L (izquierda) se codifica con $D_1=0$ y la dirección R (derecha) con $D_2=0^2=00$.
- Una transición $\delta(q_i, X_j)=(q_k, X_l, D_m)$ se codifica con la cadena $0^i10^j10^k10^l10^m$.
- Por último una codificación para M, suponiendo que tiene n transiciones, consiste en todos los códigos correspondientes a todas sus transiciones $C_1,\ldots,C_n$ separadas por pares de unos con $C_111C_211\ldots 11C_n$ donde cada C_i corresponde al código de la transición i-ésima de M.

¿Cómo se puede codificar $M=(Q=\{q_1,q_2\},\Sigma=\{0,1\},\Gamma=\{0,1,B\},\delta,q_1,B,\{q_2\})$ con $\delta(q_1,0)=(q_2,0,R)$?

Una posibilidad es 01010010100.

¿Cómo se puede codificar $M=(Q=\{q_1,q_2\},\Sigma=\{0,1\},\Gamma=\{0,1,B\},\delta,q_1,B,\{q_2\})$ con $\delta(q_1,0)=(q_1,1,L)$ y $\delta(q_1,1)=(q_2,1,R)$?

Una posible codificación es 0101010010110100100100100, otra es 0100100100100110101010010.

Sobre congruencias

¿Qué significa que a $\in N$ y b $\in Z$ sean congruentes módulo n $\in N$?

Se dice que a $\in$Z y b $\in$Z son congruentes módulo n $\in$N y se escribe a≡ b (mod n) sii al dividir a entre n se obtiene el mismo resto que al dividir b entre n.

¿Es cierto que para a,b $\in Z$ y c $\in N$, se tiene que a≡b (mod n)⇔n|(a−b)?

Sí es cierto. Ver MDM 3.4.1 de la profesora tutora Idoia.

¿Es cierto que para a,b$\in Z$ y n,c$\in N$, se tiene que ac≡bc (mod n)⇒a≡ b (mod n/c)?

No. Lo que es cierto es que a≡ b (mod n/d) con d=mcd(n,c). Ver MDM 3.4.2 de la profesora tutora Idoia.

¿Es cierto que para a,b,c,d $\in Z$ y n$\in N$, se tiene que a≡b (mod n) y c≡d (mod n)⇒ac≡ bd (mod n)?

Sí es cierto. Se parte de que a−b= $\dot{n}$ y c−d= $\dot{n}$, con lo que c(a−b)= $\dot{n}$ y b(c−d)= $\dot{n}$ y, por tanto, c(a−b) +b(c−d)= $\dot{n}$ y operando ac−bd= $\dot{n}$.

¿Es cierto que para a,b∈Z y m,n$\in \mathbb{N}$, se tiene que a≡b (mod n)⇒a^m≡ b^m(mod n)?

Sí es cierto. Se prueba por inducción utilizando la propiedad anterior.

¿Es cierto que 538≡4 (mod 11)?

Sí. Se puede probar utilizando la exponenciación rápida:

- $38_{10}=100110_2$
- 5≡5 (mod 11)
- Utilizando que 5≡5 (mod 11) y 5≡5 (mod 11), multiplicando $5^2\equiv5^2$ (mod 11), es decir $5^2\equiv3$ (mod 11)
- Utilizando que $5^2\equiv3$ (mod 11) y $5^2\equiv3$ (mod 11), multiplicando $5^4\equiv32$ (mod 11), es decir $5^4\equiv9$ (mod 11)
- Utilizando que $5^4\equiv9$ (mod 11) y $5^4\equiv9$ (mod 11), multiplicando $5^8\equiv92$ (mod 11), es decir $5^8\equiv81$ (mod 11), o lo que es lo mismo $5^8\equiv4$ (mod 11)
- Utilizando que $5^8\equiv4$ (mod 11) y $5^8\equiv4$ (mod 11), multiplicando $5^{16}\equiv42$ (mod 11), es decir $5^{16}\equiv16$ (mod 11), o lo que es lo mismo $5^{16}\equiv5$ (mod 11)
- Utilizando que $5^{16}\equiv5$ (mod 11) y $5^{16}\equiv5$ (mod 11), multiplicando $5^{32}\equiv52$ (mod 11), es decir $5^{32}\equiv25$

(mod 11), o lo que es lo mismo $5^{32} \equiv 3 \pmod{11}$

- Por último, encadenando los productos: $5^{38} = 5^{1*2^5+1*2^2+1*2^1} = 5^{32}5^4 5^2 \equiv 3 \cdot 9 \cdot 3 = 81 \equiv 4 \pmod{11}$

Además, en este caso como 11 es primo y 5 y 11 coprimos, se puede utilizar el teorema pequeño de Fermat:

- $5^{10} \equiv 1 \pmod{11}$
- $5^{38} = 5^{10}\, 5^{10}\, 5^{10}\, 5^8 \equiv 1 \cdot 1 \cdot 1 \cdot 4 = 4 \pmod{11}$

Sobre enumeración de MT

¿Cuál es el contexto de enumeración de MT? ¿Qué es la MT i-ésima?

La MT i-ésima M_i es aquella que tiene por código el dado por la i-ésima cadena w_i verificando que $1w_i=i_2$ donde i_2 es la representación binaria del número decimal i. Así la M_{1354} es la que tiene por código el dado por w_{1354}=0101001010 ya que $1354_{10}=10101001010_2$.

¿$L(M_{682})=\emptyset$?

Sí, ya que al ser $682_{10}=1010101010_2$, M_{682} tiene por código el dado por w_{682}=010101010 que no es un código válido ya que no cumple los requisitos del contexto de enumeración de MT, por lo que se considera que $L(M_{682})=\emptyset$,

¿$L(M_{1354})=L(M_{2706})$?

Sí, ya que al ser:

$1354_{10}=10101001010_2$, M_{1354} tiene por código el dado por w_{1354}=0101001010 que corresponde a $(Q=\{q_1,q_2\},\Sigma=\{0,1\},\Gamma=\{0,1,B\},\delta,q_1,B,\{q_2\})$ con $\delta(q_1,0)=(q_2,0,L)$ y por tanto $L(M_{1354})=0(0+1)^*$.

$2706_{10}=101010010010_2$, M_{2706} tiene por código el dado por w_{2706}=01010010010 que corresponde a $(Q=\{q_1,q_2\},\Sigma=\{0,1\},\Gamma=\{0,1,B\},\delta,q_1,B,\{q2\})$ con $\delta(q_1,0)=(q_2,1,L)$ y por tanto $L(M_{2706})=0(0+1)^*$.

¿$L(M_{20770})=\emptyset$?

No, ya que al ser $20770_{10}=101000100100010_2$, M_{20770} tiene por código el dado por w_{20770}=01000100100010 que corresponde a $(Q=\{q_1,q_2\},\Sigma=\{0,1\},\Gamma=\{0,1,B\},\delta,q_1,B,\{q_2\})$ con $\delta(q_1,B)=(q_2,B,L)$ y por tanto $L(M_{20770})=\{\epsilon\}\neq\emptyset$.

Sobre lenguajes L_d y L_{nd}

¿Cuál es la definición de los lenguajes L_d y L_{nd}?

El lenguaje de diagonalización L_d está formado por todas las codificaciones de MT que no forman parte del lenguaje de la MT que codifican. Es decir, $L_d = \{w_i / w_i \notin L(M_i)\}$.

Por otra parte $L_{nd} = \overline{L}_d = \{w_i / w_i \in L(M_i)\}$.

Ejemplos de cadenas de L_d son w_{100} y w_{10532} ya que $L(M_{100}) = \emptyset$ y $w_{100} \notin L(M_{100})$ y $L(M_{10532}) = 1(0+1)^*$ y $w_{10532} \notin L(M_{10532})$.

Ejemplos de cadenas de L_{nd} son w_{2706} y w_{2708}.

¿$L(M_{682})$ es L_d?

$L(M_{682})$ no es L_d ya que $L(M_{682}) = \emptyset$.

¿$w_{682} \in L_d$?

Sí, $w_{682} \in L_d$ ya que $L(M_{682}) = \emptyset$ y $w_{682} \notin L(M_{682})$

¿Las primeras cadenas de $\overline{L}_d$ son $w_1, w_2, w_3, \ldots$?

R: No, las primeras cadenas de $\overline{L}_d$ son $w_{1354}, w_{2706}, w_{2708}, \ldots$

¿L_d es RE no recursivo?

El lenguaje L_d es indecidible verificando que es no−RE.

¿L_{nd} es RE no recursivo?

El lenguaje L_{nd} es indecidible verificando que es RE no recursivo.

Sobre los lenguajes L_e y Ln_e

¿Cuál es la definición de los lenguajes L_e y L_{ne}?

El lenguaje L_e es el lenguaje formado por todas las codificaciones de MT cuyo lenguaje es el lenguaje vacío. Es decir, $L_e = \{w_i / L(M_i) = \emptyset\}$.

Por otra parte $L_{ne} = \overline{L_e} = \{w_i / L(M_i) \neq \emptyset\}$.

Ejemplos de cadenas de Le son w_{100} y w_{255} ya que $L(M_{100}) = \emptyset$ y $L(M_{255}) = \emptyset$.

Ejemplos de cadenas de L_{ne} son w_{1354} y w_{10532} ya que $L(M_{1354}) = 0(0+1)^* \neq \emptyset$ y $L(M_{10532}) = 1(0+1)^* \neq \emptyset$.

¿L_e es el lenguaje vacío?

No. El lenguaje vacío no tiene ninguna cadena, ni siquiera la cadena vacía, mientras que el lenguaje L_e sí tiene cadenas.

¿Es $L_e = L_d$?

No es cierto que $L_e = L_d$. Aunque se verifica que L_e está contenido en L_d, no se verifica que L_d está contenido en L_e, ya que, por ejemplo, $w_{10532} \in L_d$ y sin embargo $w_{10532} \notin$ Le

¿L_{ne} es RE no recursivo?

El lenguaje L_{ne} es indecidible verificando que es RE no recursivo.

¿L_e es RE no recursivo?

El lenguaje L_e es indecidible verificando que es no−RE. Si Le fuera RE, al haber probado con anterioridad que $L_{ne} \in$ RE, se tendría que tanto Le como L_{ne} serían Rec lo que entraría en contradicción con el hecho de que se ha probado con anterioridad que L_{ne} no es Rec.

Sobre los lenguajes L_u y L_{nu}

¿Cuál es la definición de los lenguajes L_u y L_{nu}?

El lenguaje universal L_u está formado por todas las cadenas que codifican el par <M,w> con w entrada que acepta M. La codificación de <M,w> es <M>111w, siendo <M> la codificación de M. Es decir,

L_u= {<M,w>/M acepta w}.

Por otra parte $L_{nu}=\overline{L}_u$={<M,w>/M no acepta w}).

Ejemplos de cadenas de L_u son 010100101001110 ya que $0 \in L(M_{01010010100})$ y 01010010100111010010100 ya que $01010010100 \in L(M_{01010010100})$.

Ejemplos de cadenas de L_{nu} son 010100101001111 ya que $1 \notin L(M_{01010010100})$ y 01010010100111ϵ ya que $\epsilon \notin L(M_{01010010100})$.

¿Es cierto que $L_d \prec L_{nu}$? ¿Es cierto que $L_{nd} \prec L_u$?

Las dos preguntas tienen respuesta afirmativa.

Para demostrar que $L_d \prec L_{nu}$ se construye una MT que transforma w_i en w_i111w_i verificando que lleva:

- instancias positivas de L_d en instancias positivas de L_{nu} , ya que si $w_i \in L_d$, $w_i111w_i \notin L_u$, con lo que $w_i111w_i \in L_{nu}$.
- instancias negativas de L_d en instancias negativas de L_{nu}, ya que si $w_i \notin L_d$, $w_i \in L(M_i)$, con lo que $w_i111w_i \in L_u$, con lo que $w_i111w_i \notin L_{nu}$.

Para demostrar que $L_{nd} \prec L_u$ se construye la misma MT que antes verificando que lleva:

- instancias positivas de L_{nd} en instancias positivas de L_u.
- instancias negativas de L_{nd} en instancias negativas de L_u.

¿L_u es RE no recursivo?

El lenguaje L_u es indecidible verificando que es RE no recursivo.

Por una parte L_u es RE ya que existe una MT que lo acepta. El esquema de demostración consiste en describir una MT U con tres cintas que acepta L_u y a apartir de ella utilizar el teorema que permite transformarla en una MT de una cinta.

La afirmación de que $L_u \notin$ Rec, se demuestra por reducción al absurdo. Si se considera que $L_u \in$ Rec, se tendría que $L_{nu} \in$ Rec y dado que $L_d \prec L_{nu}$ se tendría que $L_d \in$ Rec y se llegaría a una contradicción.

¿Lnu es RE no recursivo?

El lenguaje L_{nu} es indecidible verificando que es no−RE, ya que si $L_{nu} \in$ RE, como $L_u \in$ RE, se tendría que $L_u \in$ Rec y se llegaría a una contradicción.

¿Por qué es importante saber que Lu es RE\Rec, es decir que es RE pero no es Recursivo?

Para demostrar la indecidibilidad de un problema. Si se es capaz de demostrar que $L_u \prec P$ entonces P es indecidible, ya que si P fuera decidible, entonces L_u sería decidible y eso entraría en contradicción con el hecho de que $L_u \in RE \setminus Rec$.

Sobre los lenguajes RE y no−RE

¿Cuál es la definición de los lenguajes RE y no−RE?

Un lenguaje L es recursivo enumerable (RE)sii se puede construir una máquina de Turing M que acepta el lenguaje L. Es decir:

- Si $w \in L$, M se detiene en un estado de aceptación.
- Si $w \notin L$, M se detiene en un estado de no aceptación o no se detiene.

Es importante observar que dicha M no siempre se detiene. Evidentemente, Rec $\subset$ RE, pero RE $\subset$ Rec (por ejemplo, el lenguaje universal L_u verifica que $L_u \in RE \setminus Rec$) . Esto quiere decir que RE = (RE\Rec) $\cup$ Rec, con lo que en RE están los lenguajes RE\Rec que son indecidibles y los lenguajes Rec que son decidibles. Un lenguaje L es no recursivo enumerable (no−RE) sii no existe ninguna máquina de Turing M que lo acepte.

¿Todos los lenguajes RE son indecidibles?

No. Los lenguajes RE que pertenecen a Rec son decidibles.

¿Es posible construir una MT multicinta que acepten lenguajes que no sean RE?

No es posible. Las MT multicinta tienen igual potencia que las MT de una única cinta.

¿Si L $\in$ RE y $\overline{L} \in$ RE, entonces L$\in$R?

Sí, es cierto. Se demuestra por reducción al absurdo. Si $L \notin R$, como $L \in RE$, se tendría que $L \in RE \setminus R$, entonces $\overline{L} \notin RE$, con lo que se llegaría a una contradicción.

¿Qué propiedades son cerradas para los lenguajes RE?

Las propiedades principales que son cerradas para los lenguajes RE son: unión, intersección, concatenación, clausura de Kleen, homomorfismo inverso y homomorfismo.

¿Qué propiedades no son cerradas para los lenguajes RE?

Las propiedades principales que no son cerradas para los lenguajes RE son complementación y diferencia.

Sobre los lenguajes Rec

¿Cuál es la definición de los lenguajes Rec?

Un lenguaje L es recursivo (Rec) sii se puede construir una máquina de Turing M que decida el lenguaje L. Es decir:

- Si $w \in L$, M se detiene en un estado de aceptación.
- Si $w \notin L$, M se detiene en un estado de no aceptación.

Es importante señalar que dicha M siempre se detiene. A los lenguajes Rec se les llama también decidibles.

¿El lenguaje $L=0(0+1)^*$ es Rec?

Sí. Por ejemplo la máquina de Turing $M= (\{q_1,q_2\}, \{0,1\}, \{0,1,B\}, \delta, q_1, B, \{q_2\})$ con $\delta(q_1,0) = (q_2,0,R)$ decide a dicho lenguaje.

¿Puede ocurrir que existan distintas máquinas de Turing que sirvan para demostrar que un lenguaje es Rec?

Sí. Por ejemplo $M_1 = (\{q_1,q_2\}, \{0,1\}, \{0,1,B\}, \delta_1, q_1, B,\{q_2\})$ con $\delta_1(q_1,1)=(q_2,1,R)$ y $M_2 = (\{q_1,q_2\}, \{0,1\}, \{0,1,B\}, \delta_2, q_1, B, \{q_2\})$ con $\delta_2(q_1,1) = (q_2,1,L)$ deciden al lenguaje $1(0+1)^*$

¿El lenguaje $L=\{\epsilon\}$ es Rec?

Sí, por ejemplo $M = (\{q_1,q_2\}, \{0,1\}, \{0,1,B\}, \delta, q_1, B, \{q_2\})$ con $\delta(q_1,B) = (q_2,B,L)$ decide a dicho lenguaje.

¿Qué propiedades son cerradas para los lenguajes Rec?

Las propiedades principales que son cerradas para los lenguajes Rec son: unión, intersección, complementario, concatenación, clausura de Kleen, homomorfismo inverso y diferencia.

¿Qué propiedades no son cerradas para los lenguajes Rec?

La propiedad principal que no es cerrada para los lenguajes Rec es homomorfismo.

Sobre el teorema de Rice

¿Qué dice el teorema de Rice?

Antes de enunciar el teorema de Rice hay que dar unas definiciones previas:

- Una propiedad de los lenguajes RE es cualquier conjunto de los lenguajes RE.
- Una propiedad trivial de los lenguajes RE es el conjunto vacío o el conjunto de todos los lenguajes RE.

El teorema de Rice afirma que toda propiedad no trivial de los lenguajes RE, es indecidible.

Otra forma de enunciarlo es:

Sea C un subconjunto propio de los lenguajes RE, es decir $\emptyset \neq C \neq RE$, entonces $LC = \{w_i/L(M_i) \in C\}$ no es recursivo.

¿{w_i/M_i tiene un número par de estados} es indecidible?

Esta propiedad no hace referencia a los lenguajes, por lo que no se puede aplicar el teorema de Rice. En este caso, la mera inspección directa de w_i sirve para saber si M_i tiene un número par de estados, por lo que {w_i/M_i tiene un número par de estados} es decidible.

¿{w_i/L(M_i) tiene exactamente n elementos} es indecidible?

R: Tener exactamente n elementos es una propiedad no trivial de los lenguajes RE (los RE con exactamente n elementos son un subconjunto propio de los RE, por lo que aplicando el teorema de Rice {w_i/L(M_i) tiene exactamente n elementos} es indecidible.

¿{w_i/L(M_i) es infinito} es indecidible?

Tener infinitos elementos es una propiedad no trivial de los lenguajes RE, por lo que aplicando el teorema de Rice, {w_i/L(M_i) es infinito} es indecidible.

¿{w_i/L(M_i) está formado por al menos 37 cadenas diferentes} es indecidible?

R: Tener al menos 37 cadenas diferentes es una propiedad no trivial de los lenguajes RE. Por el teorema de Rice, {w_i/L(M_i) está formado por al menos 37 cadenas diferentes} es indecidible.

¿{w_i/M_i tiene al menos 37 estados diferentes} es indecidible?

Esta propiedad no hace referencia a los lenguajes, por lo que no se puede aplicar el teorema de Rice. Una inspección directa de w_i sirve para saber si Mi tiene al menos 37 estados diferentes, por lo que {w_i/M_i tiene al menos 37 estados diferentes} es decidible.

Sobre el teorema pequeño de Fermat

¿Qué dice el teorema pequeño de Fermat?

El teorema pequeño de Fermat tiene dos formas posibles de enunciarlo, que son equivalentes (ver M 3.4.3 de la profesora tutora Idoia):

- p primo, a y p coprimos, entonces $a^{p-1} \equiv 1 (\text{mod } p)$
- p primo, entonces $a^p \equiv a (\text{mod } p)$

¿Es cierto que si existe a con $a^p \not\equiv a$ (mod p) $\Rightarrow$p es compuesto?

Sí. En el teorema pequeño de Fermat se da que A$\Rightarrow$B, por lo que noB $\Rightarrow$noA. En este caso se dice que a es un testigo para la composición de p.

¿Es cierto que si existe a con $a^p \equiv a$ (mod p) $\Rightarrow$p es primo?

No. En el teorema pequeño de Fermat se da que A $\Rightarrow$B, por lo que no se puede afirmar que noA$\Rightarrow$noB. Por ejemplo, $2^{341} \equiv 2$ (mod 341) y sin embargo 341 no es primo.

¿Es 4 un testigo de composición de 15? ¿Y 2 es un testigo de composición de 15?

4 no es un testigo de composición de 15 porque $4^{15} \equiv 4$ (mod 15). Sin embargo 2 sí es un testigo de composición de 15 porque $2^{15} \not\equiv 2$ (mod 15).

¿Existe algún número compuesto que no tenga ningún testigo de composición?

Sí. Son los números Carmichael. El más pequeño es 561.

¿Cuántos números Carmichael existen?

Aunque sólo hay 43 números Carmichael menores que un millón, sin embargo existen infinitos números de Carmichael según se probó en 1994.

Sobre el vector característico

¿Cuál es la definición de vector característico?

Dada una M_i, se define su vector característico $v_1,v_2,\ldots,v_j,\ldots$ con $v_j=1$ sii $w_j \in L(Mi)$ y $v_j=0$ sii $w_j \notin L(M_i)$, siendo $w_1=\epsilon$, $w_2=0$, $w_3=1$, $w_4=00$, $w_5=01$, $w_6=10$, $w_7=11$,$w_8=000$,….

¿La longitud del vector característico de la M_i viene dada por la expresión decimal de w_i?

No. La longitud del vector característico es infinita.

¿Si M_i tiene por vector característico (0,1,0,0,…) significa que M_i acepta todas las cadenas que empiezan por 0?

No. Se observa que la posición 4 del vector característico de M_i indica que M_i no acepta la cadena 00.

¿Si M_i verifica que $L(M_i)=(0+1)^+$, entonces su vector característico tiene todas sus componentes igual a 1?

No, porque para $w_1= \epsilon$ se tiene que la primera componente es igual a 0.

¿Existe alguna M_i que tenga un 1 en la coordenada i-ésima de su vector característico?

Sí, por ejemplo M_{2708} verifica que $L(M_{2708})=0(0+1)^*$, y dado que $w_{2708}=01010010100 \in L(M_{2708})$, se tiene que en la coordenada 2708 del vector característico hay un 1.

¿Los vectores característicos de las 2000 primeras máquinas de Turing tienen todas sus componentes nulas?

No. La M_{1354} que verifica que $L(M_{1354})=0(0+1)^*$ tiene por vector característico el dado por (0,1,0,1,1,…).