Tarea 4 Teoría de la computación, Thomas Gómez

punto 1: Considere el problema de determinar si un autómata finito determinista y una expresión regular son equivalentes. Exprese este problema como lenguaje y muestre que es decidible.

Solución:

El lenguaje se puede expresar como $L = \{< v > - e :< v >$ es el código para un autómata finito determinista $\wedge$ e es una expresión regular $\wedge$ $< v >$ y e son equivalentes.$\}$

Hecho de clase 1: Existe una máquina de Turing N que decide $L_{eq} = \{< M, N >$: M *y* N *son* son los códigos para autómatas finitos deterministas *y* $L(M) = L(N)\}$.

La siguiente máquina de Turing M con 3 cintas decide a L.
En input $< v > -e$:
1)Primero verifica que $< v >$ sea el código para un autómata finito determinista, si lo es sigue al paso 2; si no lo es rechaza.
2) Después verifica que e representa un lenguaje regular, si lo es sigue al paso 3,si no lo es rechaza.
3) Copia e a la 2^{da} cinta.
4) En la 2^{da} cinta M convierte la expresión regular al código $< w >$, donde $< w >$ corresponde al código para el autómata finito determinista equivalente para e.
5)Copia $< w >$ y $< v >$ a la 3^{ra} cinta como $< w, v >$.
6)En la 3^{ra} cinta corre la máquina N con input $< w, v >$. Finalmente decide en base a N, si N decide que $< w >=< v >$ acepta, de lo contrario rechaza.
Q.E.D

punto 2: Dado $A = \{< R, S >: (R \ \wedge \ S)$ son expresiones regulares $\wedge$ $L(R) \subseteq L(S)\}$ muestre que A es decidible.

Solución

Lema 1: $X \subseteq Y \longleftrightarrow X \setminus Y = \phi$.

Demostración:
$\longrightarrow$:
Suponga que X es subconjunto de Y, entonces trivialmente todo elemento de X está en Y, por ende $X \setminus Y = \phi$.
$\longleftarrow$:
Suponga que $X \setminus Y = \phi$, esto implica que no hay ningún elemento de X que no este en Y o equivalentemente que todo elemento de X está en Y, pero lo anterior es la definición de $X \subseteq Y$.
Q.E.D

Hecho de clase 2: Existe una máquina de Turing N que decide si $L(< v >) = \phi$ o no, donde $< v >$ es el código para un autómata finito determinista.

La siguiente máquina de Turing M con 5 cintas, decide el lenguaje A.
En input $< R, S >$:
1)Verifica que tanto $< R >$ como $< S >$ sean códigos para expresiones regulares si lo son sigue al paso 2, si no lo son rechaza.
2) Convierta la expresión $< R >$ en $< r >$ donde r es el código para el autómata finito determinista equivalente a $< R >$ y lo coloca en la 2^{da} cinta. Análogamente convierta la expresión $< S >$ en $< s >$, donde s es el código para el autómata finito determinista equivalente y copia $< s >$ a la 3^{ra} cinta.
3)Utilice la 4^{ta} para obtener el código $< v >$ donde $< v >$ es el código para el autómata finito determinista que decide $L(< r >) \setminus L(< s >)$.
4) Utilice la 4^{ta} cinta para correr la máquina de Turing N en input $< v >$. Si N decide que $< v >= \phi$ acepte,de lo contrario rechace.
Q.E.D

punto 3: Pruebe que EQ_{DFA} es decidible probando los 2 autómatas en todos las cadenas hasta cierto longitud. Calcule esa longitud.

Solución:

Nota: Mi respuesta se basa en la siguiente fuente:
https://cs.stackexchange.com/questions/92496/proving-that-dfa-equivalence-is-decidable
Sin embargo aquí explico con mis palabras, con más detalle y más rigurosamente la solución de la fuente externa, ya que entendí a la perfección la solución planteada en el enlace adjunto.
Lema 2: $X \Delta Y = \phi \longleftrightarrow X = Y$.
Demostración:
$\longrightarrow$:
Suponga que $X \Delta Y = \phi$ entonces por la definición de la diferencia simétrica X no tiene elementos que no estén en Y y Y no tiene elementos que no estén en X,es decir $w \in Y \iff w \in X$.Esta es la definición de igualdad de conjuntos,por ende $X = Y$.
$\longleftarrow$:
Suponga que $X = Y$ entonces $X \Delta Y = (X \setminus Y) \cup (Y \setminus X) = \phi \cup \phi = \phi$.
Q.E.D

Lema 3: Sea A un autómata finito determinista con n estados, sea B un autómata finito determinista con m estados. Entonces hay un autómata finito determinista C que tiene nxm estados y que reconoce a $L\Delta = L(A)\Delta L(B)$.
Este lema no se demuestra para acortar la solución:

Lema 4: Sea C un autómata finito determinista con p estados, tal que $L(C) \neq \phi$. Entonces $L(C)$ acepta al menos una cadena w de longitud menor o igual a $p - 1$.
Demostración:
Sea C un autómata finito determinista con p estados, tal que $L(C) \neq \phi$, entonces para recorrer cualquier estado de C,se necesitan a lo sumo $p - 1$ pasos(ya que se empieza por el estado inicial y hay que recorrer a lo sumo $p - 1$ estados que sobran). Al estado final se puede llegar entonces a lo sumo por p-1 estados, por lo tanto debe existir alguna cadena con longitud menor o igual a p-1 que este en $L(C)$.
Q.E.D

Teorema 1: Sea A un autómata finito determinista con n estados y sea B un autómata finito determinista con m estados. Entonces $L(A) = L(B)$ si y solo si para todas las cadenas de longitud menor o igual a $nxm - 1$ las cadenas son aceptadas por A y B o son rechazadas por A y B.
Demostración:
$\longrightarrow$:
Suponga que tanto $L(A)$ y $L(B)$ aceptan y rechazan a la vez todas las cadenas de longitud menor o igual a $nxm - 1$, esto implica que $L\Delta$ rechaza todas las cadenas de longitud menor o igual a $nxm - 1$, como $L\Delta$ tiene nxm estados podemos utilizar la contra-reciproca del lema 4 y establecer que $L\Delta = \phi$ que es equivalente por el lema 2 a que $L(A) = L(B)$.
$\longleftarrow$:
Por contra-reciproca. Suponga que existe por lo menos una cadena de longitud menor o igual $nxm - 1$, tal que A rechaza esta cadena y B acepta esta cadena o viceversa, evidentemente esto implica que $L(A) \neq L(B)$.
Q.E.D

En conclusión:
Un Algoritmo útil para probar si 2 autómatas son equivalentes es ver si todas las cadenas de longitud menor o igual al producto de sus números de estados menos 1 son rechazadas o aceptadas simultáneamente por estos 2 autómatas finitos deterministas, si se cumple lo anterior los autómatas son equivalentes, de lo contrario no.

Punto 4: Sea A un lenguaje Turing-reconocible que consiste en las descripciones de las máquinas de Turing $\{< M_1 >, < M_2 > ....\}$ donde cada $< M_i >$ es un decider(decide un lenguaje). Pruebe que algún lenguaje decidible D no es decidible por ningún decider M_i cuya descripción aparece en A.

Solución:

Hecho de clase 3: El conjunto de todas las máquinas de Turing es enumerable, como el conjunto de todas las máquinas de Turing decider es un subconjunto infinito de todas las máquinas de Turing, entonces el anterior subconjunto también es enumerable.

Hagamos una tabla para poder diagonalizar y hallar lo deseado en el ejercicio. En la horizontal enumeramos las máquinas de Turing deciders, de igual manera en la vertical colocamos las cadenas en binario organizadas por el orden lexicográfico(w_j es la cadena j en el orden lexicográfico), se cumple algo similar para la organización de las máquinas de Turing deciders. Sea además $a_{i,j} \in \{aceptar, rechazar\}$ que nos dice si la cadena w_i es aceptada o rechazada por la máquina decider M_j.

	M_1	$M_2 ...$	M_i
0	a_{11}	$a_{12} ...$	a_{1i}
1	a_{21}	$a_{22} ...$	a_{2i}
01	a_{31}	$a_{32} ...$	a_{3i}
.	.	.	.
.	.	.	.

.	.	.	.
w_j	a_{j1}	$a_{j2}...$	a_{ji}
.	.	.	.
.	.	.	.
.	.	.	.

Definamos un lenguaje D de la siguiente manera(w_i es la i-ésima cadena binaria ordenada en orden lexicográfico): $D = \{w_i \in \{0,1\}^*$: la máquina decider M_i rechaza a w_i $\}$.

Proposición 1: D no se puede decidir por ninguna máquina decider en A.

Demostración:
Por contradicción:
Suponga que alguna máquina de Turing decider $M_i \in A$ decide D, lo anterior significa que para toda cadena $w \in D$, la máquina de Turing $M_i \in A$ acepta w y que para toda cadena $v \in \{0,1\}^* \setminus D$, la máquina de Turing M_i rechaza v.
Ahora supongamos que $w_i \in D$, por lo anterior M_i acepta a w_i, pero al $w_i \in D$, w_i es rechazado por M_i.
Por último supongamos que $w_i \notin D$, por lo anterior M_i rechaza a w_i, pero al $w_i \notin D$, w_i es aceptado por M_i.
Lo anterior es equivalente a lo siguiente: w_i es aceptado por $M_i \longleftrightarrow w_i$ es rechazado por M_i. Evidentemente lo anterior es una contradicción.
Q.E.D

Punto 5: Considere el problema de determinar si una máquina de Turing con una cinta siempre escribe un símbolo blanco sobre un símbolo no blanco durante el curso de su computación en alguna cadena de entrada. Formule este problema como lenguaje y demuestre que no es decidible.

Solución:

El lenguaje se puede plantear como sigue: $L = \{< M >:< M >$es el código para una máquina de Turing y además para cada sucesión de configuraciones de una computación de M en un input arbitrario w, M hace lo siguiente:
Si la cabeza de M está sobre un símbolo que no sea el blanco y M no se encuentra en un estado de aceptación o rechazo, M escribe sobre ese símbolo el símbolo blanco $\}$

Hecho de clase 4: $L_\phi = \{< M >: < M >$ es el código para una máquina de Turing M y $|L(M)| = \phi)\}$ (en clase se nombró el anterior conjunto como L_{empty}), es indecidible.

Suponga que existe una máquina de Turing D que decide a L, el siguiente proceso construye una máquina de Turing N tal que N decide a L_ϕ o como se nombró en clase L_{empty}(pero L_ϕ es indecidible por hecho de clase 4 y así generamos una contradicción).En otras palabras: hacemos una reducción a L_ϕ.

Supongamos que D decide a L.
La siguiente máquina de Turing N con 2 cintas decide a L_ϕ:
En input $< M >$:

1)En su primera cinta obtiene el código para $< M' >$, donde M' es una máquina de Turing tal que siempre que M no este en estado de aceptación o rechazo(en input w), M' escribe sobre un símbolo no blanco un símbolo blanco(si está en símbolo blanco puede remplazar el símbolo blanco por cualquier símbolo arbitrario de la cinta), además si M entra en estado de aceptación, M' escribe un símbolo no blanco sobre la cinta y después remplaza ese símbolo no blanco por un simbolo no blanco y acepta después el input, si M entra en estado de rechazo, M' "trabaja una última vez como la mayoría de las veces" y rechaza el input.
2) En la segunda cinta, N corre a D en input $< M' >$; Si D acepta, N acepta y si D rechaza, N rechaza.

Proposición 2: L es indecidible.
Demostración:
Notemos que la máquina de Turing N decide a L_{ϕ}, a continuación se explica el por qué.
En input $< M >$ pasa lo siguiente:
caso 1: En la computación de N, notamos que si M tiene por lo menos una cadena que acepta, N rechaza a M(esto debido a que si acepta una cadena, $< M' >$ no siempre en todas las computaciones escribe símbolos blancos sobre símbolos no blancos de la cinta, así en el segundo paso D rechaza $< M' >$ y automáticamente N rechaza a $< M >$).
caso 2: Argumentos similares se cumplen para demostrar que cuando M no tiene ninguna cadena que acepta(caso 2), N acepta, se deja la comprobación al lector.
Por lo tanto N decide a L_{ϕ}, pero por hecho de clase 4 L_{ϕ} es indecidible, contradicción. Como supusimos la existencia de la máquina de Turing N, y esta a su vez se construyó con la suposición de la existencia de la máquina de Turing D y esto a su vez generó una contradicción.No debe de existir tal máquina D para evitar la anterior contradicción.
Q.E.D

Punto 6: Demuestre que existe un subconjunto indecidible de $\{1\}^*$.

Solución:

Demostración:
Note que $|\{1\}^*| = |\mathbb{N}|$, esto se puede justificar con el hecho de que $\{1\}^*$ puede ser expresado como la unión enumerable de conjuntos enumerables.
Por otro lado tenemos que $|\mathbb{P}(\{1\}^*)| = |\mathbb{P}(\mathbb{N})| = |\mathbb{R}| > |\mathbb{N}|$
Adicionalmente sabemos por lo explicado en clase que $|L_{comp}| = |\mathbb{N}|$.
Por lo anterior y por transitividad $|\mathbb{P}(\{1\}^*)| > |L_{comp}|$, por ende existirá algún elemento de $\mathbb{P}(\{1\}^*)$ que no sea computable, en otras palabras existirá algún subconjunto de $\{1\}^*$ que no sea decidible(decidible es lo mismo por definición que computable.)
Q.E.D

Punto 7: Demuestre que $L_{GLCdisyuntas} = \{< G, H >$:G y H son GLC y además $L(G) \cap L(H) \neq \phi\}$ es indecidible.

Solución:

Lema 1 del libro::Post Correspondence Problem, es indecidible.
Explicación:
El problema de correspondencia se puede imaginar de la siguiente manera:
Definamos primero a una instancia P como un conjunto de "fichas de domino"(pueden estar repetidas). En símbolos:

$$P = \{\frac{t_1}{b_1}....\frac{t_i}{b_i}\}$$

donde cada t_i y b_i son cadenas.
Adicionalmente definamos un match como una instancia tal que

$$t_1....t_i = b_1....b_i$$

Definamos a PCP (abreviatura para Post Correspondence Problem) como:
$PCP = \{< P >$: P es un match(una instancia con la propiedad definida anteriormente)$\}$.
Este problema es indecidible, ya que se puede utilizar para decidir el problema de si una máquina de Turing arbitraria acepta una cadena también arbitraria w. Desde la página 227 del libro Sipser está a fondo la explicación de porque este problema es indecidible.

Suponga que existe una máquina de Turing D, que decide a $L_{GLCdisyuntas}$.
El siguiente proceso decide a PCP.
Para una instancia arbitraria P:

$$P = \{\frac{t1}{b1}....\frac{ti}{bi}\}$$

Nota: Todos los a_i son símbolos terminales finales nuevos.
Defina la siguiente GLC G_P así:
$S \longrightarrow t_1Sa_1....|t_nSa_n|t_1a_1|....t_na_n$
Lema 5: $L(G) = \{w : w = a_ia_j....a_kt_it_j....t_k\}$
La demostración es trivial y se deja al lector.

Análogamente defina la siguiente GLC H_Pz así:
$S \longrightarrow b_1Ta_1....|b_nTa_n|b_1a_1|....b_na_n$
Lema 6: $L(H) = \{w : w = b_ib_j....b_ka_ia_j....a_k\}$
La demostración es trivial y se deja al lector.

Lema 7: $L(G) \cap L(H) \neq \phi \longleftrightarrow t_1....t_n = b_1....b_n$
Demostración:
$\longrightarrow$:
Por contrarrecíproca:
Suponga que $t_1....t_n \neq b_1....b_n$, como las 2 cadenas son diferentes, entonces debe existir por lo menos un $t_i \neq b_i$.
De la GLC S derive a t_ia_i, trivialmente esta cadena no se puede derivar con la GLC H.
$\longleftarrow$:
Suponga que $t_1....t_n = b_1....b_n$, entonces cada $t_i = b_i$ son iguales, por lo tanto la gramática H y G son iguales y por ende tienen el mismo lenguaje, de lo anterior se sigue $L(G) \cap L(H) \neq \phi$.

Teniendo a G_P y a H_P utilice a D para decidir si $L(G) \cap L(H) \neq \phi$. Si D decide que $L(G) \cap L(H) \neq \phi$ acepte, de lo contrario rechace.
Evidentemente el anterior algoritmo decide si una instancia es un match, por ende por la tesis de Church Turing existe una máquina de Turing M que decide a PCP.
Sin embargo esto es una contradicción ya que sabemos que PCP es indecidible, por ende nuestra hipótesis es falsa y no existe una máquina de Turing D que decida a $L_{GLCdisyuntas}$
$Q.E.D$

Punto 8: Definamos un lenguaje $L \subseteq \{0,1\}^*$ como especial si para todo $k \geq 2$, hay precisamente 2 cadenas de longitud k en L. Demuestre que si L es especial y computablemente enumerable o lo que es igual, Turing-reconocible;entonces L es decidible.

Solución

Hecho de clase 5: Si $L \subseteq \{0,1\}^*$ es Turing-reconocible y co-Turing-reconocible(El complemento de L es Turing-reconocible) entonces L es decidible.
Hecho de clase 6: Si $L \subseteq \{0,1\}^*$ es un lenguaje, estas 2 propiedades son equivalentes.
1) L es Turing reconocible.
2) Existe una máquina de Turing M que "hace una enumeración de L" en el siguiente sentido:
Si dejamos correr M sobre una cinta blanca(sin input), entonces M imprime sobre la cinta $w_1 - w_2 ... w_i$ donde $w_1 - w_2 ... w_i$ son todas las cadenas en L(si L es infinito, corre para siempre.)
A demás no imprime un - hasta que termine todo el trabajo para calcular el elemento de L que precede a -, así que en algún punto en la cinta, tal que sobre ella esté escrito $w_1 - w_2 ... w_i - ....$, sabemos que $w_1 - w_2 ... w_i \in$ L.
Hecho de clase 7: Si $L \subseteq \{0,1\}^*$ es un lenguaje finito, entonces L es decidible.
Hecho de clase 8: Si $L \subseteq \{0,1\}^*$ es decidible entonces L es reconocible.
Hecho de clase 10: Si $L1 \subseteq \{0,1\}^*$ y $L1 \subseteq \{0,1\}^*$ son reconocibles, entonces $L1 \cup L2$ es reconocible.

Suponga primero que L es Turing-reconocible y además es especial.
Sea $L' = \{0,1\}^* \setminus L$.
La siguiente máquina de Turing N con 2 cintas (en input cinta blanca) enumera a un subconjunto L'' de L', tal que L'' contiene solo elementos de longitud mayor o igual a 2 de L'.
En input cinta vacía, N (con su última cinta) simula a la máquina M que enumera a L según el hecho de clase 6,sin embargo modificamos esta simulación para que solo imprima cadenas de longitud mayor o igual a 2.
Siga el siguientes paso:
Empiece con $k = 2$, y espere que en la simulación de M se impriman las 2 cadenas de longitud 2 que están en L , después agregue a la primera cinta todas las cadenas de longitud 2, excepto las 2 anteriores. Repita este proceso en orden para todos los k mayores que 2.

Podemos notar que esta máquina de Turing N enumera a L'' en input cinta blanca; por lo tanto por el hecho de clase 6, L'' es Turing-reconocible.
Definamos $L''' = L' \setminus L''$, evidentemente L''' es finito, ya que solo contiene las cadenas de longitud menor a 2 que están en L', por lo tanto L''' es decidible, lo que implica que L''' es Turing-reconocible.
Notemos también que $L'' \cup L''' = L'$ y por el hecho de clase 10, L' es Turing-reconocible.
Como L es Turing-reconocible y $L' = \{0, 1\}^* \setminus L$ es Turing-reconocible o equivalentemente L también es co-Turing-reconocible, por el hecho de clase 5 L es decidible.
Q.E.D