Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Dr. Juan Reutter

IIC2213 — Lógica para Ciencia de la Computación — Diego Iruretagoyena

Tarea 3 - Lógica preposicional

Sean $M = (Q, A, B, q_0, F, \delta)$ una máquina de turing estrictamente lineal y $w = a_1, ..., a_n$ una palabra sobre A de largo igual o mayor a 1. Explica como construir un conjunto de fórmulas en lógica proposicional tal que es satisfacible si y solo si M acepta a w. Nota que la máquina M y la palabra w son arbitrarias, tu construcción debe servir para cualquier máquina M y palabra w.

Para lograr **construir un conjunto de fórmulas $\sum$ en lógica proposicional tal que $\sum \in$ SAT ssi M acepta a w**, necesitamos nombrar ciertos conceptos. M acepta a w luego de $|w|$ pasos. $\sum$ debe representar el mundo de las ejecuciones válidas. Debemos ganar información suficiente y reducir los mundos indeseados. Debemos crear proposiciones y fórmulas que acoten el espacio. **Que M acepte a w significa que el estado en que se detuvo la máquina era un estado final.**

Tenemos $M = (Q, A, B, q_0, F, \delta)$ máquina de Turing (i.e. una tupla de elementos con Q conjunto de estados, A alfabeto, $q_0 \in Q$ estado inicial, $F \subseteq Q$ estados finales y δ función de transición) **estrictamente lineal**. No se fija si es determinista o no determinista, por lo que debemos representar ambas posibilidades.

Para maquinas no-deterministicas, el largo de la ejecución es el largo de la rama más corta que acepta a una palabra, si M acepta a w, o bien el largo de la rama más corta, cuando M no acepta a w.**L(MTND)** sobre alfabeto A se define como la existencia de una ejecucion de la máquina M con input w que se detiene en un estado final.

Además, nos dan una palabra $w = a_1, ..., a_n$ una palabra sobre A con $|w| >= 1$. Si es estrictamente lineal, significa que para M sobre A, cada palabra w (i.e. largo palabra) sobre A de largo igual o mayor a 1, la ejecución 1 de M sobre w tiene exactamente $|w|$ pasos.

Para un conjunto finito $\sum = \{\chi_1, ..., \chi_n\}$ de formulas, denotamos $\bigvee \sum$ para indicar $\chi_1 \vee ... \vee \chi_n$. Esta expresión simboliza la disyunción y se suele utilizar paraindicar que al menos un bit del vector será diferente a cero, ratificando que al menos un elemento será válido.

Similarmente para un conjunto finito $\sum = \{\chi_1, ..., \chi_n\}$ de formulas, denotamos $\bigwedge \sum$ para indicar $\chi_1 \wedge ... \wedge \chi_n$. Esta expresión simboliza la conjunción. Usamos la conjunción para "iterar" sobre todos los elementos.

Dados estados $q \in Q$, ejecuciones $1 \leq t \leq |w|$ definimos proposiciones $S_{s(q,t)}$, $T_{(q,q',t,t')}$, $F_{q,t}$, $V_{ai,t}$ y $R_{ai,q,q'}$. La intuición es que una valuación de esas variables representa una ejecucíon de w sobre M, de forma que las variables con valuación 1 tienen la siguiente intuición

- $S_{s(q,t)}$, en donde valuación $\tau(s_t) = 1$ significa que el estado de la configuracion t es q.
- $T_{(q,q',t,t')}$, en donde valuación $\tau(T_{(t,q)}) = 1$ significa que en el paso de ejecución de t a t', con $t \leq t' \leq w$ y $0 \leq t \leq w$, es posible un cambio de estado o se mantuvo en el mismo, con $q, q' \in Q$.
- $F_{q,t}$ representa el chequeo del estado q en la ejecución t, en donde valuación $\tau(F_q) = 1$ q es estado final y de aceptación.
- $V_{ai,t}$, leo caracter a_i en ejecución j, en donde valuación $\tau(V_a) = 1$ implica caracter posición j válido.
- $R_{ai,q,q'}$ es que leo caracter a_i en estado q, en donde valuación $\tau(V_a) = 1$ significa que leer dicho caracter en estado q me lleva a q'.

Luego, definimos nuestras fórmulas. Construimos $\sum \boldsymbol{= \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6 \wedge \phi_7}$ que acote nuestro espacio de mundos posibles y representen ejecuciones válidas de la palabra w sobre máquina M.

$\boldsymbol{\phi_1}$ asegura que los modelos de $\sum$ hay exactamente un estado en cada configuracion, es decir, para cada t, solo un s(q,t) puede ser verdad.

$$\phi_1 = \bigwedge_{1 \leq t \leq n} \bigvee_{q_1 \in Q} (s_{(q_1,t)} \vee \bigwedge_{q_2 \in Q, q_1 \neq q_2} \neg s_{(q_2,t)})$$

$\boldsymbol{\phi_2}$ $s_{(q_0,1)}$ nos asegura que configuración inicial MT sea válida.

$$\phi_2 = s_{(q_0,0)}$$

$\boldsymbol{\phi_3}$ hace que la última configuración, en la posición n, sea un estado final. Necesitamos esto ya que nuestra máquina debe aceptar palabras de largo w.

$$\phi_3 = \bigvee_{q \in F} s_{(q,n)}$$

Ahora tenemos que definir el concepto de las transiciones.

$$\delta : Qx(A \cup B) \rightarrow 2^{Qx(A \cup B)x\{\rightarrow, \leftarrow, \cdot\}}$$

Definimos las transiciones como movimientos en la cinta. Puede ser hacia la derecha, izquierda o quedarse.

$$D = \delta(q, a)$$

$$\bigwedge_{(q,a) \rightarrow D \in \delta} \bigvee (q', a', \{\rightarrow, \leftarrow, \cdot\}) \in D$$

$$s_{(q,a)} \vee R_{(a_i,a_j,t,\rightarrow)} \vee R_{(a_i,a_k,t,\leftarrow)} \vee R_{(a_i,a_i,t,\cdot)}$$

$\boldsymbol{\phi_4}$ asegura que los modelos de $\sum$ existe una forma de la MT para moverse en alguna dirección (o quedarse quieta). Existe una forma de cambiar de estado, i.e. **transiciones**. Para cada estado q, existe una transición posible a otro estado q, distinto o igual al actual. Dada la definición de la función transición para máquinas no deterministas.

$$\phi_4 = \bigwedge_{1 \leq t \leq n} \bigwedge_{q \in Q} \bigvee_{q' in Q} \bigvee_{a_i in A} (s(q', t) \wedge V_{a_i,t} \wedge R_{a_i,q,q'} \wedge T_{(q,q',t,t')})$$

$\boldsymbol{\phi_5}$ asegura que en cada transición donde lea caracter válido, llegaré a un estado que pertenezca a Q, es decir un estado válido, y no habrán movimientos que me lleven a estados inválidos o no definidos.

$$\phi_5 = \bigwedge_{1 \leq t \leq n} \bigvee_{q_1 \in Q} \bigvee_{a_i \in A} V_{ai,t} \wedge (\bigvee_{q_1 \in Q} \bigvee_{t \leq t' \leq |w|} T_{(q,q',t,t')}) \wedge \bigwedge_{q_2 \notin Q} \neg S_{s(q',t)}$$

Además, debe existir alguna secuencia de ejecuciones que sea válida, a partir del estado inicial y que llegue a un estado final o de aceptación.

$\boldsymbol{\phi_6}$ hace que deba existir alguna secuencia válida que llegue a un estado de aceptación, partiendo desde q_0 en w pasos. Con esto nos aseguramos que nuestras transiciones nos logran llevar efectivamente a algún estado que sea de aceptación.

$$\phi_6 = s_{(q_0,0)} \wedge (\bigvee_{q_1 \in Q} \bigvee_{q_2 \in F} \bigvee_{t \leq t' \leq |w|} T_{(q,q',t,t')} \wedge F_{(q_2,t)}) \wedge \bigwedge_{1 \leq t \leq n} \bigvee_{q_1 \in Q} F_{q_1,t}$$

$\boldsymbol{\phi_7}$ la ejecución de M sobre w tiene exactamente $|w|$ pasos al ser estrictamente lineal.

$$\phi_7 = \bigwedge_{1 \leq t \leq n} \bigvee_{q_1 \in Q} \bigvee_{q_2 \in Q} (V_{ai,t}) \wedge T_{(q,q',t,t')} \wedge \bigwedge_{n+1 \leq t_2} \neg V_{ai,t_2}$$

Demostración claúsula si y solo si.

Construimos $\sum = \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6 \wedge \phi_7$, por demostrar que $\sum$ es satisfacible si y solo si M acepta a w i.e. termina en estado final.

$\rightarrow$ Por contradicción suponemos que $\sum \in$ SAT pero no existe una codificación w que al correr sobre M llegue a un estado final o de aceptación. Podríamos entonces suponer que al partir de q_0 y después de w pasos podemos llegar a un q que no pertenezca al conjunto F. Pero si partimos de un estado inicial valido, con secuencias validas y M no aceptó a w, llegamos a una **contradicción debido a la forma en que construimos el problema**. Si pudiesemos obtener una codificacion para la cual w sobre M, no llegamos a estados aceptados, significa que existe alguna forma de satisfacer cada propiedad pero llegar a un estado que no pertenece a estados finales desde el estado inicial. Esto contradice además ϕ_6.

$\leftarrow$ Supongamos que M acepta w pero $\sum$ no pertenece a SAT. Si M acepta a w significa que todas las valuaciones para $V_{ai,t}$ fueron correctas y que llegamos a un estado final o de aceptación. Sin embargo, no pertenece a SAT es decir alguna de las clausulas no se cumple. Si pudimos tener transiciones correctas en la máquina, partiendo desde q_0 entonces cumplimos ϕ_4 y ϕ_6. Si todas estas fueron correctas significa debido a nuestra construcción que $T_{(q,q',t,t')}$, $S_{s(q,t)}$ tienen valuaciones posibles, entonces son configuraciones con asignación única. Hemos cumplido ϕ_1, ϕ_2 y ϕ_3. ϕ_7 por construcción de enunciado es verdadero. Pero eso implica que $\sum$ es satisfacible y llegamos a una contradicción.

Finalmente, por $\rightarrow$ y $\leftarrow$ se tiene que ϕ es satisfacible $\leftrightarrow$ M acepta w. $\blacksquare$