UNIVERSIDAD DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

Inteligencia Artificial - CC52A
Guía 2

Los siguientes ejercicios tratan acerca de la lógica de primer orden:

1. Sea $\mathcal{L} = \{E(\cdot,\cdot)\}$ un lenguaje utilizado para representar grafos. En cada una de las siguientes preguntas escriba una $\mathcal{L}$-oración que represente la propiedad mencionada. En otras palabras, escriba una $\mathcal{L}$-oración ϕ tal que una $\mathcal{L}$-estructura $\mathcal{A}$ satisface ϕ si y sólo si $\mathcal{A}$ tiene la propiedad mencionada.

   (a) El grafo es un clique.
   (b) El grafo contiene un clique con 4 nodos.
   (c) El grafo tiene un ciclo con 4 nodos.
   (d) Existen elementos en el grafo cuya distancia es 4.
   (e) La distancia máxima entre dos nodos del grafo es 3.

2. Sea $\mathcal{L} = \{f(\cdot), c\}$, donde f es una función unaria, y sean φ_1, φ_2, φ_3 las siguientes $\mathcal{L}$-oraciones:

$$\begin{aligned}\varphi_1 &= \forall x \forall y\ (f(x) = f(y) \rightarrow x = y),\\ \varphi_2 &= \forall x (f(x) \neq c),\\ \varphi_3 &= \forall x (x \neq c \rightarrow \exists y\ f(y) = x).\end{aligned}$$

   (a) Muestre una $\mathcal{L}$-estructura $\mathcal{A}_1$ tal que $\mathcal{A}_1 \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3$.
   (b) Muestre una $\mathcal{L}$-estructura $\mathcal{A}_2$ tal que $\mathcal{A}_2 \models \neg\varphi_1 \wedge \varphi_2 \wedge \varphi_3$.
   (c) Muestre una $\mathcal{L}$-estructura $\mathcal{A}_3$ tal que $\mathcal{A}_3 \models \varphi_1 \wedge \neg\varphi_2 \wedge \varphi_3$.
   (d) Muestre una $\mathcal{L}$-estructura $\mathcal{A}_4$ tal que $\mathcal{A}_4 \models \varphi_1 \wedge \varphi_2 \wedge \neg\varphi_3$.

3. Al igual que en la lógica proposicional, decimos que dos oraciones φ y ψ son *equivalentes*, denotado como $\varphi \equiv \psi$, si para toda estructura $\mathcal{A}$ se tiene que: $\mathcal{A} \models \varphi$ si y sólo si $\mathcal{A} \models \psi$.

   ¿Cuáles de las siguientes afirmaciones son válidas? Justifique su respuesta.

   (a) $\forall x\, \varphi \;\equiv\; \neg\exists x \neg\varphi$.
   (b) $\exists x\, \varphi \;\equiv\; \neg\forall x \neg\varphi$.
   (c) $\exists x\, (\varphi \vee \psi) \;\equiv\; (\exists x\, \varphi) \vee (\exists x\, \psi)$.
   (d) $\forall x\, (\varphi \vee \psi) \;\equiv\; (\forall x\, \varphi) \vee (\forall x\, \psi)$.
   (e) $\exists x\, (\varphi \wedge \psi) \;\equiv\; (\exists x\, \varphi) \wedge (\exists x\, \psi)$.
   (f) $\forall x\, (\varphi \wedge \psi) \;\equiv\; (\forall x\, \varphi) \wedge (\forall x\, \psi)$.

4. Al igual que en la lógica proposicional, decimos que una oración φ es *consecuencia lógica* de un conjunto de oraciones Σ, denotado como $\Sigma \models \varphi$, si para cada estructura $\mathcal{A}$ tal que $\mathcal{A} \models \Sigma$, se tiene que $\mathcal{A} \models \varphi$.

   ¿Cuáles de las siguientes afirmaciones son válidas? Justifique su respuesta.

   (a) $\{\forall x \exists y\, R(x,y)\} \models \exists x \forall y\, R(x,y)$.
   (b) $\{\exists x \forall y\, R(x,y)\} \models \forall x \exists y\, R(x,y)$.
   (c) $\{\forall x \forall y\, (R(x,y) \rightarrow R(y,x)),\ \forall x \forall y \forall z\, ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))\} \models \forall x\, R(x,x)$.
   (d) $\{\exists x\, (P(x) \wedge Q(x))\} \models (\exists x\, P(x)) \wedge (\exists x\, Q(x))$.
   (e) $\{\exists x\, P(x),\ \exists y\, Q(y))\} \models \exists x\, (P(x) \wedge Q(x))$.
   (f) $\{\forall x \exists y\, S(x,y)\} \models \exists y\, S(y,y)$.

5. Sean α y β dos $\mathcal{L}$-fórmulas. Demuestre que

$$\{\ \forall x(\alpha \rightarrow \beta),\ \forall x \alpha\ \} \models \forall x \beta.$$

6. Demuestre que ninguna de las oraciones siguientes es consecuencia lógica de las dos restantes:

   - $\forall x \forall y \forall z (P(x,y) \wedge P(y,z) \rightarrow P(x,z))$;
   - $\forall x \forall y (P(x,y) \wedge P(y,x) \rightarrow x = y)$;
   - $\forall x \forall y (P(a,x) \rightarrow P(y,b))$.

7. Sea $\mathcal{L}$ un vocabulario que consiste de una función binaria f y una constante e.

   a. Construya una $\mathcal{L}$-oración ϕ tal que $\mathcal{A} \models \phi$ si y sólo si $\mathcal{A}$ es un grupo, para toda $\mathcal{L}$-estructura $\mathcal{A}$.
   b. Demuestre que $\phi \models \forall x \forall y \exists z (f(x,z) = y)$.
   c. Demuestre que $\phi \not\models \forall x \forall y (f(x,y) = f(y,x))$.

8. Construya una oración ϕ_n tal que $\mathcal{A} \models \phi_n$ si y sólo si el dominio de $\mathcal{A}$ tiene al menos n elementos, $n \geq 1$.

9. Construya una oración ψ_n tal que $\mathcal{A} \models \psi_n$ si y sólo si el dominio de $\mathcal{A}$ tiene exactamente n elementos, $n \geq 1$.

10. Sea $\mathcal{L}$ un vocabulario que consiste de una función binaria f y una constante e. Sea ϕ la oración tal que $\mathcal{A} \models \phi$ si y sólo si $\mathcal{A}$ es un grupo, para toda $\mathcal{L}$-estructura $\mathcal{A}$.

    a) Demuestre <u>usando resolución</u> que $\forall x \forall y \exists z (f(x,z) = y)$ es consecuencia lógica de ϕ.
    b) Demuestre <u>usando resolución</u> que $\forall x \forall y \exists z \forall z (f(x,y) = f(x,z) \rightarrow y = z)$ es consecuencia lógica de ϕ.

    Es necesario primero skolemizar las oraciones, luego pasarlas a forma de cláusula, y por último integrar los axiomas para la igualdad.

11. Demuestre usando resolución que el siguiente conjunto Σ de fórmulas de la lógica de primer orden es insatisfacible (i.e. demuestre que la cláusula vacía se puede obtener mediante resolución desde Σ):

$$\Sigma = \{\forall x \exists y L(x,y),\ \forall x(\exists y L(x,y) \rightarrow \forall z L(z,x)),\ \exists x \exists y \neg L(x,y)\}$$

Para ello deberá primero convertir las fórmulas a forma normal prenex y luego skolemizar para eliminar los cuantificadores existenciales.

12. Sea $\mathcal{L}$ un vocabulario que consiste de una relación binaria R.

    a) Construya una oración ϕ que se satisface sólo en aquellas $\mathcal{L}$-estruturas donde la relación R es total (es decir, para cada elemento a existe un elemento b tal que $R(a,b)$), simétrica, y transitiva.

    b) Demuestre usando resolución que $\forall x R(x,x)$ es consecuencia lógica de ϕ.

13. Demuestre que al algoritmo visto en clases para computar el umg de un conjunto de expresiones es correcto y completo, i.e. que su salida es un unificador cuando éste existe, y que además en ese caso el unificador computado es un umg.

14. Para los siguientes conjuntos de expresiones, determine si existe un umg o no, y encuéntrelo cuando existe.

    a) $\{P(f(y), w, g(z)),\ P(u, u, v)\}$;

    b) $\{P(f(y), w, g(z)),\ P(v, u, v)\}$;

    c) $\{P(a, x, f(g(y))),\ P(z, h(z, w), f(w))\}$;

15. Una sustitución θ es *idempotente* si $\theta = \theta\theta$.

    Sea $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$, y asuma que V es el conjunto de variables mencionadas en $\{t_1, \ldots, t_n\}$. Demuestre que θ es idempotente si y sólo si $\{x_1, \ldots, x_n\} \cap V = \emptyset$.

16. Demuestre que todo umg producido por el algoritmo visto en clases es idempotente.

17. Asuma que θ_1 y θ_2 son sustituciones, y que existen sustituciones σ_1 y σ_2 tales que $\theta_1 = \theta_2\sigma_1$ y $\theta_2 = \theta_1\sigma_2$.

    Demuestre que existe una sustitución γ que es variable pura (es decir, es de la forma

    $$\{x_1/t_1, \ldots, x_n/t_n\},$$

    donde cada t_i, $1 \leq i \leq n$, es una variable), tal que $\theta_1 = \theta_2\gamma$.

18. Sea Σ un conjunto de oraciones. Demuestre que existe un conjunto Σ' de cláusulas tal que Σ es satisfacible si y sólo si Σ' es satisfacible.

19. Encuentre un conjunto de cláusulas Σ, tal que la intersección de todos sus modelos de Herbrand no es un modelo de Σ.

    Sea P un programa definitivo, y asuma que la base de Herbrand asociada a P es B.

a) Demuestre que la intersección M_P de todos los modelos de Herbrand de P también es un modelo de Herbrand de P.

b) Demuestre que $M_P = \{A \in B \mid P \models A\}$.

20. Recuerde que la clase de las expresiones regulares sobre alfabeto finito Σ se define recursivamente como sigue:

    a) $\emptyset$ y ε son expresiones regulares sobre Σ;

    b) para cada $a \in \Sigma$ se tiene que a es una expresión regular sobre Σ;

    c) si e_1 y e_2 son expresiones regulares sobre Σ, entonces $(e_1 \cup e_2)$ y $(e_1 \cdot e_2)$ son expresiones regulares sobre Σ;

    d) si e es una expresión regular sobre Σ, entonces e^* es una expresión regular sobre Σ.

    Por tanto, podemos pensar cada expresión regular sobre alfabeto Σ como un término sin variables libres sobre el vocabulario que contiene a las constantes $\emptyset$, ε, y $\{a \mid a \in \Sigma\}$, a las funciones binarias $(\cup)$ y $(\cdot)$, y a la función unaria $(\,)^*$.

    Con cada expresión regular e sobre Σ asociamos un subconjunto $L(e)$ de Σ^* (el *lenguaje regular asociado con la expresión* e). Este se define recursivamente de la siguiente forma:

    a) $L(\emptyset) = \emptyset$, y $L(\varepsilon) = \varepsilon$, asumiendo que ε es la palabra vacía;

    b) $L(a) = \{a\}$;

    c) $L(e_1 \cup e_2) = L(e_1) \cup L(e_2)$, y $L(e_1 \cdot e_2) = \{w \in \Sigma^* \mid w = w_1 w_2,\ w_1 \in L(e_1),\ w_2 \in L(e_2)\}$, asumiendo que $w_1 w_2$ es la concatenación de las palabras w_1 y w_2;

    d) $L(e^*) = \{\varepsilon\} \cup \{w \in \Sigma^* \mid \text{para algún } n \geq 1,\ w = w_1 w_2 \ldots w_n \text{ y para cada } 1 \leq i \leq n,\ w_i \in L(e)\}$.

    Para cada uno de los siguientes predicados construya un programa en PROLOG que lo defina:

    (a). El predicado $primero(x, y)$, que se satisface si y sólo si el símbolo x pertenece al siguiente conjunto: $\{a \in \Sigma \mid \text{ existe } w \in \Sigma^* \text{ tal que } aw \in L(y)\}$.

    (b). El predicado $ultimo(x, y)$, que se satisface si y sólo si el símbolo x pertenece al siguiente conjunto: $\{a \in \Sigma \mid \text{ existe } w \in \Sigma^* \text{ tal que } wa \in L(y)\}$.

    (c). El predicado $siguiente(x, y, z)$, que se satisface si y sólo si el par de símbolos (x, y) pertenece al siguiente conjunto: $\{(a, b) \in \Sigma \times \Sigma \mid \text{ existen } w_1, w_2 \in \Sigma^* \text{ tal que } w_1 ab w_2 \in L(z)\}$.

    Para esto puede asumir que tiene disponible un predicado $V(x)$ que se satisface si y sólo si $\varepsilon \in L(x)$.

21. Sea Σ un alfabeto finito. Note que los strings en Σ^* pueden verse como los términos sin variables libres sobre el vocabulario $\mathcal{L}$ tal que (1) $\mathcal{L}$ contiene como constantes a todos los elementos de Σ más el string vacío ϵ, y (2) $\mathcal{L}$ contiene como única función a la función binaria $\cdot$ tal que $w \cdot w'$ representa la concatenación de w y w'. Asuma además que $\mathcal{L}$ contiene un predicado binario $\texttt{Reverse}(\cdot, \cdot)$.

    Construya un programa definitivo P (i.e. un conjunto de cláusulas de Horn) – que solo ocupe a términos del vocabulario $\mathcal{L}$, pero que pueda ocupar más predicados que tan solo `Reverse`

– tal que si M_P es el modelo mínimo de P entonces $\texttt{Reverse}(t, t') \in M_P$ si y solo si t, t' son strings en Σ^* y t' es el string *inverso* de t, i.e. t' es el string que se obtiene al leer t de derecha a izquierda.

22. Sea Σ un alfabeto finito. Note que los strings en Σ^* pueden verse como los términos sin variables libres sobre el vocabulario $\mathcal{L}$ tal que (1) $\mathcal{L}$ contiene como constantes a todos los elementos de Σ más el string vacío ϵ, y (2) $\mathcal{L}$ contiene como única función a la función binaria $\cdot$ tal que $w \cdot w'$ representa la concatenación de w y w'. Asuma además que $\mathcal{L}$ contiene un predicado unario Pal.

    Construya un programa definitivo P (i.e. un conjunto de cláusulas de Horn) – que solo ocupe a términos del vocabulario $\mathcal{L}$, pero que pueda ocupar más predicados que tan solo Pal – tal que si M_P es el modelo mínimo de P entonces $Pal(t) \in M_P$ si y solo si t es un string en Σ^* que es un palíndrome.

23. Estudie PROLOG con detención, tanto los problemas presentados en clase y ayudantía como otros que pueda encontrar en la web (la página

    `http://wiki.python.org/moin/ProblemSets/99_Prolog_Problems_Solutions`

    es particularmente interesante).