PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Lógica para Ciencia de la Computación - IIC2213
Guía 1 - Lógica proposicional

1. Dados dos strings w_1 y w_2, se dice que w_1 es un sub-string de w_2 si existen strings u y v tales que $uw_1v = w_2$. Por ejemplo, *aa* es un sub-string de *bbaacb*, *aacc*, *bcaa* y *aa*. Dada una fórmula φ, la lista de sub-fórmulas de φ se define como la lista de strings ψ tal que: ψ es una fórmula y ψ es un sub-string de φ. Por ejemplo, la siguiente es la lista de sub-fórmulas de $((p \vee q) \rightarrow (p \wedge (\neg r)))$:

   $((p \vee q) \rightarrow (p \wedge (\neg r)))$
   $(p \vee q)$
   p
   q
   $(p \wedge (\neg r))$
   p
   $(\neg r)$
   r

   Defina de manera inductiva la función $\text{nsf}(\varphi)$ que retorna la cantidad de elementos en la lista de sub-fórmulas de φ. Por ejemplo, $\text{nsf}(((p \vee q) \rightarrow (p \wedge (\neg r)))) = 8$.

2. Defina $\text{la}(\alpha)$ como el largo de la fórmula proposicional α. Demuestre que para cada fórmula φ se tiene que $\text{nsf}(\varphi) \leq \text{la}(\varphi)$.

3. Sea $P = \{p, q\}$. Defina el conjunto de números naturales LARGO de la siguiente forma:

   $$\text{LARGO} \quad = \quad \{n \in \mathbb{N} \mid \text{existe una fórmula } \beta \in L(P) \text{ tal que } \text{la}(\beta) = n\},$$

   donde $\text{la}(\beta)$ es el largo de β (ver ejercicio 2). Demuestre que $\text{LARGO} = \{1, 4, 5\} \cup \{n \in \mathbb{N} \mid n \geq 7\}$.

4. Sea EQ un conectivo ternario definido como $\text{EQ}(p, q, r) = 1$ si y sólo si $3 \cdot p - 2 \cdot (q + r) \geq 0$. Defina el conectivo EQ utilizando los conectivos $\wedge$, $\vee$ y $\neg$.

5. Formalice el siguiente argumento en el cálculo proposicional:

   > "Si Superman fuera capaz y deseara prevenir el mal, entonces lo haría. Si Superman fuera incapaz de prevenir el mal, entonces sería impotente, y si no deseara prevenir el mal, entonces sería malévolo. Si Superman existe, no es ni impotente ni malévolo. Superman no previene el mal. Entonces, Superman no existe."

   Demuestre usando tablas de verdad que Superman no existe.

6. Dada una matriz C de 3×3 que contiene números entre 0 y 3, decimos que C es *completable* si es que existe una manera de reemplazar los números 0 por números entre 1 y 3 de tal forma que la suma de cada fila y de cada columna es la misma. Por ejemplo, la siguiente matriz es completable:

2	0	0
0	2	0
0	0	3

puesto que podemos reemplazar los valores 0 por los siguientes valores:

2	2	1
2	2	1
1	1	3

de manera tal que la suma de cada fila y de cada columna es 5. En cambio, la siguiente matriz no es completable:

1	1	1
0	0	0
3	0	0

Dada una matriz C de 3×3, construya una fórmula φ en lógica proposicional tal que C es completable si y sólo si φ es satisfacible. En particular, φ tiene que ser construida de tal forma que cada valuación σ que satisface a φ represente una forma de completar C.

7. Decimos que una fórmula φ está en 3-CNF si φ está en CNF y cada una de sus claúsulas contiene a lo más tres literales. Por ejemplo, $(p \vee q \vee \neg r) \wedge (\neg p \vee s)$ está en 3-CNF mientras que $(p \vee \neg q \vee \neg r \vee s)$ no está en 3-CNF.

   Demuestre que existen fórmulas que no son equivalentes a ninguna fórmula en 3-CNF.

8. Decimos que una fórmula φ está en k-CNF ($k \geq 2$) si φ está en CNF y cada una de sus claúsulas contiene a lo más k literales. ¿Existe algún valor de k para el cual toda fórmula es equivalente a una fórmula en k-CNF?

9. En esta pregunta usted va a demostrar que existen fórmulas proposicionales para las cuales toda fórmula equivalente en DNF es de tamaño exponencial.

   Suponga que $n \geq 1$ es un número fijo. Defina $P_n = \{x_1, \ldots, x_n, y_1, \ldots, y_n\}$ como un conjunto compuesto de $2 \cdot n$ variables proposicionales distintas, y defina φ_n de la manera siguiente:

$$\varphi_n \quad = \quad (x_1 \vee y_1) \wedge \cdots \wedge (x_n \vee y_n).$$

   Además, suponga que ψ_n es la fórmula en DNF más corta tal que $\varphi_n \equiv \psi_n$. Vale decir, para toda fórmula θ_n en DNF tal que $\varphi_n \equiv \theta_n$, se tiene que $\mathrm{la}(\psi_n) \leq \mathrm{la}(\theta_n)$, donde $\mathrm{la}(\alpha)$ es el largo de una fórmula α (ver ejercicio 2). En el resto de esta pregunta usted va a demostrar que $\mathrm{la}(\psi_n) \geq 2^n$, es decir, en esta pregunta usted va a demostrar que el largo de ψ_n es exponencial en el largo de φ_n, dado que $\mathrm{la}(\varphi_n) = 6 \cdot n - 1$.

   Defina Γ_n como el siguiente conjunto de asignaciones:

$$\Gamma_n \;=\; \{\sigma : P_n \to \{0,1\} \mid \text{para cada } i \in \{1,\ldots,n\} \text{ se tiene que } (\sigma(x_i) = 1 \text{ o } \sigma(y_i) = 1) \text{ y } (\sigma(x_i) = 0 \text{ o } \sigma(y_i) = 0)\}.$$

   Vale decir, Γ_n es el conjunto de valuaciones σ tal que para cada $i \in \{1, \ldots, n\}$, se tiene que σ asigna el valor 1 a x_i o a y_i, pero no a ambos. Nótese que cada valuación en Γ_n satisface a φ_n, y este conjunto

está compuesto por 2^n asignaciones. Además, suponga que $\psi_n = (D_1 \vee \cdots \vee D_\ell)$, donde cada D_i $(1 \leq i \leq \ell)$ es una conjunción de literales, y suponga que $f : \Gamma_n \to \{1, \ldots, \ell\}$ es una función tal que para cada $\sigma \in \Gamma_n$ se tiene que $\sigma(D_{f(\sigma)}) = 1$. Nótese que esta función f existe ya que para cada $\sigma \in \Gamma_n$ se tiene que $\sigma(\psi_n) = 1$ (dado que $\sigma(\varphi_n) = 1$ y $\varphi_n \equiv \psi_n$). En el resto de esta pregunta usted va a demostrar que f es una función inyectiva, de lo cual se concluye en primer lugar que $\ell \geq |\Gamma_n|$, y en segundo lugar que $\mathrm{la}(\psi_n) \geq 2^n$ ya que $\mathrm{la}(\psi_n) \geq \ell$ y $|\Gamma_n| = 2^n$.

(a) Sea $i \in \{1, \ldots, \ell\}$. Demuestre que D_i es una fórmula consistente (vale decir, no tiene literales complementarios).

(b) Sea $i \in \{1, \ldots, \ell\}$. Demuestre que para cada $j \in \{1, \ldots, n\}$, se tiene que x_j es uno de los literales mencionados en D_i (vale decir, D_i es de la forma $\alpha \wedge x_j \wedge \beta$) o y_j es uno de los literales mencionados en D_i (o ambos son mencionados, no se pide en esta parte demostrar que sólo uno de los dos es mencionado).

(c) Suponga que la función f no es inyectiva, vale decir, suponga que existen $\sigma_1 \in \Gamma_n$ y $\sigma_2 \in \Gamma_n$ tales que $\sigma_1 \neq \sigma_2$ y $f(\sigma_1) = f(\sigma_2) = i$. Obtenga una contradicción utilizando (b) y los siguientes hechos: $\sigma_1 \in \Gamma_n$, $\sigma_2 \in \Gamma_n$, $\sigma_1 \neq \sigma_2$, $\sigma_1(D_i) = 1$ y $\sigma_2(D_i) = 1$.

10. Una *claúsula de Horn* es una claúsula que tiene a lo más un literal positivo. Por ejemplo $\neg p$, $(\neg p \vee \neg q)$ y $(p \vee \neg q \vee \neg r \vee \neg s)$ son todas claúsulas de Horn, mientras que $(p \vee q \vee \neg r)$ no es una claúsula de Horn porque tiene dos literales positivos. Demuestre que existe una fórmula que no es equivalente a ningún conjunto de claúsulas de Horn.

11. Un conjunto Σ de fórmulas proposicionales se dice *redundante* si existe una fórmula $\varphi \in \Sigma$ tal que $\Sigma \setminus \{\varphi\} \models \varphi$. Además, Σ se dice *fuertemente redundante*, si para cada $\Sigma' \subseteq \Sigma$ tal que Σ y Σ' son equivalentes, se tiene que Σ' es redundante. Construya un conjunto de fórmulas que sea fuertemente redundante.

12. Sea Σ un conjunto satisfacible de fórmulas proposicionales, y φ una fórmula proposicional que no es una tautología. Además, suponga que Σ y φ no tienen variables proposicionales en común. ¿Es cierto que $\Sigma \not\models \varphi$? Demuestre o de un contraejemplo.

13. Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$, demuestre que $\Sigma \models \alpha \to \beta$ si y sólo si $\Sigma \cup \{\alpha\} \models \beta$.

14. Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$, demuestre que si α es una tautología, entonces $\Sigma \cup \{\alpha\} \models \beta$ si y sólo si $\Sigma \models \beta$.

15. Dado $\Sigma \subseteq L(P)$ y $\varphi, \psi, \theta \in L(P)$, demuestre que si $\varphi \to \psi$ es una tautología, entonces $\Sigma \cup \{\varphi, \psi\} \models \theta$ si y sólo si $\Sigma \cup \{\varphi\} \models \theta$.

16. Dado $\Sigma \subseteq L(P)$ y $\alpha, \beta \in L(P)$ tal que α y $\Sigma \cup \{\beta\}$ no tienen variables proposicionales en común. ¿Es cierto que $\Sigma \models \beta$ si y sólo si $\Sigma \cup \{\alpha\} \models \beta$?

En los ejercicios 12 - 22 decimos que un algoritmo es eficiente si el número de pasos ejecutado por el algoritmo es n^c cuando la entrada tiene largo n, donde c es una constante. Por ejemplo, un algoritmo que funciona en tiempo n^2 es eficiente mientras que un algoritmo que funciona en tiempo 2^n no lo es.

12. Encuentre un algoritmo eficiente que dada una fórmula φ en CNF construya una fórmula ψ en 3-CNF tal que φ es satisfacible si y sólo si ψ es satisfacible.

13. Encuentre un algoritmo eficiente que verifique si una fórmula en DNF es satisfacible.

14. Encuentre un algoritmo eficiente que verifique si una fórmula en CNF es una tautología.

15. Un grafo G es una tupla (N, A), donde N es un conjunto de nodos y $A \subseteq N \times N$ es un conjunto de arcos. Un grafo es no dirigido si cada vez que $(a, b) \in A$ se tiene que $(b, a) \in A$.

    Un grafo no dirigido $G = (N, A)$ es 3-coloreable si existe una asignación de colores para los nodos tal que nodos adyacentes reciben colores distintos. Formalmente, G es 3-coloreable si existe una función $f : N \to \{blanco, azul, rojo\}$ tal que para cada $(a, b) \in A$ se tiene que $f(a) \neq f(b)$.

    Demuestre que el problema de 3-coloración puede ser reducido eficientemente al problema de satisfacibilidad. Vale decir, encuentre un algoritmo eficiente que dado un grafo G construya una fórmula φ tal que G es 3-coloreable si y sólo si φ es satisfacible. Estime el número de pasos de su algoritmo cuando el grafo G tiene n nodos y m arcos.

16. ¿Qué pasa en el problema anterior si también consideramos algoritmos no eficientes? ¿Cuáles son las fórmulas de menor largo que podríamos usar en la reducción?

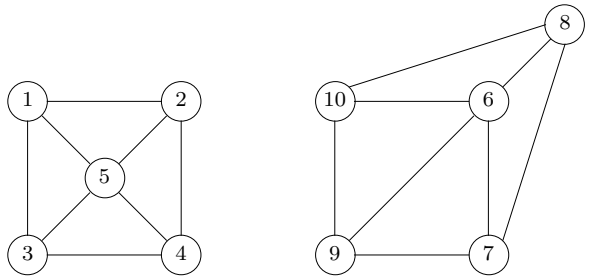
17. Un grafo no dirigido $G = (N, A)$ es k-coloreable ($k \geq 2$) si existe una función $f : N \to \{1, 2, \ldots, k\}$ tal que para cada $(a, b) \in A$ se tiene que $f(a) \neq f(b)$.

    Demuestre que el problema de k-coloración puede ser reducido eficientemente al problema de satisfacibilidad. Estime el número de pasos de su algoritmo para un grafo con n nodos y m arcos.

18. Suponga que existe un algoritmo eficiente para el problema de satisfacibilidad, es decir, un algoritmo que dada una fórmula φ con n letras proposicionales, en n^c pasos retorna 1 si la fórmula es satisfacible y 0 en caso contrario, donde c es una constante fija.

    Encuentre un algoritmo eficiente que dada una fórmula φ retorna una valuación σ tal que $\sigma(\varphi) = 1$ si φ es satisfacible, y retorna 0 en caso contrario.

19. Decimos que dos grafos $G_1 = (N_1, A_1)$ y $G_2 = (N_2, A_2)$ son *isomorfos* si existe una biyección $f : N_1 \to N_2$ tal que para todo a y b en A_1 se tiene que $(a, b) \in A_1$ si y sólo si $(f(a), f(b)) \in A_2$. Por ejemplo, los siguientes grafos son isomorfos:



G_1 G_2

    Encuentre un algoritmo eficiente que dados dos grafos G_1 y G_2 construya una fórmula φ tal que G_1 y G_2 son isomorfos si y sólo si φ es satisfacible. Estime el número de pasos de su algoritmo cuando G_1 tiene n_1 nodos y m_1 arcos, y G_2 tiene n_2 nodos y m_2 arcos.

20. Dado un grafo $G = (N, A)$, una secuencia de nodos $(a_1, \ldots, a_n)$ es un *camino en* G si para todo $i \in [1, n-1]$ se tiene que $(a_i, a_{i+1}) \in A$. Decimos que G contiene un *circuito Hamiltoniano* si existe un camino $(a_1, \ldots, a_n)$ en G tal que:

    - n es el número de nodos de N,
    - $a_i \neq a_j$ para cada $i, j \in \{1, \ldots, n\}$ tales que $i \neq j$,
    - $(a_n, a_1) \in A$.

Encuentre un algoritmo eficiente que dado un grafo G, construya una fórmula proposicional φ tal que G contiene un circuito Hamiltoniano si y sólo si φ es satisfacible.

21. Un grafo $G' = (N', A')$ es un subgrafo de un grafo $G = (N, A)$ si $N' \subseteq N$ y $A' \subseteq A$. Dado un grafo $G = (N, A)$ no dirigido y un número entero $k \geq 2$, se dice que G tiene un k-clique si existe un subgrafo $G' = (N', A')$ de G tal que N' contiene k elementos y para cada par de nodos distintos $a, b \in N'$, se tiene que $(a, b) \in A'$ y $(b, a) \in A'$.

    Encuentre un algoritmo eficiente que dado un grafo G y un número entero $k \geq 2$, construya una fórmula proposicional φ tal que G contiene un k-clique si y sólo si φ es satisfacible.

22. Sea $n \geq 2$ un número natural fijo. La estructura $\mathbb{Z}_n = (\{0, \ldots, n-1\}, +, \cdot)$ se define de la siguiente forma. Sean a y b dos números en el conjunto $\{0, \ldots, n-1\}$. Entonces se tiene que $a + b$ es el resto módulo n de la suma usual entre a y b, y $a \cdot b$ es el resto módulo n de la multiplicación usual entre a y b. Por ejemplo, las siguientes son las tablas de las operaciones $+$ y $\cdot$ para $n = 5$:

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

    Sea V un conjunto infinito de variables. El conjunto de expresiones aritméticas $E(V)$ es definido como el menor conjunto que satisface las siguiente reglas:

    - $V \subseteq E(V)$.
    - Si $e_1 \in E(V)$ y $e_2 \in E(V)$, entonces $(e_1 + e_2) \in E(V)$ y $(e_1 \cdot e_2) \in E(V)$.

    Por ejemplo, $(u \cdot (v + w))$ y $(u + (u \cdot v))$ son expresiones en $E(V)$, suponiendo que u, v y w son variables en V.

    Sea e una expresión aritmética que menciona a las variables $x_1, \ldots, x_k$. Dados $i_1, \ldots, i_k \in \{0, \ldots, n-1\}$, decimos que $x_1 \to i_1, \ldots, x_k \to i_k$ es una raíz para e en $\mathbb{Z}_n$ si la evaluación de e en $\mathbb{Z}_n$ con cada x_j reemplazado por i_j ($j \in \{1, \ldots, k\}$) da como resultado 0. Por ejemplo, $u \to 2$, $v \to 4$ es una raíz para la expresión aritmética $(u + (u \cdot v))$ en $\mathbb{Z}_5$, como también lo es $u \to 0$, $v \to 0$. Finalmente, decimos que una raíz $x_1 \to i_1, \ldots, x_k \to i_k$ para e en $\mathbb{Z}_n$ es *no trivial* si al menos uno de los números i_j ($j \in \{1, \ldots, k\}$) es distinto de 0. Así, por ejemplo, $u \to 2$, $v \to 4$ es una raíz no trivial de $(u + (u \cdot v))$ en $\mathbb{Z}_5$, mientras que $u \to 0$, $v \to 0$ no lo es.

    (a) Encuentre un algoritmo eficiente que dada una expresión aritmética e, genere una fórmula φ tal que e tiene una raíz no trivial en $\mathbb{Z}_n$ si y sólo si φ es satisfacible.

    (b) Aplique la metodología desarrollada en (a) para el caso de $n = 3$ y la expresión aritmética $(x + (y \cdot (x + y)))$.

23. Encuentre un método que no use tablas de verdad para demostrar que una fórmula es una tautología. Use su método para demostrar que $(p \leftrightarrow q) \leftrightarrow (\neg p \leftrightarrow \neg q)$ es una tautología.

24. Dado un conjunto de letras proposicionales P, $L_\infty(P)$ es el menor conjunto que satisface las siguientes reglas:

    - Si $\Sigma \subseteq L(P)$, entonces $(\bigwedge \Sigma) \in L_\infty(P)$.
    - Si $\varphi \in L_\infty(P)$, entonces $(\neg\varphi) \in L_\infty(P)$.
    - Si $\varphi, \psi \in L_\infty(P)$, entonces $(\varphi \vee \psi) \in L_\infty(P)$ y $(\varphi \wedge \psi) \in L_\infty(P)$.

Por ejemplo, si $P = \{p_i \mid i \in \mathbb{N}\}$ y $\Sigma = \{(\neg p_i) \mid i \in \mathbb{N}\}$, entonces $(\bigwedge \Sigma)$ y $(\neg(\bigwedge \Sigma))$ son fórmulas en $L_\infty(P)$. Note que la primera fórmula nos dice que todas las letras proposicionales son falsas, mientras que la segunda nos dice que al menos una es verdadera. También note que $((\bigwedge \Sigma) \vee (\neg(\bigwedge \Sigma)))$ es una fórmula en $L_\infty(P)$ que debería ser siempre verdadera (tautología).

Dada una valuación $\sigma : P \to \{0, 1\}$ y una fórmula $\varphi \in L_\infty(P)$, definimos $\sigma(\varphi)$ de la siguiente manera:

- Si $\varphi = (\bigwedge \Sigma)$, donde $\Sigma \subseteq L(P)$, entonces

$$\sigma(\varphi) = \begin{cases} 1 & \text{si para cada } \psi \in \Sigma \text{ se tiene que } \sigma(\psi) = 1 \\ 0 & \text{en caso contrario} \end{cases}$$

- Si $\varphi = (\neg\alpha)$, entonces

$$\sigma(\varphi) = \begin{cases} 1 & \text{si } \sigma(\alpha) = 0 \\ 0 & \text{si } \sigma(\alpha) = 1 \end{cases}$$

- Si $\varphi = (\alpha \vee \beta)$, entonces

$$\sigma(\varphi) = \begin{cases} 1 & \text{si } \sigma(\alpha) = 1 \text{ o } \sigma(\beta) = 1 \\ 0 & \text{si } \sigma(\alpha) = 0 \text{ y } \sigma(\beta) = 0 \end{cases}$$

- Si $\varphi = (\alpha \wedge \beta)$, entonces

$$\sigma(\varphi) = \begin{cases} 1 & \text{si } \sigma(\alpha) = 1 \text{ y } \sigma(\beta) = 1 \\ 0 & \text{si } \sigma(\alpha) = 0 \text{ o } \sigma(\beta) = 0 \end{cases}$$

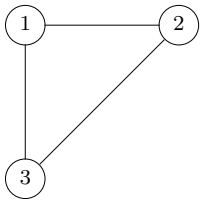
Conteste las siguientes preguntas.

(a) Dado $P = \{p_i \mid i \in \mathbb{N}\}$, demuestre que existe una fórmula $\varphi \in L_\infty(P)$ tal que φ no es equivalente a ningún $\Sigma \subseteq L(P)$.

(b) ¿Es el teorema de compacidad válido para la lógica proposicional infinitaria? Vale decir, dado $\Gamma \subseteq L_\infty(P)$, ¿Es cierto que si todo subconjunto finito de Γ es satisfacible, entonces Γ es satisfacible?

25. Considere el argumento dado en el problema 5 y su formalización en el cálculo proposicional. Demuestre usando resolución que Superman no existe.

26. Usando la fórmula construida en el problema 17 y resolución, demuestre que el siguiente grafo no es 2-coloreable:



27. El *principio de los cajones* establece que si $n+1$ objetos son distribuidos en n cajones, entonces al menos habrá un cajón con más de un objeto. Demuestre el principio para $n = 2$ usando cálculo proposicional y resolución.

28. La siguiente es una tautología generada utilizado el segundo esquema en el sistema de Hilbert:

$$((p \to (q \to r)) \to ((p \to q) \to (p \to r))).$$

Utilizando resolución demuestre que esta fórmula es efectivamente una tautología.

29. En este problema se pide al alumno demostrar que si junto con las reglas de resolución y factorización podemos usar tautologías en cualquier parte de una demostración, entonces el sistema resultante es completo, vale decir, si C es una claúsula, Σ es un conjunto de claúsulas y $\Sigma \models C$, entonces $\Sigma \vdash C$.

    Formalmente, dada una claúsula C y un conjunto de claúsulas Σ, una demostración por resolución de C a partir de Σ es una secuencia de claúsulas C_1, C_2, ..., C_n tal que:

    - Para cada $i \leq n$:
        - $C_i \in \Sigma$ o
        - C_i es una tautología o
        - existe $j < i$ tal que C_i es obtenido desde C_j usando la regla de factorización o
        - existen $j, k < i$ tales que C_i es obtenido desde C_j y C_k usando la regla de resolución.
    - $C_n = C$.

    Si C es demostrable desde Σ por resolución, entonces usamos la notación $\Sigma \vdash C$.

    Nótese que si C es una tautología, entonces se deduce que $\Sigma \vdash C$ trivialmente. Así, para demostrar que el sistema de demostración es completo, sólo necesitamos considerar dos casos.

    (a) Demuestre que si Σ es inconsistente, entonces $\Sigma \vdash C$ para toda claúsula C.

    (b) Demuestre que si Σ es consistente, C no es una tautología y $\Sigma \models C$, entonces $\Sigma \vdash C$.

30. En clases se demostró que es necesario agregar la regla de factorización a la regla de resolución para obtener un sistema de demostración que satisface la noción de completidad débil. Suponga que se reemplaza estas dos reglas por la *regla de resolución extendida:*

$$\frac{\begin{array}{l} C_1 \vee \ell \vee C_2 \vee \ell \vee \cdots \vee C_{n-1} \vee \ell \vee C_n \\ D_1 \vee \bar{\ell} \vee D_2 \vee \bar{\ell} \vee \cdots \vee D_{m-1} \vee \bar{\ell} \vee D_m \end{array}}{C_1 \vee C_2 \vee \cdots \vee C_{n-1} \vee C_n \vee D_1 \vee D_2 \vee \cdots \vee D_{m-1} \vee D_m}$$

    donde $n \geq 2$, $m \geq 2$, cada C_i ($i \in \{1, \ldots, n\}$) es una claúsula y cada D_j ($j \in \{1, \ldots, m\}$) también es una claúsula. Por ejemplo, las siguientes son dos aplicaciones de la regla de resolución extendida:

$$\frac{\begin{array}{l} p \vee p \\ \neg p \vee \neg p \vee \neg p \end{array}}{\square} \qquad \frac{\begin{array}{l} q \vee p \vee \neg r \vee p \vee p \\ \neg p \vee \neg r \vee s \vee t \vee \neg p \end{array}}{q \vee \neg r \vee \neg r \vee s \vee t}$$

    Dada una claúsula C y un conjunto de claúsulas Σ, una demostración por resolución extendida de C a partir de Σ es una secuencia de claúsulas C_1, C_2, ..., C_n tal que:

    - Para cada $i \leq n$:
        - $C_i \in \Sigma$ o
        - existen $j, k < i$ tales que C_i es obtenido desde C_j y C_k usando la regla de resolución extendida.
    - $C_n = C$.

    Si C es demostrable desde Σ por resolución extendida, entonces usamos la notación $\Sigma \vdash_E C$. Nótese que la regla de resolución extendida es correcta, por lo que se tiene que el sistema es correcto: Si $\Sigma \vdash_E C$, entonces $\Sigma \models C$. En esta pregunta usted debe verificar si la regla de resolución extendida satisface la noción de completidad débil. Vale decir, demuestre o de un contraejemplo para la siguiente afirmación: Si $\Sigma \models \square$, entonces $\Sigma \vdash_E \square$.