Técnicas de Resolución

Pablo Barceló

Resolución: Problemas

Si no somos cuidadosos al elegir resolventes el proceso de resolución puede volverse muy ineficiente.

Ejercicio: Considere el conjunto Σ que contiene a las cláusulas $\{\neg p, \neg q\}, \{\neg p, q\}, \{\neg q, p\}, \{p\}$.

- Demuestre usando resolución que Σ es insatisfacible.
- ¿Qué sucede si eligiéramos otra estrategia de resolución en este ejemplo? ¿Existen loops?

Resolución: Problemas

Si no somos cuidadosos al elegir resolventes el proceso de resolución puede volverse muy ineficiente.

Ejercicio: Considere el conjunto Σ que contiene a las cláusulas $\{\neg p, \neg q\}, \{\neg p, q\}, \{\neg q, p\}, \{p\}$.

- Demuestre usando resolución que Σ es insatisfacible.
- ¿Qué sucede si eligiéramos otra estrategia de resolución en este ejemplo? ¿Existen loops?

Pregunta: ¿Cómo podríamos evitar los loops en el ejemplo anterior?

Resolución: Problemas

Si no somos cuidadosos al elegir resolventes el proceso de resolución puede volverse muy ineficiente.

Ejercicio: Considere el conjunto Σ que contiene a las cláusulas $\{\neg p, \neg q\}, \{\neg p, q\}, \{\neg q, p\}, \{p\}$.

- Demuestre usando resolución que Σ es insatisfacible.
- ¿Qué sucede si eligiéramos otra estrategia de resolución en este ejemplo? ¿Existen loops?

Pregunta: ¿Cómo podríamos evitar los loops en el ejemplo anterior?

- Ya que la cláusula p está en Σ podríamos haber sacado a $\{\neg q, p\}$ antes de comenzar el proceso.

Estrategias de resolución: Vista general

Estudiaremos 2 tipos de estrategias para implementar resolución:

- Estrategias de ordenación: Definen el orden en que se realizan las resoluciones.

  Primero en anchura, primero en profundidad, resolución unitaria.

- Estrategias de refinamiento: Imponen restricciones en los tipos de resoluciones permitidos.

  Eliminación de clausulas, conjunto soporte, resolución lineal.

Estrategia de ordenación: Primero en anchura

Definimos los resolventes de nivel $i \geq 0$ como sigue:

- Los resolventes de nivel 0 son las cláusulas en Σ.
- Los resolventes de nivel $i + 1$ son aquellas cláusulas obtenidas aplicando resolución a clásulas C_1 de nivel i y C_2 de nivel $j \leq i$.

La búsqueda primero en anchura calcula primero los resolventes de primer nivel, luego los de segundo nivel, etc. Se detiene cuando encuentra la clásula vacía en algún nivel.

Estrategia de ordenación: Primero en anchura

Ejercicio: Realice búsqueda primero en anchura para conjunto Σ de cláusulas: $\{p, q\}, \{\neg q, r\}, \{\neg p, s\}, \{\neg p, \neg s\}, \{\neg r\}$.

Estrategia de ordenación: Primero en anchura

Ejercicio: Realice búsqueda primero en anchura para conjunto Σ de cláusulas: $\{p, q\}, \{\neg q, r\}, \{\neg p, s\}, \{\neg p, \neg s\}, \{\neg r\}$.

Ventajas y desventajas de la búsqueda en profundidad:

- Ventaja: No entra en loops, y por tanto, siempre finaliza con la cláusula vacía en el caso de que el conjunto sea insatisfacible.
- Desventaja: Es exponencial tanto en tiempo como en espacio.

Estrategia de ordenación: Primero en profundidad

Se almacena la demostración como una lista de cláusulas.

En cada paso resolvemos utilizando la última cláusula generada y alguna cláusula en la lista.

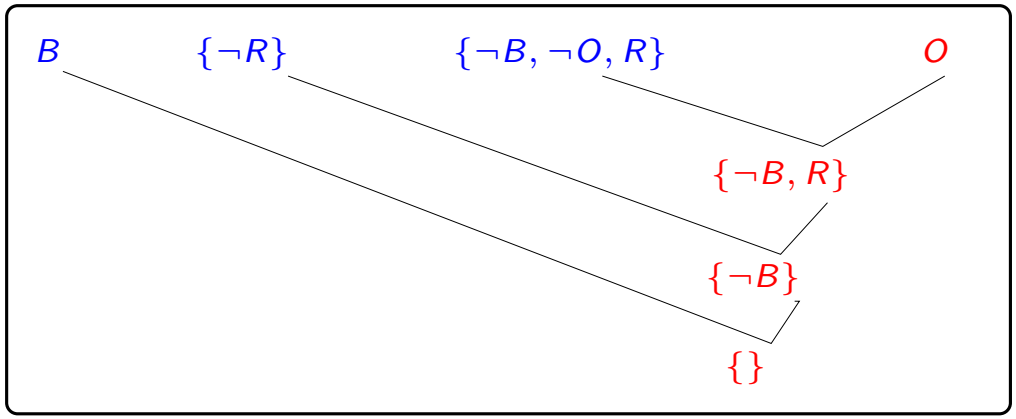
El problema es que podemos caer en loops si no elegimos bien la estrategia.

Ejemplo anterior: $\Sigma = \{\{\neg p, \neg q\}, \{\neg p, q\}, \{\neg q, p\}, \{p\}\}$.

Comúnmente se pone una cota en el largo de las demostraciones para evitar loops.

Estrategias de ordenaión: Primero en profundidad

Ejemplo: Búsqueda en profundidad.



Estrategias de ordenaión: Primero en profundidad

Ventajas y deventajas de primero en profundidad:

- Ventaja: Requiere menos memoria ya que sólo almacenamos cláusulas en la lista.
- Desventaja: Debemos incluir estrategias de control para detección de loops (no trivial).

Estrategias de ordenación: Resolución unitaria

La resolución unitaria le entrega siempre prioridad a aquellas resoluciones donde al menos una de las cláusulas tiene un sólo literal.

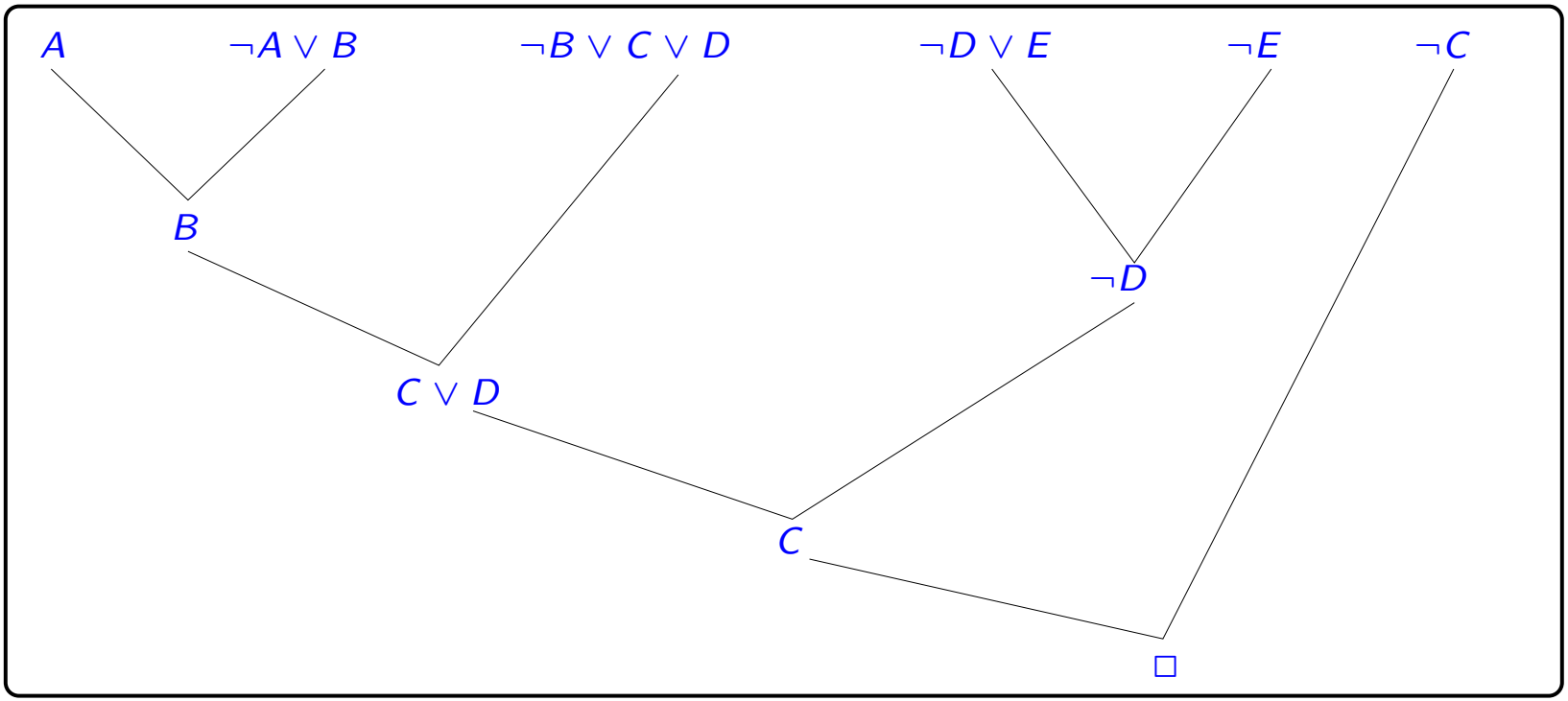
Estrategias de ordenación: Resolución unitaria

La resolución unitaria le entrega siempre prioridad a aquellas resoluciones donde al menos una de las cláusulas tiene un sólo literal.

Justificación: Al resolver una cláusula con k literales y una cláusula con 1 literal obtenemos una cláusula con $k-1$ literales.
Al obtener cláusulas más y más pequeñas aumenta la probabilidad de llegar a la cláusula vacía $\square$.

Resolución unitaria: Ejemplo

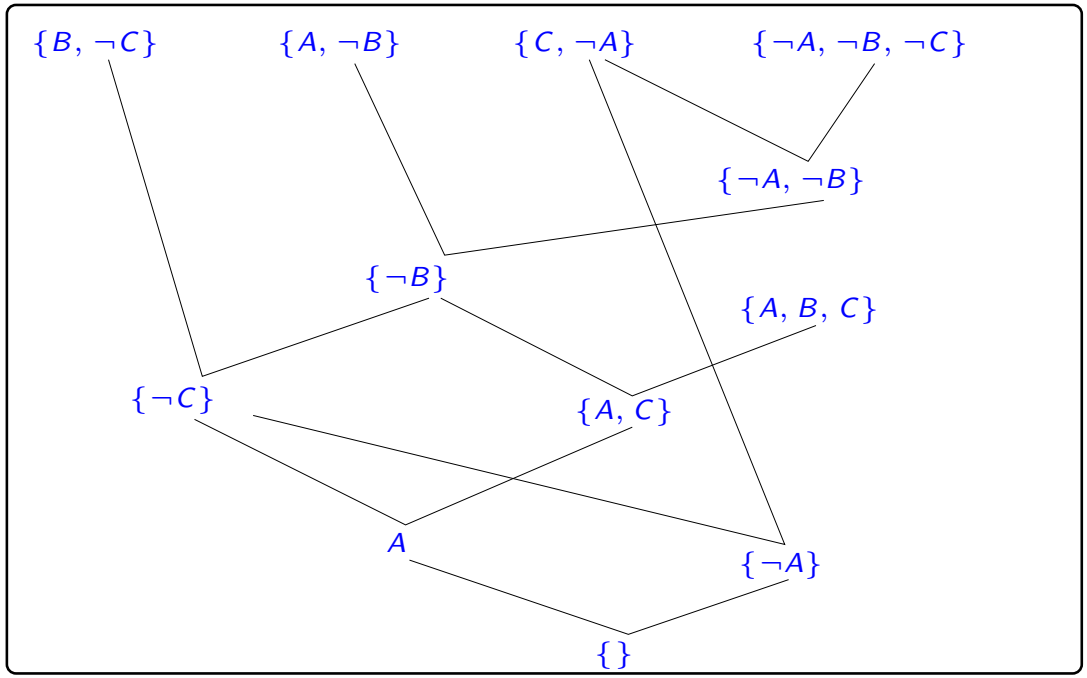
El siguiente es un ejemplo de resolución donde todos los pasos son unitarios:



Nota: Por supuesto esto no es siempre posible!

Estrategias de refinamiento: Eliminación de cláusulas

En general, no podemos eliminar cláusulas generadas en un proceso de resolución porque podrían ser utilizadas posteriormente.



Sin embargo, podríamos intentar mantener el número de cláusulas tan pequeño como se pueda sin alterar la satisfacibilidad del conjunto.

Estrategias de refinamiento: Eliminación de cláusulas

En un proceso de resolución podemos eliminar los siguientes tipos de cláusulas sin afectar la satisfacibilidad del conjunto:

- Cláusulas puras: para algún literal l en la cláusula se tiene que $\overline{l}$ no aparece en ninguna otra cláusula.
- Tautologías: clásulas que contienen literales l y $\overline{l}$.
- Cláusulas subsumidas: aquellas para las cuales existe otra cláusula que contiene exactamente un subconjunto de sus literales.

Nos enfocaremos principalmente en este último tipo de cláusulas.

Estrategias de refinamiento: Cláusulas subsumidas

Ejercicio: ¿Por qué podemos eliminar las cláusulas subsumidas sin afectar la satisfacibilidad del conjunto?

Ejercicio: Realice resolución sobre nuestro ejemplo anterior, pero esta vez eliminando a cada paso todas las cláusulas subsumidas.

Cláusulas subsumidas y resolución unitaria

Note que después de cada paso unitario de resolución de la forma:

$$\frac{\begin{array}{l} l \vee C_1 \\ \bar{l} \end{array}}{C_1}$$

podemos eliminar la cláusula $l \vee C_1$ (¿por qué?).

Estrategias de refinamiento: Conjunto soporte

Una suposición razonable al momento de verificar si $\Sigma \models \phi$ es que Σ es satisfacible:

> Σ corresponde a nuestra descripción del mundo, y por tanto, debería tener al menos un modelo.

Por tanto, toda demostración de insatisfacibilidad de $\Sigma \cup \{\neg\phi\}$ mediante resolución debe contener alguna cláusula de $\neg\phi$.

La idea del conjunto soporte es por tanto guíar nuestra búsqueda usando $\neg\phi$.

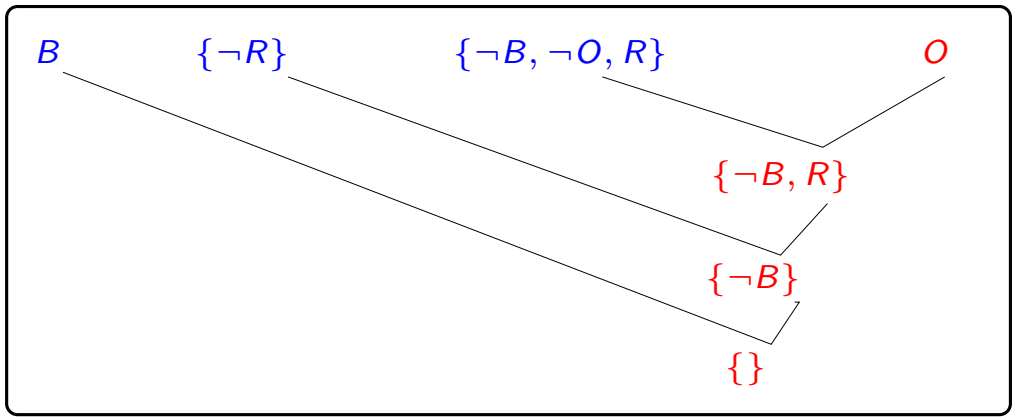
Conjunto soporte: Definición

El conjunto soporte se define como sigue:

- ϕ pertenece al conjunto soporte;
- la resolución de cualquier elemento en el conjunto soporte y otra cláusula pertenece al conjunto soporte.

La estrategia del conjunto soporte: En cada resolución usar al menos una cláusula en el conjunto soporte.

Ejemplo: $\{B\}, \{\neg R\}, \{\neg B, \neg O, R\} \models \neg O$ usando estrategia de conjunto soporte:



Conjunto soporte: Corrección y completidad

Teorema

Para el caso en que Σ es satisfacible la estrategia del conjunto soporte es correcta y completa.

Esto quiere decir que si $\Sigma \cup \{\neg\phi\} \vdash \square$, entonces $\square$ puede ser obtenido desde $\Sigma \cup \{\neg\phi\}$ usando resolución mediante la estrategia del conjunto soporte.

Ejercicio (difícil): Demuestre el teorema anterior.

Ejercicio: Demuestre que $\neg A$ es consecuencia lógica de

$$\{\neg A, B\}, \{\neg B, C, D\}, \{\neg D, E\}, \{\neg E\}, \{\neg C\}$$

usando la estrategia del conjunto soporte.

Estrategias de refinamiento: Resolución lineal

Una resolución lineal es una secuencia $C_0, C_1, \ldots, C_n$ tal que $C_0 \in \Sigma$ y para cada $1 \leq i \leq n$, C_i se obtiene se obtiene:

- por resolución desde C_{i-1} y algún elemento en Σ; o
- por resolución desde C_{i-1} y C_j, $j < i - 1$.

Ejemplo: El ejemplo que usamos para mostrar la estrategia del conjunto soporte es también una resolución lineal.

Ejercicio: Demuestre que la resolución lineal es completa.

Estrategias de refinamiento: Resolución lineal

Si eliminamos el caso en que C_i se obtiene por resolución de C_j y C_{i-1} entonces hablamos de resolución de entrada.

Ejercicio: Demuestre que la estrategia de resolución de entrada no es completa.

Complejidad de resolución

Por muy buenas que sean nuestras estrategias hay dos resultados matemáticos que muestran que resolución es (en el caso general) un método poco eficiente:

- El problema de satisfacibilidad es NP-completo.
- Existen casos en que el *largo* de la refutación más corta es aún exponencial [Haken'85].

Intentar convertir a DNF – o procedimientos parecidos – no funcionan, ya que las transformaciones entre estos fragmentos está demostrado que toman tiempo exponencial en el peor caso.

Complejidad de resolución

Pero... estamos hablando de peor caso: Por ejemplo, la complejidad *promedio* del problema de satisfacibilidad es polinomial.

La complejidad de satisfacibilidad para 2-SAT también es polinomial.

A continuación demostraremos que ésto también es cierto para un fragmento de SAT muy útil en programación en lógica: Cláusulas de Horn.

Cláusulas de Horn

Estudiaremos un tipo de cláusulas para el que podemos resolver eficientemente el problema de satisfacibilidad: Cláusulas de Horn.

Una cláusula es de Horn si tiene a lo más un literal positivo.

Ejemplo: $\{p, \neg q, \neg r\}$, $\{p\}, \{\neg q, \neg r\}$.

Mostraremos que verificar satisfacibilidad para esta clase de cláusulas se puede hacer en tiempo polinomial.

Ejercicio: Demuestre que existe una cláusula que no es equivalente a ningún conjunto de clásulas de Horn.

Cláusulas de Horn

Por lo antes dicho, las cláusulas de Horn pueden ser de 3 tipos:

(1) $p \leftarrow$	átomo
(2) $\leftarrow q_1, \ldots, q_n$	cláusula negativa
(3) $p \leftarrow q_1, \ldots, q_n$	cláusula positiva

En una cláusula positiva $p \leftarrow q_1, \ldots, q_n$ decimos que p es la cabeza y $q_1, \ldots, q_n$ es el cuerpo.

Procedimiento para chequear satisfacibilidad

El siguiente procedimiento falla si y sólo si el conjunto de cláusulas de Horn es insatisfacible. Si no falla, entrega además el conjunto de proposiciones atómicas que son consecuencia lógica del programa:

Dado conjunto Σ de cláusulas de Horn:

1. Inicialice $L := \emptyset$ (L conjunto de átomos que se deducen de Σ).
2. Realice lo siguiente hasta que el procedimiento falle o hasta que el valor de L no cambie:
   - Si existe cláusula $\leftarrow q_1, \ldots, q_n$ en Σ tal que $q_i \in L$, para todo $i \leq n$, entonces el procedimiento falla.
   - De otra forma, para cada cláusula $p \leftarrow q_1, \ldots, q_n$ en Σ tal que $p \notin L$ y $q_i \in L$, para todo $i \leq n$, agregue p a L.

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

El procedimiento anterior entrega lo siguiente (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{d, e\}$;
- $L_2 = \{d, e, b, c\}$;
- $L_3 = \{d, e, b, c, a\}$;
- $L_4 = L_3$ (procedimiento para).

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

El procedimiento anterior entrega lo siguiente (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{d, e\}$;
- $L_2 = \{d, e, b, c\}$;
- $L_3 = \{d, e, b, c, a\}$;
- $L_4 = L_3$ (procedimiento para).

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

El procedimiento anterior entrega lo siguiente (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{d, e\}$;
- $L_2 = \{d, e, b, c\}$;
- $L_3 = \{d, e, b, c, a\}$;
- $L_4 = L_3$ (procedimiento para).

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

El procedimiento anterior entrega lo siguiente (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{d, e\}$;
- $L_2 = \{d, e, b, c\}$;
- $L_3 = \{d, e, b, c, a\}$;
- $L_4 = L_3$ (procedimiento para).

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

El procedimiento anterior entrega lo siguiente (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{d, e\}$;
- $L_2 = \{d, e, b, c\}$;
- $L_3 = \{d, e, b, c, a\}$;
- $L_4 = L_3$ (procedimiento para).

Procedimiento para satisfacibilidad: Ejemplo

Ejemplo: Sea Σ el siguiente conjunto de cláusulas:

$$\{d\}, \{e\}, \{a \leftarrow b, c\}, \{b \leftarrow d, e\},$$
$$\{b \leftarrow c\}, \{b \leftarrow g, e\}, \{c \leftarrow e\}, \{f \leftarrow a, g\}$$

El procedimiento anterior entrega lo siguiente (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{d, e\}$;
- $L_2 = \{d, e, b, c\}$;
- $L_3 = \{d, e, b, c, a\}$;
- $L_4 = L_3$ (procedimiento para).

Procedimiento para satisfacibilidad: Ejemplo

Concluimos que Σ es satisfacible y que $L_3 = \{d, e, b, c, a\}$ es el conjunto de variable proposicionales que son consecuencia lógica de Σ.

Que Σ sea satisfacible no es muy sorprendente. De hecho,

Ejercicio: Demuestre que todo conjunto de átomos $(p \leftarrow)$ y cláusulas positivas $(p \leftarrow q_1, \ldots, q_n)$ es satisfacible.

Cláusulas de Horn y consecuencia lógica

Más que satisfacibilidad nosotros queremos verificar consecuencia lógica.

Si queremos usar este procedimiento para chequear si $\Sigma \models \phi$, donde Σ es conjunto de clásulas de Horn, se requiere que $\neg\phi$ sea una cláusula de Horn:

> Asumimos que ϕ es una conjunción de literales positivos (llamada objetivo). Por tanto, $\neg\phi$ es de la forma $\leftarrow q_1, \ldots, q_n$.

En este caso $\Sigma \cup \{\neg\phi\}$ es también un conjunto de cláusulas de Horn.

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Basta demostrar que el procedimiento anterior falla sobre $\Sigma \cup \{\leftarrow \text{Girl}\}$ (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{\text{Female}, \text{Baby}\}$;
- $L_2 = \{\text{Female}, \text{Baby}, \text{Child}\}$;
- $L_3 = \{\text{Female}, \text{Baby}, \text{Child}, \text{Girl}\}$;
- En el 4to paso el pocedimiento falla debido a la presencia de $\leftarrow \text{Girl}$.

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Basta demostrar que el procedimiento anterior falla sobre $\Sigma \cup \{\leftarrow \text{Girl}\}$ (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{\text{Female}, \text{Baby}\}$;
- $L_2 = \{\text{Female}, \text{Baby}, \text{Child}\}$;
- $L_3 = \{\text{Female}, \text{Baby}, \text{Child}, \text{Girl}\}$;
- En el 4to paso el pocedimiento falla debido a la presencia de $\leftarrow \text{Girl}$.

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Basta demostrar que el procedimiento anterior falla sobre $\Sigma \cup \{\leftarrow \text{Girl}\}$ (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{\text{Female}, \text{Baby}\}$;
- $L_2 = \{\text{Female}, \text{Baby}, \text{Child}\}$;
- $L_3 = \{\text{Female}, \text{Baby}, \text{Child}, \text{Girl}\}$;
- En el 4to paso el pocedimiento falla debido a la presencia de $\leftarrow \text{Girl}$.

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Basta demostrar que el procedimiento anterior falla sobre $\Sigma \cup \{\leftarrow \text{Girl}\}$ (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{\text{Female}, \text{Baby}\}$;
- $L_2 = \{\text{Female}, \text{Baby}, \text{Child}\}$;
- $L_3 = \{\text{Female}, \text{Baby}, \text{Child}, \text{Girl}\}$;
- En el 4to paso el pocedimiento falla debido a la presencia de $\leftarrow \text{Girl}$.

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Basta demostrar que el procedimiento anterior falla sobre $\Sigma \cup \{\leftarrow \text{Girl}\}$ (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{\text{Female}, \text{Baby}\}$;
- $L_2 = \{\text{Female}, \text{Baby}, \text{Child}\}$;
- $L_3 = \{\text{Female}, \text{Baby}, \text{Child}, \text{Girl}\}$;
- En el 4to paso el pocedimiento falla debido a la presencia de $\leftarrow \text{Girl}$.

Procedimiento para consecuencia lógica: Ejemplo

Ejemplo: Queremos demostrar que Girl es consecuencia lógica del siguiente conjunto Σ de cláusulas de Horn:

$$\{\text{Baby}\},\ \{\text{Child} \leftarrow \text{Baby}\},\ \{\text{Boy} \leftarrow \text{Child}, \text{Male}\}$$
$$\{\text{Female}\},\ \{\text{Girl} \leftarrow \text{Child}, \text{Female}\},\ \{\text{Child} \leftarrow \text{Infant}\}$$

Basta demostrar que el procedimiento anterior falla sobre $\Sigma \cup \{\leftarrow \text{Girl}\}$ (L_i es el valor de L en paso i):

- $L_0 = \emptyset$;
- $L_1 = \{\text{Female}, \text{Baby}\}$;
- $L_2 = \{\text{Female}, \text{Baby}, \text{Child}\}$;
- $L_3 = \{\text{Female}, \text{Baby}, \text{Child}, \text{Girl}\}$;
- En el 4to paso el pocedimiento falla debido a la presencia de $\leftarrow \text{Girl}$.

Corrección y completitud del procedimiento

¿Cómo podemos demostrar que el procedimiento es correcto y completo?

Es decir, queremos demostrar que Σ es satisfacible si y sólo si el procedimiento anterior no falla con entrada Σ.

Corrección y completitud del procedimiento

¿Cómo podemos demostrar que el procedimiento es correcto y completo?

Es decir, queremos demostrar que Σ es satisfacible si y sólo si el procedimiento anterior no falla con entrada Σ.

Asuma primero que el procedimiento no falla y su salida es L.

Ejercicio: Demuestre que la valuación σ tal que $\sigma(p) = 1 \Leftrightarrow p \in L$ satisface a Σ.

Corrección y completitud del procedimiento

Asuma ahora que Σ es satisfacible por valuación σ, y (por contradicción) que el procedimiento falla.

Sea L la salida del algoritmo hasta el paso inmediatamente anterior a fallar.

Primero demostramos que para toda variable proposicional p en L se tiene que $\sigma(p) = 1$.

> Podemos demostrar esto por inducción en el número de pasos ejecutados por el algoritmo hasta antes de fallar.

Pero entonces para toda regla $\leftarrow q_1, \ldots, q_n$ tal que $q_i \in L$, para todo $i \leq n$, se tiene que $\sigma(q_i) = 1$, para todo $i \leq n$.

> Contradicción, porque $\sigma(\Sigma) = 1$.

Complejidad del algoritmo

La complejidad del algoritmo es la siguiente:

- El algoritmo realiza a lo más $|\Sigma|$ pasos.
- Cada paso tiene que buscar una regla seleccionable entre $|\Sigma|$ reglas.
- Verificar si una regla es seleccionable toma tiempo n^2, donde n es el número de variables proposicionales en Σ.
- Complejidad final: $O(|\Sigma|^2 \cdot n^2)$, i.e. polinomial.
- Se puede hacer lineal con un buen manejo de la estructura de datos.

Cláusulas de Horn y resolución unitaria

Pregunta: ¿Qué tipo de resolución es la que realiza este procedimiento?

Cláusulas de Horn y resolución unitaria

Pregunta: ¿Qué tipo de resolución es la que realiza este procedimiento?

Resolución unitaria en cada paso!

Cláusulas de Horn y resolución unitaria

Pregunta: ¿Qué tipo de resolución es la que realiza este procedimiento?

Resolución unitaria en cada paso!

Corolario

Un conjunto Σ de cláusulas de Horn es insatisfacible si y sólo si $\Sigma \vdash \square$ si y sólo si $\square$ puede ser obtenida mediante resolución unitaria en cada paso a partir de Σ.

Otro procedimiento para consecuencia lógica

El procedimiento anterior se conoce como forward chaining pues es dirigido desde los átomos hacia el objetivo.

Presentaremos otro procedimiento que dado conjunto Σ de cláusulas de Horn y conjunción ϕ de variables proposicionales determina si ϕ es consecuencia lógica de Σ.

Este procedimiento se conoce como backward chaining pues es dirigido desde el objetivo hacia los átomos.

- Es menos eficiente que el anterior.
- Lo presentamos porque generaliza bien al caso de Prolog con variables y funciones.

Backward chaining

Dado conjunto Σ de cláusulas de Horn átomicas y positivas, y conjunción $\phi = q_1 \wedge \cdots \wedge q_n$ de variables proposicionales. El procedimiento retorna un SI si y sólo si $\Sigma \cup \{\neg\phi\}$ es insatisfacible.

procedure RESOLVER$[q_1, \ldots, q_n]$
    if $n = 0$ **then return** SI.
    for cada cláusula $q_1 \leftarrow r_1, \ldots, r_m$ en Σ **do**
        if RESOLVER$[r_1, \ldots, r_m, q_2, \ldots, q_n]$
        then return SI.
    end for
return NO.

Backward chaining

Dado conjunto Σ de cláusulas de Horn átomicas y positivas, y conjunción $\phi = q_1 \wedge \cdots \wedge q_n$ de variables proposicionales. El procedimiento retorna un SI si y sólo si $\Sigma \cup \{\neg\phi\}$ es insatisfacible.

procedure RESOLVER$[q_1, \ldots, q_n]$
  if $n = 0$ **then return** SI.
  for cada cláusula $q_1 \leftarrow r_1, \ldots, r_m$ en Σ **do**
    if RESOLVER$[r_1, \ldots, r_m, q_2, \ldots, q_n]$
    then return SI.
  end for
return NO.

Ejercicio: A parrtir de nuestro último ejemplo demuestre que $\Sigma \models \mathrm{Girl}$ usando backward chaining.

Backward chaining: Observaciones

El procedimiento se dice que es:

- Depth-first: Intenta resolver los nuevos objetivos $r_1, \ldots, r_m$ antes que los viejos objetivos $q_1, \ldots, q_n$.
- Left-to-right: Intenta resolver los objetivos q_i en orden: $1, 2, 3, \cdots$.

Este procedimiento también se denomina SLD:

- Selection: Selecciona el primer elemento del objetivo.
- Linear: Efectúa resolución lineal.
- Definite: Funciona para cláusulas definitivas o de Horn.

Backward chaining: Observaciones

Para el procedimiento:

- La elección del átomo a resolver en el objetivo no importa: Todos los átomos deben ser resueltos.
- La elección de la regla sí importa: Malas elecciones pueden llevar a fracasos, *loops* y *backtracking*.

Ejercicio: Investigue que sucede en nuestro primer ejemplo de cláusulas de Horn si usamos SLD.

Ejercicio: Demuestre que existe un conjunto Σ de cláusulas de Horn y un objetivo ϕ para los cuales SLD entra en un loop infinito.

Corrección y completitud de SLD

Pregunta: ¿Cómo podemos demostrar que el procedimiento SLD es correcto y completo?

Corrección y completitud de SLD

Pregunta: ¿Cómo podemos demostrar que el procedimiento SLD es correcto y completo?

No hay nada que demostrar, el procedimiento SLD es simplemente el de forward chaining pero recorremos el grafo en sentido contrario (desde el objetivo hacia los átomos).

Corolario

Un conjunto de cláusulas de Horn es insatisfacible si y sólo si $\Sigma \vdash \square$ *si y sólo si la cláusula vacía* $\square$ *puede ser obtenida por resolución lineal y de entrada desde* Σ.