Ingeniería del Software II
9 - Algoritmos para verificar satisfactibilidad en lógica proposicional

SAT en Computación

- La satisfactibilidad de fórmulas en lógica proposicional tiene aplicaciones en análisis de especificaciones.
- Su aplicación involucra otras áreas tales como la verificación y diseño de hardware, constraint solving en inteligencia artificial (planning), etc.
- En lo que sigue, veremos con algo más de detalle cómo funcionan los algoritmos de decisión para satisfactibilidad de fórmulas proposicionales.

Algoritmos Simples

Tablas de Verdad

La verificación de satisfactibilidad de fórmulas proposicionales mediante la construcción de tablas de verdad puede describirse de la siguiente manera:

- Dada una fórmula F, para la cual se desea saber si es satisfactible o no, construir la tabla de verdad que tenga por columnas todas las subfórmulas de F, ordenadas por complejidad.
- Llenar las columnas correspondientes a variables proposicionales con todas las posibles combinaciones. Luego, para cada fila, llenar el resto de las columnas de acuerdo a la semántica del conectivo lógico en cuestión, y los valores para sus subfórmulas.
- Si la última columna tiene $\top$ para alguna fila, entonces la fórmula es satisfactible, y la asignación de valores de verdad a las letras proposicionales en esa fila es la interpretación correspondiente.

Algoritmos Simples

Tablas de Verdad (cont.)

- Ejemplo: Supongamos que queremos saber si la siguiente fórmula es satisfactible.

$$F : \; P \wedge Q \;\rightarrow\; P \vee \neg Q$$

- Construimos la siguiente tabla de verdad:

P	Q	$P \wedge Q$	$\neg Q$	$P \vee \neg Q$	F
0	0	0	1	1	1
0	1	0	0	0	1
1	0	0	1	1	1
1	1	1	0	1	1

- Observando la última columna podemos concluir que la fórmula no sólo es satisfactible, sino que es válida.

Algoritmos Simples

Tablas de Verdad (cont.)

- Ejemplo:

$$F : \; P \vee Q \;\rightarrow\; P \wedge Q$$

- Construimos la siguiente tabla de verdad:

P	Q	$P \vee Q$	$P \wedge Q$	F
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	1	1	1

- Observando la última columna podemos concluir que la fórmula es satisfactible. Las dos primeras columnas dan las interpretaciones de P y Q que hacen verdadera la fórmula.

Algoritmos Simples

Argumento Semántico (método de tableaux)

El segundo mecanismo simple para la verificación de satisfactibilidad es a través del argumento semántico:

- Dada una fórmula F, para la cual se desea saber si es válida o no, comenzar suponiendo que la misma es insatisfactible.
- Descomponer la suposición, de acuerdo a reglas de inferencia basadas en la semántica de los conectivos lógicos hasta que ya no puedan aplicarse más reglas.
- Si todas las ramas (las reglas pueden dar lugar a ramificaciones) terminan en contradicción, la fórmula es válida.
- Caso contrario, cada rama que no termine en contradicción corresponde a una valuación que hace verdadera a la negación de F.

Algoritmos Simples

Argumento Semántico: Reglas de descomposición

Negación:

$$\frac{I \models \neg F}{I \not\models F} \qquad \frac{I \not\models \neg F}{I \models F}$$

Conjunción:

$$\frac{I \models F \wedge G}{\begin{array}{l} I \models F \\ I \models G \end{array}} \qquad \frac{I \not\models F \wedge G}{I \not\models F \quad | \quad I \not\models G}$$

Implicación:

$$\frac{I \models F \rightarrow G}{I \not\models F \quad | \quad I \models G} \qquad \frac{I \not\models F \rightarrow G}{\begin{array}{l} I \models F \\ I \not\models G \end{array}}$$

Disjunción:

$$\frac{I \models F \vee G}{I \models F \quad | \quad I \models G} \qquad \frac{I \not\models F \vee G}{\begin{array}{l} I \not\models F \\ I \not\models G \end{array}}$$

Equivalencia:

$$\frac{I \models F \leftrightarrow G}{I \models F \wedge G \quad | \quad I \not\models F \vee G} \qquad \frac{I \not\models F \leftrightarrow G}{I \models F \wedge \neg G \quad | \quad I \models \neg F \wedge G}$$

Contradicción:

$$\frac{\begin{array}{l} I \models F \\ I \not\models F \end{array}}{I \models \bot}$$

Algoritmos Simples

Argumento Semántico (cont.)

- Ejemplo:

$$F: \; P \wedge Q \;\rightarrow\; P \vee \neg Q$$

- El argumento semántico correspondiente es el siguiente:

$$\begin{array}{llll}
1. & I \not\models P \wedge Q \rightarrow P \vee \neg Q & & \{\text{suposición}\} \\
2. & I \models P \wedge Q & & \{\text{semántica de la implicación en 1}\} \\
3. & I \not\models P \vee \neg Q & & \{\text{semántica de la implicación en 1}\} \\
4. & I \models P & & \{\text{semántica de la conjunción en 2}\} \\
5. & I \models Q & & \{\text{semántica de la conjunción en 2}\} \\
6. & I \not\models P & & \{\text{semántica de la disyunción en 3}\} \\
7. & I \not\models \neg Q & & \{\text{semántica de la disyunción en 3}\} \\
8. & I \models Q & & \{\text{semántica de la negación en 7}\}
\end{array}$$

- La fórmula es válida (tenemos sólo una rama con una contradicción: líneas 4 y 6)

Algoritmos Simples

Argumento Semántico (cont.)

- Ejemplo:

$$F: \; P \wedge Q \;\rightarrow\; P \vee \neg Q$$

- El argumento semántico correspondiente es el siguiente:

1.	I	$\not\models$	$P \wedge Q \rightarrow P \vee \neg Q$	{supos
2.	I	$\models$	$P \wedge Q$	{sen
3.	I	$\not\models$	$P \vee \neg Q$	{sen
4.	I	$\models$	P	{semá
5.	I	$\models$	Q	{se ... ica de la ... 2}
6.	I	$\not\models$	P	{semántica de la disyunción en 3}
7.	I	$\not\models$	$\neg Q$	{semántica de la disyunción en 3}
8.	I	$\models$	Q	{semántica de la negación en 7}

No es necesario descomponer todo, alcanza con llegar a la contradicción.

- La fórmula es válida (tenemos sólo una rama con una contradicción: líneas 4 y 6)

Algoritmos Simples

Argumento Semántico (Otro ejemplo)

$$F:\ (P \rightarrow Q) \wedge (Q \rightarrow R)\ \rightarrow\ (P \rightarrow R)$$

- El argumento semántico correspondiente es el siguiente:

1.	I	$\not\models$	F	{suposición}
2.	I	$\models$	$(P \rightarrow Q)\ \wedge\ (Q \rightarrow R)$	{semántica de la implicación en 1}
3.	I	$\not\models$	$P \rightarrow R$	{semántica de la implicación en 1}
4.	I	$\models$	P	{semántica de la implicación en 3}
5.	I	$\not\models$	R	{semántica de la implicación en 3}
6.	I	$\models$	$P \rightarrow Q$	{semántica de la conjunción en 2}
7.	I	$\models$	$Q \rightarrow R$	{semántica de la conjunción en 2}

- de 6, tenemos dos casos:

$8a.$	I	$\not\models$	P	{6 y semántica de →}	$8b.$	I	$\models$	Q	{6 y semántica de →}
$9a.$	I	$\models$	$\perp$	{4 y 8a son contradict.}					

y de 7, dos casos más

$9ba.$	I	$\not\models$	Q	{7 y semántica de →}	$9bb.$	I	$\models$	R	{7 y semántica de →}
$10ba.$	I	$\models$	$\perp$	{8b y 9ba contradict.}	$10bb.$	I	$\models$	$\perp$	{5 y 9bb contradict.}

Algoritmos Simples

Argumento Semántico (Otro ejemplo más)

$$F:\ P \vee Q \ \rightarrow\ P \wedge Q$$

- El argumento semántico correspondiente es el siguiente:

1.	I	$\not\models$	$P \vee Q \ \rightarrow\ P \wedge Q$	{suposición}
2.	I	$\models$	$P \vee Q$	{semántica de la implicación en 1}
3.	I	$\not\models$	$P \wedge Q$	{semántica de la implicación en 1}

- A partir de 2, tenemos que considerar dos casos, uno de los cuales es el siguiente:

$4a.$	I	$\models$	P	{semántica de la disjunción en 2}

- Para continuar debemos considerar dos casos a partir de 3 uno de los cuales es:

$5a.$	I	$\not\models$	Q	{semántica de la conjunción en 3}

- En este caso, no podemos seguir aplicando reglas y no llegamos a ninguna contradicción.

Algoritmos Simples

Argumento Semántico (Otro ejemplo más)

$$F: \; P \vee Q \;\rightarrow\; P \wedge Q$$

- El argumento semántico correspondiente es el siguiente:

1.	I	$\not\models$	$P \vee Q \;\rightarrow\; P \wedge Q$	{suposición}
2.	I	$\models$	$P \vee Q$	{semántica de la implicación en 1}
3.	I	$\not\models$	$P \wedge Q$	{semántica de la implicación en 1}

- A partir de 2, tenemos que considerar dos casos, uno de los cuales es el siguiente:

$4a.$	I	$\models$	P	{semántica de l

- Para continuar debemos considera
los cuales es:

$5a.$	I	$\not\models$	Q	{semántica

- En este caso, no podemos seguir aplicando reglas y no llegamos a ninguna contradicción.

Luego, F no es válida y la valuación que la hace falsa es

$$I: \; \{P \mapsto \text{true}, \;\; Q \mapsto \text{false}\}$$

Algoritmos más avanzados

Tablas de verdad reconsideradas

- Para hacer al algoritmo de tablas de verdad más eficiente respecto al espacio ocupado podemos considerar de a una variable proposicional por vez, en lugar de construir la tabla de verdad en su totalidad.
- Esto puede resumirse en el siguiente algoritmo:

$$
\begin{array}{l}
\texttt{let rec } \textsc{sat}\ F = \\
\quad \texttt{if } F = \top \texttt{ then true} \\
\quad \texttt{else if } F = \bot \texttt{ then false} \\
\quad \texttt{else} \\
\quad\quad \texttt{let } P = \textsc{choose}\ \mathsf{vars}(F) \texttt{ in} \\
\quad\quad (\textsc{sat}\ F\{P \mapsto \top\})\ \lor\ (\textsc{sat}\ F\{P \mapsto \bot\})
\end{array}
$$

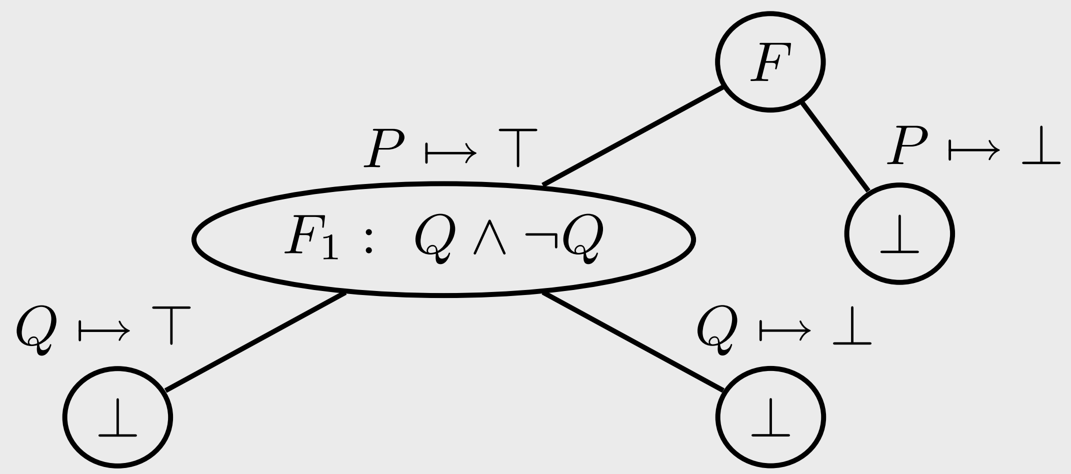
Algoritmos más avanzados

Tablas de verdad reconsideradas (Ejemplo)

- Verifiquemos la satisfactibilidad de la siguiente fórmula:

$$F : \ (P \rightarrow Q) \wedge P \wedge \neg Q$$

- La ejecución del algoritmo anterior para esta fórmula puede graficarse de la siguiente manera:



Algoritmos más e

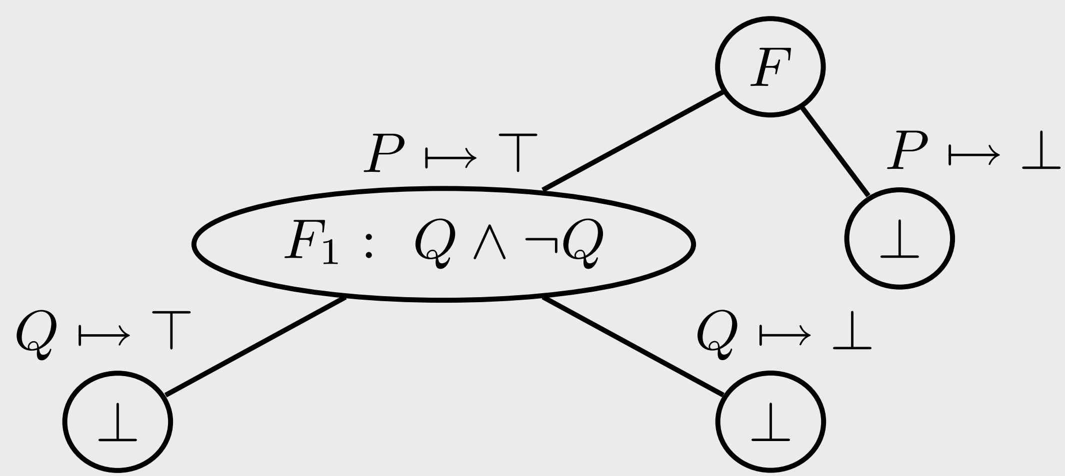
Observar que en este algoritmo se necesitan hacer muchas evaluaciones.

Tablas de verdad reconsideradas (Ejemplo)

- Verifiquemos la satisfactibilidad de la siguiente fórmula:

$$F: \ (P \to Q) \wedge P \wedge \neg Q$$

- La ejecución del algoritmo anterior para esta fórmula puede graficarse de la siguiente manera:



Algoritmos más avanzados

Forma Normal Conjuntiva

- Un literal es una proposición atómica P o su negación $\neg P$.
- Una fórmula proposicional está en forma normal conjuntiva (CNF) si es de la forma:

$$\bigwedge_i \bigvee_j \ell_{ij}$$

donde cada ℓ_{ij} es un literal.

Algoritmos más avanzados

Forma Normal Conjuntiva

- Un literal es una proposición atómica P o su negación $\neg P$.
- Una fórmula proposicional está en forma normal conjuntiva (CNF) si es de la forma:

Cada bloque de disjunción se denomina cláusula

$$\bigwedge_i \bigvee_j \ell_{ij}$$

donde cada ℓ_{ij} es un literal.

Algoritmos más avanzados

Equisatisfactibilidad

Dos fórmulas F y G son equisatisfactibles si
F es satisfactible sii G es satisfactible.

F y G no son necesariamente equivalentes.
Por ejemplo $P \vee Q$ y $(P \vee R) \wedge (Q \vee \neg R)$,
son equisatisfactible pero no son equivalentes.

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF

- Los algoritmos que veremos a continuación trabajan sobre fórmulas en CNF.
- Por consiguiente es importante tener un algoritmo efectivo de reducción a fórmulas en CNF.
- Por supuesto, la fórmula en CNF deben ser equisatisfactible a la original.
- La forma usual que utiliza doble negación, De Morgan y distributividad genera fórmulas **exponencialmente** grandes:

$$(F_1 \wedge G_1) \vee (F_2 \wedge G_2) \vee \cdots \vee (F_n \wedge G_n)$$

$$\longrightarrow \quad (F_1 \vee \cdots \vee F_{n-1} \vee F_n) \wedge (F_1 \vee \cdots \vee F_{n-1} \vee G_n) \wedge$$
$$\cdots \wedge (G_1 \vee \cdots \vee G_{n-1} \vee F_n) \wedge (G_1 \vee \cdots \vee G_{n-1} \vee G_n)$$

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF

- Los algoritmos que veremos a c… fórmulas en CNF.
- Por consiguiente es importante … …ivo de reducción a fórm…
- Por supuesto, la … …sfactible a la original.

¿Existe un algoritmo más eficiente?

En general:

$$\bigvee_{i\in I}\bigwedge_{j\in J} F_{i,j} \rightarrow \bigwedge_{f:I\to J}\bigvee_{i\in I} F_{i,f(i)}$$

- La forma usual que utiliza doble negación, De Morgan y distributividad genera fórmulas **exponencialmente** grandes:

$$(F_1 \wedge G_1) \vee (F_2 \wedge G_2) \vee \cdots \vee (F_n \wedge G_n)$$

$$\longrightarrow \quad (F_1 \vee \cdots \vee F_{n-1} \vee F_n) \wedge (F_1 \vee \cdots \vee F_{n-1} \vee G_n) \wedge \cdots \wedge (G_1 \vee \cdots \vee G_{n-1} \vee F_n) \wedge (G_1 \vee \cdots \vee G_{n-1} \vee G_n)$$

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- Sí, existe.
- El siguiente algoritmo genera una fórmula equisatisfactible que sólo crece **linealmente** respecto de la dada.
- La idea es introducir una variable por cada subfórmula de la fórmula original asegurándose de que cada variable introducida es equivalente a la fórmula que representa.
- Entonces tendremos dos funciones:

• $\mathsf{Rep} : \mathsf{PL} \to \mathcal{V} \cup \{\top, \bot\}$	Asigna variables proposicionales a las subfórmulas de F.
• $\mathsf{En} : \mathsf{PL} \to \mathsf{PL}$	A cada subfórmula G de F asigna una fórmula que asegura que G es equivalente a la variable $\mathsf{Rep}(G)$ que la representa.

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- Sí, existe.
- El siguiente algoritmo genera una fórmula ... que sólo crece **linealmente** respecto ...
- La idea es introducir una variable p... fórmula original asegurándose de q... introducida es equivalente a la fórmu...
- Entonces tendremos dos funciones:

Es decir:

$$\text{En}(G) \quad \equiv \quad \text{Rep}(G) \leftrightarrow G$$

bajo la hipótesis de que lo mismo se cumple para las subfórmulas de G

• $\text{Rep} : \text{PL} \to \mathcal{V} \cup \{\top, \bot\}$	Asigna variables proposicionales a las subfórmulas de F.
• $\text{En} : \text{PL} \to \text{PL}$	A cada subfórmula G de F asigna una fórmula que asegura que G es equivalente a la variable $\text{Rep}(G)$ que la representa.

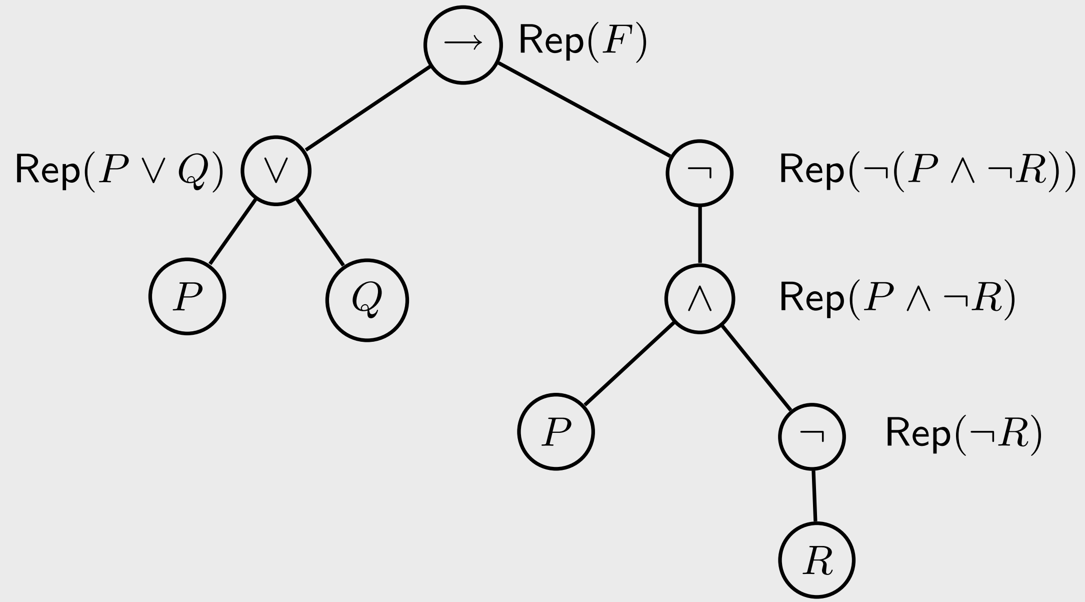
Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- Ejemplo: Para

$$F : P \vee Q \rightarrow \neg(P \wedge \neg R)$$

- tendremos nuevas variables proposicionales representando cada una de las subfórmulas de F:



Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- La definición de Rep y En tiene los siguientes casos elementales:

$$\begin{aligned} \mathsf{Rep}(\top) &= \top & \qquad \mathsf{En}(\top) &= \top \\ \mathsf{Rep}(\bot) &= \bot & \qquad \mathsf{En}(\bot) &= \top \\ \mathsf{Rep}(P) &= P & \qquad \mathsf{En}(P) &= \top \end{aligned}$$

- Notar que en todos estos casos $\mathsf{En}(F)$ es verdadero dado que $\mathsf{Rep}(F)$ y F son exactamente lo mismo.
- Para cualquier subfórmula F que no sea un átomo, $\mathsf{Rep}(F)$ será una letra proposicional nueva:

$$\mathsf{Rep}(F) = P_F$$

- El caso inductivo de $\mathsf{En}(F)$ se define a continuación.

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

$\mathsf{En}(F_1 \wedge F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \wedge F_2) \texttt{ in}$
  $(\neg P \vee \mathsf{Rep}(F_1)) \;\wedge\; (\neg P \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2) \vee P)$

$\mathsf{En}(\neg F) =$
  $\texttt{let } P = \mathsf{Rep}(\neg F) \texttt{ in}$
  $(\neg P \vee \neg \mathsf{Rep}(F)) \;\wedge\; (P \vee \mathsf{Rep}(F))$

$\mathsf{En}(F_1 \vee F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \vee F_2) \texttt{ in}$
  $(\neg P \vee \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg \mathsf{Rep}(F_1) \vee P) \;\wedge\; (\neg \mathsf{Rep}(F_2) \vee P)$

$\mathsf{En}(F_1 \rightarrow F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \rightarrow F_2) \texttt{ in}$
  $(\neg P \vee \neg \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2)) \;\wedge\; (\mathsf{Rep}(F_1) \vee P) \;\wedge\; (\neg \mathsf{Rep}(F_2) \vee P)$

$\mathsf{En}(F_1 \leftrightarrow F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \leftrightarrow F_2) \texttt{ in}$
  $(\neg P \vee \neg \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg P \vee \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2))$
  $\wedge\; (P \vee \neg \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2)) \;\wedge\; (P \vee \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2))$

Observar:

$$(\neg P \vee \mathsf{Rep}(F_1)) \;\wedge\; (\neg P \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2) \vee P)$$

está en CNF y es equivalente a

$$\mathsf{Rep}(F_1 \wedge F_2) \;\leftrightarrow\; \mathsf{Rep}(F_1) \wedge \mathsf{Rep}(F_2)$$

Conversión ... (cont.)

$\mathsf{En}(F_1 \wedge F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \wedge F_2) \texttt{ in}$
  $(\neg P \vee \mathsf{Rep}(F_1)) \;\wedge\; (\neg P \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2) \vee P)$

$\mathsf{En}(\neg F) =$
  $\texttt{let } P = \mathsf{Rep}(\neg F) \texttt{ in}$
  $(\neg P \vee \neg \mathsf{Rep}(F)) \;\wedge\; (P \vee \mathsf{Rep}(F))$

Ocurre lo mismo con las otras definiciones (Verificarlo)

$\mathsf{En}(F_1 \vee F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \vee F_2) \texttt{ in}$
  $(\neg P \vee \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg \mathsf{Rep}(F_1) \vee P) \;\wedge\; (\neg \mathsf{Rep}(F_2) \vee P)$

$\mathsf{En}(F_1 \rightarrow F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \rightarrow F_2) \texttt{ in}$
  $(\neg P \vee \neg \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2)) \;\wedge\; (\mathsf{Rep}(F_1) \vee P) \;\wedge\; (\neg \mathsf{Rep}(F_2) \vee P)$

$\mathsf{En}(F_1 \leftrightarrow F_2) =$
  $\texttt{let } P = \mathsf{Rep}(F_1 \leftrightarrow F_2) \texttt{ in}$
  $(\neg P \vee \neg \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2)) \;\wedge\; (\neg P \vee \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2))$
  $\wedge\; (P \vee \neg \mathsf{Rep}(F_1) \vee \neg \mathsf{Rep}(F_2)) \;\wedge\; (P \vee \mathsf{Rep}(F_1) \vee \mathsf{Rep}(F_2))$

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- Finalmente la fórmula equisatisfactible a la original F en CNF es:

$$F' : \ \mathsf{Rep}(F) \ \wedge \bigwedge_{G \in S_F} \mathsf{En}(G)$$

- donde S_F es el conjunto de todas las subfórmulas de F.

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- Ejemplo:

$$F : \ (Q_1 \wedge Q_2) \ \vee \ (R_1 \wedge R_2)$$

- El conjunto de todas sus subfórmulas es:

$$S_F : \ \{Q_1, \ Q_2, \ Q_1 \wedge Q_2, \ R_1, \ R_2, \ R_1 \wedge R_2, \ F\}$$

Rep

$$Q_1 \quad Q_2 \quad P_{(Q_1 \wedge Q_2)} \quad R_1 \quad R_2 \quad P_{(R_1 \wedge R_2)} \quad P_{(F)}$$

Algoritmos más avanzados

Conversión a una fórmula equisatisfactible en CNF (cont.)

- Ejemplo (cont.):

$$\begin{aligned}
\mathsf{En}(Q_1) &= \top \\
\mathsf{En}(Q_2) &= \top \\
\mathsf{En}(Q_1 \wedge Q_2) &= (\neg P_{(Q_1 \wedge Q_2)} \vee Q_1) \;\wedge\; (\neg P_{(Q_1 \wedge Q_2)} \vee Q_2) \\
&\quad \wedge\; (\neg Q_1 \vee \neg Q_2 \vee P_{(Q_1 \wedge Q_2)}) \\
\mathsf{En}(R_1) &= \top \\
\mathsf{En}(R_2) &= \top \\
\mathsf{En}(R_1 \wedge R_2) &= (\neg P_{(R_1 \wedge R_2)} \vee R_1) \;\wedge\; (\neg P_{(R_1 \wedge R_2)} \vee R_2) \\
&\quad \wedge\; (\neg R_1 \vee \neg R_2 \vee P_{(R_1 \wedge R_2)})
\end{aligned}$$

$$\begin{aligned}
\mathsf{En}(F) = {} & (\neg P_{(F)} \vee P_{(Q_1 \wedge Q_2)} \vee P_{(R_1 \wedge R_2)}) \\
& \wedge\; (\neg P_{(Q_1 \wedge Q_2)} \vee P_{(F)}) \\
& \wedge\; (\neg P_{(R_1 \wedge R_2)} \vee P_{(F)})
\end{aligned}$$

- Luego:

$$F' :\; P_{(F)} \;\wedge\; \bigwedge_{G \in S_F} \mathsf{En}(G)$$

Algoritmos más avanzados

El procedimiento de resolución

- Este procedimiento para la verificación de satisfactibilidad se aplica sólo a fórmulas proposicionales en CNF.
- Por lo tanto, la fórmula sobre la que se aplicará el procedimiento debe transformarse previamente a una equisatisfactible en CNF.
- Este procedimiento se basa en la siguiente observación:

  Si la fórmula F contiene dos cláusulas $C_1[P]$ y $C_2[\neg P]$ que hacen referencia a cierta variable proposicional P en forma negativa en una y positiva en la otra, entonces para que F sea satisfactible, también debe serlo la cláusula:

$$C_1[\bot] \vee C_2[\bot]$$

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- El procedimiento de resolución considera entonces la siguiente regla de prueba, denominada regla de resolución clausal:

$$\frac{C_1[P] \qquad C_2[\neg P]}{C_1[\bot] \vee C_2[\bot]}$$

- El procedimiento de resolución consiste en aplicar esta regla para obtener resoluciones, y agregarlas como nuevas cláusulas en la conjunción de la fórmula original.
- Si en algún momento se agrega $\bot$, entonces F es insatisfactible.
- Caso contrario, F es satisfactible.

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Ejemplo:

$$(P \to Q) \land P \land \neg Q$$

- Su equivalente en CNF es:

$$F: \ (\neg P \lor Q) \ \land \ P \ \land \ \neg Q$$

- Aplicando la regla de resolución causal calculamos:

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Ejemplo:

$$(P \to Q) \wedge P \wedge \neg Q$$

- Su equivalente en CNF es:

$$F: \underbrace{(\neg P \vee Q)} \wedge \underbrace{P} \wedge \neg Q$$

- Aplicando la regla de resolución causal calculamos:

$$\frac{(\neg P \vee Q) \qquad P}{Q}$$

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Ejemplo:

$$(P \rightarrow Q) \wedge P \wedge \neg Q$$

- Su equivalente en CNF es:

$$F: \ (\neg P \vee Q) \ \wedge \ P \ \wedge \ \neg Q$$

- Aplicando la regla de resolución causal calculamos:

$$\frac{(\neg P \vee Q) \qquad P}{Q}$$

$$F_1: \ (\neg P \vee Q) \ \wedge \ P \ \wedge \ \neg Q \ \wedge \ Q$$

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Ejemplo:

$$(P \rightarrow Q) \wedge P \wedge \neg Q$$

- Su equivalente en CNF es:

$$F : \ (\neg P \vee Q) \ \wedge \ P \ \wedge \ \neg Q$$

- Aplicando la regla de resolución causal calculamos:

$$\frac{(\neg P \vee Q) \qquad P}{Q}$$

$$F_1 : \ (\neg P \vee Q) \ \wedge \ P \ \wedge \ \underbrace{\neg Q}_{} \ \wedge \ \underbrace{Q}_{}$$

$$\frac{\neg Q \qquad Q}{\perp}$$

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Ejemplo:

$$(P \to Q) \land P \land \neg Q$$

- Su equivalente en CNF es:

$$F: \ (\neg P \lor Q) \ \land \ P \ \land \ \neg Q$$

- Aplicando la regla de resolución causal calculamos:

$$\frac{(\neg P \lor Q) \qquad P}{Q}$$

$$F_1: \ (\neg P \lor Q) \ \land \ P \ \land \ \neg Q \ \land \ Q$$

$$\frac{\neg Q \qquad Q}{\perp}$$

Luego F es insatisfactible.

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Otro ejemplo:

$$F: \ (\neg P \vee Q) \ \wedge \ \neg Q$$

- La fórmula ya está en CNF.
- Aplicando la regla de resolución causal sólo podemos calcular:

$$\frac{(\neg P \vee Q) \qquad \neg Q}{\neg P}$$

- Haciendo la conjunción del resultado con la F, tenemos:

$$F_1: \ (\neg P \vee Q) \ \wedge \ \neg Q \ \wedge \ \neg P$$

- Dado que no podemos aplicar otra vez la regla de resolución, y no alcanzamos una contradicción, la fórmula es satisfactible.
- De hecho:

Algoritmos más avanzados

El procedimiento de resolución (cont.)

- Otro ejemplo:

$$F: \; (\neg P \vee Q) \; \wedge \; \neg Q$$

- La fórmula ya está en CNF.
- Aplicando la regla de resolución causal sólo podemos calcular:

$$\frac{(\neg P \vee Q) \qquad \neg Q}{\neg P}$$

- Haciendo la conjunción del resultado con la F, tenemos:

$$F_1: \; (\neg P \vee Q) \; \wedge \; \neg Q \; \wedge \; \neg P$$

- Dado que no podemos aplicar otra vez la regla de resolución, y no alcanzamos una contradicción, la fórmula es satisfactible.
- De hecho:

$$I: \; \{P \mapsto \text{false}, \; Q \mapsto \text{false}\}$$

Algoritmos más avanzados

Cláusulas de Horn: un caso particular de resolución

- Son cláusulas con a lo sumo un literal positivo:

$$\left(\bigvee_{i\in I} \neg p_i\right) \vee q$$

- La resolución de dos cláusulas de Horn resuelve a otra cláusula de Horn:

$$\frac{\left(\bigvee_{i\in I} \neg p_i\right) \vee q_k \qquad \left(\bigvee_{j\in J} \neg q_j\right) \vee r}{\left(\bigvee_{i\in I} \neg p_i\right) \vee \left(\bigvee_{j\in J-\{k\}} \neg q_j\right) \vee r}$$

Algoritmos más avanzados

Cláusulas de Horn: un caso particula

I puede ser vacío y
q puede no estar

- Son cláusulas con a lo sumo un literal positivo:

$$\left(\bigvee_{i\in I} \neg p_i\right) \vee q$$

- La resolución de dos cláusulas de Horn resuelve a otra cláusula de Horn:

$$\frac{\left(\bigvee_{i\in I} \neg p_i\right) \vee q_k \qquad\qquad \left(\bigvee_{j\in J} \neg q_j\right) \vee r}{\left(\bigvee_{i\in I} \neg p_i\right) \vee \left(\bigvee_{j\in J-\{k\}} \neg q_j\right) \vee r}$$

Algoritmos más avanzados

Cláusulas de Horn: un caso particular de resolución

- Una cláusula de Horn es equivalente a una de estas tres formas

$\left(\bigwedge_{i\in I} p_i\right) \Rightarrow q$	$\left(\bigwedge_{i\in I} p_i\right) \Rightarrow \bot$	$\top \Rightarrow q$
$\equiv$	$\equiv$	$\equiv$
$\left(\bigvee_{i\in I} \neg p_i\right) \vee q$	$\bigvee_{i\in I} \neg p_i$	q

Regla

Objetivo negado

Hecho

Algoritmos más avanzados

Cláusulas de Horn: un caso particular de resolución

- Un conjunto de cláusulas de Horn (las interesantes son reglas y hechos) establecen la especificación de un problema.
- Se desea ver si un objetivo dado se deduce de este conjunto.

  => se niega el objetivo y se aplica resolución.

- Notar que la resolución entre un objetivo negado y una regla o hecho resuelve a otro objetivo negado.

$$\frac{(\bigvee_{i\in I} \neg p_i) \vee q_k \qquad (\bigvee_{j\in J} \neg q_j)}{(\bigvee_{i\in I} \neg p_i) \vee (\bigvee_{j\in J-\{k\}} \neg q_j)} \qquad\qquad \frac{q_k \qquad (\bigvee_{j\in J} \neg q_j)}{(\bigvee_{j\in J-\{k\}} \neg q_j)}$$

- Si la resolución termina en una contradicción, entonces el objetivo se deduce de las cláusulas originales.

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

¿$hay\_gol\_de\_Messi$?

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$$juega\_Messi$$

$$juega\_Nesta$$

$$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$$

$$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$$

$$hay\_falta \Rightarrow hay\_tiro\_libre$$

$$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$$

$$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$$

$$\neg hay\_gol\_de\_Messi$$

Suponemos que:

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \land buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \land hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$$\neg hay\_gol\_de\_Messi$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$$\neg patea\_Messi$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$$juega\_Messi$$

$$juega\_Nesta$$

$$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$$

$$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$$

$$hay\_falta \Rightarrow hay\_tiro\_libre$$

$$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$$

$$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$$

$$\neg juega\_Messi \vee \neg hay\_tiro\_libre$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$$\neg juega\_Messi \vee \neg hay\_falta$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$$\neg juega\_Messi \vee \neg juega\_Nesta \vee \neg buen\_jugador\_en\_cancha$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \land buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \land hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$$\neg juega\_Messi \lor \neg juega\_Nesta$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$\boxed{juega\_Messi}$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$$\neg juega\_Messi$$

Algoritmos más avanzados

Cláusulas de Horn: ejemplo

$juega\_Messi$

$juega\_Nesta$

$juega\_Messi \Rightarrow buen\_jugador\_en\_cancha$

$(juega\_Nesta \wedge buen\_jugador\_en\_cancha) \Rightarrow hay\_falta$

$hay\_falta \Rightarrow hay\_tiro\_libre$

$(juega\_Messi \wedge hay\_tiro\_libre) \Rightarrow patea\_Messi$

$patea\_Messi \Rightarrow hay\_gol\_de\_Messi$

$\perp$

Por consiguiente, *hay_gol_de_Messi*

Algoritmos más avanzados

Resolución de cláusulas de Horn

- Trabaja sobre un subconjunto de fórmulas proposicionales (conjunción de cláusulas de Horn)
- La resolución de este tipo de fórmulas se puede realizar en tiempo lineal en el tamaño de la fórmula completa.
- Se puede extender a lógica de primer orden (LPO) donde las variables están cuantificadas universalmente.
- Aunque aún indecidible en general para LPO, son más “controlables” y con un mínimo de ayuda extra se pueden calcular muchas cosas.
- El lenguaje de programación PROLOG está basado en cláusulas de Horn de primer orden y en el algoritmo de resolución ya presentado.

Algoritmos más avanzados

El Algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

- Todos los SAT solvers modernos se basan en este algoritmo.
- DPLL también se aplica a fórmulas en CNF.
- DPLL combina el algoritmo SAT "clásico" con una forma restringida de resolución, llamada resolución de unidad:

$$\frac{\ell \qquad C[\neg \ell]}{C[\bot]}$$

- Notar que una de las cláusulas a la cual se aplica debe ser un único literal y la otra debe contener el literal negado.
- El resultado de la aplicación nos da una subcláusula de una de las cláusulas a las cuales se aplicó.
- A diferencia del algoritmo de resolución, la subcláusula $C[\bot]$ reemplaza a la cláusula original $C[\neg \ell]$ en la fórmula original.

Algoritmos más avanzados

El Algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

- Ejemplo (resolución de unidad):
- Dada la fórmula:

$$(P \vee Q) \wedge (\neg P \vee R) \wedge (\neg R \vee S) \wedge P$$

$$(P \vee Q) \wedge R \wedge (\neg R \vee S) \wedge P$$

$$(P \vee Q) \wedge R \wedge S \wedge P$$

Algoritmos más avanzados

El Algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

- El proceso de repetidamente aplicar resolución de unidad sobre una fórmula en CNF hasta que ya no sea posible se denomina boolean constraint propagaition (BCP).
- Ejemplo:

$$F: \quad P \quad \wedge \ (\neg P \vee Q) \ \wedge \ (R \vee \neg Q \vee S)$$

- Aplicando BCP calculamos:
  - $$\frac{P \qquad (\neg P \vee Q)}{Q} \qquad\qquad F': \quad P \ \wedge \ Q \ \wedge (R \vee \neg Q \vee S)$$
  - $$\frac{Q \qquad R \vee \neg Q \vee S}{R \vee S} \qquad\qquad F'': \quad P \ \wedge \ Q \ \wedge (R \vee S)$$

Algoritmos más avanzados

El Algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

- Finalmente, el algoritmo completo es:

$$
\begin{array}{l}
\texttt{let rec}\ \textsc{dpll}\ F = \\
\quad \texttt{let}\ F' = \textsc{bcp}\ F\ \texttt{in} \\
\quad \texttt{if}\ F' = \top\ \texttt{then true} \\
\quad \texttt{else if}\ F' = \bot\ \texttt{then false} \\
\quad \texttt{else} \\
\qquad \texttt{let}\ P = \textsc{choose}\ \mathsf{vars}(F')\ \texttt{in} \\
\qquad (\textsc{dpll}\ F'\{P \mapsto \top\})\ \lor\ (\textsc{dpll}\ F'\{P \mapsto \bot\})
\end{array}
$$

- Notar como se utiliza BCP para simplificar la fórmula antes de realizar las llamadas recursiva a DPLL.

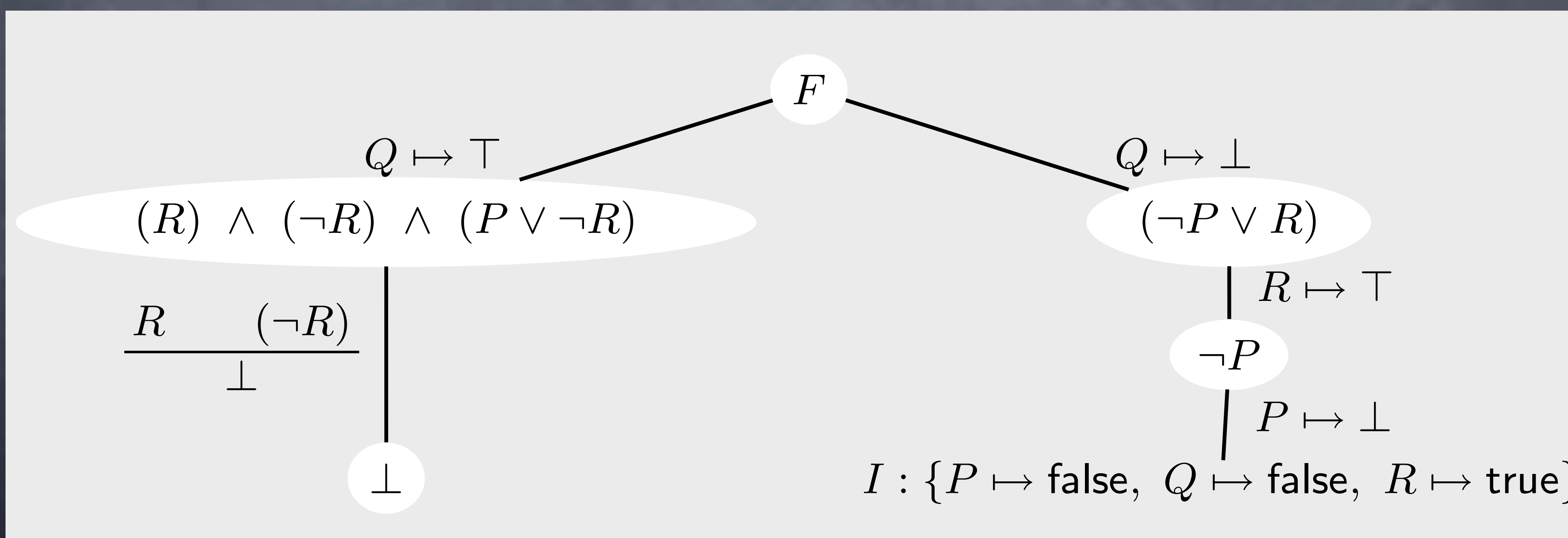
Algoritmos más avanzados

El Algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

- Ejemplo:

$$F: \ (\neg P \vee Q \vee R) \ \wedge \ (\neg Q \vee R) \ \wedge \ (\neg Q \vee \neg R) \ \wedge \ (P \vee \neg Q \vee \neg R)$$

- La aplicación de DPLL a esta fórmula puede graficarse de la siguiente manera:



Algoritmos más avanzados

El Algoritmo de Davis-Putnam-Logemann-Loveland (DPLL)

Una simple optimización del algoritmo es la siguiente:

- Si la variable P aparece sólo de manera positiva (o sólo de manera negativa) en F, no debería ser seleccionada por el algoritmo en $\mathrm{CHOOSE}\ \mathsf{vars}(F')$.
- En cualquier caso, F es equisatisfactible a la fórmula G que se obtiene de eliminar todas las cláusulas de F que contienen a P.
- Cuando sólo queden en F variables de este tipo, la fórmula será satisfactible.
- La interpretación correspondiente se construye asignando $\top$ a cada variable que aparece positiva, y $\perp$ a cada variable que aparece negativa.

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((P \land Q) \lor \neg R) \to (R \to Q))$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((\underbrace{(P \wedge Q)}_{S} \vee \neg R) \rightarrow (R \rightarrow Q))$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((\underbrace{(P \wedge Q)}_{S} \vee \neg R) \rightarrow (R \rightarrow Q))$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R}_{T}) \rightarrow (R \rightarrow Q))$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R}_{T}) \rightarrow (R \rightarrow Q))$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((\underbrace{P \wedge Q}_{S}) \vee \underbrace{\neg R}_{T}) \rightarrow \underbrace{(R \rightarrow Q)}_{U})$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(((\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R)}_{T} \to \underbrace{(R \to Q)}_{U}))$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(\underbrace{(\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R}_{T})}_{V} \rightarrow \underbrace{(R \rightarrow Q)}_{U})$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(\underbrace{(\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R}_{T})}_{V} \rightarrow \underbrace{(R \rightarrow Q)}_{U})$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$
$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg\underbrace{(\underbrace{(\underbrace{(P \land Q)}_{S} \lor \underbrace{\neg R}_{T})}_{V} \to \underbrace{(R \to Q)}_{U})}_{W}$$

$$(\neg S \lor P) \land (\neg S \lor Q) \land (\neg P \lor \neg Q \lor S) \land$$
$$(\neg T \lor \neg R) \land (R \lor T) \land$$
$$(\neg U \lor \neg R \lor Q) \land (R \lor U) \land (\neg Q \lor U) \land$$
$$(\neg V \lor S \lor T) \land (\neg S \lor V) \land (\neg T \lor V) \land$$

Ejemplo: Conversion a CNF equisatisfactible

$$\neg(\underbrace{(\underbrace{(\underbrace{(P \land Q)}_{S} \lor \underbrace{\neg R}_{T})}_{V} \to \underbrace{(R \to Q)}_{U})}_{W})$$

$$(\neg S \lor P) \land (\neg S \lor Q) \land (\neg P \lor \neg Q \lor S) \land$$

$$(\neg T \lor \neg R) \land (R \lor T) \land$$

$$(\neg U \lor \neg R \lor Q) \land (R \lor U) \land (\neg Q \lor U) \land$$

$$(\neg V \lor S \lor T) \land (\neg S \lor V) \land (\neg T \lor V) \land$$

$$(\neg W \lor \neg V \lor U) \land (V \lor W) \land (\neg U \lor W) \land$$

Ejemplo: Conversion a CNF equisatisfactible

$$\underbrace{\neg(\underbrace{(\underbrace{(\underbrace{P \wedge Q}_{S}) \vee \underbrace{\neg R}_{T}}_{V}) \rightarrow (\underbrace{R \rightarrow Q}_{U})}_{W})}_{X}$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\underbrace{\neg(\underbrace{\underbrace{(\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R}_{T})}_{V} \to \underbrace{(R \to Q)}_{U}}_{W})}_{X}$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge$$

$$(\neg X \vee \neg W) \wedge (X \vee W) \wedge$$

Ejemplo: Conversion a CNF equisatisfactible

$$\underbrace{\neg(\underbrace{\underbrace{(\underbrace{(P \wedge Q)}_{S} \vee \underbrace{\neg R}_{T})}_{V} \rightarrow \underbrace{(R \rightarrow Q)}_{U}}_{W})}_{X}$$

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge$$

$$(\neg X \vee \neg W) \wedge (X \vee W) \wedge$$

$$X$$

Ejemplo: DPLL

$$
\begin{aligned}
&(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge \\
&(\neg T \vee \neg R) \wedge (R \vee T) \wedge \\
&(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge \\
&(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge \\
&(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge \\
&(\neg X \vee \neg W) \wedge (X \vee W) \wedge \\
&X
\end{aligned}
$$

Ejemplo: DPLL

$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$

$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$

$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$

$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$

$(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge$

$\underline{(\neg X \vee \neg W)} \wedge (X \vee W) \wedge$

$\underline{X}$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge$$

$$(\qquad\quad \neg W) \wedge (X \vee W) \wedge$$

$$X$$

Ejemplo: DPLL

$$
\begin{aligned}
&(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge \\
&(\neg T \vee \neg R) \wedge (R \vee T) \wedge \\
&(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge \\
&(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge \\
&(\neg W \vee \neg V \vee U) \wedge (V \vee W) \wedge (\neg U \vee W) \wedge \\
&(\qquad\quad \neg W) \wedge (X \vee W) \wedge \\
&X
\end{aligned}
$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge \underline{(V \vee W)} \wedge \underline{(\neg U \vee W)} \wedge$$

$$(\qquad \underline{\neg W)} \wedge \underline{(X \vee W)} \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge \underline{(V \qquad\quad)} \wedge \underline{(\neg U \qquad\quad)} \wedge$$

$$(\qquad\quad \underline{\neg W}) \wedge \underline{(X \qquad\quad)} \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$
$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \vee \neg V \vee U) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$
$$(\qquad \neg W) \wedge (X \qquad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \vee U) \wedge (\neg Q \vee U) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \vee U) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$

$$(\qquad \neg W) \wedge (X \qquad) \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \lor P) \land (\neg S \lor Q) \land (\neg P \lor \neg Q \lor S) \land$$
$$(\neg T \lor \neg R) \land (R \lor T) \land$$
$$(\neg U \lor \neg R \lor Q) \land (R \qquad) \land (\neg Q \qquad) \land$$
$$(\neg V \lor S \lor T) \land (\neg S \lor V) \land (\neg T \lor V) \land$$
$$(\neg W \lor \neg V \qquad) \land (V \qquad) \land (\neg U \qquad) \land$$
$$(\qquad \neg W) \land (X \qquad) \land$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \vee Q) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$
$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \vee \neg V \qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$
$$(\qquad \neg W) \wedge (X \qquad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$

$$(\neg V \vee S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \vee \neg V \qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$

$$(\qquad \neg W) \wedge (X \qquad) \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \vee Q) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$
$$(\qquad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$
$$(\qquad \neg W) \wedge (X \qquad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \vee Q) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$
$$(\qquad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \qquad\qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$
$$(\qquad \neg W) \wedge (X \qquad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \vee Q) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \vee Q) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$$

$$(\quad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$$

$$(\quad \neg W) \wedge (X \quad) \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \quad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$
$$(\neg U \vee \neg R \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$$
$$(\quad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$$
$$(\quad \neg W) \wedge (X \quad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \quad\quad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \quad\quad) \wedge (R \quad\quad) \wedge (\neg Q \quad\quad) \wedge$$

$$(\quad\quad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \quad\quad\quad\quad) \wedge (V \quad\quad) \wedge (\neg U \quad\quad) \wedge$$

$$(\quad\quad \neg W) \wedge (X \quad\quad) \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \qquad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \vee \neg R) \wedge (R \vee T) \wedge$$

$$(\neg U \vee \neg R \qquad) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$

$$(\qquad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \qquad\qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$

$$(\qquad \neg W) \wedge (X \qquad) \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \qquad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \qquad) \wedge (R \vee T) \wedge$$
$$(\neg U \qquad) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$
$$(\qquad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$
$$(\qquad \neg W) \wedge (X \qquad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \quad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$
$$(\neg T \quad) \wedge (R \vee T) \wedge$$
$$(\neg U \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$$
$$(\quad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$
$$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$$
$$(\quad \neg W) \wedge (X \quad) \wedge$$
$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \qquad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \qquad) \wedge (R \vee T) \wedge$$

$$(\neg U \qquad) \wedge (R \qquad) \wedge (\neg Q \qquad) \wedge$$

$$(\qquad S \vee T) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \qquad) \wedge (V \qquad) \wedge (\neg U \qquad) \wedge$$

$$(\qquad \neg W) \wedge (X \qquad) \wedge$$

$$X$$

Ejemplo: DPLL

$(\neg S \vee P) \wedge (\neg S \quad) \wedge (\neg P \vee \neg Q \vee S) \wedge$

$(\neg T \quad) \wedge (R \quad) \wedge$

$(\neg U \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$

$(\quad S \quad) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$

$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$

$(\quad \neg W) \wedge (X \quad) \wedge$

X

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \quad) \wedge (\neg P \vee \neg Q \vee S) \wedge$$

$$(\neg T \quad) \wedge (R \quad) \wedge$$

$$(\neg U \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$$

$$(\quad S \quad) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$$

$$(\quad \neg W) \wedge (X \quad) \wedge$$

$$X$$

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\underline{\neg S} \quad) \wedge \underline{(\neg P \vee \neg Q \vee S)} \wedge$$

$$(\neg T \quad) \wedge (R \quad) \wedge$$

$$(\neg U \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$$

$$\underline{(\quad S \quad)} \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$$

$$(\quad \neg W) \wedge (X \quad) \wedge$$

$$X$$

Ejemplo: DPLL

$(\neg S \vee P) \wedge (\underline{\neg S} \quad) \wedge (\underline{\neg P \vee \neg Q \quad}) \wedge$

$(\neg T \quad) \wedge (R \quad) \wedge$

$(\neg U \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$

$(\underline{\quad\quad}) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$

$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$

$(\quad \neg W) \wedge (X \quad) \wedge$

X

Ejemplo: DPLL

$$(\neg S \vee P) \wedge (\neg S \quad) \wedge (\neg P \vee \neg Q \quad) \wedge$$

$$(\neg T \quad) \wedge (R \quad) \wedge$$

$$(\neg U \quad) \wedge (R \quad) \wedge (\neg Q \quad) \wedge$$

$$(\quad \perp \quad) \wedge (\neg S \vee V) \wedge (\neg T \vee V) \wedge$$

$$(\neg W \quad) \wedge (V \quad) \wedge (\neg U \quad) \wedge$$

$$(\quad \neg W) \wedge (X \quad) \wedge$$

$$X$$