Informática
UNIVERSIDAD DE MURCIA

FUNDAMENTOS LÓGICOS DE LA INFORMÁTICA

El problema y las técnicas SAT

PRIMER CURSO DEL GRADO DE INGENIERÍA INFORMÁTICA

PRIMER CUATRIMESTRE

2018-19

Índice del documento

Facultad de Informática

El Problema SAT

- **Problema de la Satisfacibilidad booleana (en adelante, SAT):** dada una oración lógica (f.b.f. en L0) formada por una serie de proposiciones atómicas (variables lógicas o booleanas), ¿existe una combinación de atribuciones de valores de verdad a las proposiciones atómicas, es decir, una interpretación, en la que la evaluación de la oración sea verdadera?
- El problema tiene su complementario, el Problema UNSAT: ¿cuál es la combinación de atribuciones de valores de las proposiciones atómicas que hacen a la oración falsa?
- **Es un problema de decision!!!!** SAT es un problema fundamental en lógica formal, teoría de computación, inferencia, razonamiento automatizado, ingeniería VLSI, entre otros. Stephen Cook, en 1971, demostró que era un problema NP-COMPLETO.
- Los problemas NP-COMPLETO son problemas para los cuales no se conoce un algoritmo de tiempo polinómico determinístico que los pueda resolver. Todos los problemas de esta clase tienen la propiedad de que, no obstante, cualquier solución suya puede ser verificada efectivamente.

- Su importancia reside básicamente en el hecho de que es el primer problema, y uno de los más simples, para el cual se demostró (teorema de Cook, 1971) que pertenece a la clase de complejidad NP-COMPLETO. Además muchos problemas prácticos, tales como la demostración automática de teoremas, la verificación y la detección de fallos de hardware, la detección de isomorfismo de grafos, el problema del viajante de comercio, etc. , son problemas NP-COMPLETO y se pueden transformar de manera eficiente a SAT. Por lo tanto, buenos algoritmos para SAT son indudablemente de gran utilidad. Dado que es poco probable que se pueda encontrar un algoritmo de tiempo polinómico para resolver SAT, buenas soluciones parecen importantes.
- **Realmente un problema de la clase NP-COMPLETO es un problema intratable**. No hay un algoritmo mejor que realizar una búsqueda exhaustiva de todas las posibilidades en la búsqueda de una solución al problema.
- Un caso particular del problema SAT es el 3-SAT en el que el objetivo del problema es determinar si una oración lógica (f.b.f. en L0) en forma normal conjuntiva con exactamente tres literales por cláusula, es satisfacible. Un aspecto muy importante en este contexto es que cualquier fórmula proposicional en forma normal conjuntiva puede ser transformada a una fórmula en forma normal conjuntiva con tres literales por cláusula.

La complejidad y las clases de problemas (I)

Clases de complejidad: Los problemas se clasifican en conjuntos de complejidad comparable llamados clases de complejidad.

- *Definición*: "La clase de complejidad **P** es el conjunto de los **problemas de decisión** que pueden ser resueltos en una máquina (de Turing) determinista en tiempo polinómico, lo que corresponde intuitivamente a problemas que pueden ser resueltos aún en el peor de sus casos".
- *Definición*: "La clase de complejidad **NP** es el conjunto de los **problemas de decisión** que pueden ser resueltos en tiempo polinómico por una máquina (de Turing) no-determinista". NP es el acrónimo en inglés de Polinómico No determinista (Non-Deterministic Polynomial-time).

  Imaginemos una máquina que ante varias alternativas, puede elegir la que quiera, y si una de ellas lleva a la solución de una manera "mágica" elegirá esa. Es una máquina **no determinista.**

  La clase NP es importante porque contiene muchos problemas de búsqueda y optimización para los que se desea saber si existe cierta solución o si existe una mejor solución que las conocidas. Dada su importancia, se han hecho muchos esfuerzos para encontrar algoritmos que resuelvan algún problema de NP en tiempo polinómico.

  Aún así, para algunos problemas de NP no es posible encontrar siquiera un algoritmo mejor que simplemente realizar una búsqueda exhaustiva (éstos son los problemas del conjunto NP-completo).
- *Definición*: "La clase de complejidad **NP-COMPLETO** es el subconjunto de los problemas de decisión en NP tal que todo problema en NP se puede transformar polinomialmente en cada uno de los problemas de NP-COMPLETO". Una transformación polinomial es un algoritmo determinista que transforma instrucciones de un problema en instrucciones del otro.

  Se puede decir que los problemas NP-COMPLETO son los problemas más difíciles de NP y, muy probablemente, no formen parte de la clase de complejidad P. La razón es que de tenerse una solución polinómica para un problema de NP-COMPLETO, todos los problemas de NP tendrían también una solución en tiempo polinómico.

  Como consecuencia de esta definición, de tenerse un algoritmo en P para uno de los problemas NP-COMPLETO, se tendría una solución en P para todos los problemas de NP. Esta definición dada fue propuesta por Stephen Cook en 1971.

La complejidad y las clases de problemas (II)

Ejemplos de Problemas de la clase P (polinomiales)

- Extraer cualquier elemento de un vector. La indexación en un vector lleva el mismo tiempo sea cual fuere el índice que se quiera buscar. Por tanto, es una operación de complejidad constante O(1).
- Buscar en un diccionario tiene complejidad logarítmica. Se puede iniciar la búsqueda de una palabra por la mitad del diccionario. Inmediatamente se sabe si se ha encontrado la palabra o, en el caso contrario, en cuál de las dos mitades hay que repetir el proceso (es un proceso recursivo) hasta llegar al resultado. En cada (sub)búsqueda el problema (las páginas en las que la palabra puede estar) se ha reducido a la mitad, lo que se corresponde con la función logarítmica. Este procedimiento de búsqueda (conocido como búsqueda binaria o quicksort) en una estructura ordenada tiene complejidad logarítmica O(logn).
- El proceso más común para ordenar un conjunto de elementos tiene complejidad cuadrática. El procedimiento consiste en crear una colección vacía de elementos. A ella se añade, en orden, el menor elemento del conjunto original que aún no haya sido elegido, lo que implica hacer un recorrido completo del conjunto original (O(n), siendo n el número de elementos del conjunto). Este recorrido sobre el conjunto original se realiza hasta que queda todos sus elementos están en la secuencia de resultado. Se puede ver que hay que hacer n selecciones (se ordena todo el conjunto) cada una con un coste n de ejecución: el procedimiento es de orden cuadrático $O(n^2)$. Hay que aclarar que hay diversos algoritmos de ordenación con mejores resultados que el descrito.

La complejidad y las clases de problemas (III)

Los Problemas de la clase NP-COMPLETO

- Como otro ejemplo de un problema NP-COMPLETO está el problema de la suma de subconjuntos que se puede enunciar como sigue: dado un conjunto S de enteros, ¿existe un subconjunto no vacío de S cuyos elementos sumen cero? Es fácil verificar si una respuesta es correcta, pero no se conoce mejor solución que explorar todos los 2n-1 subconjuntos posibles hasta encontrar uno que cumpla con la condición. Un ejemplo de esto sería, dado el conjunto { −7, −3, −2, 5, 8}, la pregunta anterior tiene la respuesta SI, porque el subconjunto { −3, −2, 5} está formado por unos elementos cuya suma es cero. Este problema, conjuntamente con SAT, es probablemente el más simple de explicar de los problemas NP-COMPLETO.
- Este es un problema importante en la teoría de la complejidad y en la criptografía. Un problema equivalente es: dado un conjunto de enteros y un entero *s*, ¿existe algún subconjunto cuya suma sea *s*? La suma de subconjuntos también puede verse como un caso especial del problema de la mochila.
- Actualmente, todos los algoritmos conocidos para problemas NP-COMPLETO utilizan tiempo exponencial con respecto al tamaño de la entrada. Se desconoce si hay mejores algoritmos, por la cual, para resolver un problema NP-COMPLETO de tamaño arbitrario se utiliza uno de los siguientes enfoques:
    - Aproximación: Un algoritmo que rápidamente encuentra una solución no necesariamente óptima, pero dentro de un cierto rango de error. En algunos casos, encontrar una buena aproximación es suficiente para resolver el problema, pero no todos los problemas NP-COMPLETO tienen buenos algoritmos de aproximación.
    - Probabilístico: Una algoritmo probabilístico obtiene en promedio una buena solución al problema planteado, para una distribución de los datos de entrada dada.
    - Casos particulares: Cuando se reconocen casos particulares del problema para los cuales existen soluciones rápidas.
    - Heurísticas: Un algoritmo que trabaja razonablemente bien en muchos casos. En general son rápidos, pero no existe medida de la calidad de la respuesta.

Facultad de Informática

La complejidad y las clases de problemas (y IV)

- En la siguiente tabla, N es el número de entradas al problema, y el COSTE o complejidad logarítmica es el número de pasos secuenciales para culminarlo o resolverlo en el peor de los casos (en una búsqueda exhaustiva).
- Los problemas NP-COMPLETO tienen coste exponencial.
- A título de comparación de las magnitudes que se manejan: El número de protones en el universo tiene 80 dígitos. El número de microsegundos transcurridos desde el "Big Bang" tiene 24 dígitos

	N / COSTE	10	50	100	300	1.000
Polinomial	**5N**	50	250	500	1.500	5.000
	NlogN	33	282	665	2.469	9.966
	N^2	100	2.500	10.000	90.000	1.000.000
	N^3	1.000	125.000	1.000.000	27 millones	Mil millones
Exponencial	**2^N**	1.024	16 dígitos	31 dígitos	91 dígitos	302 dígitos
	N!	3,6 millones	65 dígitos	161 dígitos	623 dígitos	inimaginable
	N^N	10 mil millones	85 dígitos	201 dígitos	744 dígitos	inimaginable

Facultad de Informática

Las Técnicas SAT

- En L0 el problema SAT y el problema UNSAT son decidibles.
- Algoritmos completos (1): Los que proporcionando una interpretación, ésta es un modelo o un contramodelo.
  - Tablas de verdad.
  - Algoritmo DP (Davis; Martin; Putnam; y Hillary. 1960)
  - Algoritmo DPLL (Davis; Putnam; Logemann; y Loveland. 1962)
  - Árboles semánticos.
  - Tableaux semánticos.
  - ...
- Algoritmos no Completos (2):
  - GSAT (Greedy SAT)
  - WalkSat (GSAT con probabilidades)
  - SA-SAT (Simulating Annealing SAT)
  - ...

(1) Determinan tanto la satisfacibilidad como la insatisfacibilidad.

(2) Puede que no encuentre un modelo, aún existiendo.

Tipos de Técnicas SAT

- Las Técnicas que vamos a estudiar para resolver el problema SAT son aquellas que implementan aproximaciones fértiles, generalmente, mediante búsquedas en profundidad, a la pregunta (decisión) de si para una fórmula proposicional cualquiera existirá una interpretación que la haga verdadera (o, falsa).
  - Si una interpretación determinada hace a la fórmula verdadera, diremos que la oración es SATISFACIBLE.
  - No sabremos que una oración será TAUTOLOGÍA, en general, si no podemos asegurar que en todas las interpretaciones la fórmula es evaluada como verdadera. En todo caso, no será TAUTOLOGÍA si es FALSEABLE.
  - Si una interpretación determinada hace a la fórmula falsa, diremos que la oración es FALSEABLE.
  - No sabremos que una oración es una CONTRADICCIÓN (Insatisfacible), en general, si no podemos garantizar que será FALSEABLE en todas las interpretaciones.
- Vamos a clasificar las Técnicas SAT en función del tipo de oración de partida al aplicar la técnica. Básicamente, tenemos dos tipos de técnicas: el primero, aquellas que parten de oraciones expresadas en su FORMA CLAUSAL (previa transformación en FNC); y el segundo, aquellas que parten de la fórmula original sin necesidad de transformar en FNC.
  - Las Técnicas SAT aplicables a partir de FORMAS CLAUSALES:
    - Algoritmo DPLL
    - Algoritmo de Resolución
  - Las Técnicas SAT aplicables a fórmulas cualesquiera
    - Tablas de Verdad
    - Árboles Semánticos
    - Tableaux Semánticos

Formas Normales Conjuntivas y Cláusulas (I)

Definiciones

- Un literal es una proposición atómica o su negada
- Una cláusula es la disyunción de literales.
- Una cláusula es un conjunto finito de literales.
- Una cláusula con sólo un literal se llama cláusula unitaria.
- Una cláusula sin literales se llama cláusula vacía y se denota por $\square$.
- Una cláusula de Horn es una cláusula con a lo sumo un literal positivo.
    - La expresión general es de la forma: **C** = $P \vee \neg Q1 \vee \neg Q2 \vee \ldots \vee \neg Qn$
    - Si n = 0, se llama hecho.
    - Si no hay literal positivo P, se llama objetivo.
    - Si n > 0 y existe el literal positivo P, se llama regla. Se utiliza para razonamiento "hacia atrás".
- Una fórmula, **α**, está en forma normal conjuntiva (FNC) si responde a la expresión **ξ** = $(\wedge_i (\vee_j P_{ij}))$ donde Pij son literales (de L0). LO VIMOS AL TRATAR LA EQUIVALENCIA LÓGICA!!!
- Un conjunto clausal o clausulado es un conjunto de cláusulas expresadas como conjuntos de literales. El conjunto clausal **{{**¬ p; q**}**; **{** r**}}** es la forma clausal de las fórmulas $((p \rightarrow q) \wedge r)$ y $((\neg q \rightarrow \neg p) \wedge r)$
- Una fórmula está en forma clausal si se expresa como un conjunto clausal. La forma clausal de la FNC $((p \vee q) \wedge \neg p)$, es **{{** p; q**}**; **{**¬ p**}}**. **APLICACIÓN DE FÓRMULAS EN CONJUNTOS CLAUSALES. OJO!!!!**
- **La cláusula vacía $\square$, la disyunción de ningún literal, es una oración insatisfacible. POR DEFINICIÓN!!!!**
- **El conjunto vacío de oraciones {Ø} (ya sea o no de cláusulas), es un conjunto satisfacible. MUY IMPORTANTE!!!!**

Formas Normales Conjuntivas y Cláusulas (II)

Construcción de una FNC aplicando transformaciones por equivalencia lógica. En las páginas 13 y 14 del documento 03-SAT-B disponible en RECURSOS tienen detallados cada uno de los pasos secuenciales para llegar a la FNC de una oración dada.

- Tengamos la siguiente oración: $\mathbf{((p \rightarrow (q \rightarrow r)) \rightarrow (p \land q \rightarrow r))}$
  - Aplicando la interdefinición en el antecedente: $((\neg p \lor (\neg q \lor r)) \rightarrow (p \land q \rightarrow r))$
  - Aplicando la interdefinición en el consecuente: $((\neg p \lor (\neg q \lor r)) \rightarrow (\neg(p \land q) \lor r))$
  - Aplicando la interdefinición a la implicación central: $(\neg(\neg p \lor (\neg q \lor r)) \lor (\neg(p \land q) \lor r))$
  - Aplicando la ley de D'Morgan: $((\neg\neg p \land \neg(\neg q \lor r)) \lor (\neg p \lor \neg q) \lor r)$
  - Aplicando la ley de D'Morgan: $((\neg\neg p \land (\neg\neg q \land \neg r)) \lor (\neg p \lor \neg q) \lor r)$
  - Aplicando la idempotencia y optimizando paréntesis: $((p \land q \land \neg r) \lor (\neg p \lor \neg q \lor r))$
  - Aplicando la propiedad distributiva de la disyunción con respecto a la conjunción, tomando los paréntesis como son: $((p \lor \neg p \lor \neg q \lor r) \land (q \lor \neg p \lor \neg q \lor r) \land (\neg r \lor \neg p \lor \neg q \lor r))$
  - Aplicando la eliminación de literales opuestos: $((V \lor \neg q \lor r) \land (V \lor \neg p \lor r) \land (V \lor \neg p \lor \neg q))$
  - Y de ahí, la FNC será $((V) \land (V) \land (V)) \equiv (V)$, con la que sabemos que **la oración de partida es una Tautología** al ser las tres cláusulas oraciones verdaderas. Con lo cual, su Forma Clausal es: **{Ø}**, el conjunto vacío de cláusulas.

Formas Normales Conjuntivas y Cláusulas (III)

Construcción de una FNC aplicando transformaciones por equivalencia lógica. En las páginas 13 y 14 del documento 03-SAT-B disponible en RECURSOS tienen detallados cada uno de los pasos secuenciales para llegar a la FNC de una oración dada.

- Tengamos ahora la siguiente oración: $\mathbf{((\neg p \lor \neg q \rightarrow p \land r) \land (r \rightarrow \neg q))}$
  - Aplicando la interdefinición: $((\neg(\neg p \lor \neg q) \lor (p \land r)) \land (\neg r \lor \neg q))$
  - Aplicando la ley de D'Morgan: $(((\neg\neg p \land \neg\neg q) \lor (p \land r)) \land (\neg r \lor \neg q))$
  - Aplicando la idempotencia: $(((p \land q) \lor (p \land r)) \land (\neg r \lor \neg q))$
  - Aplicando la propiedad distributiva de la disyunción con respecto a la conjunción en el primer paréntesis: $(((p \lor p) \land (p \lor r) \land (q \lor p) \land (q \lor r)) \land (\neg r \lor \neg q))$
  - Simplificando paréntesis por asociatividad: $((p \lor p) \land (p \lor r) \land (q \lor p) \land (q \lor r) \land (\neg r \lor \neg q))$
  - Simplificando al eliminar literales iguales: $((p) \land (p \lor r) \land (q \lor p) \land (q \lor r) \land (\neg r \lor \neg q))$
  - Y, finalmente, subsumiendo cláusulas: $\mathbf{((p) \land (q \lor r) \land (\neg r \lor \neg q))}$
  - Esta es la FNC, con la que podremos resolver la satisfacibilidad de la oración de origen. Sabemos que la **oración de partida NO es una Tautología. La asignación v(p) = F, la hace Falseable. Veremos si es Satisfacible!!!**
  - Su forma clausal es: $\{\{p\}; \{q; r\}; \{\neg r; \neg q\}\}$

La Propagación de Literales (I)

La Propagación de Literales puede verse en las páginas 16 y 17 del documento 03-SAT-B disponible en RECURSOS.

- Cuando tenemos una fórmula en su expresión FNC = **ξ**, la acción de propagar un determinado literal (ℓ), consiste en simplificar dicha expresión teniendo en cuenta la asignación a dicho literal del valor VERDAD. La nueva FNC la denotaremos por **ξ (** ℓ **).**
  - Así, las cláusulas en las que aparezca dicho literal tendremos la disyunción de la constante V con el resto de los literales, lo que es equivalente lógicamente a VERDAD. Al ser la constante V el elemento neutro de la conjunción, dichas cláusulas pueden cancelarse o suprimirse de la nueva FNC = **ξ (** ℓ **).**
  - Al mismo tiempo, en las cláusulas en las que aparezca dicho literal negado, lo sustituiremos por la constante F, con lo que nos aparecerá una disyunción de dicha constante con el resto de la cláusula. Dado que la constante F es el elemento neutro para la disyunción, la ocurrencia de dicho literal negado puede suprimirse de las cláusulas en las que aparezca, culminando la formación de la nueva FNC = **ξ (** ℓ **).**
- Si tenemos la oración: **α =** $((\neg p \vee \neg q \rightarrow p \wedge r) \wedge (r \rightarrow \neg q))$
- Hemos visto que su FNC es: **ξ =** $((p) \wedge (q \vee r) \wedge (\neg r \vee \neg q))$
  - Si propagamos el literal **(** ℓ **) = p**, la nueva FNC ξ(p) resultante es: **ξ(p) =** $((q \vee r) \wedge (\neg r \vee \neg q))$
  - Si en esta FNC **ξ(p)**, propagamos el literal **(** ℓ **) = q**, la nueva resultante es: **ξ(p, q) = (**$\neg$ r)
  - Si, finalmente, propagamos el literal **(** ℓ **) =** $\neg$ **r**, la nueva resultante es el conjunto vacío de cláusulas: **ξ(p, q,** $\neg$ **r) = {Ø}**
  - ❖ Si en un diferente intento de propagación, desde la FNC original propagamos inicialmente el literal **(** ℓ **) =** $\neg$ **q**, la nueva resultante es: **ξ(**$\neg$ **q) =** $((p) \wedge (r))$
  - ❖ Si propagamos ahora **(** ℓ **) = p**, nos quedaría **ξ(**$\neg$ **q, p) = (r).** Y si, finalmente, propagamos **(** ℓ **) = r**, nos quedaría como antes que la nueva FNC es: **ξ(**$\neg$ **q, p, r) = {Ø}**

La Propagación de Literales (II)

INTERP			EVALUACIONES: $\alpha = ((\neg p \vee \neg q \rightarrow p \wedge r) \wedge (r \rightarrow \neg q))$ y su FNC: $\xi = ((p) \wedge (q \vee r) \wedge (\neg r \vee \neg q))$														
p	q	r	$\neg p$	$\neg q$	$\neg r$	(1) $\neg p \vee \neg q$	(2) $p \wedge r$	$r \rightarrow \neg q$	(1)$\rightarrow$(2)	α	p	$q \vee r$	$\neg r \vee \neg q$	ξ	ξ (p)	ξ (p, q)	ξ (p, q, $\neg$r)
V	V	V	F	F	F	F	V	F	V	F	V	V	F	F	F	F	
V	V	F	F	F	V	F	F	V	V	V	V	V	V	V	V	V	V
V	F	V	F	V	F	V	V	V	V	V	V	V	V	V	V	F	
V	F	F	F	V	V	V	F	V	F	F	V	F	V	F	F	V	
F	V	V	V	F	F	V	F	F	F	F	F	V	F	F	F	F	
F	V	F	V	F	V	V	F	V	F	F	F	V	V	F	V	V	
F	F	V	V	V	F	V	F	V	F	F	F	V	V	F	V	F	
F	F	F	V	V	V	V	F	V	F	F	F	F	V	F	F	V	

- **ξ (p)** $= ((q \vee r) \wedge (\neg r \vee \neg q))$; **$\xi$ (p, q) $= (\neg r)$; ξ (p, q, $\neg$ r) $= \{\emptyset\}$**
- Podemos darnos cuenta que **α** $\equiv \xi$, pero **$\xi \not\equiv \xi$ (p) $\not\equiv \xi$ (p, q) $\not\equiv \xi$ (p, q, $\neg$ r). La propagación de literales no mantiene la equivalencia lógica en sentido estricto, pero sí podemos decir que $\alpha(v(p)=V) \equiv \xi$ (p), y así, en las sucesivas propagaciones.**
- **La interpretación: v(p) = V, v(q) = V, v(r) = F, hace a ξ (p, q, $\neg$ r) SATISFACIBLE, cuando ξ, y por tanto α, lo es para esa misma interpretación. Podemos decir que la propagación mantiene la satisfacibilidad, y que el conjunto de las cláusulas de la forma clausal de la oración α es un conjunto satisfacible para dicha interpretación, por lo que será un modelo.**

La Propagación de Literales (y III)

INTERP			EVALUACIONES: α = ((¬p ∨ ¬q→p ∧ r) ∧ (r → ¬q)) y su FNC: ξ = ((p) ∧(q ∨ r) ∧ (¬ r ∨ ¬q))														
p	q	r	¬p	¬q	¬r	(1) ¬p ∨ ¬q	(2) p ∧ r	r → ¬q	(1)→(2)	α	p	q ∨ r	¬r ∨ ¬q	ξ	ξ (¬q)	ξ (¬q, p)	ξ (¬q, p, r)
V	V	V	F	F	F	F	V	F	V	F	V	V	F	F	V	V	
V	V	F	F	F	V	F	F	V	V	V	V	V	V	V	F	F	
V	F	V	F	V	F	V	V	V	V	V	V	V	V	V	V	V	V
V	F	F	F	V	V	V	F	V	F	F	V	F	V	F	F	F	
F	V	V	V	F	F	V	F	F	F	F	F	V	F	F	F	V	
F	V	F	V	F	V	V	F	V	F	F	F	V	V	F	F	F	
F	F	V	V	V	F	V	F	V	F	F	F	V	V	F	F	V	
F	F	F	V	V	V	V	F	V	F	F	F	F	V	F	F	F	

- **ξ (¬ q) =** ((p) ∧ (r)); **ξ (¬ q, p) =** (r); **ξ (¬ q, p, r) = {Ø}**
- Podemos darnos cuenta que **α** ≡ **ξ**, pero **ξ ≢ ξ (¬ q) ≢ ξ (¬ q, p) ≢ ξ (¬ q, p, r). La propagación de literales no mantiene la equivalencia lógica en sentido estricto, pero sí podemos decir que α(v(¬ q)=V)** ≡ **ξ (¬ q), y así, en las sucesivas propagaciones.**
- **La interpretación: v(p) = V, v(q) = F, v(r) = V, hace a ξ (¬ q, p, r) SATISFACIBLE, cuando ξ, y por tanto α, lo es para esa misma interpretación. Podemos decir que la propagación mantiene la satisfacibilidad, y que el conjunto de las cláusulas de la forma clausal de la oración α es un conjunto satisfacible para dicha interpretación, por lo que será un modelo.**

El Algoritmo DPLL (I)

- Para estudiar la satisfacibilidad de **α** en su FNC = **ξ (**mejor, en su FORMA CLAUSAL) se aplica un algoritmo que consiste en realizar transformaciones sucesivas que manteniendo la satisfacibilidad, sean equivalentes en los términos explicados antes a la fórmula original.
- El método considera cinco reglas:
  - **Regla de la tautología:** Eliminar las cláusulas que contengan un par de literales complementarios.
  - **Regla de la inclusión:** (o de eliminación de cláusulas subsumidas): Si en **ξ** existen conjuntos clausales C1 y C2 tales que C1 es un subconjunto de C2, eliminar C2 de **ξ**. De ahí que, si existe una cláusula C1 cuyos literales están también en otra cláusula C2, eliminar C2 de **ξ**.
  - **Regla de propagación unitaria:** Si existe una cláusula unitaria formada por un único literal (ℓ), propagar el literal (ℓ).
  - **Regla del literal puro:** Si existe un literal puro (ℓ) (para el que no existe su complementario en otras cláusulas), propagar el literal (ℓ).
  - **Regla de ramificación:** Si no existe una cláusula unitaria y no existen literales puros, considerar un literal (ℓ) de alguna cláusula (de forma no determinística) y reducir el problema de satisfacibilidad a resolver uno de estos dos problemas:
    - o bien, la satisfacibilidad de $\xi \cup \{\ell\}$ (propagando (ℓ))
    - o bien, la satisfacibilidad de $\xi \cup \{\neg \ell\}$ (propagando ($\neg \ell$))

El Algoritmo DPLL (II)

En las **páginas 20-22** del documento 03-SAT-B que tienen en RECURSOS, tienen desarrollado **el algoritmo que implementa la técnica DPLL**.

- Si al propagar sucesivamente un conjunto de literales la **ξ** $((\ell)_1, (\ell)_2, \ldots, (\ell)_n)$ resultante es el **{Ø}**, entonces diremos que la FNC es SATISFACIBLE para la interpretación correspondiente al conjunto de los literales propagados (y su conjunto clausulado tendrá un modelo para la misma, por lo que será un conjunto SATISFACIBLE).
- Si al propagar sucesivamente un conjunto de literales la **ξ** $((\ell)_1, (\ell)_2, \ldots, (\ell)_n)$ resultante es la cláusula vacía **☐**, entonces diremos que la FNC es FALSEABLE para la interpretación correspondiente al conjunto de los literales propagados (y su conjunto clausulado tendrá un contramodelo para la misma).
- Únicamente si al aplicar la Regla de la Tautología al inicio del algoritmo nos quedamos como resultado el **{Ø}**, el conjunto vacío de cláusulas, diremos que la FNC **ξ** es una TAUTOLOGÍA. En cualquier otro caso, sólo podremos decir que es SATISFACIBLE.
- Una FNC **ξ** será CONTRADICCIÓN O INSATISFACIBLE, si al ser FALSEADA en la propagación **ξ** $(\ldots (\ell)_i)$ también lo es en la **ξ** $(\ldots (\neg\ell)_i)$, siendo el literal $()_i$ la primera aplicación de la Regla de Ramificación.
- Es decir, superadas las Reglas de Tautología, de Inclusión, de Propagación Unitaria y de Literales Puros, para comprobar si una FNC (y por tanto su oración original) es INSATISFACIBLE, necesariamente debemos hacer "backtraking" (vuelta atrás) en la primera Regla de Ramificación.
- Al propagar un literal puro nunca se obtendrá la cláusula vacía. Tampoco la obtendremos, en general, al propagar el literal de una cláusula unitaria. Para obtener una FNC INSATISFACIBLE de manera trivial deberíamos tener dos cláusulas unitarias de literales opuestos en el conjunto clausal original o de partida.
- Si al aplicar la "n" vez (la enésima vez) la Regla de Ramificación, resultaran falseadas su dos propagaciones, debemos hacer backtraking a la "n-1" aplicación, y propagar el literal negado: puede ser que en esta alternativa de propagación la FNC resultara SATISFACIBLE. Esta operación de vuelta atrás debe hacerse cuantas veces haga falta.

El Algoritmo DPLL (III)

Dos ejemplos:

- **ξ = {{**p; ¬q; ¬r**}**; {p; r}; **{**¬ p; ¬q**}}**
- No podemos aplicar la regla de la Tautología, tampoco la regla de la inclusión. No existen cláusulas unitarias, pero si tenemos un literal puro (¬q), así que propagamos dicho literal:
- **ξ (**¬q) = **{{**p; r**}}**
- En esta nueva forma clausal tenemos una cláusula con dos literales puros, por tanto, propagamos cualquiera de ellos:
- **ξ (**¬q, p) = **{Ø}**
- Obtenemos el conjunto vacío de cláusulas, lo que nos indica que el conjunto clausal es **SATISFACIBLE** y la oración que representa, también lo es.

- **ϕ = {{**¬ p**}**; {p; q}; **{**p; ¬q**}}**
- No podemos aplicar la regla de la Tautología, tampoco la regla de la inclusión. Tenemos una cláusula unitaria cuyo único literal el (¬ p); por lo tanto, propagamos dicho literal:
- **ϕ** (¬ p) = **{{**q**}**; **{**¬q**}}**
- Ahora tenemos dos cláusulas unitarias, propagamos cualquiera de los dos literales:
- **ϕ** (¬ p, q) = **{**□**}**
- Al resultar la cláusula vacía, se ha FALSEADO la oración, por lo que debemos hacer backtraking y propagar (¬q):
- **ϕ** (¬ p, ¬q) = **{**□**}**
- Como en las dos ramas el resultado es la cláusula vacía, esto quiere decir que el conjunto clausulado es **INSATISFACIBLE**, y la FNC es una oración INSATISFACIBLE. Hay que advertir que si volvemos al primer paso y en vez de propagar (¬ p), propagamos su literal complementario (p), entonces la cláusula unitaria de partida se transforma en la cláusula vacía, es decir, asimismo interpretación falseable.

El Algoritmo DPLL (y IV)

Ejemplo en 3-SAT:

- **ξ = {**{p; ¬q; r}; {¬ p; q; s}; **{**¬ p; ¬q; ¬r**}; {**¬ p; ¬q; r**}; {**q; r; ¬s**}}**
- Tenemos cinco cláusulas, cada una con tres literales, y ninguna es trivial (cláusula en la que aparece un literal y su negado). No podemos aplicar la regla de inclusión. Tampoco tenemos cláusulas unitarias, y tampoco literales puros. Debemos aplicar la Regla de Ramificación, propagando cualquiera de los literales, pero con la idea puesta en que deberemos hacer backtraking, si llega el caso. Propagamos el literal (p), es decir el valor v(p) = V:
- **ξ (**p**)** = **{{**q; s**}; {**¬q; ¬r**}; {**¬q; r**}; {**q; r; ¬s**}}**
- Seguimos teniendo una forma clausal sin cláusulas unitarias, ni literales puros. Propagamos el literal (q), es decir v(q) = V:
- **ξ (**p, q**)** = **{{**¬r**}; {**r**}}**
- Obtenemos una forma clausal con dos cláusulas unitarias cuyos literales son opuestos o complementarios. Al propagar v(r) = V obtenemos la clásula vacía, y al hacer backtraking y propagar v(¬r) = V, obtenemos de nuevo la clásula vacía, lo que nos dice que se falsean las dos líneas de propagación.
- **ξ (**p, q, r**)** = **ξ (**p, q, ¬ r**)** = **{**□**}**
- ESTO NOS DICE QUE HEMOS LLEGADO A UN CONJUNTO CLAUSULADO INSATISFACIBLE??? PUES NO!!!! Debemos volver a la segunda propagación, es decir aquella en la que propagamos el literal (q), y ahora propagar el literal negado (¬q):
- **ξ (**p, ¬q**)** = **{{**s**}; {**r; ¬s**}}**
- Ahora en esta nueva alternativa de propagación, tenemos la cláusula unitaria formada por el literal (s), con lo que propagamos dicho literal, v(s) = V
- **ξ (**p, ¬q, s**)** = **{{**r**}}**
- Tenemos una forma clausal que es una única cláusula unitaria de literal (r), lo propagamos y entonces la nueva forma clausal será el conjunto vacío de cláusulas.
- **ξ (**p, ¬q, s, r**)** = **{Ø}**
- Esto nos dice que bajo esta propagación de literales, y por tanto, en esta interpretación, **el conjunto clausal es SATISFACIBLE**. Si bien para otras propagaciones (interpretaciones) quedaba FALSEADO, tenía contramodelos, pero no hemos podido garantizar que todas las interpretaciones falsearan el conjunto clausal, y con ello, la oración de la que se partía: **Por tanto, no es INSATISFACIBLE**.

El Método de Resolución (I)

Recordemos el razonamiento básico VÁLIDO que vimos en el tema anterior, llamado el DILEMA CONSTRUCTIVO

- $\{\alpha \lor \beta; \alpha \to \delta; \beta \to \gamma\} \vDash \delta \lor \gamma$
- Si definimos a $\alpha = p$ y a $\beta = \neg p$, tendremos $\{p \lor \neg p; p \to \delta; \neg p \to \gamma\} \vDash \delta \lor \gamma$. Aplicando la equivalencia de interdefinición de la implicación con la disyunción: $\{p \lor \neg p; \neg p \lor \delta; p \lor \gamma\} \vDash \delta \lor \gamma$
- Suponiendo que δ y que γ sean expresiones clausules, es decir, disyunción de literales, el conjunto de premisas y la conclusión de la consecuencia lógica anterior, son conjuntos clausales. En las premisas, tenemos una cláusula trivial, la primera de ellas, que puede ser cancelada o suprimida sin problemas. Las otra dos, son cláusulas en la que aparece un literal en una, que en la otra aparece negado: dos literales complementarios u opuestos, uno en cada una de las dos premisas. El resultado es una conclusión, que será una cláusula formada por el resto de los literales de las dos premisas, una vez canceladas o suprimidas las ocurrencias tanto del literal **(p)** como de su negado **($\neg$p)**, es decir, **($\delta \lor \gamma$)**.
- Esto mismo podíamos haberlo obtenido partiendo del SILOGISMO DISYUNTIVO AMPLIADO.
- **Esta es la esencia del método de Resolución. Procedimiento que puede enunciarse de la siguiente manera:**

- Se define la **resolvente** de las clausulas C1 y C2 respecto del literal (ℓ) a la cláusula R_ℓ(C1 ; C2), dada por las siguientes condiciones y procedimiento:
  - Tanto en C1 como en C2 no existen literales iguales ni literales opuestos.
  - Existen $(\ell)_i$ en C1 y $(\ell)_j$ en C2 tales que $(\ell)_i = \neg(\ell)_j = (\ell)$, que son literales complementarios u opuestos.
  - Si es así: Construir R_ℓ(C1 ; C2) = **{{C1 - $(\ell)_i$} ∪ {C2 - $(\ell)_j$}}**
- C1 y C2 se llaman las cláusulas padres de R_ℓ(C1 ; C2), y podemos escribir que: {C1; C2} $\vDash$ R_ℓ(C1 ; C2)

- **Muy importante: {C1; C2} es SATISFACIBLE si y sólo si R_ℓ(C1 ; C2) es SATISFACIBLE. Si R_ℓ(C1 ; C2) fuera la cláusula vacía, entonces {C1; C2} será un conjunto INSATISFACIBLE.**

El Método de Resolución (II)

En la página 25 del documento 03-SAT-B que tienen en RECURSOS pueden ver el formalismo del ALGORITMO DE RESOLUCIÓN planteado por Robinson en 1965. En la página 26 pueden leer las definiciones de Grafo de Resolución y Grafo de Refutación.

- La resolución de Robinson evita la generación de conjuntos de instancias básicas como exigen otros métodos. Realmente, el algoritmo de Robinson concreta su función como **la UNIFICACIÓN de expresiones clausales** (en L1 para que se pueda aplicar la Unificación por Resolución previamente hay que realizar sustituciones de variables). Por eso, es conocido como el Algoritmo de Unificación de Robinson. Dicha Unificación viene definida en L0 por la cláusula Resolvente $\mathbf{R_{\ell}}$ **(C1 ; C2)** de las cláusulas **C1** y **C2**, vista anteriormente aplicando una relación semántica de Consecuencia Lógica.
- Puede ser aplicada directamente a cualquier conjunto **Ω** de cláusulas para sondear su insatisfacibilidad. La idea esencial del álgoritmo de resolución es indagar si **Ω** contiene la cláusula vacía □ o no: si **Ω** contiene □, entonces es insatisfacible. Si **Ω** no contiene □, la siguiente cosa es saber cuando la cláusula vacía puede ser derivada de dicho conjunto. Ciertamente el algoritmo de resolución puede ser entendido como un método semántico para generar nuevas cláusulas a partir de las existentes y añadirlas al conjunto clausal previo. La clave del asunto está en que el "agrandamiento" (fundamentado en la propiedad de Monotonía y en la propiedad reflexiva de la Consecuencia Lógica) de **Ω** con sus consecuencias lógicas trasmite la Insatisfacibilidad; de forma que si llegamos a generar la cláusula vacía por el algoritmo de resolución, **Ω** será INSATISFACIBLE. Y si, finalmente, no llegamos a generar la cláusula vacía tras contrastar todas las cláusulas del conjunto **Ω** agrandado, podremos decir que el **Ω** de partida es SATISFACIBLE.
- **El método de Resolución, y el algoritmo de Unificación de Robinson, es adecuado para encontrar la Insatisfacibilidad** de las oraciones en su forma clausal. El método no encuentra las interpretaciones que falsean la oración, sino que directamente garantiza que el conjunto clausal de partida es insatisfacible y con ello, que la oración original será INSATISFACIBLE.
- Al no posibilitar una forma de identificar interpretaciones en las que se falsee, o bien en las que se satisfaga el conjunto clausal, no es un método adecuado para responder a la pregunta de si una oración será una TAUTOLOGÍA.
- Un conjunto clausal será SATISFACIBLE, y con ello la oración original, cuando no se obtenga como resolvente la cláusula vacía en ninguna de las resoluciones. Para que se obtenga ésta, las cláusulas padres serán dos cláusulas unitarias con literales opuestos o complementarios, por ejemplo: $\{\{p\}; \{\neg p\}\} \models R_{p} (\{p\}; \{\neg p\}) = \square$

El Método de Resolución (III)

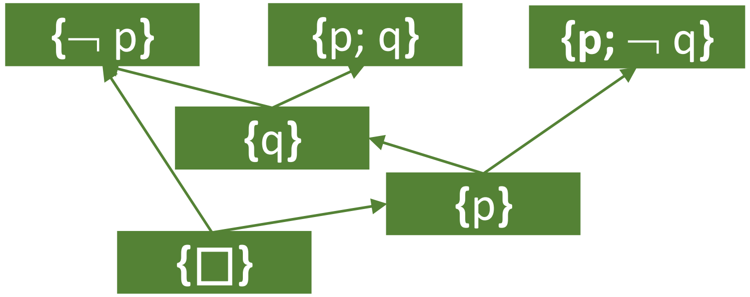
En la página 27 del documento 03-SAT-B que tienen en RECURSOS pueden ver la notación FITTING que podemos utilizar para expresar el desarrollo del algoritmo de Resolución.

- Veamos un ejemplo que ya vimos en DPLL, resuelto por Resolución con notación FITTING:

  $\Omega = \{\{\neg p\}; \{p; q\}; \{p; \neg q\}\}$

1.	$\{\neg p\}$	Cláusula del conjunto inicial	NIVEL 0
2.	$\{p; q\}$	Cláusula del conjunto inicial	NIVEL 0
3.	$\{p; \neg q\}$	Cláusula del conjunto inicial	NIVEL 0
4.	$\{q\}$	$\{\{\neg p\}; \{p; q\}\} \vDash R_{\not p}(\{\neg p\}; \{p; q\}) = \{q\}$	NIVEL 1
5.	$\{p\}$	$\{\{p; \neg q\}; \{q\}\} \vDash R_{\not q}(\{p; \neg q\}; \{q\}) = \{p\}$	NIVEL 2
6.	$\{\square\}$	$\{\{p\}; \{\neg p\}\} \vDash R_{\not p}(\{p\}; \{\neg p\}) = \{\square\}$	NIVEL 3

GRAFO DE RESOLUCIÓN COMPLETO

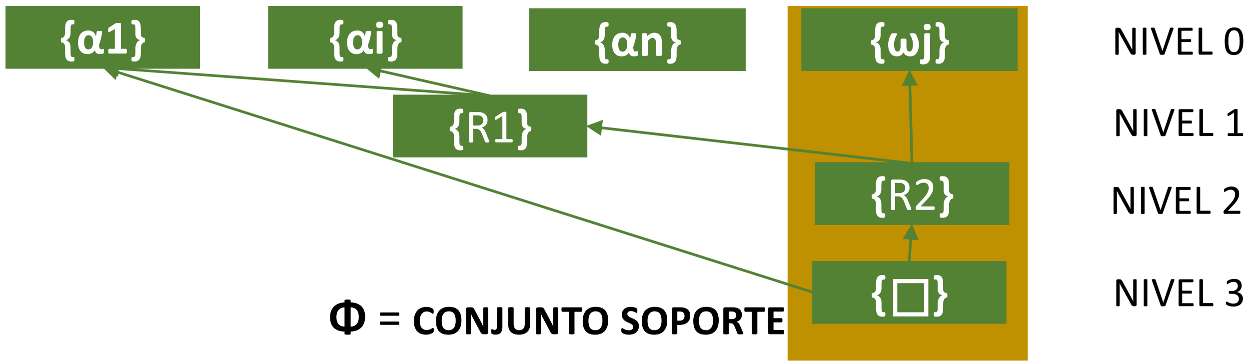


- Como vemos, en el último paso de resolución se ha obtenido una resolvente que es la cláusula vacía □. Con ello, podemos garantizar que el conjunto clausal es **INSATISFACIBLE**, cuestión que ya sabíamos por DPLL.
- Podemos apreciar que los distintos pasos de resolución conducen a la obtención de una pareja de cláusulas unitarias con literales opuestos. En el caso que tenemos, entre las cláusulas iniciales ya existe una cláusula unitaria **{**$\neg$ p**}**, y en el paso 5 hemos obtenido una resolvente **{**p**}.** El conjunto de las dos cláusulas citadas es Insatisfacible y obtiene una nueva resolvente que es la cláusula vacía.
- Los niveles van marcando la profundidad de la Resolución. Cada resolvente representa un Grafo de Resolución desde ella a sus dos cláusulas padres. El Grafo de Resolución que parte de la cláusula vacía se llama Grafo de Refutación. El conjunto de todos los grafos se llama Grafo de Resolución Completo. Si en este Grafo de Resolución Completo se incluye un Grafo de Refutación, el conjunto clausal de origen (el que representa las cláusulas del NIVEL 0) será INSATISFACIBLE. En cualquier otro caso, será SATISFACIBLE.

El Método de Resolución (IV)

En la páginas 29-32 del documento 03-SAT-B que tienen en RECURSOS pueden ver las diferentes aproximaciones o estrategias de resolución. Entre ellas, vamos a destacar: La llamada del CONJUNTO SOPORTE. Es una técnica SAT completa.

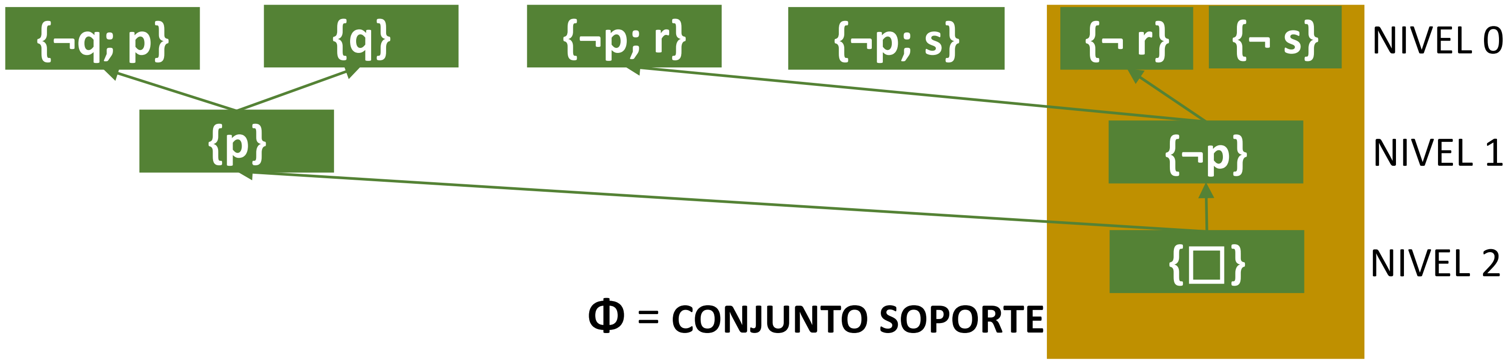
- La estrategia del CONJUNTO SOPORTE se utiliza cuando requerimos demostrar la Insatisfacibilidad de un conjunto Clausal que proviene de una oración formulada para resolver el problema de Validez de una Consecuencia Lógica mediante el planteamiento de Refutación. Recordemos, este planteamiento se elabora cuando queremos demostrar que una oración formada por la conjunción de las premisas con la negación de la conclusión es una oración INSATISFACIBLE: su conjunto clausal será, por tanto, Insatisfacible.
- Es decir, cuando al querer demostrar **$\{\alpha 1, \alpha 2, \alpha 3, \ldots, \alpha n\} \vDash \beta$**, el planteamiento que hacemos es demostrar que la oración **$(\alpha 1 \wedge \alpha 2 \wedge \alpha 3 \wedge \ldots \wedge \alpha n \wedge \neg \beta)$** es una contradicción.
- Supongamos que todas las oraciones premisas **αi** están en FNC. Y que la oración **$\neg \beta$** la transformamos en FNC. La oración que queremos demostrar que es contradicción, se puede expresar como un conjunto clausal. Dicho conjunto deberá ser INSATISFACIBLE, Y PARA ELLO DEBE DERIVAR LA CLÁUSULA VACÍA.
- Sea **$\Psi = \{\omega j\}$** el conjunto de cláusulas que conforman la FNC de la oración **$\neg \beta$** . Se llama conjunto soporte **Φ** al formado por dichas cláusulas y todas las resolventes de nivel **n** que provengan de resoluciones de al menos una cláusula del conjunto **Ψ** o de una cláusula de nivel **n-1** o inferiores de dicho conjunto soporte. Es decir, todas las cláusulas del Conjunto Soporte tienen al menos una cláusula padre en dicho conjunto.



El Método de Resolución (y V)

Tomemos el siguiente ejemplo: Queremos comprobar si la siguiente relación de consecuencia lógica es válida

- **{(¬q ∨ p), (q), (¬p ∨ r), (¬p ∨ s)} ⊨ (r ∨ s)**
- Tenemos cuatro premisas que son cláusulas cada una de ellas, así como la conclusión. El planteamiento de Refutación nos dice que la oración **((¬q ∨ p) ∧ (q) ∧ (¬p ∨ r) ∧ (¬p ∨ s) ∧ ¬ (r ∨ s))** debe ser una CONTRADICCIÓN u oración INSATISFACIBLE.
- La oración no es una FNC. Para serlo debemos aplicar la transformación por equivalencia a la última parte de la conjunción, precisamente la que nos presenta la negación de la conclusión.
- Si lo hacemos: **((¬q ∨ p) ∧ (q) ∧ (¬p ∨ r) ∧ (¬p ∨ s) ∧ (¬ r) ∧ (¬ s)).** Ahora sí que tenemos una FNC con seis cláusulas, las dos últimas son las que provienen de la negación de la conclusion, que conformarán el inicio (NIVEL 0) del Conjunto Soporte**.** Podemos escribir su forma clausal **{{¬q; p}; {q}; {¬p; r}; {¬p; s}; {¬ r}; {¬ s}}.**
- Como vemos, tenemos el Grafo de Resolución Completo, con dos niveles de profundidad en resolución en el Conjunto Soporte y obtenemos la cláusula vacía (un Grafo de Refutación). Por lo tanto, el conjunto clausal de partida **es INSATISFACIBLE**. De esta manera, la oración original es una CONTRADICCIÓN, y al llevar a cabo un planteamiento por Refutación, **la Consecuencia Lógica inicial es VÁLIDA.**



Algoritmos para expresiones no clausales

En las páginas 33-44 del documento 03-SAT-B que pueden encontrar en RECURSOS tienen explicadas estas técnicas SAT, así como su procedimiento de aplicación.

- Las tres técnicas son: **Tablas de Verdad, Árboles Semánticos y Tableaux Semánticos**.
- Las Tablas de verdad ya las hemos visto en el tema anterior con suficiente extensión y ejemplos.
- En cuanto a los Árboles Semánticos y los Tableaux Semánticos, el desarrollo de sus contenidos en dichas páginas es suficientemente explicativo y didáctico. No obstante, es conveniente resaltar algunas cuestiones relevantes de dichas técnicas:
  - Con los Árboles Semánticos podemos encontrar si una oración es una TAUTOLOGÍA, si todos los nodos hoja son ÉXITO. Si una oración es CONTRADICCIÓN, si todos los nodos hoja son FALLO. Y podemos saber todas las interpretaciones que hacen SATISFACIBLE a una oración, y todas las interpretaciones que hacen FALSEABLE a una oración dada.
  - Con los Tableaux Semánticos podemos encontrar si una oración es CONTRADICCIÓN, si una vez completado el mismo todos los nodos hoja quedan cerrados, es decir, encierran una contradicción. Si un nodo hoja queda abierto, las proposiciones atómicas o sus negadas que lo compongan representan la interpretación que hace a la oración SATISFACIBLE.
  - Con la técnica de los Tableaux Semánticos no podemos saber directamente si una oración es una TAUTOLOGÍA. Para comprobarlo, debemos negar la oración y comprobar que dicha oración negada es una CONTRADICCIÓN, mediante lo dicho anteriormente.
  - Los Tableaux Semánticos no determinan todas las posibles interpretaciones de una oración dada. Los Árboles Semánticos si determinan todas las posibles interpretaciones de la misma.
  - La toma de decisión de la Satisfacibilidad, tanto en Árboles Semánticos, como en Tableaux Semánticos, se toma al nivel de los nodos hoja. Tenemos un nodo hoja en un Árbol Semántico cuando la expresión del nodo es la constante VERDAD o la constante FALSO. En los Tableaux Semánticos, los nodos hoja son los formados únicamente por literales.

Facultad de Informática

Árboles Semánticos

Veamos el desarrollo de la aplicación de la técnica de Árboles Semánticos.

- Nos preguntamos por la Satisfacibilidad de la oración: $\mathbf{(p \rightarrow (q \wedge \neg r) \rightarrow (p \rightarrow q))}$

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

p

$(V \rightarrow (q \wedge \neg r) \rightarrow (V \rightarrow q))$

q

$v(V \wedge \neg r) = v(\neg r)$

$v(V \rightarrow \neg r \rightarrow V) = V$

Consecuente verdadero

$(V \rightarrow (V \wedge \neg r) \rightarrow (V \rightarrow V))$

$\equiv (V \rightarrow \neg r \rightarrow V)$

$\equiv (V)$

$\neg q$

$(V \rightarrow (F \wedge \neg r) \rightarrow (V \rightarrow F))$

$v(F \wedge \neg r) = F$

$v(V \rightarrow F) = F$

$\equiv (V \rightarrow F \rightarrow F) \equiv (F \rightarrow F)$

$\equiv (V)$

$\neg p$

$(F \rightarrow (q \wedge \neg r) \rightarrow (F \rightarrow q))$

$v(F \rightarrow (q \wedge \neg r)) = V$

$v(F \rightarrow q) = V$

$\equiv (V \rightarrow V) \equiv (V)$

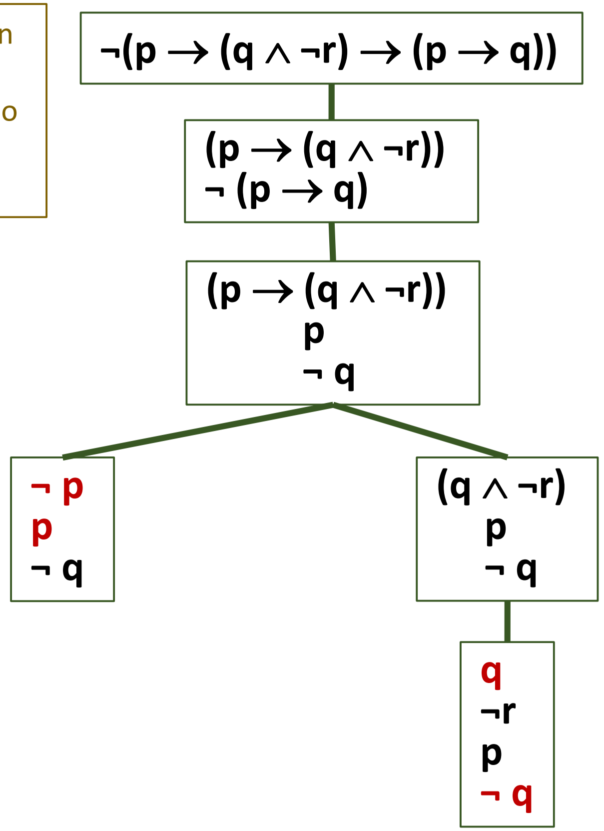
- Los tres nodos hoja del Árbol Semántico son equivalentes lógicamente a la constante VERDAD, por tanto, son todos los nodos hoja ÉXITO, **POR LO QUE LA ORACIÓN ES UNA TAUTOLOGÍA.**
- **Y podemos decir que hemos obtenido todas las interpretaciones!!!!!!** Todas las interpretaciones que contienen la asignación v(¬p) = V, es decir, v(p) = F, con independencia de las asignaciones a las proposiciones atómicas (q) y (r), hacen a la oración satisfacible. Y son cuatro dichas interpretaciones. También, independientemente de la asignación de (r), las interpretaciones que asignan v(p) = V y v(q) = V, que son dos interpretaciones, o que asignan v(p) = V y v(q) = F, que son otras dos interpretaciones, hacen a la oración satisfacible.

Tableaux Semánticos (I)

Tengan presente la tabla de la transformación de fórmulas mediante las llamadas α-fórmulas y β-fórmulas que aparece en la página 42 del documento 03-SAT-B que tienen en RECURSOS

- Nos preguntamos por la Satisfacibilidad de la oración: $\boldsymbol{\phi = \neg(p \rightarrow (q \wedge \neg r) \rightarrow (p \rightarrow q))}$

La expresión **ϕ** es la negación de una implicación que constituye una **α-fórmula** equivalente a **$(p \rightarrow (q \wedge \neg r)) \wedge \neg (p \rightarrow q)$**. Se construye un nodo nuevo desdoblando en serie las dos partes conjuntadas



La expresión **$\neg (p \rightarrow q)$** es la negación de una implicación que constituye una **α-fórmula** equivalente a **$(p \wedge \neg q)$**. Se construye un nuevo nodo desdoblando en serie las dos partes conjuntadas

La expresión **$(p \rightarrow (q \wedge \neg r))$** es una implicación que constituye una **β -fórmula** equivalente a **$(\neg p \vee (q \wedge \neg r))$**. Se abren dos ramas con nuevos nodos, colocando en cada uno de ellos una de las partes disyuntadas

La expresión **$(q \wedge \neg r)$** es una conjunción que constituye una **α -fórmula** por lo que se construye un nuevo nodo desdoblando en serie las dos partes conjuntadas

- **Los dos nodos hoja del Tableaux Semántico Completado quedan CERRADOS al contener cada uno de ellos un literal y su negado (en rojo). Por tanto, la oración es INSATISFACIBLE. Y no hemos obtenido interpretaciones!!!!!!**

Tableaux Semánticos (II)

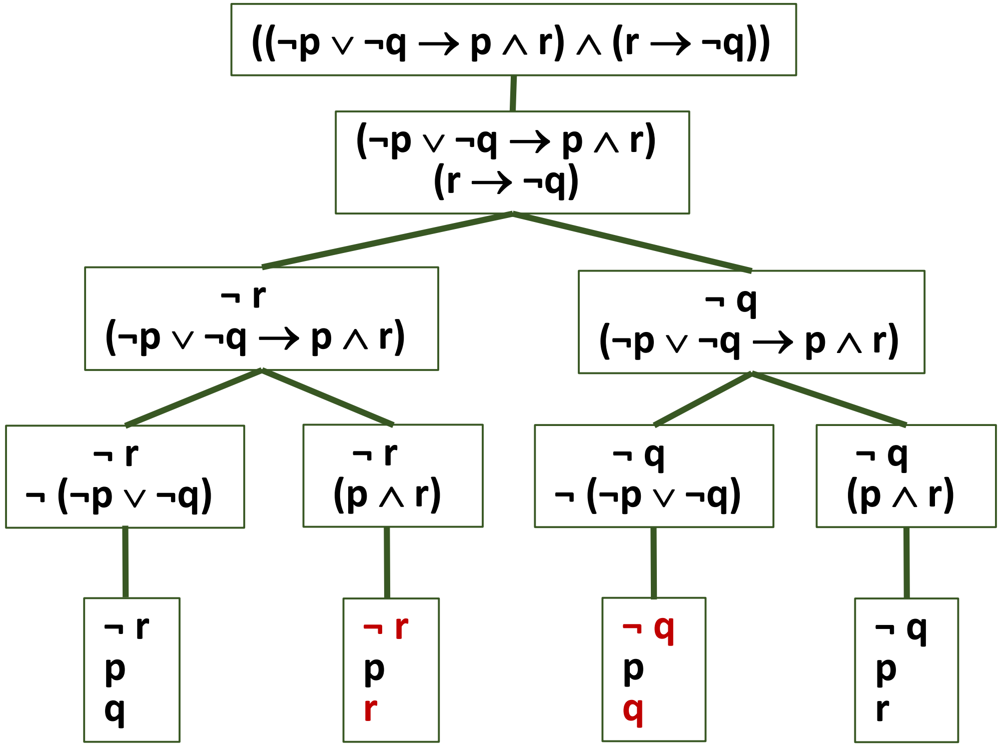
Tengan presente la tabla de la transformación de fórmulas mediante las llamadas α-fórmulas y β-fórmulas que aparece en la página 42 del documento 03-SAT-B que tienen en RECURSOS

- Nos preguntamos por la Satisfacibilidad de la oración: $\boldsymbol{\phi = ((\neg p \vee \neg q \to p \wedge r) \wedge (r \to \neg q))}$

La expresión **ϕ** es una conjunción que constituye una **α-fórmula.** Se construye un nodo nuevo desdoblando en serie las dos partes conjuntadas

La expresión **$(\neg p \vee \neg q \to p \wedge r)$** es una implicación que constituye una **β -fórmula** equivalente a **$(\neg (\neg p \vee \neg q) \vee (p \wedge r))$.** Se abren dos ramas con nuevos nodos, colocando en cada uno de ellos una de las partes disyuntadas

La expresión **$\neg (\neg p \vee \neg q)$** es la negación de una disyunción que constituye una **α -fórmula** equivalente a **$(p \wedge q)$** por lo que se construye un nuevo nodo desdoblando en serie las dos partes conjuntadas



La expresión **$(r \to \neg q)$** es una implicación que constituye una **β -fórmula** equivalente a **$(\neg r \vee \neg q)$.** Se abren dos ramas con nuevos nodos, colocando en cada uno de ellos una de las partes disyuntadas

La expresión **$(p \wedge r)$** es una conjunción que constituye una **α -fórmula** por lo que se construye un nuevo nodo desdoblando en serie las dos partes conjuntadas

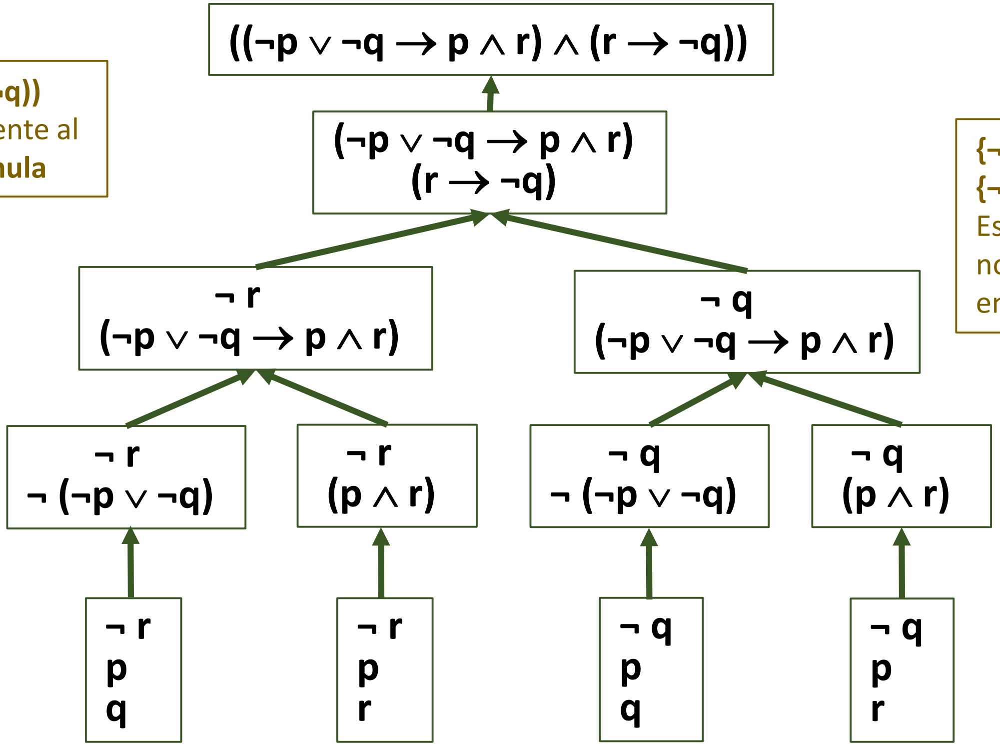
- **De los cuatro nodos hoja del Tableaux Semántico Completado, quedan CERRADOS los dos centrales al contener cada uno de ellos un literal y su negado (en rojo). Los otros dos nodos hoja quedan abiertos. Por tanto, la oración es SATISFACIBLE. En este caso, los dos nodos hoja abiertos presentan las dos interpretaciones que hacen satisfacible la oración!!!!!! Pueden ver las tablas de verdad de las páginas 14 y 15 de estos apuntes en las que resolvíamos la evaluación de esta oración.**

Tableaux Semánticos (y III)

Si leemos el Tableaux Semántico desde los nodos hoja hasta el nodo raíz, podemos darnos cuenta de que lo que realmente estamos aplicando son las consecuencias lógicas: **$\{\alpha 1, \alpha 2\} \vDash \alpha 1 \wedge \alpha 2$** para las transformaciones de las **α-fórmula;** y las consecuencias lógicas: **$\{\alpha 1\} \vDash \alpha 1 \vee \alpha 2$ y $\{\alpha 2\} \vDash \alpha 1 \vee \alpha 2$** para las transformaciones de las **β –fórmula.** En el ejemplo que hemos resuelto anteriormente:

$\{(\neg p \vee \neg q \rightarrow p \wedge r), (r \rightarrow \neg q)\} \vDash ((\neg p \vee \neg q \rightarrow p \wedge r) \wedge (r \rightarrow \neg q))$
Esta es la consecuencia lógica aplicable entre el nodo siguiente al raíz y su padre, el propio nodo raíz, en aplicación de **α-fórmula**

$\{\neg r, \neg(\neg p \wedge \vee \neg q)\} \vDash \neg r \wedge (\neg(\neg p \wedge \vee \neg q) \vee (p \wedge r))$
$\{\neg r, (p \wedge r)\} \vDash \neg r \wedge (\neg(\neg p \wedge \vee \neg q) \vee (p \wedge r))$
$\{\neg q, \neg(\neg p \wedge \vee \neg q)\} \vDash \neg q \wedge (\neg(\neg p \wedge \vee \neg q) \vee (p \wedge r))$
$\{\neg q, (p \wedge r)\} \vDash \neg q \wedge (\neg(\neg p \wedge \vee \neg q) \vee (p \wedge r))$
Estas son las consecuencias lógicas aplicables entre los nodos anteriores a los hoja y sus correspondientes nodos padres en aplicación de **β -fórmula**



$\{\neg r, (\neg p \vee \neg q \rightarrow p \wedge r)\} \vDash (\neg r \vee \neg q) \wedge (\neg p \vee \neg q \rightarrow p \wedge r)$
$\{\neg q, (\neg p \vee \neg q \rightarrow p \wedge r)\} \vDash (\neg r \vee \neg q) \wedge (\neg p \vee \neg q \rightarrow p \wedge r)$
Estas son las consecuencias lógicas aplicables entre los nodos anteriores y sus correspondientes nodos padres en aplicación de **β -fórmula**

$\{\neg q, p, r\} \vDash \neg q \wedge (p \wedge r)$
$\{\neg q, p, q\} \vDash \neg q \wedge \neg(\neg p \wedge \vee \neg q)$
$\{\neg r, p, r\} \vDash \neg r \wedge (p \wedge r)$
$\{\neg r, p, q\} \vDash \neg r \wedge \neg(\neg p \wedge \vee \neg q)$
Estas son las consecuencias lógicas aplicables entre los nodos hoja y sus correspondientes nodos padres en aplicación de **α-fórmula**

Técnicas SAT y Razonamientos (I)

- Las Técnicas SAT son Métodos Semánticos:
  - Las técnicas SAT que hemos estudiado se engloban en lo que llamamos MÉTODOS SEMÁNTICOS. Lo son, pues se establecen a partir de la aplicación de las dos relaciones semánticas entre oraciones: la relación de EQUIVALENCIA LÓGICA y la relación de CONSECUENCIA LÓGICA.
  - En la base de estos métodos están los dos principios epistemológicos de la Lógica Proposicional y de la Lógica de Predicados: EL PRINCIPIO DEL TERCIO EXCLUIDO Y EL PRINCIPIO DE NO CONTRADICCIÓN.
  - Asimismo, manejamos los conceptos de INTERPRETACIÓN Y DE EVALUACIÓN de oraciones lógicas.
- Cuando nos planteamos demostrar si un Razonamiento Lógico es VÁLIDO, lo hacemos a través del concepto de CONSECUENCIA LÓGICA, teniendo en cuenta dos planteamiento: el Directo y el de Refutación. Vamos a resumir los aspectos claves y las Técnicas SAT óptimas para resolver cada uno de estos planteamientos.
- Si queremos comprobar si la oración β es una conclusión del conjunto de premisas **$\{\alpha1, \alpha2, \alpha3, \ldots, \alpha n\}$**, es decir, si se establece entre ellas un Razonamiento Válido, lo hacemos demostrando la siguiente Consecuencia Lógica **$\{\alpha1, \alpha2, \alpha3, \ldots, \alpha n\} \vDash \beta$**. Y para hacerlo, por el planteamiento directo, hemos de demostrar que la oración **$(\alpha1 \land \alpha2 \land \alpha3 \land \ldots. \land \alpha n \rightarrow \beta)$** es una **TAUTOLOGÍA**. Por otra lado, mediante el planteamiento por Refutación, demostraremos si la oración **$(\alpha1 \land \alpha2 \land \alpha3 \land \ldots. \land \alpha n \land \neg \beta)$** es una **CONTRADICCIÓN**.
- En el planteamiento directo, **$(\alpha1 \land \alpha2 \land \alpha3 \land \ldots. \land \alpha n \rightarrow \beta)$** será una Tautología, si aplicando:
  - El algoritmo DPLL, una vez transformada la oración citada a su FNC (o su Forma Clausal) aplicando SÓLO la regla de la Tautología, con todas las cláusulas triviales, nos queda el CONJUNTO VACÍO DE CLÁUSULAS.
  - Tablas de Verdad, se obtiene que la evaluación de la oración para todas sus Interpretaciones es VERDAD.
  - El Árbol Semántico a la oración, se obtienen todos sus nodos hoja como nodos ÉXITO, es decir, equivalentes a la constante VERDAD.
  - NO son aplicables directamente la técnica del Algoritmo de Resolución, ni la técnica de los Tableaux Semánticos.

Técnicas SAT y Razonamientos (y II)

- En el planteamiento por Refutación, **($\alpha 1 \land \alpha 2 \land \alpha 3 \land .... \land \alpha n \land \neg \beta$)** será una CONTRADICCIÓN, si aplicando:
  - El algoritmo DPLL, una vez transformada la oración citada a su FNC (o su Forma Clausal) aplicando las Reglas de Propagación, se obtienen en todas las propagaciones la CLÁUSULA VACÍA. Recordemos que en el caso de aplicar la Regla de Ramificación debemos hacer Backtraking o vuelta atrás.
  - El Algoritmo de Resolución , una vez transformada la oración en su Forma Clausal, el Grafo de Resolución Completo contiene un Grafo de Refutación, es decir, obtenemos como resolvente la CLÁUSULA VACÍA.
  - Tablas de Verdad, se obtiene que la evaluación de la oración para todas sus Interpretaciones es FALSO.
  - El Árbol Semántico a la oración, se obtienen todos sus nodos hoja como nodos FALLO, es decir, equivalentes a la constante FALSO.
  - Tableaux Semántico, se obtiene el Tableaux Semántico Completado con todos sus nodos hoja CERRADOS, es decir, con literales opuestos o contradictorios.
- Hemos visto que las Técnicas SAT mantienen la SATISFACIBILIDAD o la INSATISFACIBILIDAD.
- **L0 como Lógica Formal es totalmente DECIDIBLE**, es decir, siempre disponemos de una técnica o método para demostrar tanto la Satisfacibilidad como la Insatisfacibilidad de una determinada oración, si bien las Técnicas SAT estudiadas no son todas ellas COMPLETAS. Es decir, aún sabiendo que una oración es Satisfacible con algunas técnicas no estamos seguros de poderlo demostrar: es el caso de algunas de las estrategias del Algoritmo de Resolución. Y hemos visto que en el caso de las Tautologías no lo son directamente la técnica del Algoritmo de Resolución, ni la de Tableaux Semánticos.
- Por último, si queremos saber si un conjunto de oraciones (como puede ser el conjunto de premisas de un razonamiento) es SATISFACIBLE o INSATISFACIBLE: con DPLL o Resolución basta con transformar todas las oraciones a su Forma Clausal, generar el conjunto de cláusulas formado por todas ellas, y aplicar la técnica como sabemos. Con Tablas de Verdad, Árboles Semánticos o Tableaux Semánticos, basta con conjuntar todas las oraciones y a la oración resultante aplicarle la técnica elegida.