Fundamentos Lógicos de la Informática

Problema SAT

Grupo docente FLI

Dpto. Ingeniería de al Información y las Comunicaciones
Facultad de Informática
Universidad de Murcia

Desarrollo

Problema SAT

¿Cómo sé si un conjunto de oraciones es satisfacible?

- Es un problema intratable del tipo NP-completo.
  No hay un algoritmo mejor que realizar una búsqueda exahustiva de todas las posibilidades en la búsqueda de una solución al problema.
- Todo problema NP-completo se pueden reducir al problema SAT.
- Si se resuelve SAT como problema P, se resuelven todos.
- Algunos tipos de problemas SAT
  - SAT(k): Dada una FNC en L0 con a lo sumo k literales por cláusula ¿tiene una asignación de valores que haga a la expresión verdadera?
    Ejemplo: ¿existen valores para x_1, x_2, x_3, x_4 tales que sea cierta la expresión: $(x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$?
  - UNSAT: ¿cuál es la combinación de valores de las variables que hacen a la expresion falsa?
  - SAT*(k): Dada una FNC en L0 con exactamente k literales por cláusula, determinar si es satisfacible.
  - MAX-SAT: Número máximo de cláusulas que pueden satisfacerse para una asignación.

Algoritmos de resolución del problema SAT

En L0 el problema es decidible.

- Algoritmos completos[1].
  Los que proporcionando una asignación, ésta es un modelo.
  - Tablas de verdad.
  - Algoritmo DP (Davis, Martin; Putnam, Hillary. 1960)
  - Algoritmo DPLL (Davis, Putnam, Logemann y Loveland, 1962)
  - Árboles semánticos.
  - Tableros 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.

Desarrollo

Profundizamos en ...

Con más detalle ...

Cláusulas

Definición (Cláusula)

- *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 $\square$.

Definición (Cláusula de Horn)

*Una **cláusula de Horn** es una cláusula con a lo sumo un literal positivo.*

La expresión general es de la forma $\alpha = P \vee \neg Q_1 \vee \neg Q_2 \vee \cdots \vee \neg Q_n$

- Si $n = 0$, α se llama **hecho**.
- Si no hay literal positivo P, α se llama **objetivo**.
- Si $n > 0$ y existen el literal positivo P, α se llama **regla**.

Se utiliza para razonamiento “hacia atrás”.

Forma Normal Conjuntiva

Definición

- *Una fórmula, ξ, está en* ***forma normal conjuntiva*** *si responde a la expresión $\wedge_i(\vee_j P_{i_j})$ donde P_{i_j} son literales (de L0).*
- *Un* ***conjunto clausal*** *o clausulado es un conjunto de cláusulas expresadas como conjuntos de literales.*
- *Una fórmula está en* ***forma clausal*** *si se expresa como un conjunto clausal.*

Ejemplo:La forma clausal de $(p \vee q) \wedge \neg p$ es $\{\{p, q\}, \{\neg p\}\}$
Ejemplo:El conjunto clausal $\{\{\neg p, q\}, \{r\}\}$ es la forma clausal de las fórmulas $(p \to q) \wedge r$ y $(\neg q \to \neg p) \wedge r$.
Ejemplo:$\{\neg p \vee q, r\}$=Conjunto de cláusulas. No es una forma clausal, ni está en FNC.

Conceptualmente son equivalentes
(cambia la forma de decirlo y su representación)
F.N. conjuntiva (o clausal). Conjunción de disyunciones de literales. Conjunción de cláusulas. Conjunto clausulado. Forma clausal.

El “loco” conjunto vacío

- La cláusula vacía, $\square$, la disyunción de ningún literal es
  una oración insatisfacible.
- El conjunto vacío de oraciones, $\emptyset$, (ya sea o no de cláusulas) es:
  un conjunto satisfacible.

¿Es satisfacible $\{\square\}$?

Con más detalle ...

Construcción de la FNC - I

1. Eliminar todas los operadores binarios salvo $\wedge$ y $\vee$.
   Aplicar las reglas de eliminación de $\leftrightarrow$ y $\rightarrow$ de *L0*.

$$\alpha \leftrightarrow \beta \equiv [\neg\alpha \vee \beta] \wedge [\alpha \vee \neg\beta]$$
$$\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$$

2. Aplicar D'Morgan: $\begin{cases} \neg(\alpha \vee \beta) & \equiv & \neg\alpha \wedge \neg\beta \\ \neg(\alpha \wedge \beta) & \equiv & \neg\alpha \vee \neg\beta \end{cases}$
   Reducir el alcance de las negaciones. Las negaciones solo deben afectar a fórmulas atómicas.
3. Eliminar las negaciones múltiples aplicando idempotencia: $\neg\neg\alpha \equiv \alpha$
4. Aplicar distributividad. $\begin{cases} (\alpha \wedge \beta) \vee \gamma & \equiv & (\alpha \vee \gamma) \wedge (\beta \vee \gamma) \\ \alpha \vee (\beta \wedge \gamma) & \equiv & (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \end{cases}$

Construcción de la FNC - II

5. Reducir la cantidad de paréntesis: $\begin{cases} (\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge \beta \wedge \gamma \\ (\alpha \vee \beta) \vee \gamma \equiv \alpha \vee \beta \vee \gamma \end{cases}$

6. Eliminar información redundante.

   i) Eliminar literales opuestos

   $$\alpha \vee \neg\alpha \equiv \mathit{VERDAD}$$

   $$\alpha \wedge \neg\alpha \equiv \mathit{FALSO}$$

   ii) Eliminar constantes

   $$\mathit{FALSO} \vee \alpha \equiv \alpha$$

   $$\mathit{VERDAD} \wedge \alpha \equiv \alpha$$

   iii) Eliminar literales iguales

   $$\alpha \vee \alpha \equiv \alpha$$

   $$\alpha \wedge \alpha \equiv \alpha$$

   iv) Quedarnos con expresiones subsumidas (o literales incluidos)

   $$(\alpha \vee \beta) \wedge \alpha \equiv \alpha$$

Profundizamos en ...

Propagación de literales - I

Una aproximación intuitiva al algoritmo DPLL.

Dada $\xi = (p \vee q \vee \neg t) \wedge (\neg q \vee \neg r \vee s) \wedge (q \vee t)$, y considerando el átomo q:

- Si se le asigna el valor de verdad V, son equivalentes:

$$\begin{aligned}\xi_{v(q)=V} \equiv \xi(q) &= \overbrace{\cancel{(p \vee q \vee \neg t)}}^{\text{Cláusula Cancelada.}} \wedge \overbrace{(\cancel{\neg q} \vee \neg r \vee s)}^{\text{Ocurrencia eliminada.}} \wedge \overbrace{\cancel{(q \vee t)}}^{\text{C. cancelada}} \\ &= \neg r \vee s\end{aligned}$$

- Si se le asigna el valor de verdad F, son equivalentes:

$$\begin{aligned}\xi_{v(q)=F} \equiv \xi(\neg q) &= \overbrace{(p \vee \cancel{q} \vee \neg t)}^{\text{Ocurrencia Eliminada.}} \wedge \overbrace{\cancel{(\neg q \vee \neg r \vee s)}}^{\text{Cláusula Cancelada.}} \wedge \overbrace{(\cancel{q} \vee t)}^{\text{O. Elim.}} \\ &= (p \vee \neg t) \wedge (t)\end{aligned}$$

Propagación de literales - II

Una aproximación intuitiva al algoritmo DPLL.

Definición (Propagación del literal ℓ)

Dada una expresión ξ, en FNC, y un literal ℓ de una cláusula de ξ, la propagación del literal ℓ en ξ es una nueva expresión $\xi(\ell)$ que se obtiene reescribiendo ξ pero: (a) sin escribir las cláusulas canceladas, y (b) eliminando las ocurrencias "negadas" del literal ℓ en aquellas cláusulas donde aparezca.

Composición de propagación de literales. Dada una expresión ξ, en FNC,

- $\xi(p)$ denota a la expresión resultante de la propagación del literal p en ξ
- $\xi(p, q)$ indica a la expresión resultante de realizar la propagación del literal p en ξ para a continuación realizar la propagación de q en $\xi(p)$. Es decir, es la expresión: $(\xi(p))\,(q)$ ....
- $\xi(p, q, r, \ldots, s)$ denota a: $(\ldots(((\xi(p))\,(q))\,(r))\ldots)\,(s)$

Algoritmo DPLL - I

El método de Davis-Logemann y Loveland

La idea: Para estudiar la satisfacibilidad de ξ en FNC realizar transformaciones sucesivas que sean equivalentes a la fórmula original.

El método: Considera 5 reglas:

Regla de propagación unitaria
: Si existe una cláusula unitaria formada por un literal ℓ, propagar el literal ℓ.

Regla del literal puro
: Si existe un literal puro ℓ (para el que no existe su complemento en otras cláusulas), propagar el literal ℓ.

Regla de la tautología
: Eliminar las cláusulas que contengan un par de literales complementarios.

Regla de la inclusión
: (o de eliminación de clásulas subsubmidas) Si en Ω existen conjuntos clausales C_1 y C_2 tales que $C_1 \subset C_2$, eliminar C_2 de Ω.

Algoritmo DPLL - II

El método de Davis-Logemann y Loveland

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 $\Omega \cup \{\ell\}$ (propagar ℓ)
- o bien la satisfacibilidad de $\Omega \cup \{\neg\ell\}$ (propagar $\neg\ell$)

Algoritmo DPLL - III

El método de Davis-Logemann y Loveland

1. Considerar Ω el conjunto de cláusulas.
2. Definir `valor=`$DPLL(\Omega)$.
3. Si `(valor == true)`, Ω es satisfacible.
4. Si `(valor == false)`, Ω es insatisfacible.

$DPLL(\Omega)$ (Algoritmo con backtracking)

1. Aplicar regla de la tautología. Ir al paso 2.
2. Aplica la R. de propagación unitaria. Si se aplicó hacer los siguientes pasos, y si no ir al siguiente paso.
   - asignar `valor:=`$Satisf(\Omega)$
   - si `valor == indeterminado` ir al paso 2, si no `return valor`.
3. Aplica la R. del literal puro. Si se aplicó hacer los siguientes pasos, y si no ir al siguiente paso.
   - asignar `valor:=`$Satisf(\Omega)$

Algoritmo DPLL - IV

El método de Davis-Logemann y Loveland

- si `valor == indeterminado` ir al paso 3, si no `return valor`.

4. Mientras que se pueda, aplicar la regla de cláusulas subsumidas.
5. Si Ω' es el conjunto resultante de los pasos anteriores, aplicar la regla de ramificación y asignar `valor:=`$DPLL(\Omega' \cup \{\ell\})$
   Si (`valor` $==$ `true`) entonces `return true`
   Si (`valor` $==$ `false`) entonces `return` $(DPLL(\Omega' \cup \{\neg\ell\}))$. Observa que si se vuelve a la primera ramificación, que solo es posible cuando se hayan estudiado todas las ramificaciones "hijas", y ya se han "consultado" las dos ramas, se retorna el valor `false`.

$Satisf(\Omega)$

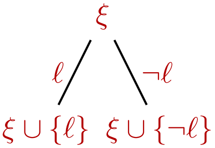
1. Si $\Omega = \emptyset$, `return true`
2. Si $\Box \in \Omega$, `return false`
3. `return indeterminado`

Uso de grafos para DPLL

Dada una expresión ξ en FNC

- Cada nodo tiene asociada una expresión clausal que se asume que es cierta, salvo que aparezca la cláusula vacía.
- Cada nodo no hoja tiene asociado una letra ℓ de la fórmula ξ y se le asocia una o dos ramas (en función de la regla a aplicar) para representar la propagación del literal de ℓ que etiqueta la rama.

Regla de ramificación



Todas las demás reglas

Profundizamos en ...

Resolución y cláusulas

Definición

Se define la **resolvente** de las clausulas $\psi = \{\ell_i\}$ y $\phi = \{\ell'_j\}$ **respecto del literal** ℓ a la cláusula:

$$R_\ell(\psi, \phi)$$

dada por las siguientes condiciones y procedimiento.

- En ψ no existen i, i' tales que $\neg\ell_i = \ell_{i'}$ (literales opuestos) ni $\ell_i = \ell_{i'}$ (literales iguales).
- En ϕ no existen j, j' tales que $\neg\ell'_j = \ell'_{j'}$ (literales opuestos) ni $\ell'_j = \ell'_{j'}$ (literales iguales).
- Existen ℓ_i y ℓ'_j que son literales contrarios. P.e. $\neg\ell_i = \ell'_j = \ell$.
- Construir $R_\ell(\psi, \phi) = (\psi - \{\ell_i\}) \cup (\phi - \{\ell'_j\})$.

- ψ y ϕ se llaman las **cláusulas padres** de $R_\ell(\psi, \phi)$.
- **Muy importante:** $\{\psi, \phi\}$ es satisfacible sii $R_\ell(\psi, \phi)$ es satisfacible.

Algoritmo de Robinson I

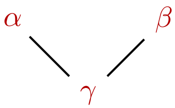
Método de resolución

1. Definir Ω_0 como el conjunto clausal de ξ.
2. **Elegir dos cláusulas** $C_1 = \ell \vee \alpha$ **y** $C_2 = \neg\ell \vee \beta$ que aún no hayan sido seleccionadas.
3. Aplicar la regla de resolución: $C = R_\ell(C_1, C_2) = \alpha \vee \beta$
4. Si $C = \square$, retornar que Ω_0 es `Insatisfacible`.
5. Si C no es la cláusula trivial, $\Omega_{i+1} = \Omega_i \cup \{C\}$; en otro caso $\Omega_{i+1} = \Omega_i$.
6. Si todas las cláusulas han sido "enfrentadas", retornar que Ω_0 es `Satisfacible`.

Se eliminan del proceso: cláusulas triviales, cláusulas subsumidas y cláusulas con literales puros.

Grafo de Resolución

- Se define el **grafo de resolución básico** asociado a $\{\alpha, \beta\} \vdash_{\mathsf{RR}} \gamma = R_\ell(\alpha, \beta)$ como el siguiente grafo:



- Un **grafo de resolución completo** asociado al conjunto $\mathcal{F}$ es la unión de todos los grafos básicos de su conjunto clausulado.
- Un **grafo de resolución** para el conjunto $\mathcal{F}$ es cualquier subconjunto del grafo de resolución completo de $\mathcal{F}$.
- Un **grafo de refutación** es un grafo de resolución en el que aparece la cláusula vacía $\square$.

IMPORTANTE

Los **padres**, α y β, de la resolvente γ, son los **hijos** en el grafo. Las cláusulas de $\mathcal{F}$ son **padres** y ascendiente desde el punto de las resolventes, pero son nodos **hijos** (son hojas) en el grafo.

Representación Fitting

1	$\vdots$	
2	C_t	Premisa
3	$\vdots$	
4	$\vdots$	
5	ξ	R_ℓ(i,j)
6	$\cancel{\ell, p, \ldots}$	Puro(ℓ)
7	$\cancel{\ell', \ldots}$	Subsumida(i)
8	$\cancel{\ell, \neg\ell, \ldots}$	Tautología
9	$\vdots$	

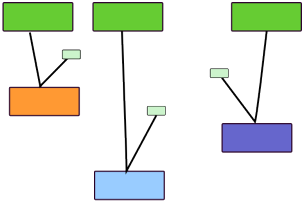
- C_t es una cláusula inicial.
- $\xi, \psi, \ldots$ son cláusulas derivadas.
- $R_\ell(i, j)$, aplicación de la regla de resolución sobre las cláusulas de las líneas i y j.
- *Puro*(ℓ), eliminación de la cláusula por ser ℓ un literal puro.
- *Subsumida*(j), eliminación de la cláusula por estar subsumida en la cláusula de la línea i.
- *Tautologa*, eliminación de la cláusula por ser trivial..

Con más detalle ...

Antes de seguir. Definiciones importantes - I

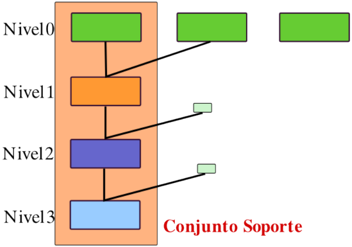
- Una cláusula C es **unitaria** sii está formada por un sólo literal.
  Ejemplo: $C = \{P\}, C = \{Q\}, C = \{\neg P\}, \ldots$

- Dado un conjunto clausal $\mathcal{F}$ y una resolvente ξ ($\mathcal{F} \vdash \xi$) se dice que:
  - Es una **resolvente de entrada** si uno de sus padres pertenece al conjunto (de entrada) original.



Antes de seguir. Definiciones importantes - II

- Es una **resolvente de nivel** i si un padre es de nivel $i-1$ y el otro de nivel $j \leq i-1$. A nivel 0 no hay resolventes y se entiende que hace referencia a las clausulas de $\mathcal{F}$.
- Dado el conjunto clausal $\mathcal{F}' = \{\mathcal{F} \cup \{\neg\phi\}\}_{FNC}$ se define el **conjunto soporte** como el formado por el teorema negado y todas las resolventes descendientes de él.
  $Sop = \{\xi | \xi \in \{\neg\phi\}_{FNC}$ o (si $\xi = R(C_i, C_j)$ entonces $C_i \in Sop)\}$



Estrategias de Resolución

- **Estrategias a ciegas**
  - Primero en anchura.
  - Primero en profundidad.
- **Estrategias de refinamiento.**
  - Preferencia de unidades
  - Entrada.
  - Filtrado de antepasados.
- **Basadas en Resolución semántica** (J.R. Slagle en 1967)
  - Conjunto soporte
  - Hiper resolución.
- **Resolución lineal.**
  - Resolución de resolución lineal.
  - Resolución SLD, (Kowalski)
- **Otras estrategias.**

Estrategias Básicas

- Estrategias a ciegas
  1. **Primero en anchura.**
     - Otros nombres: Estr. a lo ancho, estr. en amplitud.
     - Criterio: Obtener todas las resolvente nivel a nivel.
       Obtener en cada paso, i, todas las posibles resolventes a nivel i.
     - Es el procedimiento de búsqueda de consistencias.
     - Es completa, pues calcula el cierre clausulado.
  2. **Primero en profundiad.**
     - Criterio: Obtenida una resolvente de nivel i, intentar generar a partir de ésta una resolvente de nivel $i+1$. Se puede fijar una cota de profundidad para volver atrás y seguir por otro camino.
     - No es completa.
- Estrategias de Resolución semántica
  - **Estrategia de Conjunto soporte.**
    - Criterio: un padre siempre pertenece al conjunto soporte.
    - Es completa.

Desarrollo

Profundizamos en ...

Tablas de verdad

Dada una expresión ξ.

1. Construir una tabla de verdad por filas.
2. Si en un fila (para una interpretación) la oración es cierta, retornar que es `satisfacible`.
3. Si en todas las filas (para todas las interpretaciones) la oración es falsa, retornar que es `insatisfacible`.

Profundizamos en ...

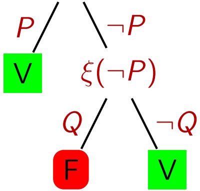
Árboles semánticos

Un árbol semántico asociado a una expresión ξ es una árbol donde:

- Cada nodo no hoja
  - se asocia a una letra, ℓ, de la fórmula ξ.
  - tiene dos hijos con ramas etiquetas.
  - Una rama representa la asignación del valor V a la letra asociada y la otra rama la asignación del valor F. Se etiquetan con ℓ y $\neg\ell$.
- Cada nodo hoja representa el valor de la expresión α para la asignación de verdad realizada en el camino desde la raíz hasta este nodo y se etiqueta con la evaluación.

P	Q	α
V	$*$	V
F	V	F
F	F	V

$$\xi = (P \to Q) \to (\neg P \to \neg Q)$$



Decisión con árboles semánticos

Definición (Nodo Fallo / Nodo Éxito)

Un nodo hoja de un árbol semántico de una expresión ξ se dice que es:

- ***Nodo fallo**, si la etiqueta asociada es F (la expresión es falsa).*
- ***Nodo éxito**, si la etiqueta asociada es V (la expresión es verdadera).*

Teorema

Si ξ es una expresión y Υ su árbol semántico asociado, entonces

- *ξ es satisfactible sii Υ tiene (al menos) un nodo éxito.*
- *ξ es no-válida sii Υ tiene (al menos) un nodo fallo.*
- *ξ es válida sii todos los nodos de Υ son nodo éxito.*
- *ξ es insatisfactible sii todos los nodos de Υ son nodo fallo.*

Profundizamos en ...

Tableau Semántico

La idea

Hastar ahora, dada ξ se busca $v(\xi)$ estableciendo un valor de verdad sobre un literal ℓ de ξ. Es decir, fijando (paso a paso) un interpretación hasta evaluar ξ.
Ahora, dada ξ se evalúa directamente $v(\xi) = V$ y se descompone en sus subfórmulas hasta encontrar las interpretaciones donde se cumple.

Ejemplo:$\xi = P \wedge (\neg Q \vee \neg P)$ cumplirá $v(\xi) = V$ sii

1. o bien $v(P) = V$ y $v(\neg P) = V$, pero esto no es una interpretación posible pues $\{P, \neg P\}$ es insatisfacible.
2. o bien $v(P) = V$ y $v(\neg Q) = V$, que sí es posible porque el conjunto $\{P, Q\}$ es satisfactible.

Tableaux Semántico

La idea en forma de árbol

- Utilizar un árbol que represente a la descomposición de la expresión lógica.
- La raíz será la expresión inicial.
- Para ir de la raíz a las hojas: Si un nodo tiene expresiones $\{\psi, \xi\}$, generar hijos con expresiones más sencillas $\{\psi', \xi\}$.
- Las hojas de los árboles contendrán a conjuntos de literales.
- Con solo observar las hojas se pueda deducir la satisfactibilidad de la expresión si no está ℓ y $\neg\ell$.

α y β fórmulas

Una fórmula es de tipo α (comportamiento conjuntivo) o es de tipo β (comportamiento disyuntivo) y siempre tiene asociada dos componentes de acuerdo a la siguiente tabla.

α fórmulas			β fórmulas		
α	α_1	α_2	β	β_1	β_2
$\neg\neg\xi$	ξ				
$\xi \wedge \varphi$	ξ	φ	$\neg(\xi \wedge Q)$	$\neg\xi$	$\neg\varphi$
$\neg(\xi \vee \varphi)$	$\neg\xi$	$\neg\varphi$	$\xi \vee Q$	ξ	φ
$\neg(\xi \rightarrow \varphi)$	ξ	$\neg\varphi$	$\xi \rightarrow Q$	$\neg\xi$	φ
$\xi \leftrightarrow \varphi$	$\xi \rightarrow \varphi$	$\varphi \rightarrow \xi$	$\neg(\xi \leftrightarrow \varphi)$	$\neg(\xi \rightarrow \varphi)$	$\neg(\varphi \rightarrow \xi)$

Teorema

- *Para toda α-fórmula* $\alpha \equiv \alpha_1 \wedge \alpha_2$
- *Para toda β-fórmula* $\beta \equiv \beta_1 \vee \beta_2$

Construcción de un tableau semántico

1. Construir el nodo raíz del árbol, etiquetarlo con el conjunto $\{\xi\}$ y marcarlo como no-resuelto.
2. Para cada nodo, n, no resuelto del árbol
   1. Considerar el conjunto de fórmulas $U(n)$ del nodo y marcarlo como resuelto.
   2. Si $U(n)$ está formado sólo por literales,
      a) si existe un literal y su negado en dicho conjunto, etiquetarlo como cerrado ✘.
      b) en otro caso, etiquetarlo como abierto ✔.
   3. Si $U(n)$ no está formado sólo por literales elegir una fórmula que no sea un literal,
      a) Si es una α-fórmula crear un nodo hijo, l, y etiquetarlo con el conjunto $U(l) = (U(n) - \{\alpha\}) \cup \{\alpha_1, \alpha_2\}$
      b) Si es una β-fórmula crear dos nodos hijo, l y l', y etiquetarlos con los conjuntos $U(l) = (U(n) - \{\beta\}) \cup \{\beta_1\}$ y $U(l') = (U(n) - \{\beta\}) \cup \{\beta_2\}$
3. Retornar el árbol (tableau semántico) Υ.

Tableaux semánticos y satisfactibilidad en $L0$

Definición (Árbol completado, abierto y cerrado)

- *Un tableau cuya construcción termina se llama (está)* ***tableau completado****.*
- *Un tableau completado es* ***cerrado*** *sii todas las hojas están marcadas como cerradas.*
- *Un tableau completado es* ***abierto*** *sii no es cerrado.*

Teorema

Si ξ es una expresión y Υ su tableau completado, entonces

- *ξ es satisfactible sii Υ es abierto.*
- *ξ es válido sii el tableau de $\neg\xi$ es cerrado (todas las hojas tienen contradicciones).*
- *ξ es insatisfactible sii todos los nodos de Υ son cerrados.*

Desarrollo

Objetivo de esta sección

Cómo se pueden utilizar las técnicas mostradas anteriormente para demostrar que *"Si H_1 y H_2 y .... y H_n entonces C"*

Recuerda

- Relación entre $\models$ y Tautologías/Validez.
  a) $\mathcal{F} \models \beta$ sii $\models \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \rightarrow \beta$.
  b) $\mathcal{F} \models \beta$ sii $\neg(\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \wedge \neg\beta)$ es válida.

- Relación entre $\models$ y Contradicciones/Insatisfictibilidad.
  $\mathcal{F} \models \beta$ sii $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \wedge \neg\beta$ es insatisfactible.

Tablas de verdad

- Demostración directa de $\models$: (1) en todas las interprataciones donde el conjunto de premisas sea satisfacible, la consecuencia también es satisfacible (varias tablas); (2) o bien comprobando la tautología "$\rightarrow$" asociada (una tabla).
- Por refutación.
  1. Asumir que β es falsa y determinar la asignaciones $\{I_i\}$ de las letras de β para las que $v_{I_i}(\beta) = F$.
  2. Construir una tabla de verdad considerando únicamente las interpretaciones I_i donde se evaluarán cada una de las premisas.
  3. Si existe una fila I_i donde todas las premisas son ciertas, la fila responde a una situación donde todas las premisas son ciertas y la conclusión es falsa. Retornar: Razonamiento `no válido`.
  4. Si en todas las tablas el conjunto de premisas es insatisfacible, retornar que el razonamiento es `válido`.

Resto de las técnicas

$\mathcal{F} \models \beta$ sii

- $(\mathcal{F} \cup \{\neg\beta\})_{FNC}$ es insatisfacible por DPLL, es decir retorna `false`.
- $(\mathcal{F} \cup \{\neg\beta\})_{FNC}$ es insatisfacible por resolución, es decir se deriva la clásula vacía.
- se cumple una de estas dos condiciones:
  - todos los nodos de árbol semántico de $\mathcal{F} \to \beta$ son nodo éxito.
  - todos los nodos de árbol semántico de $\mathcal{F} \wedge \neg\beta$ son nodo fallo.
- el tableau semántico de $\mathcal{F} \wedge \neg\beta$ es completo y cerrado.

Desarrollo

Expresiones restringidas

- Una fbf es una cláusula sii es V, F, un literal o una disyunción (posiblemente vacía) de literales. Una cláusula es restringida si no contienen literales repetidos ni pares de literales opuestos.
- Una fbf está en forma normal conjuntiva sii es V, F, una cláusula o una conjunción (posiblemente vacía) de cláusulas. Una fnc es restringida si ninguna cláusula contienen literales repetidos ni pares de literales opuestos y ninguna cláusula contiene a otra. La fnc restringida de α como conjunto de cláusulas se denomina la forma clausal de α.
- Una fbf es un cubo sii es V, F, un literal o una conjunción (posiblemente vacía) de literales. Una cubo es restringido si no contienen literales repetidos ni pars de literales opuestos.
- Una fbf está en forma normal disyuntiva sii es V, F, un cubo o una disyunción (posiblemente vacía) de cubos. Una fnd es restringida si ningún cubo contienen literales repetidos ni pares de literales opuestos y ningún cubo contiene a otro.
- Toda expresión es equivalente a su fnc y a su fnd (restringida o no).

Tableaus semánticos

Teorema (Correctitud y completitud)

- *Correcto.* *Si* $\{\xi_1, \xi_2, \ldots, \xi_n, \neg\xi\}$ *puede ser refutado (su tableau está completado y es cerrado), entonces* $\{\xi_1, \xi_2, \ldots, \xi_n\} \models \xi$.
- *Completo.* *Si* $\{\xi_1, \xi_2, \ldots, \xi_n\} \models \xi$, *entonces existe una refutación para* $\{\xi_1, \xi_2, \ldots, \xi_n, \neg\xi\}$

Método DPLL

Dada una expresión ξ con conjunto clausal Ω_ξ, denotamos por $\Omega_{\xi(\ell)}$ al resultado de propagar el literal ℓ en ξ. Recuerda, propagar ℓ es obtener una expresión equivalente supuesto que $v(\ell) = V$.
$S \cong S'$ denota que S es satisfacible sii S' s satisfacible.

Teorema

- *Si* ℓ *es una cláusula unitaria en* Ω_ξ*, entonces* $\Omega_\xi \cong \Omega_{\xi(\ell)}$.
- *Si* ℓ *es un literal puro en* Ω_ξ*, entonces* $\Omega_\xi \cong \Omega_{\xi(\ell)}$.
- *Si* ℓ *no es puro ni unitario entonces si* Ω_ξ *es satisfacible lo es* $\Omega_{\xi(\ell)}$ *o lo es* $\Omega_{\xi(\neg\ell)}$
- *Si* $C_1, C_2 \in \Omega_\xi$ *tales que* $C_1 \subseteq C_2$ *(*C_1 *subsume a* C_2*), entonces* $\Omega_\xi \cong \Omega_\xi - \{C_2\}$.

Resolución

Teorema

Dado un conjunto de cláusulas Ω

- Ω *es insatisfacible sii* $\Omega \models \Box$
- Ω *es satisfacible sii* $\Omega \cup \{R_\ell(C_1, C_2)\}$

Teorema (Correctitud y completitud)

Dado un conjunto de cláusulas Ω

- *Correcto. Si existe una refutación por resolución de* Ω*, entonces* Ω *es insatisfacible.*
- *Completo.* Ω *es insatisfacible sii existe una refutación por resolución de* Ω*.*

Estrategias a Ciegas

1. **Primero en anchura.**
   - Otros nombres: Estr. a lo ancho, estr. en amplitud.
   - Criterio: Obtener todas las resolvente nivel a nivel.
     Obtener en cada paso, i, todas las posibles resolventes a nivel i.
   - Es el procedimiento de búsqueda de consistencias.
   - Es completa, pues calcula el cierre clausulado.
2. **Primero en profundiad.**
   - Criterio: Obtenida una resolvente de nivel i, intentar generar a partir de ésta una resolvente de nivel $i+1$. Se puede fijar una cota de profundidad para volver atrás y seguir por otro camino.
   - No es completa.

Estrategias de Refinamiento- I

1. **Estrategia de preferencia por unidades.**
    - Otros nombres: de resolución unitaria.
    - Criterio: un padre debe de ser unitario (formado por un sólo literal).
    - Tiene el mismo poder de inferencia que la estrategia de entrada (más adelante).
    - En general es no-completa. Falla cuando no hay cláusulas unitarias. En $\{PQ, \bar{P}R, \bar{Q}R, \bar{R}\}$ sí encuentra $\square$ pero no en $\{P\bar{Q}, \bar{Q}P\}$.
    - Es completa con cláusulas de Horn (con un literal positivo como máximo).

2. **Estrategia de entrada.**
    - Criterio: todas las resolventes son de entrada.
    - Tiene el mismo poder de inferencia que la estrategia por unidades.
    - En general es no-completa.

Estrategias de Refinamiento- II

- Es completa con cláusulas de Horn (con un literal positivo como máximo).

3. **Filtrado de antepasados.**
   - Criterio: Cada resolvente tiene un padre que:
     - o se encuentra en el conjunto inicial, (es la estrategia de entrada)
     - o es el antepasado del otro padre.
   - Es completa.

El conjunto $\{\bar{Q}\bar{P}, Q\bar{P}, \bar{Q}P, QP\}$ es insatisfactible. La estrategia de entrada no lo detecta pero sí filtrado de antepasados.

Fundamentos de la Resolución semántica

Observa los siguientes hechos

- Generalmente se aplica resolución para obtener una contradicción en $\mathcal{F}' = \mathcal{F} \cup \{\neg\phi\}$.
- No existe ninguna valuación que haga verdadero al conjunto $\mathcal{F}'$.
- Elegida una valuación, habrá cláusulas, $\mathcal{F}_1$, que serán ciertas y cláusulas, $\mathcal{F}_2$, que serán falsas.
- Cumplen que no son vacíos y que $\mathcal{F}' = \mathcal{F}_1 \cup \mathcal{F}_2$.
- Para producir la contradicción, no tiene sentido utilizar resolventes obtenidos solo de $\mathcal{F}_1$ o solo de $\mathcal{F}_2$.
- La contradicción solo se encuentra obteniendo resolventes con un padre de $\mathcal{F}_1$ y otro de $\mathcal{F}_2$.

Resolución semántica

Proceso general

- Para un interpretación dada, dividir $\mathcal{F}'$ en dos conjuntos:
  - $\mathcal{F}_1$: las cláusulas verdaderas para la interpretación, y
  - $\mathcal{F}_2$: las cláusulas falsas para la interpretación.

  Entonces:

  1. Tomar un padre de $\mathcal{F}_1$ y otro de $\mathcal{F}_2$, para obtener una resolvente R.
  2. La nueva resolvente nueva, R, formará parte o de $\mathcal{F}_1$ o de $\mathcal{F}_2$.
  3. Repetir el primer paso hasta encontrar $\square$.

- Si $\mathcal{F}$ es consistente y estamos interesados en $\mathcal{F}' = \mathcal{F} \cup \{\neg\phi\}$, no encontraremos la contradicción usando sólo $\mathcal{F}$, habrá que usar $\neg\phi$ o sus sucesores (conjunto $\mathcal{F}_2$).

Estrategias de Resolución semántica

1. **Estrategia de Conjunto soporte.**
    - Criterio: un padre siempre pertenece al conjunto soporte.
    - Es completa.
2. **Estrategia basada en Hiper resolución positiva.**
    - Hiper resolución positiva: establecer una interpretación que haga **falsas** a todas las expresiones atómicas. lo que define:
        - $\mathcal{F}_2 = \{\text{cláusulas falsas}\} =$las que solo tienen literales positivos.
        - $\mathcal{F}_1 = \{\text{cláusulas verdaderas}\} =$las que tiene al menos un literal negado.
    - Criterio: Escoger n-cláusulas de $\mathcal{F}_2$ (llamadas satélites) y una cláusula de $\mathcal{F}_1$ (llamada núcleo) y realizar las resolventes.
    - Es completa.
3. **Estrategia basada en Hiper resolución negativa**.
   Igual que la anterior pero estableciendo una interpretación que haga **verdadera** a todas las expresiones atómicas.

Estrategias de Resolución Lineal

1. **Estrategia de resolución lineal.**
   - Criterio: Se escoge una cláusula cláusula inicial llamada **cláusula cabeza**, C_0, y se forma una cadena de resolventes $R_0, R_1, R_3, \ldots, R_n$ donde:
     - $R_0 = C_0$
     - $R_{i+1} = R(R_i, C_i)$ tal que $C_i \in C$ o $C_i = R_j$ $(j \leq i)$
   - No es completa, depende de la cláusual cabeza.
   - Teorema: Si $\mathcal{F}$ es satisfactible y $\mathcal{F} \cup C$ es insatisfactible, la estretegia de resolución por resolución lineal es completa si se toma como cláusula cabeza a una de C.
2. **Resolución SLD**, (Kowalski)
   - Otros nombres: de entrada lineal, conjunto soporte con resolución lineal.
   - Criterio: Conjunto soporte + Conjunto entrada. Es decir,
     - Un padre es del conjunto soporte.
     - El otro es del conjunto inicial.

Otras estrategias - I

1. **Combinación de estrategias.** La resolución SLD realmente es una mezcla de 2 estrategias. Otra es:
   - **Conjunto soporte con filtrado de antepasados.**
   - Criterio:
     - Un padre es del conjunto soporte.
     - El otro es del conjunto inicial o predecesor del otro padre.

Otras estrategias - II

2. **Resolución ordenada.**
   - Otros nombres: Resolución selectiva.
   - Preparación: Cada cláusula se considera como un conjunto de literales ordenados.
   - Criterio: Realizar la resolución sólo con el primer literal de cada cláusula.
   - Observación: Los literales del resolvente mantienen el orden de las cláusulas padre con los literales del padre positivo (la cláusula que contenía el literal por el que se resuelve afirmado) seguidos de los literales del padre negativo (la cláusula que contenía el literal por el que se resuelve negado).
   - Es la más eficiente.
   - No es completa; pero sí lo es con cláusulas de Horn.