Representación del Conocimiento

Lógica Proposicional

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana

Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097

mailto:aguerra@uv.mx
https://www.uv.mx/personal/aguerra/rc

Maestría en Inteligencia Artificial 2025

Organización

Universidad Veracruzana

Metavariables y fórmulas

- Las reglas de derivación que se han introducido en el capítulo precedente, son válidas para cualquier fórmula que podamos formar con el lenguaje de la lógica proposicional.
- A partir de los átomos proposicionales podemos usar conectivos lógicos para crear fórmulas lógicas más complejas.
- El objetivo del capítulo anterior era entender la mecánica de las reglas de la deducción natural.

Universidad Veracruzana

Ejemplo

- Consideren este caso, una aplicación de la regla de prueba ($\implies e$):

1.	$p \implies q$	premisa
2.	p	premisa
3.	q	($\implies e$) 1,2

Universidad Veracruzana

Continuación del ejemplo

- Su aplicación es válida aún si substituimos $p \vee \neg r$ por p y q con $r \implies p$:

$$\begin{array}{lll} 1. & p \vee \neg r \implies (r \implies p) & \text{premisa} \\ 2. & p \vee \neg r & \text{premisa} \\ 3. & r \implies p & \end{array}$$

- Escribimos las reglas de prueba como esquemas de razonamiento que usan símbolos griegos que pueden ser substituidos por fórmulas genéricas.
- Estas meta-variables no son fórmulas bien formadas (fbf) del sistema.

Universidad Veracruzana

Atomos proposicionales

- Lo primero que necesitamos es una fuente no acotada de átomos proposicionales: $p, q, r, \ldots$ o $p_1, p_2, p_3, \ldots$
- La cuestión del si tal conjunto es infinito no debería preocuparnos.
- El carácter no acotado de la fuente es una forma de confrontar que si bien, normalmente necesitaremos una gran cantidad finita de proposiciones para describir un programa de computadora, no sabemos de antemano cuantos vamos a necesitar.

Universidad Veracruzana

Primer aproximación

- Que las fórmulas de nuestra lógica proposicional deben ser cadenas de caracteres formadas a partir del alfabeto

$$\{p, q, r, \dots\} \cup \{p_1, p_2, p_3, \dots\} \cup \{\neg, \wedge, \vee, \implies, (,)\}$$

es una observación trivial, que no provee la información necesaria.

- Ejemplo: $(\neg)$ y $pq \implies$ son cadenas construidas a partir de este alfabeto, aunque no tienen sentido en la lógica proposicional.

Universidad Veracruzana

Fórmulas bien formadas (definición inductiva)

1. Todo átomo proposicional $p, q, r, \ldots$ es una fórmula bien formada (fbf).
2. Si ϕ es una fórmula bien formada, también lo es $(\neg\phi)$.
3. Si ϕ y ψ son fbf, también lo es $(\phi \wedge \psi)$.
4. Si ϕ y ψ son fbf, también lo es $(\phi \vee \psi)$.
5. Si ϕ y ψ son fbf, también lo es $(\phi \implies \psi)$.

Universidad Veracruzana

Fórmula bien formada (Backus-Naur Form)

- BNF, adaptada de Huth y Ryan [3]:

$$\phi \quad ::= \quad p \mid (\neg\phi) \mid (\phi \land \phi) \mid (\phi \lor \phi) \mid (\phi \implies \phi) \tag{1}$$

donde p denota cualquier átomo proposicional y cada ocurrencia de ϕ a la derecha del símbolo $::=$ denota una fórmula bien formada previamente construida.

Universidad Veracruzana

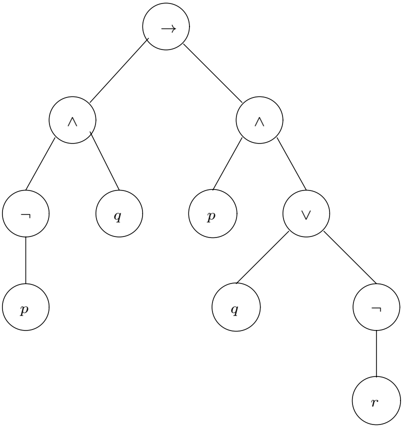
Principio de inversión

- Solo una regla se aplica a la vez.
- Ejemplo: ¿Es $(((\neg p) \wedge q) \implies (p \wedge (q \vee (\neg r))))$ una fbf?
  - La última regla aplicada fue (5) dado que la fbf es una implicación con $((\neg p) \wedge q)$ como antecedente y $(p \wedge (q \vee (\neg r)))$ como conclusión.
  - El antecedente es una conjunción (3) entre $(\neg p)$ y q, donde el primer operando es una negación (2) de p que es un átomo (1); igual que q.
  - De igual forma podemos proceder con la conclusión de la expresión original y demostrar que ésta es una fbf.

Universidad Veracruzana

Arbol sintáctico

- Las fórmulas tienen una estructura de árbol:



Sintaxis y árboles sintácticos

- La raíz del árbol es una implicación, de manera que la expresión en cuestión es, a su nivel más alto una implicación.
- Ahora basta probar recursivamente que los sub-árboles izquierdo y derecho son también fórmulas bien formadas.
- Observen que las fórmulas bien formadas en el árbol o bien tienen como raíz un átomo proposicional; o un operador con el número adecuado de operandos.
- Otro ejemplo de definición inductiva.

Universidad Veracruzana

Sub-fórmulas y Sub-árboles

- Las sub-fórmulas se corresponden a los sub-árboles de un árbol sintáctico.
- Siguiendo el ejemplo, esto incluye las hojas p y q que ocurren dos veces; así como r.
- Luego $(\neg p)$ y $((\neg p) \wedge q)$ en el sub-árbol izquierdo de la implicación.
- Así como $(\neg r)$, $(p \vee (\neg r))$ y $((p \wedge (q \vee (\neg p))))$ en el sub-árbol derecho de la implicación.
- El árbol entero es un sub-árbol de si mismo.

Universidad Veracruzana

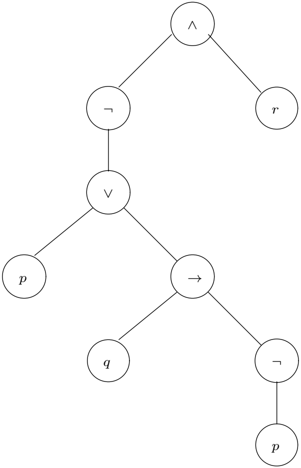
Ejemplo: sub-fórmulas

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

Universidad Veracruzana

Ejemplo: validación sintáctica

- ¿Porqué representa una fbf?



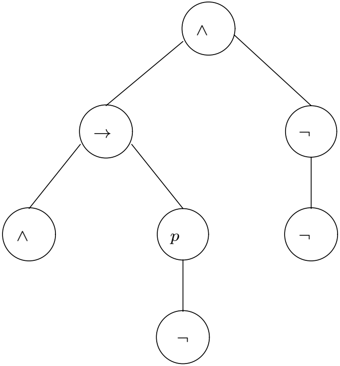
Continuación del ejemplo

- Todas sus hojas son átomos proposicionales: p dos veces; q y r una.
- Todos sus nodos internos son operadores lógicos ($\neg$ dos veces, $\wedge$, $\vee$ y $\implies$) y
- El número de sus sub-árboles es el correcto en todos los casos (un sub-árbol para la negación, dos para los demás operadores).
- La expresión linearizada del árbol puede obtenerse recorriendo el árbol de manera en orden: $((\neg(p \vee (q \implies (\neg p)))) \vee r)$.

Universidad Veracruzana

Ejemplo de fórmula no bien formada

- ¿Cómo sabemos que el árbol de la figura no representa una fbf?



Universidad Veracruzana

Continuación del ejemplo

- Hay dos razones para ello: primero, las hojas $\wedge$ y $\neg$, lo cual puede arreglarse diciendo que el árbol está parcialmente construido;
- Segundo, y definitivo, el nodo para el átomo proposicional p no es una hoja, es un nodo interno.

Universidad Veracruzana

Validez y Consecuencia Lógica

- Desarrollamos un cálculo del razonamiento que nos permite verificar si las inferencias de la forma $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ son válidas, lo cual quiere decir que a partir de las premisas $\phi_1, \phi_2, \ldots, \phi_n$ podemos demostrar ψ usando las reglas de prueba.
- Desarrollaremos una nueva relación de consecuencia lógica entre las premisas y la consecuencia de las inferencias:

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi$$

  que denota que siempre que las premisas son verdaderas, la consecuencia también lo es.

Universidad Veracruzana

Consecuencia Lógica y Valores de Verdad

- Esta relación se basa en los valores de verdad de las proposiciones atómicas que ocurren en las premisas y la conclusión;
- así como la forma en que los operadores lógicos manipulan estos valores de verdad.
- ¿Qué es un valor de verdad? Los enunciados declarativos que representan las proposiciones atómicas se corresponden con la realidad (verdaderos); o no (falsos).

Universidad Veracruzana

Ejemplo: Conjunción

- El valor de verdad de $p \wedge q$ está determinado por tres aspectos: el valor de verdad de p, el valor de verdad de q y el significado de $\wedge$.
- El significado de la conjunción captura la observación de que $p \wedge q$ es verdadera si y solo si (ssi) p y q son ambas verdaderas; en cualquier otro caso $p \wedge q$ es falsa.
- De forma que, desde la perspectiva de $\wedge$ todo lo que debemos saber es si p y q son verdaderos.
- Lo mismo para los demás operadores!

Universidad Veracruzana

Modelos y Valores de Verdad

- El conjunto de valores de verdad contiene dos elementos T y F donde T representa verdadero, y F falso.
- Un modelo de una fórmula ϕ es una asignación de valores de verdad a las proposiciones atómicas que ocurren en ϕ.
- Ejemplo: La función que asigna $q \mapsto T$ y $p \mapsto F$ es un modelo de la fórmula $p \vee \neg q$.

Universidad Veracruzana

Tabla de verdad: Conjunción

ϕ	ψ	$\phi \land \psi$
T	T	T
T	F	F
F	T	F
F	F	F

Universidad Veracruzana

Observaciones

- Como cada argumento puede tener dos valores de verdad, el número de combinaciones posibles es 2^n donde n es el número de argumentos del operador.
- Ejemplo: $\phi \wedge \psi$ tiene $2^2 = 4$ casos a considerar, los que se listan en la tabla de verdad anterior.
- Pero $\phi \wedge \psi \wedge \chi$ tendría $2^3 = 8$ casos a considerar.

Universidad Veracruzana

Tabla de verdad: Disyunción

ϕ	ψ	$\phi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

Universidad Veracruzana

Tabla de verdad: Implicación

ϕ	ψ	$\phi \implies \psi$
T	T	T
T	F	F
F	T	T
F	F	T

Universidad Veracruzana

Semántica de la implicación

- Se puede pensar en ella como una relación que preserva la verdad.
- Es evidente que $T \implies F$ no preserva la verdad, por lo que la entrada correspondiente en la tabla de verdad da como salida falso.
- También es evidente que $T \implies T$ preserva la verdad.
- Pero los casos donde el primer argumento tiene valor de verdad F también lo hacen, porque no tienen verdad a preservar dado que el supuesto de la implicación es falso.

Universidad Veracruzana

La implicación como abreviatura

- La expresión $\phi \implies \psi$ puede verse como una abreviatura de $\neg\phi \vee \psi$.
- Las tablas de verdad pueden usarse para probar que cuando la primer expresión mapea a verdadero, también lo hace la segunda.
- Esto quiere decir que ambas expresiones son semánticamente equivalentes.
- Aunque claro, las reglas de la deducción natural tratan a ambas fórmulas de manera muy diferente, dado que su sintaxis es bien diferente.

Universidad Veracruzana

Tabla de verdad: Negación y Contradicción

- Observen que en este caso tenemos $2^1 = 2$ casos que considerar.

ϕ	$\neg\phi$
T	F
F	T

- También pueden definirse tablas de verdad para la contradicción y su negación: $\perp \mapsto F$ y $\neg\perp \mapsto T$.

Universidad Veracruzana

Valor de verdad de una expresión

- Ejemplo: ¿Cual es el valor de verdad de la expresión

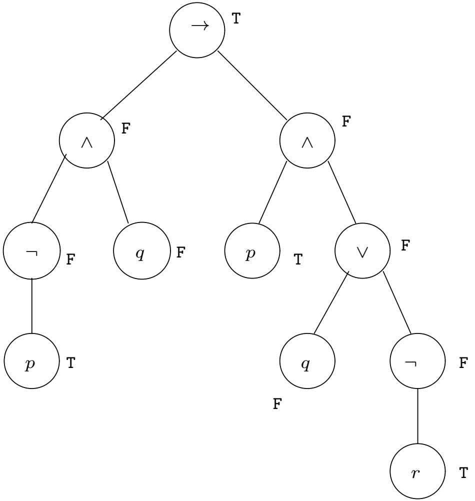
$$\neg p \wedge q \implies p \wedge (q \vee \neg r)$$

si el modelo es $q \mapsto F$, $p \mapsto T$ y $r \mapsto T$?

- Nuestra semántica es composicional: Si sabemos los valores de verdad de $\neg p \wedge q$ y $p \wedge (q \vee \neg r)$, entonces podemos usar la tabla de verdad de la implicación para saber el valor de verdad de toda la expresión.
- De forma que podemos ascender el árbol sintáctico de la expresión propagando los valores de verdad.

Universidad Veracruzana

Arboles sintácticos y evaluación de una expresión



Universidad Veracruzana

Tabla de verdad y evaluación de una expresión

- Para la expresión $(p \implies \neg q) \implies (q \vee \neg p)$:

p	q	$\neg p$	$\neg q$	$p \implies \neg q$	$q \vee \neg p$	$(p \implies \neg q) \implies (q \vee \neg p)$
T	T	F	F	F	T	T
T	F	F	T	T	F	F
F	T	T	F	T	T	T
F	F	T	T	T	T	T

Universidad Veracruzana

El castigo de Gauss

- ¿Cual es la suma de los números del 1 al 100? El rumor dice que Gauss respondió 5050 a los pocos segundos de haber sido castigado, para sorpresa de su profesor
- ¿Cómo logró Gauss librarse tan fácilmente del castigo? Quizá sabía que:

$$1 + 2 + 3 + 4 + \cdots + n = \frac{n(n+1)}{2} \tag{2}$$

de forma que:

$$\begin{aligned} 1 + 2 + 3 + 4 + \cdots + 100 &= \frac{100(101)}{2} \\ &= 5050 \end{aligned}$$

Universidad Veracruzana

La inducción matemática

- Nos permite probar que todo número natural satisface cierta propiedad M.
- Supongamos que sabemos lo siguiente de M:
  1. Caso base: Tenemos una prueba de que el número natural 1 tiene la propiedad M, *i.e.*, $M(1)$.
  2. Paso inductivo: Si n es un número natural con la propiedad $M(n)$, entonces podemos probar que $M(n+1)$ es el caso; es decir, una prueba de que $M(n) \implies M(n+1)$.
- De lo anterior se sigue que todo número natural n tiene la propiedad $M(n)$.
- Al hecho de suponer $M(n)$ en el paso inductivo, se le conoce como hipótesis de la inducción.

Universidad Veracruzana

Ejemplo: Caso base

- Queremos demostrar que la suma de $1 + 2 + 3 + 4 + \cdots + n$ es igual a $n(n+1)/2$, para todo número natural n.
- Denotamos por LIE_n a $1 + \cdots + n$; y por LDE_n a $n(n+1)/2$.
- Caso base: Si $n = 1$ entonces $LIE_1 = 1$ (solo hay un sumando), que es igual a $LDE_1 = 1(1+1)/2 = 1$.
- Paso inductivo: Asumamos que $LIE_n = LDE_n$ y probemos que $LIE_{n+1} = LDE_{n+1}$, esto es, que:

$$1 + 2 + 3 + 4 \cdots + n + (n+1) = (n+1)(((n+1)+1)/2)$$

Universidad Veracruzana

Demostración

$$
\begin{aligned}
LIE_{n+1} &= 1+2+3+4+\cdots+n+(n+1) \\
&= LIE_n+(n+1) \quad \text{reagrupando la suma} \\
&= LDE_n+(n+1) \quad \text{por la hipótesis inductiva} \\
&= \frac{n(n+1)}{2}+(n+1) \\
&= \frac{n(n+1)}{2}+\frac{2(n+1)}{2} \\
&= \frac{(n+2)(n+1)}{2} \\
&= \frac{(n+1)(n+2)}{2} \\
&= \frac{(n+1)((n+1)+1)}{2} = LDE_{n+1}
\end{aligned}
$$

Final

- Como hemos probado:
  - El caso base $M(1)$ y
  - El paso inductivo $M(n) \implies M(n+1)$

  inferimos (*modus ponens*) que todo número natural n satisface la propiedad M. $\square$
- Observen el paralelismo con la introducción de la implicación en la deducción natural.

Universidad Veracruzana

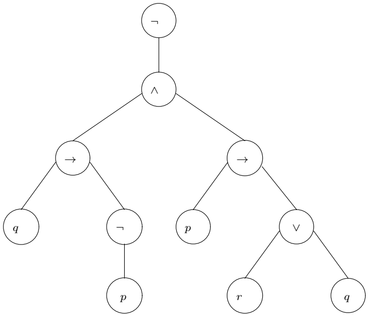
Curso de valores

- Existe una variante de inducción matemática en la que la hipótesis inductiva para probar $M(n+1)$ no es solo $M(n)$, sino la conjunción $M(1) \wedge M(2) \wedge \cdots \wedge M(n)$.
- En esta variante, llamada curso de valores, no es necesario tener un caso base explícito, todo puede hacerse en el paso inductivo.

Universidad Veracruzana

Altura de un árbol sintáctico

- Dada una fbf ϕ, definimos su altura como 1 más la longitud de la rama más larga del árbol.
- Ejemplo. La altura del siguiente árbol es 5:



Universidad Veracruzana

Inducción estructural

- Observen que la altura de una proposición atómica es $1 + 0 = 1$.
- Puesto que toda fbf tiene una altura finita, podemos demostrar enunciados sobre las fbfs haciendo inducción matemática sobre su altura.
- Este truco suele conocerse como inducción estructural, una importante técnica de razonamiento de Ciencias de la Computación.
- Es un caso especial de la inducción por curso de valores.

Universidad Veracruzana

Ejemplo

- Toda fbf proposicional tiene sus paréntesis balanceados.
- Procederemos por inducción por curso de valores sobre la altura del árbol sintáctico de la fbf ϕ.
- Denotemos por $M(n)$ que "todas las fórmulas de altura n tienen el mismo número de paréntesis que abren y cierran." Asumimos $M(k)$ para cada $k < n$ y tratamos de probar $M(n)$. Tomemos una fórmula ϕ de altura n.

Universidad Veracruzana

Continuación

- Caso base: Cuando $n = 1$, ϕ es un átomo proposicional por lo que no hay paréntesis en la expresión y $M(1)$ se satisface: $0 = 0$.
- Paso inductivo por curso de valores: Para $n > 1$ la raíz del árbol sintáctico de ϕ debe ser $\neg$, $\implies$, $\vee$ o $\wedge$. Supongamos que es $\implies$, ϕ tiene la forma $(\phi_1 \implies \phi_2)$. Usando la hipótesis de inducción asumimos que ϕ_1, cuya altura es menor que n, tiene el mismo número de paréntesis que abren y cierran; lo mismo que ϕ_2. Como en $(\phi_1 \implies \phi_2)$ agregamos un paréntesis que abre y uno que cierra, ϕ está balanceada. $\square$
- Observen que hemos probado $M(1) \wedge \cdots \wedge M(n)$.

Universidad Veracruzana

Planteamiento

- Las reglas de la deducción natural permiten desarrollar rigurosos hilos de argumentación a través de los cuales concluimos que ψ es el caso, asumiendo otras proposiciones $\phi_1, \phi_2, \ldots, \phi_n$.
- En ese caso decimos que la argumentación $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ es válida.
- ¿Tenemos evidencia de que las reglas de prueba son todas correctas? *i.e.*, ¿Preservan la verdad computada con nuestra semántica basada en tablas de verdad?

Universidad Veracruzana

Preservación de la verdad

- Dada una prueba de $\phi_1, \phi_2, \ldots \phi_n \vdash \psi$ ¿Es concebible una asignación de valores de verdad donde ψ es falso y todas las fórmulas $\phi_1, \phi_2, \ldots \phi_n$ son verdaderas?
- Afortunadamente ese no es el caso y lo demostraremos a continuación.

Universidad Veracruzana

Consecuencia Lógica

- Si, para todas las asignaciones de valores de verdad donde todas las proposiciones $\phi_1, \phi_2, \ldots \phi_n$ son verdaderas, ψ también es verdadera, decimos que

$$\phi_1, \phi_2, \ldots \phi_n \models \psi$$

es el caso. El símbolo $\models$ denota la relación de consecuencia lógica (*semantic entailment*).

Universidad Veracruzana

Ejemplos I

- ¿Es el caso que $p \wedge q \models p$? Para responder debemos inspeccionar todas las asignaciones de verdad para p y q. Cuando la asignación de valores compute T para $p \wedge q$ debemos asegurarnos de que ese también es el caso para p. Pero $p \wedge q$ solo computa T cuando p y q son verdaderas, por lo que p es consecuencia lógica de $p \wedge q$.
- ¿Qué hay acerca de $p \vee q \models p$? Hay tres asignaciones de verdad donde $p \vee q$ es T, de forma que p debería ser T en todas ellas. Sin embargo, si asignamos T a q y F a p la disyunción computa T pero p es falsa. De forma que la relación $p \vee q \models p$ no se sostiene.

Universidad Veracruzana

Ejemplos II

- ¿Qué sucede si modificamos la argumentación anterior para que sea $\neg q, p \vee q \models p$. Observe que esto obliga a focalizar en evaluaciones donde q es falsa, lo cual obliga a que p sea verdadera si queremos que la disyunción lo sea. Por tanto, es el caso que $\neg q, p \vee q \models p$
- Observen que $p \models q \vee \neg q$ es el caso, aún cuando no existen ocurrencias de q en las premisas.

Universidad Veracruzana

Correctez, definición formal

- Sean $\phi_1, \phi_2, \ldots, \phi_n$ y ψ fbfs de la lógica proposicional.
- Si $\phi_1, \phi_2, \ldots \phi_n \vdash \psi$ es válida, entonces $\phi_1, \phi_2, \ldots \phi_n \models \psi$ es el caso.

Universidad Veracruzana

Prueba

- Si $\phi_1, \phi_2, \ldots \phi_n \vdash \psi$ es válida, conocemos una prueba de ψ a partir de las premisas $\phi_1, \phi_2, \ldots \phi_n$.
- Razonaremos por inducción matemática sobre la longitud de esta prueba, *i.e.*, su número de líneas.
- $M(k)$: Para toda argumentación $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi \quad (n \geq 0)$ que tiene una prueba de longitud k, es el caso que $\phi_1, \phi_2, \ldots \phi_n \models \psi$.
- Procedemos por curso-de-valores sobre el número natural k.

Universidad Veracruzana

Ejemplo

- La inferencia $p \wedge q \implies r \vdash p \implies (q \implies r)$ tiene una prueba:

Universidad Veracruzana

Prueba

1.	$p \wedge q \implies r$	premisa
2.	p	supuesto
3.	q	supuesto
4.	$p \wedge q$	$(\wedge i)$ 2,3
5.	r	$(\implies e)$ 1,4
6.	$q \implies r$	$(\implies i)$ 3-5
7.	$p \implies (q \implies r)$	$(\implies i)$ 2-6

Universidad Veracruzana

Continuación...

- Si eliminamos las última línea, ya no tenemos una prueba.
- Sin embargo, podemos obtener una prueba re-escribiendo el supuesto de la caja más externa como una premisa:

1.	$p \wedge q \implies r$	premisa
2.	p	premisa
3.	q	supuesto
4.	$p \wedge q$	$(\wedge i)$ 2,3
5.	r	$(\implies e)$ 1,4
6.	$q \implies r$	$(\implies i)$ 3-5

Universidad Veracruzana

Continuación....

- Tenemos una prueba de que $p \wedge q \implies r, p \vdash q \implies r$.
- La hipótesis inductiva garantiza que $p \wedge q \implies r, p \models q \implies r$.
- Entonces podemos razonar que $p \wedge q \implies r \models p \implies (q \implies r)$.

Universidad Veracruzana

Prueba por curso de valores

- Asumamos que $M(k')$ para cada $k' < k$ y tratemos de probar $M(k)$.
- Caso base: Pruebas de longitud 1. Si $k = 1$ entonces la prueba debe ser de la forma:

$$1. \quad \phi \quad \text{premisa}$$

  puesto que todas las demas reglas involucran más de una línea.
- Este es el caso cuando en la argumentación $n = 1$ y ϕ_1 y ψ son iguales a ϕ, *i.e.*, $\phi \vdash \phi$.
- Evidentemente si ϕ evalua a T, es el caso que $\phi \models \phi$.

Universidad Veracruzana

Continuación...

- Paso inductivo por curso-de-valores: Asumamos que la prueba $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ tiene una longitud $k > 1$.
- Nuestra prueba tiene la siguiente estructura:

$$
\begin{array}{lll}
1. & \phi_1 & \text{premisa} \\
2. & \phi_2 & \text{premisa} \\
 & \vdots & \\
n. & \phi_n & \text{premisa} \\
 & \vdots & \\
k. & \psi & \text{justificación}
\end{array}
$$

- ¿Cual fue la justificación de la última línea?

Universidad Veracruzana

Casos: Introducción de la conjunción

- Supongamos que la última regla es $(\wedge i)$, entonces ψ tiene la forma $\psi_1 \wedge \psi_2$ y la justificación de la línea k hace referencia a dos líneas precedentes que tienen a ψ_1 y ψ_2 respectivamente, como conclusiones.
- Supongamos que esas líneas son k_1 y k_2. Dado que $k_1, k_2 < k$ existen pruebas de $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi_1$ y $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi_2$ con una longitud menor que k.
- Usando la hipótesis inductiva concluimos que $\phi_1, \phi_2, \ldots, \phi_n \models \psi_1$ y $\phi_1, \phi_2, \ldots, \phi_n \models \psi_2$.
- Estas dos relaciones implican que $\phi_1, \phi_2, \ldots, \phi_n \models \psi_1 \wedge \psi_2$.

Universidad Veracruzana

Casos: Eliminación de la disyunción

- Supongamos que la última regla es $(\vee e)$, entonces $\eta_1 \vee \eta_2$ es una premisa o supuesto en alguna línea $k' < k$, referenciada en la línea k.
- Por lo tanto, tenemos una prueba más corta del la argumentación $\phi_1, \phi_2, \ldots \phi_n \vdash \eta_1 \vee \eta_2$, obtenido al convertir los supuestos de las cajas que se abren en la línea k' en premisas.
- De forma similar obtenemos pruebas de las argumentaciones $\phi_1, \phi_2, \ldots \phi_n, \eta_1 \vdash \psi$ y $\phi_1, \phi_2, \ldots \phi_n, \eta_2 \vdash \psi$.
- Por la hipótesis inductiva $\phi_1, \phi_2, \ldots \phi_n \models \eta_1 \vee \eta_2$, $\phi_1, \phi_2, \ldots \phi_n, \eta_1 \models \psi$ y $\phi_1, \phi_2, \ldots \phi_n, \eta_2 \models \psi$.
- Por lo que $\phi_1, \phi_2, \ldots \phi_n \models \psi$.

Universidad Veracruzana

Resto de los casos.

- La argumentación continua probando que todas las reglas de prueba se comportan semánticamente de la misma forma que las tablas de verdad correspondiente. $\square$

Universidad Veracruzana

No existencia de prueba

- Digamos que queremos probar que $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ es válida, pero no lo conseguimos ¿Cómo podemos estar seguros de que no hay una prueba para ese caso?
- Basta con encontrar un modelo en donde ϕ_i evalua a T mientras que ψ evalua a F.
- Entonces $\phi_1, \phi_2, \ldots \phi_n \not\models \psi$ y, usando la robustez, esto significa que $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ no es válido
- Y por tanto, no tiene una prueba.

Universidad Veracruzana

Planteamiento

- En esta sección probaremos que las reglas de la deducción natural de la lógica proposicional son completas.
- Cuando es el caso que $\phi_1, \phi_2, \ldots, \phi_n \models \psi$, entonces existe una prueba de deducción natural para la argumentación $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.
- Combinando esto con el resultado anterior obtenemos que $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ es válida, si y sólo si $\phi_1, \phi_2, \ldots, \phi_n \models \psi$.
- Esto nos da libertad sobre qué método usar:
  - Una búsqueda de prueba al estilo de la programación lógica; o
  - La construcción de la tabla de verdad.

Universidad Veracruzana

Pasos de la demostración

- El argumento que construiremos en esta sección se da en tres pasos, asumiendo que $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ es el caso:

  1. Mostraremos que $\models \phi_1 \implies (\phi_2 \implies (\ldots(\phi_n \implies \psi)))$ es el caso.
  2. Mostraremos que $\vdash \phi_1 \implies (\phi_2 \implies (\ldots(\phi_n \implies \psi)))$ es válida.
  3. Finalmente, mostraremos que $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ es válida.

Universidad Veracruzana

Paso 1

- Una fórmula de la lógica proposicional ϕ es llamada tautología ssi evalua T bajo toda asignación de verdad, *i.e.*, $\models \phi$.
- Supongamos que $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ es el caso.
- Entonces $\phi_1 \implies (\phi_2 \implies (\ldots(\phi_n \implies \psi)))$ es una tautología: Solo puede evaluar F ssi todas las ϕ_i evalúan a T y ψ evalua a F.
- Pero esto contradice el hecho de que $\phi_1, \phi_2, \ldots, \phi_n \models \psi$; por lo tanto $\models \phi_1 \implies (\phi_2 \implies (\ldots(\phi_n \implies \psi)))$.

Universidad Veracruzana

Paso 2

- Si $\models \eta$, entonces $\vdash \eta$ es válida. En otras palabras, si η es una tautología, entonces η es un teorema.
- Asumamos que $\models \eta$.
- Dado que η contiene n distintos átomos proposicionales $p_1, p_2, \ldots, p_n$ sabemos que η es T para todas las 2^n líneas de su tabla de verdad.
- La clave está en codificar cada línea de la tabla de verdad de η como una argumentación.
- Entonces construimos pruebas para las 2^n argumentaciones y las ensamblamos en la prueba de η.

Universidad Veracruzana

Proposición

- Sea ϕ una fbf cuyos átomos proposicionales únicos son $p_1, p_2, \ldots, p_n$.
- Sea l cualquier línea en la tabla de verdad de
- Entonces, para $1 \leq i \leq n$:

$$\hat{p}_i = \begin{cases} p_i & \text{si } p_i \mapsto T \\ \neg p_i & \text{si } p_i \mapsto F \end{cases}$$

- Y es el caso que:

1. $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi$ es demostrable si la entrada para ϕ en la línea l es verdadera.
2. $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\phi$ es demostrable si la entrada para ϕ en la línea l es falsa.

Universidad Veracruzana

Prueba: Átomos proposicionales

- La prueba se lleva a cabo por inducción estructural sobre la altura del árbol sintáctico de la fbf ϕ.
- Si la altura del árbol sintáctico de ϕ es 1, entonces ϕ es un átomo proposicional.
- En ese caso ϕ tiene la forma p, y debemos mostrar que $p \vdash p$ y que $\neg p \vdash \neg p$.
- La prueba es trivial: de la premisa se deriva la premisa misma. $\square$

Universidad Veracruzana

Prueba: Negación

- Si la altura del árbol sintáctico de ϕ es mayor a uno, entonces ϕ es una fbf compuesta.
- Si ϕ es de la forma $\neg\phi_1$, tenemos dos casos a considerar:

  $\phi \mapsto T$ Entonces $\phi_1 \mapsto F$. Como ϕ_1 tiene las mismas proposiciones atómicas únicas que ϕ, por hipótesis inductiva $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\phi_1$. Ahora $\neg\phi_1$ es ϕ;

  $\phi \mapsto F$ Entonces $\phi_1 \mapsto T$ y por hipótesis inductiva $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1$. Usando $(\neg\neg i)$ obtenemos $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\neg\phi_1$. Ahora $\neg\neg\phi_1$ es $\neg\phi$. $\square$

Universidad Veracruzana

Los demás casos

- Para los demás casos ϕ tiene la forma $\phi_1 \circ \phi_2$, donde $\circ \in \{\implies, \wedge, \vee\}$.
- Sean $\{q_1, \dots, q_l\}$ los átomos proposicionales de ϕ_1 y $\{r_1, \dots, r_k\}$ los de ϕ_2.
- Entonces $\{q_1, \dots, q_l\} \cup \{r_1, \dots, r_k\} = \{p_1, \dots, p_n\}$.
- De manera que, cuando $\hat{q}_1, \dots, \hat{q}_l \vdash \phi_1$ y $\hat{r}_1, \dots, \hat{r}_k \vdash \phi_2$ son válidos, también lo es $\hat{p}_1, \dots, \hat{p}_n \vdash \phi_1 \wedge \phi_2$ por $(\wedge i)$.
- Usaremos la hipótesis inductiva para que estas conjunciones nos permitan probar si ϕ o $\neg\phi$ son el caso.

Universidad Veracruzana

Prueba: Implicación falsa

- Sea ϕ de la forma $\phi_1 \implies \phi_2$.
- Si $\phi \mapsto F$, entonces sabemos que $\phi_1 \mapsto T$ y que $\phi_2 \mapsto F$.
- Usando nuestra hipótesis inductiva tenemos que $\hat{q}_1, \ldots, \hat{q}_l \vdash \phi_1$ y que $\hat{r}_1, \ldots, \hat{r}_k \vdash \neg\phi_2$.
- De forma que $\hat{p}_1, \ldots, \hat{p}_n \vdash \phi_1 \wedge \neg\phi_2$.
- El resto consiste en probar que de tal conjunción se sigue $\neg\phi$, *i.e.*, demostrar que $\phi_1 \wedge \neg\phi_2 \vdash \neg(\phi_1 \implies \phi_2)$ es válida.

Universidad Veracruzana

Prueba: Implicación verdadera (caso 1)

- Cuando $\phi_1 \mapsto F$ y $\phi_2 \mapsto F$.
- Por hipótesis inductiva tenemos que: $\hat{q}_1, \ldots, \hat{q}_l \vdash \neg\phi_1$ y $\hat{r}_1, \ldots, \hat{r}_k \vdash \neg\phi_2$.
- De forma que $\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\phi_1 \wedge \neg\phi_2$.
- Solo queda probar que $\neg\phi_1 \wedge \neg\phi_2 \vdash \phi_1 \implies \phi_2$ es válida.

Universidad Veracruzana

Implicación verdadera (casos 2 y 3)

- Si $\phi_1 \mapsto F$ y $\phi_2 \mapsto T$ por hipótesis inductiva $\hat{p}_1, \ldots, \hat{p}_n \vdash \neg\phi_1 \wedge \phi_2$. Solo resta probar que $\neg\phi_1 \wedge \phi 2 \vdash \phi_1 \implies \phi_2$ es válida.
- Si $\phi_1 \mapsto T$ y $\phi_2 \mapsto T$, por hipótesis inductiva $\hat{p}_1, \ldots, \hat{p}_n \vdash \phi_1 \wedge \phi_2$. Solo resta probar que $\phi_1 \wedge \phi 2 \vdash \phi_1 \implies \phi_2$. $\square$
- Nota. El libro de Huth y Ryan [3] (p. 52) incluye las pruebas para la conjunción y la disyunción.

Universidad Veracruzana

Verificando tautologías

- Apliquemos la técnica a $\models \phi_1 \implies (\phi_2 \implies (\ldots(\phi_n \implies \psi)\ldots))$.
- Como es una tautología, evalúa T en las 2^n líneas de su tabla de verdad.
- Obtendremos entonces 2^n pruebas de $\hat{p}_1, \ldots, \hat{p}_n \vdash \eta$.
- Hay que ensamblar todas estas pruebas en una sola prueba de η que no tiene premisas.
- Existe una manera de hacerlo uniformemente.

Universidad Veracruzana

Ejemplo

- Para el caso de la tautología $p \wedge q \implies p$ debemos que considerar que las cuatro líneas (2^2) de su tabla de verdad.
- Por ello tendríamos cuatro pruebas del tipo $\hat{p}_1, \hat{p}_2 \vdash \eta$:

$$
\begin{aligned}
p, q &\vdash p \wedge q \implies p \\
\neg p, q &\vdash p \wedge q \implies p \\
p, \neg q &\vdash p \wedge q \implies p \\
\neg p, \neg q &\vdash p \wedge q \implies p
\end{aligned}
$$

- Pero recuerden, debemos deshacernos de las premisas ¿Cómo podemos hacer eso?

Universidad Veracruzana

Estructura de la prueba (Gracias LEM)

1.	$p \vee \neg p$	(LEM)
2.	p	supuesto
3.	$q \vee \neg q$	(LEM)
4.	q	supuesto
5.	$\vdots$	$\vdots$
6.	$p \wedge q \implies p$	
7.	$\neg q$	supuesto
8.	$\vdots$	$\vdots$
9.	$p \wedge q \implies p$	
10.	$p \wedge q \implies p$	$(\vee e)$ 3,4-6,7-9
11.	$\neg p$	supuesto
12.	$q \vee \neg q$	(LEM)
13.	q	supuesto
14.	$\vdots$	$\vdots$
15.	$p \wedge q \implies p$	
16.	$\neg q$	supuesto
17.	$\vdots$	$\vdots$
18.	$p \wedge q \implies p$	
19.	$p \wedge q \implies p$	$(\vee e)$ 12,13-15,16-18
20.	$p \wedge q \implies p$	$(\vee e)$ 1,2-10,11-19

Universidad Veracruzana

Paso 3

- Finalmente, necesitamos encontrar una prueba de que $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ es una argumentación válida.
- Tomamos la prueba de que $\vdash \phi_1 \implies (\phi_2 \implies (\ldots(\phi_n \implies \psi)))$ obtenida en el paso 2 y aumentamos la prueba introduciendo $\phi_1, \ldots, \phi_n$ como premisas.
- Aplicamos $(\implies e)$ n veces sobre cada una de las premisas y llegaremos a la conclusión que ψ lo cual nos da la prueba buscada. $\square$

Universidad Veracruzana

Corolario (robustez y completitud)

- Sean $\phi_1, \ldots, \phi_n$ y ψ fbf de la lógica proposicional.
- $\phi_1, \ldots \phi_n \models \psi$ es el caso, si y sólo si la argumentación $\phi_1, \ldots \phi_n \vdash \psi$ es válida.

Universidad Veracruzana

De lo sintáctico a lo semántico

- La correctez implica que aquello que demostremos será un hecho verdadero, con base en la semántica de tablas de verdad.
- La completitud implica que toda consecuencia lógica tiene una prueba en el sistema de deducción natural.
- Esta conexión nos permite usar indistintamente las nociones de prueba ($\vdash$) y consecuencia lógica ($\models$).
- Ahora bien, la deducción natural es solo una forma de demostración de muchas posibles.
- Exploraremos otras explotando la equivalencia semántica.

Universidad Veracruzana

Equivalencia semántica

- Decimos que dos fbf son equivalentes si tienen el mismo "significado", pero eso necesita precisarse.
- ¿Si las dos fbf tienen la misma tabla de verdad? Eso no siempre es el caso. Ej. $p \land q \implies p$ y $r \lor \neg r$.
- Def. Dos fbf ϕ y ψ son semánticamente equivalentes ssi $\phi \models \psi$ y $\psi \models \phi$, lo que solemos escribir $\phi \equiv \psi$.
- Decimos que ϕ es válida si $\models \phi$. Las tautologías son el conjunto de fbfs válidas.
- Pudimos definir $\phi \equiv \psi$ para denotar que $\models (\phi \implies \psi) \land (\psi \implies \phi)$.
- Dadas la robustez y la completitud, la equivalencia semántica y de prueba son idénticas.

Universidad Veracruzana

Ejemplos

- $p \implies q \equiv \neg q \implies \neg p$
- $p \implies q \equiv \neg p \lor q$
- $p \land q \implies p \equiv r \lor \neg r$
- $p \land q \implies r \equiv p \implies (q \implies r)$

Universidad Veracruzana

Lema

- El siguiente lema expresa que cualquier procedimiento de decisión para tautologías, es también un procedimiento de decisión para la validez de los argumentos.
- Lema. Sean $\phi_1, \phi_2, \ldots, \phi_n$ y ψ fbfs de la lógica proposicional, entonces $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ ssi $\models \phi_1 \implies (\phi_2 \implies \cdots \implies (\phi_n \implies \psi))$.
- Prueba: Supongamos que $\models \phi_1 \implies (\phi_2 \implies \ldots (\phi_n \implies \psi))$ es el caso. Si las ϕ_i son verdaderas bajo una valuación, también lo debe ser ψ. A la inversa, la demostración es el paso 1 de la prueba de completitud. $\square$

Universidad Veracruzana

CNF

- Una literal L es un átomo p o su negación (3). Una fórmula C está en Formal Normal Conjuntiva (CNF) si es una conjunción de cláusulas (5), donde cada cláusula es una disyunción de literales (4).

$$L ::= p \mid \neg p \tag{3}$$

$$D ::= L \mid L \vee D \tag{4}$$

$$C ::= D \mid D \wedge C \tag{5}$$

- Ejemplos
  - $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$
  - $(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$

Universidad Veracruzana

Relevancia de la CNF

- Permite verificar validez fácilmente, proceso que de otra forma es exponencial en el número de átomos de la fbf a verificar.
- Lema. Una disyunción de literales $L_1 \vee \ldots \vee L_m$ es válida ssi existen $1 \leq i, j \leq m$ tal que L_i es $\neg L_j$.
- Ej. $p \vee q \vee r \vee \neg q$ no puede ser falso.
- ¿El costo? Convertir una fbf proposicional a su CNF.

Universidad Veracruzana

Satisfacción

- Sea ϕ una fbf proposicional. Decimos que ϕ es satisfacible si tiene una asignación de valores en la que evalúa a *True*.
- ϕ es satisfacible ssi $\neg\phi$ no es válida.
- Prueba: Asumimos que ϕ es satisfacible. Por definición, existe una valuación donde ϕ es verdadera, pero eso implica que $\neg\phi$ sería falsa para la misma valuación, por lo que $\neg\phi$ no pude ser válida; Asumimos que $\neg\phi$ no es válida, debe haber una valuación donde $\neg\phi$ sea falsa, en cuyo caso ϕ es verdadera y por tanto satisfacible. $\square$
- Solo necesitamos un procedimiento de decisión para ambos conceptos!

Universidad Veracruzana

Ideas

- Comenzaremos por un algoritmo determinista que siempre computa la misma CNF para una fbf ϕ de entrada.
- Características:

  1. CNF termina para toda fbf de la lógica proposicional.
  2. Para toda fbf de la lógica proposicional, CNF computa una fbf equivalente en forma normal conjuntiva.

Universidad Veracruzana

Estrategia

- Proceder por inducción estructural sobre la fbf ϕ.
- Ejemplo: Si ϕ es de la forma $\phi_1 \wedge \phi_2$ computar la CNF η_i para ϕ_i, $i = 1, 2$; de forma que $\eta_1 \wedge \eta_2$ es la CNF equivalente a ϕ.
- La inducción estructural garantiza las características deseables del algoritmo.

Universidad Veracruzana

Pre-procesamiento

free_impl. Convierte las implicaciones en disyunciones, usando $\phi \implies \psi \equiv \neg\phi \vee \psi$.

nnf. Convierte las fbf en su equivalente bajo negación en forma normal (solo los átomos están negados). Se usan las leyes de De Morgan.

cnf. Computa la forma conjuntiva normal de $nnf(impl\_free(\phi))$.

Universidad Veracruzana

CNF

- Se resuelve por casos:
  1. Si ϕ es una literal, por definición está en CNF y la salida es ϕ.
  2. Si ϕ es de la forma $\phi_1 \wedge \phi_2$ se llama a CNF recursivamente sobre cada ϕ_i para obtener η_1 y η_2. La CNF es $\eta_1 \wedge \eta_2$.
  3. Si ϕ tiene la forma $\phi_1 \vee \phi_2$ se llama a CNF sobre cada ϕ_i, pero no regresamos $\eta_1 \vee \eta_2$ puesto que esta solo es una forma normal si η_i son literales.
  4. Se debe distribuir la disyunción sobre la conjunción, garantizando que la disyunciones generadas sean de literales. Una función $distr(\eta_1, \eta_2)$ haría ese trabajo.

Universidad Veracruzana

CNF

function $\mathrm{CNF}(\phi)$ ▷ ϕ es una fbf sin implicación en NNF
    switch ϕ **do**
        case literal(ϕ)
            return ϕ
        case $\phi_1 \wedge \phi_2$
            return $CNF(\phi_1) \wedge CNF(\phi_2)$
        case $\phi_1 \vee \phi_2$
            return $DISTR(CNF(\phi_1), CNF(\phi_2))$
end function

Universidad Veracruzana

DISTR

function $\mathrm{DISTR}(\eta_1, \eta_2)$ ▷ η_1 y η_2 están en CNF

  if $\eta_1 = \eta_{11} \wedge \eta_{12}$ **then**

    return $\mathit{DISTR}(\eta_{11}, \eta_2) \wedge \mathit{DISTR}(\eta_{12}, \eta_2)$

  else if $\eta_2 = \eta_{21} \wedge \eta_{22}$ **then**

    return $\mathit{DISTR}(\eta_1, \eta_{21}) \wedge \mathit{DISTR}(\eta_1, \eta_{22})$

  else ▷ No hay conjunciones

    return $\eta_1 \vee \eta_2$

  end if

end function

NNF

```
function NNF(φ)                                   ▷ φ es libre de implicaciones
    switch φ do
        case literal(φ)
            return φ
        case ¬¬φ1
            return φ1
        case φ1 ∧ φ2
            return NNF(φ1) ∧ NNF(φ2)
        case φ1 ∨ φ2
            return NNF(φ1) ∨ NNF(φ2)
        case ¬(φ1 ∧ φ2)
            return NNF(¬φ1) ∨ NNF(¬φ2)
        case ¬(φ1 ∨ φ2)
            return NNF(¬φ1) ∧ NNF(¬φ2)
end function
```

Universidad Veracruzana

Ejemplo

- $\phi = (\neg p \wedge q \implies p \wedge (r \implies q))$

Universidad Veracruzana

IMPL_FREE

$$
\begin{aligned}
\texttt{IMPL\_FREE}\,(\phi) &= \neg\texttt{IMPL\_FREE}\,(\neg p \wedge q) \vee \texttt{IMPL\_FREE}\,(p \wedge (r \to q)) \\
&= \neg((\texttt{IMPL\_FREE}\,\neg p) \wedge (\texttt{IMPL\_FREE}\,q)) \vee \texttt{IMPL\_FREE}\,(p \wedge (r \to q)) \\
&= \neg((\neg p) \wedge \texttt{IMPL\_FREE}\,q) \vee \texttt{IMPL\_FREE}\,(p \wedge (r \to q)) \\
&= \neg(\neg p \wedge q) \vee \texttt{IMPL\_FREE}\,(p \wedge (r \to q)) \\
&= \neg(\neg p \wedge q) \vee ((\texttt{IMPL\_FREE}\,p) \wedge \texttt{IMPL\_FREE}\,(r \to q)) \\
&= \neg(\neg p \wedge q) \vee (p \wedge \texttt{IMPL\_FREE}\,(r \to q)) \\
&= \neg(\neg p \wedge q) \vee (p \wedge (\neg(\texttt{IMPL\_FREE}\,r) \vee (\texttt{IMPL\_FREE}\,q))) \\
&= \neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee (\texttt{IMPL\_FREE}\,q))) \\
&= \neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee q)).
\end{aligned}
$$

Universidad Veracruzana

NNF

$$
\begin{aligned}
\texttt{NNF}\ (\texttt{IMPL\_FREE}\ \phi) &= \texttt{NNF}\ (\neg(\neg p \land q)) \lor \texttt{NNF}\ (p \land (\neg r \lor q)) \\
&= \texttt{NNF}\ (\neg(\neg p) \lor \neg q) \lor \texttt{NNF}\ (p \land (\neg r \lor q)) \\
&= (\texttt{NNF}\ (\neg\neg p)) \lor (\texttt{NNF}\ (\neg q)) \lor \texttt{NNF}\ (p \land (\neg r \lor q)) \\
&= (p \lor (\texttt{NNF}\ (\neg q))) \lor \texttt{NNF}\ (p \land (\neg r \lor q)) \\
&= (p \lor \neg q) \lor \texttt{NNF}\ (p \land (\neg r \lor q)) \\
&= (p \lor \neg q) \lor ((\texttt{NNF}\ p) \land (\texttt{NNF}\ (\neg r \lor q))) \\
&= (p \lor \neg q) \lor (p \land (\texttt{NNF}\ (\neg r \lor q))) \\
&= (p \lor \neg q) \lor (p \land ((\texttt{NNF}\ (\neg r)) \lor (\texttt{NNF}\ q))) \\
&= (p \lor \neg q) \lor (p \land (\neg r \lor (\texttt{NNF}\ q))) \\
&= (p \lor \neg q) \lor (p \land (\neg r \lor q)).
\end{aligned}
$$

Universidad Veracruzana

Corrida en Prolog

```
?- impl_free(~p ^ q => p ^ (r => q), IMPLFREE).
IMPLFREE = (~ (~p^q)v p^(~r v q))

?- impl_free(~p ^ q => p ^ (r => q), IMPLFREE), nnf(IMPLFREE,NNF).
IMPLFREE = (~ (~p^q)v p^(~r v q)),
NNF = ((p v ~q)v p^(~r v q))

?- cnf(~p ^ q => p ^ (r => q), CNF).
CNF = ((p v ~q)v p)^((p v ~q)v~r v q)

?- cnf(r => (s => (t ^ s => r)), CNF).
CNF = (~r v ~s v (~t v ~s)v r)
```

Universidad Veracruzana

SAT (Davis, Logemann y Loveland [1])

```
function DPLL(f,θ)                          ▷ f es una fbf en CNF, θ es una asignación de verdad
    θ1 ← θ ∪ unit-propagation(f, θ)
    if is-satisfied(f, θ1) then
        return θ1
    else if is-conflicting(f, θ1) then
        return ⊥
    else
        x ← choose-free-variable(f, θ1)
        θ2 ← DPLL(f, θ1 ∪ {x ↦ true})
        if θ2 ≠ ⊥ then
            return θ2
        else
            return DPLL(f, θ1 ∪ {x ↦ false})
        end if
    end if
end function
```

Universidad Veracruzana

Propiedades

- Las entradas al algoritmo son:

  f una fbf en CNF; y

  θ una función de asignación de verdad parcial, normalmente vacía, *i.e.*, $\theta : Vars \mapsto \{true, false\}$, t.q. $Vars$ es el conjunto de proposiciones atómicas únicas que ocurren en f.

- El algoritmo regresa $\perp$ si la fbf f no puede satisfacerse (fallo).
- En cualquier otro caso, θ satisface f al terminar.

Universidad Veracruzana

Propagación unitaria

- Consiste en extender θ deduciendo una asignación de variables θ_1 que satisface f.
- Ejemplo: Sea $f = (\neg x \vee z) \wedge (u \vee \neg v \vee w) \wedge (\neg w \vee y \vee \neg z)$. $Vars = \{u, v, w, x, y, z\}$. Sea $\theta = \{x \mapsto true, y \mapsto false\}$:
  1. Al considerar $(\neg x \vee z)$ necesariamente debe ser el caso que $z \mapsto true$.
  2. Para $(\neg w \vee y \vee \neg z)$ se sigue que $w \mapsto false$.
  3. Para $(u \vee \neg v \vee w)$ no aporta información porque hay dos variables sin asignación, u y v.

  La función propagación unitaria debe regresar $\{w \mapsto false, z \mapsto true\}$ para extender θ en θ_1.

Universidad Veracruzana

Literales observadas

- Solo podemos derivar información de una cláusula si ésta no contiene dos incógnitas.
- Vigilar cada cláusula monitoreando dos de sus incógnitas para implementar la propagación unitaria.
- Ejemplo Para $(u \vee \neg v \vee w)$, u y v son monitores adecuados. Para $(\neg w \vee y \vee \neg z)$, w y z lo son. Para $(\neg x \vee z)$ necesariamente x y z son sus monitores. Estos son todos los monitores necesarios.
- Cuando θ es extendida con $x \mapsto \mathit{true}$, no hay otro monitor para $(\neg x \vee z)$, la propagación unitaria infiere que $z \mapsto \mathit{true}$ debe ser el caso.
- Esta asignación es detectada por los monitores de la cláusula $(\neg w \vee y \vee \neg z)$, por lo que sus nuevos monitores serán w e y.

Universidad Veracruzana

Terminación

- $is\_satisfied(f, \theta)$ regresa *true* si para toda cláusula de f al menos una literal se satisface.
- En cambio, un conflicto se da si f no se puede satisfacer, *i.e.*, una de las disyunciones de la CNF no se satisface. En ese caso, $is\_satisfied/2$ regresa $\perp$.

Universidad Veracruzana

Búsqueda y casos recursivos

- Cuando no podemos establecer si f se satisface o no, una variable $\alpha \in \mathit{Vars}$ es seleccionada para etiquetarla.
- Se hace una llamada recursiva sobre f y θ_1 extendida, primero con $\alpha \mapsto \mathit{true}$ y luego con $\alpha \mapsto \mathit{false}$.
- La terminación se garantiza porque *Vars* se reduce estrictamente en cada llamada.

Universidad Veracruzana

Implementación en Prolog I

```
% Test:
% Clauses = [[false-X, true-Y],[false-X, false-Z]], sat(Clauses,[X,Y,Z]).

sat(Clauses, Vars) :-
    problem_setup(Clauses), elim_var(Vars).

elim_var([]).
elim_var([Var | Vars]) :-
    elim_var(Vars), assign(Var).

assign(true).
assign(false).

problem_setup([]).
problem_setup([Clause | Clauses]) :-
    clause_setup(Clause),
    problem_setup(Clauses).

clause_setup([Pol-Var | Pairs]) :-
    set_watch(Pairs, Var, Pol).

set_watch([], Var, Pol) :-
```

Universidad Veracruzana

Implementación en Prolog II

```
    Var = Pol.
set_watch([Pol2-Var2 | Pairs], Var1, Pol1) :-
    when((nonvar(Var1); nonvar(Var2)),watch(Var1, Pol1, Var2, Pol2, Pairs)).

watch(Var1, Pol1, Var2, Pol2, Pairs) :-
    nonvar(Var1) ->
  update_watch(Var1, Pol1, Var2, Pol2, Pairs);
  update_watch(Var2, Pol2, Var1, Pol1, Pairs).

update_watch(Var1, Pol1, Var2, Pol2, Pairs) :-
    Var1 == Pol1 -> true; set_watch(Pairs, Var2, Pol2).
```

Universidad Veracruzana

Ideas principales

- Se basa en el artículo de Howe y King [2], explotando tres aspectos de Prolog para ello:
    1. El uso de variables lógicas.
    2. La reconsideración.
    3. La suspensión y continuación de metas.

Universidad Veracruzana

Representación

- f, la fbf en CNF, será representada como una lista de listas.
- Ejemplo $(\neg x \vee y) \wedge (\neg x \vee \neg z)$ se representará como `[[false-X, true-Y], [false-X, false-Z]]`.
- Cada lista interna representa una cláusula y la lista completa es la conjunción de estas.
- Las literales se representan como pares `Pol-Var`, donde `Var` es una variable lógica y `Pol` es `true` o `false` para indicar si la literal es positiva o negativa.

Universidad Veracruzana

Literales observadas

- *sat*/2 se basa en lanzar una meta *watch*/5 para cada cláusula, observando dos de sus literales.
- Como la polaridad de las literales se conoce, esto se reduce a bloquear la ejecución de la meta hasta que una de las dos incógnitas observadas sea instanciada, vía *when*/2.
- Si la variable instanciada tiene el mismo valor que su polaridad, no hay que hacer nada más, y regresa *true*, en cualquier otro caso habrá que explorar el resto de las variables vía *set_watch*/3.

Universidad Veracruzana

Propagación unitaria I

- El primer caso de *set_watch*/3 se da cuando no hay más variables para observar: Si la variable restante no está instanciada, ocurre la propagación unitaria, asignando a la variable un valor que satisface la cláusula donde ocurre.
- Si la polaridad de la variable es *true*, se le asigna *true*, en caso contrario se le asigna *false*.
- Una sola unificación es suficiente para contender con ambos casos. Si *Var* y *Pol* no puede unificar, entonces *f* no se puede satisfacer.
- Una vez que *problem_setup* lanzo procesos para cada una de las cláusulas en *f*, *elim_var*/1 liga las variables en *Vars* con un valor de verdad vía *assign*/1.

Universidad Veracruzana

Propagación unitaria II

- El control de la ejecución regresa a *watch*/5 en cuanto una de las literales observadas es instanciada (Se puede trazar una meta para ver este efecto).

Universidad Veracruzana

Búsqueda

- Prolog permite deshacer ligaduras conflictivas por medio de la reconsideración. Si $Var = Pol$ falla, se puede hacer *backtracking* a otra llamada a $elim\_var/1$, *i.e.*, otra asignación de valores de verdad es intentada.
- Evidentemente, eso hace posible también que *sat* encuentre varias soluciones vía reconsideración (cinco para el ejemplo).

Universidad Veracruzana

Corrida

- Probar que $\neg x \vee (y \wedge \neg z)$ es satisfacible.
- En CNF: $(\neg x \vee y) \wedge (\neg x \vee \neg z)$.

```
?- Clauses = [[false-X, true-Y],[false-X, false-Z]], sat(Clauses,[X,Y,Z]).
Clauses = [[false-false, true-true], [false-false, false-true]],
X = false,
Y = Z, Z = true ;
Clauses = [[false-false, true-false], [false-false, false-true]],
X = Y, Y = false,
Z = true ;
Clauses = [[false-true, true-true], [false-true, false-false]],
X = Y, Y = true,
Z = false ;
Clauses = [[false-false, true-true], [false-false, false-false]],
X = Z, Z = false,
Y = true ;
Clauses = [[false-false, true-false], [false-false, false-false]],
X = Y, Y = Z, Z = false.
```

Universidad Veracruzana

Aplicaciones

- Marques-Silva [4] lista:
  - Verificación de equivalencia combinatoria en el diseño de circuitos electrónicos.
  - Generación automática de patrones de prueba para circuitos electrónicos.
  - Verificación de modelos con lógicas temporales (*BMC, bounded model checking*).
  - Planeación sobre un sistema de transición entre estados definidos con variables booleanas. La idea tras BMC.
  - Inferencia de haplotipos en bioinformática.
- *Scheduling* y, en general, satisfacción de restricciones.
- Aprendizaje automático, ver este tutorial.

Universidad Veracruzana

Referencias I

[1] M Davis, D Logemann y D Loveland. "A machine program for theorem proving". En: *Communications of the ACM* 5.7 (1962), págs. 394-397.

[2] JM Howe y A King. "A pearl in SAT and SMT solving in Prolog". En: *Theoretical Computer Science* 435.2012 (2012), págs. 43-55.

[3] M Huth y M Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge, UK: Cambridge University Press, 2004.

[4] J Marques-Silva. "Practical applications of boolean satisfiability". En: *2008 9th International Workshop on Discrete Event Systems*. Los Alamitos, CA, USA: IEEE CSP, 2008, págs. 74-80.

Universidad Veracruzana