Lógica Computacional 2022-1, nota de clase 11
Resolución binaria

Favio Ezequiel Miranda Perea   Araceli Liliana Reyes Cabello
Lourdes Del Carmen González Huesca   Pilar Selene Linares Arévalo

18 de enero de 2022

1. Discusión preliminar

Vamos a trabajar en la lógica sin igualdad. Recordemos que cualquier fórmula φ puede transformarse a una forma normal especial llamada forma cláusular que se representa como un conjunto de cláusulas:

$$Cl(\varphi) =_{def} \mathcal{C}_1, \ldots, \mathcal{C}_k$$

donde todas las cláusulas tienen variables ajenas y todas las variables se consideran cuantificadas universalmente, aunque los cuantificadores no se escriben.

Por otra parte, el proceso de unificación permite transformar, mediante sustituciones dos términos o fórmulas atómicas en una misma de manera sintáctica.

Con estas dos herramientas podemos extender el método de resolución binaria a la lógica de predicados. Por ejemplo si se tienen las siguientes dos clásulas:

1. $Qayfx \vee \neg Ryga$
2. $\neg Pwz \vee Sbv \vee \neg Qzcv$

se observa que tenemos dos literales, las que tienen al predicado Q, que nos gustaría considerar como complementarias, pues en una figura Q y en la otra $\neg Q$, aunque no lo son realmente debido a sus términos.

Recordemos que todas las variables que figuran en cláusulas son cuantificadas universalmente y que las cláusulas se consideran verdaderas, por lo que podemos sustituir sus variables a nuestro conveniencia. Cada claúsula obtenida por sustitución se llama una instancia de la cláusula original. Aquí entra en acción el mecanismo de unificación, ¿ Cómo hacer que las literales $Qayfx$ y $\neg Qzcv$ sean contrarias?

Obsérvese que este problema se reduce a lograr que las literales $Qayfx$ y $Qzcv$ sean idénticas, pero es esta clase de problemas los que resuelve el algoritmo de unificación. Basta hallar un unificador más general del conjunto $\{Qayfx, Qzcv\}$ y considerar las instancias de las cláusulas $1, 2$ que aplican dicho unificador, que según el algoritmo de Marteli-Montanari es: $\mu = [y, z, v := c, a, fx]$

3. $Qacfx \vee \neg Rcga$ μ en 1.
4. $\neg Pwa \vee Sbfx \vee \neg Qacfx$ μ en 2.

Obsérvese que el unificador debe aplicarse a toda la cláusula y NO solamente a las literales que nos interesan. Con esto se ha resuelto el problema anterior, ahora sí tenemos dos literales contrarias, una en cada cláusula, a saber $Qacfx$ en la cláusula 3 y $\neg Qacfx$ en la cláusula 4. Por lo que podemos aplicar la regla de resolución binaria, obteniendo:

5. $\neg Rcga \vee \neg Pwa \vee Sbfx$ Res 3,4

Esta claúsula 5 es un resolvente binario de 1 y 2. Por lo general, el paso intermedio de hallar el unificador se hace "al vuelo", es decir, no se indican los pasos 3 y 4 por separado.

2. Resolución binaria con unificación

La regla de resolución binaria es de primordial importancia para los sistemas de programación lógica y razonamiento automatizado, al proporcionar un método mecánico para decidir la consecuencia lógica $\Gamma \vdash \varphi$, mediante la obtención de la cláusula vacía $\square$ a partir de las formas clausulares de las fórmulas del conjunto $\Gamma \cup \{\neg\varphi\}$.

La regla de resolución para la lógica de predicados toma como premisas dos cláusulas $\mathcal{C}, \mathcal{D}$ con variables ajenas tales que existe una literal ℓ en una de ellas y una literal ℓ' en la otra de manera que ℓ^c y ℓ' son **unificables** mediante un σ. La regla devuelve como conclusión la disyunción de las literales de $\mathcal{C}, \mathcal{D}$ después de eliminar las literales contrarias $\ell\sigma$ y $\ell'\sigma$, pero aplicando a cada una de ellas el unificador σ.

Esquemáticamente tenemos lo siguiente:

$$\frac{\mathcal{C} =_{def} \mathcal{C}_1 \vee \ell \qquad \mathcal{D} =_{def} \mathcal{D}_1 \vee \ell' \qquad Var(\mathcal{C}) \cap Var(\mathcal{D}) = \varnothing \qquad \sigma \text{ umg de } \{\ell^c, \ell'\}}{(\mathcal{C}_1 \vee \mathcal{D}_1)\sigma}$$

En tal caso decimos que $(\mathcal{C}_1 \vee \mathcal{D}_1)\sigma$ es un **resolvente** de $\mathcal{C}$ y $\mathcal{D}$.
Obsérvese que la restricción acerca de las variables siempre puede obtenerse al renombrar las variables de alguna de las cláusulas, lo cual es válido pues las variables de una cláusula en realidad estaban cuantificadas universalmente en una forma normal de Skolem. Por esto se pide, al final del proceso para obtener la forma clausular, que se renombren variables para que cada cláusula tenga variables distintas.
Veamos un ejemplo

$$\frac{\mathcal{C} = Pxy \vee \neg Qax \qquad \mathcal{D} = Rz \vee Qzb}{Pby \vee Ra}$$

en tal caso $\ell = \neg Qax$, $\ell' = Qzb$ y $\sigma = [x := b, z := a]$.

Veamos ahora un ejemplo de decisión de consecuencia lógica utilizando resolución binaria:

Ejemplo 1 Verificar que $\forall x \exists y(Cx \to Py \wedge Axy) \models \forall z(Cz \to \exists w(Pw \wedge Azw))$.
La transformación a formas clausulares de la premisa y la negación de la conclusión resulta en el siguiente conjunto:

$$\{\neg Cx \vee Pfx,\ \neg Cx \vee Axfx,\ Ca,\ \neg Pw \vee \neg Aaw\}$$

La derivación mediante resolución es:

1.	$\neg Cx \vee Pfx$	$Hip.$
2.	$\neg Cx \vee Axfx$	$Hip.$
3.	Ca	$Hip.$
4.	$\neg Pw \vee \neg Aaw$	$Hip.$
5.	Pfa	$Res(1, 3, [x := a])$
6.	$\neg Aafa$	$Res(5, 4, [w := fa])$
7.	$\neg Ca$	$Res(6, 2, [x := a])$
8.	$\square$	$Res(3, 7, \varnothing)$

Dado que la cláusula vacía fue obtenida podemos concluir que la consecuencia lógica original es válida.

Es importante mencionar que la restricción acerca de que las variables de las dos cláusulas sobre las que se va a aplicar la resolución sean ajenas es primordial para demostrar la completud refutacional, sin tal restricción tendríamos por ejemplo que el conjunto insatisfacible $\{\forall x Px, \forall x \neg Pfx\}$ cuyas formas clausulares son $\{Px, \neg Pfx\}$, no permite derivar la cláusula vacía, pues el conjunto $\{Px, Pfx\}$ no es unificable.
El renombre de variables nos lleva al conjunto $\{Py, Pfx\}$ unificable mediante $\sigma = [y := fx]$.

Veamos un ejemplo más elaborado:

Cualquier coyote persigue a algún correcaminos. Los correcaminos que dicen bip-bip son listos. Ningún coyote atrapa a correcaminos listos. Cualquier coyote que persigue a algún correcaminos pero no lo atrapa está frustrado. Luego entonces, si todos los correcaminos dicen bip-bip. entonces todos los correcaminos están frustrados.

Las formas cláusulares de las premisas de este argumento y de la negación de las conclusión son las siguientes, enlistadas en un orden distinto a como aparecen las premisas.

1. $\neg Rx \vee \neg Bx \vee Lx$
2. $\neg Rx \vee Bx$
3. Ca
4. $\neg Cx \vee \neg Ry \vee \neg Pxy \vee Axy \vee Fx$
5. $\neg Fa$
6. $\neg Cx \vee Rfx$
7. $\neg Cx \vee \neg Ry \vee \neg Ly \vee Axy$
8. $\neg Cx \vee Pxfx$

A continuación se busca la cláusula vacía $\square$ mediante resolución binaria: las unificaciones y el renombre de variables entre dos cláusulas se hace al vuelo:

9. $\neg Ca \vee \neg Ry \vee \neg Pay \vee Aay$
10. $\neg Ry \vee \neg Pay \vee Aay$
11. $\neg Cx \vee \neg Pafx \vee Aafx$
12. $\neg Pafa \vee Aafa$
13. $\neg Ca \vee Aafa$
14. $Aafa$
15. $\neg Ca \vee \neg Rfa \vee \neg Lfa$
16. $\neg Rfa \vee \neg Lfa$
17. Rfa
18. $\neg Lfa$

19. $\neg Rfa \vee \neg Bfa$

20. $\neg Rfa \vee \neg Rfa$

21. $\neg Rfa$

22. $\neg Ca$

23. $\Box$

Para mejorar la comprensión del método de resolución binaria, sugerimos realizar los siguientes ejercicios relacionados al ejemplo anterior.

1. De acuerdo al argumento en español y a los predicados que figuran en la especificación formal, defina el glosario.

2. Usando el glosario anterior realice la especificación formal en lógica de predicados de cada una de las premisas y conclusión.

3. Haga la transformación a cada una de las fórmas normales.

4. Haga una correspondencia de las formas clausulares obtenidas con las que aparecen en los pasos 1 a 8 del ejemplo.

5. Justifique adecuadamente cada uno de los pasos 9 a 23. Debe anotar el unificador utilizado en cada paso.

6. Hay un paso qué no se justifica mediante resolución binaria, diga cúal es y cómo se justifica.

7. ¿Son necesarios todos los pasos de esta derivación de la cláusula vacía?

8. Busque una derivación lo más sencilla posible.