IIC2213 — Lógica para Ciencia de la Computación

Notas Semana 5

1. Lógica proposicional

La lógica proposicional nos sirve para estudiar el comportamiento de ciertas *proposiciones*, o pedazos de información, cuando son combinadas para crear otras proposiciones nuevas. Combinando distintas proposiciones podemos mejorar o hacer más específico el conocimiento que estamos modelando. Por ejemplo, si tengo las proposiciones

$$p = \text{Sócrates es hombre}$$
$$q = \text{Sócrates es mortal}$$

Puedo construir la fórmula $p \to q$, que en español se leería como *si Sócrates es hombre, entonces sócrates es mortal*. Si asumo verdaderas las fórmulas p (representando la información que Sócrates es hombre) y $p \to q$, entonces vamos a ver cómo deducir que Sócrates debe necesariamente ser mortal.

En computación usamos la lógica en miles de aplicaciones como inteligencia artificial, hardware, bases de datos, etc. Esta semana vamos a ver lo básico de la lógica proposicional, la forma más simple de lógica que usamos. Vamos a ver también como podemos usar lógica para modelar funciones booleanas, circuitos y el conocimiento de un agente inteligente.

Sintaxis de la lógica proposicional

Sea P un conjunto de proposiciones atómicas (p. ej. $\{p, q, r, s, \dots\}$). Definimos por inducción el conjunto $L(P)$ de fórmulas en lógica proposicional sobre P:

1. Todo símbolo $p \in P$ es una fórmula en $L(P)$;
2. Si φ y ψ son fórmulas en $L(P)$, entonces:
   - $(\varphi \wedge \psi)$ es una fórmula en $L(P)$;
   - $(\varphi \vee \psi)$ es una fórmula en $L(P)$;
   - $(\varphi \to \psi)$ es una fórmula en $L(P)$;
   - $(\varphi \leftrightarrow \psi)$ es una fórmula en $L(P)$; y
   - $(\neg\varphi)$ es una fórmula en $L(P)$

Como siempre, vamos a omitir los paréntesis cuando no exista riesgo de doble lectura, por ejemplo escribiremos $\neg\varphi \wedge \psi$ en vez de $((\neg\varphi) \wedge \psi)$.

Semántica de la lógica proposicional

La semántica se define en términos de *valuaciones*. Estas son funciones $\tau : P \to \{0, 1\}$, que intuitivamente representan un valor de verdad de cada una de las proposiciones atómicas en P. La semántica de la lógica proposicional entonces viene dada por la relación $\models$. Sea φ una fórmula en $L(P)$. La idea es definir cuando la valuación τ *satisface* a φ, lo que escribimos como $\tau \models \varphi$. Definimos $\models$ de forma inductiva:

- $\tau \models p$ cuando $\tau(p) = 1$.
- $\tau \models (\varphi \wedge \psi)$ cuando $\tau \models \varphi$ y $\tau \models \psi$.
- $\tau \models (\varphi \vee \psi)$ cuando $\tau \models \varphi$ o $\tau \models \psi$.
- $\tau \models (\varphi \to \psi)$ cuando $\tau \models \varphi$ implica que $\tau \models \psi$.
- $\tau \models (\varphi \leftrightarrow \psi)$ cuando $\tau \models \varphi$ si y solo si $\tau \models \psi$.
- $\tau \models (\neg\varphi)$ cuando no es verdad que $\tau \models \varphi$. En ese caso escribimos $\tau \not\models \varphi$.

Antes de seguir vamos a introducir un par de conceptos útiles.

- Una fórmula φ en $L(P)$ es una *tautología* si para toda valuación $\tau : P \to \{0, 1\}$ se tiene que $\tau \models \varphi$.
- Una fórmula φ en $L(P)$ es una *contradicción* si para toda valuación $\tau : P \to \{0, 1\}$ se tiene que $\tau \not\models \varphi$, o alternativamente, que no existe valuación τ tal que $\tau \models \varphi$.
- Dos fórmulas φ y ψ en $L(P)$ son *logicamente equivalentes* si para si para toda valuación $\tau : P \to \{0, 1\}$ se tiene que $\tau \models \varphi$ y $\tau \models \psi$, o bien $\tau \not\models \varphi$ y $\tau \not\models \psi$.

Ejercicio. Muestre un ejemplo de una tautología y una contradicción. Demuestre que φ es una tautología si y solo si $(\neg\varphi)$ es una contradicción.

Solucion. Los ejemplos los dejamos para los lectores. Para la demostración, asumamos que φ es una fórmula en $L(P)$ que es tautología. Por la definición anterior, para toda valuación $\tau : P \to \{0, 1\}$ se tiene que $\tau \models \varphi$. Ahora analicemos $(\neg\varphi)$. Queremos demostrar que es una contradicción, y por tanto debemos mostrar que no existe valuación τ tal que $\tau \models (\neg\varphi)$. Sea entonces $\tau : P \to \{0, 1\}$ una valuación arbitraria, vamos a demostrar que no satisface $(\neg\varphi)$. En efecto, como φ es tautología, sabemos que $\tau \models \varphi$. Luego, por la definición de la relación $\models$ para fórmulas con negación, vemos que no es verdad que $\tau \models (\neg\varphi)$ (pues $\tau \models \varphi$). Eso era lo que buscábamos demostrar para τ. Como τ fue elegida arbitrariamente, mostramos que no existe valuación τ tal que $\tau \models (\neg\varphi)$.

Distintas formas de ver $L(P)$

Dependiendo de la aplicación, podemos mirar a las fórmulas de $L(P)$ de distintas formas.

Fórmulas como funciones. Imagina que el conjunto P tiene n proposiciones, es decir, $P = \{p_1, \ldots, p_n\}$. Luego cada fórmula de $L(P)$ puede verse como una función de $\{0,1\}^n \to \{0,1\}$, que toma un vector binario de tamaño n y entrega un bit de respuesta.

Más específicamente, dada una fórmula φ en $L(P)$, definimos la función asociada

$$f_\varphi : \{0,1\}^n \to \{0,1\}$$

de la siguiente forma. Sea $\bar{b} = b_1, b_2, \ldots, b_n$ un vector binario de tamaño n (es decir, cada b_i es 0 o 1), y define la valuación $\tau_{\bar{b}} : P \to \{0,1\}$ como $\tau(p_i) = b_i$, para $1 \leq i \leq n$ (en otras palabras, τ le asigna a p_i el valor del i-ésimo bit de $\bar{b}$). Luego

$$f_\varphi(\bar{b}) = 1 \text{ si y solo si } \tau_{\bar{b}} \models \varphi.$$

Podemos también dar una definición inductiva de f_φ, tal como lo hicimos para las fórmulas en $L(P)$ (completa la definición para los conectivos $\to$ y $\leftrightarrow$):

- Si $\varphi = p$ entonces $f_\varphi(\bar{b}) = \tau_{\bar{b}}(p)$.
- Si $\varphi = \psi \wedge \chi$, entonces $f_\varphi(\bar{b}) = \text{mín}(f_\psi(\bar{b}), f_\chi(\bar{b}))$.
- Si $\varphi = \psi \vee \chi$, entonces $f_\varphi(\bar{b}) = \text{máx}(f_\psi(\bar{b}), f_\chi(\bar{b}))$.
- Si $\varphi = \neg\psi$, entonces $f_\varphi(\bar{b}) = 1 - f_\psi(\bar{b})$.

Ejercicio. Considera la función $<$ que opera sobre bits: la función recibe dos bits b_1 y b_2, y entrega un 1 si $b_1 < b_2$ o un 0 en otro caso. ¿Cómo puedes representar esa función como una proposición? Representa también la función $<$ pero ahora asumiendo que recibes números binarios de tres dígitos.

Fórmulas como mundos posibles. Imagina nuevamente que el conjunto P tiene n proposiciones. ¿Cuántas valuaciones distintas hay para P? Podemos asignarle a cada valuación τ un subconjunto S_τ de P, de forma que S_τ solo contiene las proposiciones que τ hace verdaderas: $S_\tau = \{p \in P \mid \tau(p) = 1\}$. La intuición es que S_τ contiene las proposiciones de P que son *verdaderas*, y por lo tanto puede verse como un mundo posible.

Por ejemplo, si $P = \{\texttt{está\_lloviendo}, \texttt{hace\_calor}, \texttt{hay\_smog}\}$, entonces la valuación τ tal que $\tau(\texttt{está\_lloviendo}) = 1$, $\tau(\texttt{hace\_calor}) = 1$ y $\tau(\texttt{hay\_smog}) = 0$ se refiere al mundo posible en donde estamos en medio de una tormenta tropical (llueve y hace calor, pero no hay smog). Ese mundo posible es representado por el conjunto

$$S_\tau = \{\texttt{está\_lloviendo}, \texttt{hace\_calor}\}$$

de proposiciones que τ hace verdaderas.

Podemos usar esta visión para encontrar una semántica para la lógica proposicional que sea independiente de la sintaxis! Sea φ una formula en $L(P)$. Definimos el conjunto *modelos*(φ) como el conjunto que tiene a todos los conjuntos S_τ tal que $\tau \models \varphi$. Más formalmente,

$$\textit{modelos}(\varphi) = \{S_\tau \mid \tau : P \to \{0,1\} \text{ y } \tau \models \varphi\}.$$

Recuerda que la cantidad de valuaciones distintas depende del tamaño de P, y por lo tanto *modelos*(φ) tiene a lo más tantos elementos como valuaciones posibles.

Decimos que dos fórmulas φ y ψ en $L(P)$ son equivalentes si para cada valuación $\tau : P \to \{0,1\}$ se tiene que $\tau \models \varphi$ si y solo si $\tau \models \psi$. O, dicho de otra manera, si *modelos*(φ) = *modelos*(ψ).

Ejercicio. Supón que P tiene n proposiciones. ¿Cuántas fórmulas distintas en $L(P)$ existen (sin contar dos veces formulas que sean lógicamente equivalentes)?

Ejercicio. Seguramente en matemáticas discretas viste la noción de tablas de verdad para la lógica proposicional. ¿Cómo puedes relacionar las tablas de verdad con la semántica de mundos posibles?

2. Fórmulas como circuitos, y evaluar fórmulas

El problema de *evaluar* una fórmula φ en $L(P)$ consiste en saber si una valuación $\tau : P \to \{0,1\}$ satisface a φ o no. Más especificamente, consideramos este problema:

Problema:	$L(P)$-EVAL
Input:	Fórmula φ en $L(P)$, valuación $\tau : P \to \{0,1\}$
Output:	¿Es verdad que $\tau \models \varphi$?

Veamos como resolver esto de una forma más algorítmica.

- Primero, vamos a reemplazar las formulas para que en vez de variables proposicionales tengamos números 0 (por falso) y 1 (por verdadero).
- Luego, en la primera pasada por φ, vamos a reemplazar cada $p \in P$ por $\tau(p)$.
- Lo que nos queda no es una fórmula propiamente tal, pero se parece: son conectivos sobre fórmulas y/o caracteres 0 y 1. Ahora podemos ir resolviendo cada conectivo unido a dos caracteres 0 o 1. Es decir, para $a, b \in \{0,1\}$, reemplazamos conectivos de la forma $(a \wedge b)$ por el valor mín(a,b), el conectivo $(a \vee b)$ por el valor máx(a,b), $(\neg a)$ por el valor $1 - a$.
- Repetimos esto hasta que no queden ningún conectivo unido a dos caracteres 0 y 1. Notar que la sintaxis de la lógica nos garantiza que cuando no pase esto, tendremos un solo caracter, ya sea 0 o 1.
- Si el valor final es 1, decimos que $\tau \models \varphi$, de lo contrario, $\tau \not\models \varphi$.

- Por la sintaxis de la lógica, siempre hay al menos un conectivo listo para ser evaluado, y en total tenemos que hacer un reemplazo por cada conectivo en φ. Siguiendo este argumento, podemos mostrar que existe un algoritmo para evaluar φ con τ que está en $O(n)$[1].

Noción de consecuencia lógica

De forma intuitiva, vamos a escribir que $\{p, p \to q\} \models q$ cuando desde el conjunto $\{p, p \to q\}$ de fórmulas podamos deducir q. En ese caso diremos *de* $\{p, p \to q\}$ *es consecuencia lógica* q, o q *es consecuencia lógica de* $\{p, p \to q\}$.

Definición. Sea P un conjunto de proposiciones y φ y ψ dos fórmulas en $L(P)$. Entonces $\varphi \models \psi$ si y solo si para cada valuación $\tau : P \to \{0, 1\}$ tal que $\tau \models \varphi$ se tiene que $\tau \models \psi$.

Ejercicio. Demuestre que $\varphi \models \psi$ si y solo si $\varphi \to \psi$ es una tautología.

Ahora extendemos nuevamente el operador de consecuencia lógica $\models$ para que opere sobre conjuntos de fórmulas. Dado un conjunto Σ de fórmulas en $L(P)$ y una valuación τ sobre P, decimos que $\tau \models \Sigma$ si para cada fórmula φ en Σ se tiene que $\tau \models \varphi$ (es decir, $\tau \models \Sigma$ si τ hace verdad a cada fórmula en Σ).

Definición. Sea P un conjunto de proposiciones, Σ un conjunto de fórmulas en $L(P)$ y φ una fórmula en $L(P)$. Entonces $\Sigma \models \varphi$ si y solo si para cada valuación $\tau : P \to \{0, 1\}$ tal que $\tau \models \Sigma$ se tiene que $\tau \models \varphi$.

Una primera observación interesante es que todo conjunto para el que podamos deducir cualquier cosa debe necesariamente ser una contradicción. Pero sorprendentemente, el reverso también es verdad: ¡de una contradicción podemos deducir cualquier cosa! Dejamos la demostración de esta observación como un ejercicio:

Ejercicio. Muestra que un conjunto Σ de fórmulas en $L(P)$ es una contradicción si y solo si para cada fórmula φ en $L(P)$ se tiene que $\Sigma \models \varphi$.

[1]Esto asume que contamos con la fórmula en una estructura de datos adecuada, lo que escapa a este curso. Si solo tratamos a φ como un string, lo de arriba nos da un algoritmo cuadrático que es directo de implementar: por cada reemplazo tenemos que pasar por el string entero, lo que da un total de n^2 lecturas.