Introducción a la Programación

Algoritmos y Estructuras de Datos I

Primer cuatrimestre de 2023

Lógica proposicional

Definición (Especificación) de un problema

problema *nombre*(*parámetros*) :tipo de dato del resultado{
  requiere *etiqueta* { condiciones sobre los parámetros de entrada }
  asegura *etiqueta* { condiciones sobre los parámetros de salida }
}

- *nombre*: nombre que le damos al problema
  - será resuelto por una función con ese mismo nombre
- *parámetros*: lista de parámetros separada por comas, donde cada parámetro contiene:
  - Nombre del parámetro
  - Tipo de datos del parámetro
- *tipo de dato del resultado*: tipo de dato del resultado del problema (inicialmente especificaremos funciones)
  - En los asegura, podremos referenciar el valor devuelto con el nombre de res
- *etiquetas*: son nombres opcionales que nos servirán para nombrar declarativamente a las condiciones de los requiere o aseguras.

Definición (Especificación) de un problema

- *Sobre los requiere*
  - Describen todas las condiciones y posibles valores o casuísticas de los parámetros de entrada.
  - Puede haber más de un requiere (recomendamos una condición por renglón). Se asume que valen todos juntos (es una conjunción).
  - Evitar contradicciones (un requiere no debería contradecir a otro).
- *Sobre los asegura*
  - Describen todas las condiciones y posibles valores o casuísticas de los parámetros de salida y entrada/salida en función de los parámetros de entrada.
  - Puede haber más de un asegura (recomendamos una condición por renglón). Se asume que valen todos juntos (es una conjunción).
  - Evitar contradicciones (un asegura no debería contradecir a otro).

Antes de continuar... hablemos de lógica proposicional

- Si bien no utilizaremos un lenguaje formal para especificar... ¿Es lo mismo decir...?
  - Mañana llueve e iré a comprar un paragüas
  - Si mañana llueve iré a comprar un paragüas
  - O mañana no llueve o no iré a comprar un paragüas
  - Compraré un paragüas por si mañana llueve
  - Si compro un paragüas, mañana llueve

Lógica proposicional - Sintaxis

- Símbolos:

$$\text{True} \; , \; \text{False} \; , \; \neg \; , \; \wedge \; , \; \vee \; , \; \rightarrow \; , \; \leftrightarrow \; , \; (\; , \;)$$

- Variables proposicionales (infinitas)

$$p \; , \; q \; , \; r \; , \; \dots$$

- Fórmulas

  1. True y False son fórmulas
  2. Cualquier variable proposicional es una fórmula
  3. Si A es una fórmula, $\neg A$ es una fórmula
  4. Si $A_1, A_2, \dots, A_n$ son fórmulas, $(A_1 \wedge A_2 \wedge \cdots \wedge A_n)$ es una fórmula
  5. Si $A_1, A_2, \dots, A_n$ son fórmulas, $(A_1 \vee A_2 \vee \cdots \vee A_n)$ es una fórmula
  6. Si A y B son fórmulas, $(A \rightarrow B)$ es una fórmula
  7. Si A y B son fórmulas, $(A \leftrightarrow B)$ es una fórmula

Semántica clásica

- Dos valores de verdad: “verdadero” (V) y “falso” (F).
- Conociendo el valor de las variables proposicionales de una fórmula, podemos calcular el valor de verdad de la fórmula.

p	$\neg p$
V	F
F	V

p	q	$(p \wedge q)$
V	V	V
V	F	F
F	V	F
F	F	F

p	q	$(p \vee q)$
V	V	V
V	F	V
F	V	V
F	F	F

p	q	$(p \to q)$
V	V	V
V	F	F
F	V	V
F	F	V

p	q	$(p \leftrightarrow q)$
V	V	V
V	F	F
F	V	F
F	F	V

Tautologías, contradicciones y contingencias

- Una fórmula es una tautología si siempre toma el valor V para valores definidos de sus variables proposicionales.

  Por ejemplo, $((p \wedge q) \to p)$ es tautología:

p	q	$(p \wedge q)$	$((p \wedge q) \to p)$
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	V

- Una fórmula es una contradicción si siempre toma el valor F para valores definidos de sus variables proposicionales.

  Por ejemplo, $(p \wedge \neg p)$ es contradicción:

p	$\neg p$	$(p \wedge \neg p)$
V	F	F
F	V	F

- Una fórmula es una contingencia cuando no es ni tautología ni contradicción.

Equivalencias entre fórmulas

- **Teorema.** Las siguientes fórmulas son tautologías.

  1. Doble negación
     $(\neg\neg p \leftrightarrow p)$
  2. Idempotencia
     $((p \wedge p) \leftrightarrow p)$
     $((p \vee p) \leftrightarrow p)$
  3. Asociatividad
     $(((p \wedge q) \wedge r) \leftrightarrow (p \wedge (q \wedge r)))$
     $(((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r)))$
  4. Conmutatividad
     $((p \wedge q) \leftrightarrow (q \wedge p))$
     $((p \vee q) \leftrightarrow (q \vee p))$
  5. Distributividad
     $((p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r)))$
     $((p \vee (q \wedge r)) \leftrightarrow ((p \vee q) \wedge (p \vee r)))$
  6. Reglas de De Morgan
     $(\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q))$
     $(\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q))$

Relación de fuerza

- Decimos que A es más fuerte que B cuando $(A \rightarrow B)$ es tautología.
- También decimos que A fuerza a B o que B es más débil que A.
- Por ejemplo,

  1. ¿$(p \wedge q)$ es más fuerte que p? Sí
  2. ¿$(p \vee q)$ es más fuerte que p? No
  3. ¿p es más fuerte que $(q \rightarrow p)$? Sí
     Pero notemos que si q está indefinido y p es verdadero entonces $(q \rightarrow p)$ está indefinido.
  4. ¿p es más fuerte que q? No
  5. ¿p es más fuerte que p? Sí
  6. ¿hay una fórmula más fuerte que todas? Sí, False
  7. ¿hay una fórmula más débil que todas? Sí, True

Expresión bien definida

- Toda expresión está bien definida si todas las proposiciones valen T o F.
- Sin embargo, existe la posibilidad de que haya expresiones que no estén bien definidas.
  - Por ejemplo, la expresión $x/y = 5$ no está bien definida si $y = 0$.
- Por esta razón, necesitamos una lógica que nos permita decir que está bien definida la siguiente expresión
  - $y = 0 \vee x/y = 5$
- Para esto, introducimos tres valores de verdad:
  1. verdadero (V)
  2. falso (F)
  3. indefinido ($\perp$)

Semántica trivaluada (secuencial)

Se llama secuencial porque ...

- los términos se evalúan de izquierda a derecha,
- la evaluación termina cuando se puede deducir el valor de verdad, aunque el resto esté indefinido.

Introducimos los operadores lógicos $\wedge_L$ (y-luego, o *conditional and*, o **cand**), $\vee_L$ (o-luego o *conditional or*, o **cor**).

p	q	$(p \wedge_L q)$
V	V	V
V	F	F
F	V	F
F	F	F
V	$\perp$	$\perp$
F	$\perp$	F
$\perp$	V	$\perp$
$\perp$	F	$\perp$
$\perp$	$\perp$	$\perp$

p	q	$(p \vee_L q)$
V	V	V
V	F	V
F	V	V
F	F	F
V	$\perp$	V
F	$\perp$	$\perp$
$\perp$	V	$\perp$
$\perp$	F	$\perp$
$\perp$	$\perp$	$\perp$

Semántica trivaluada (secuencial)

¿Cuál es la tabla de verdad de $\rightarrow_L$?

p	q	$(p \rightarrow_L q)$
V	V	V
V	F	F
F	V	V
F	F	V
V	$\perp$	$\perp$
F	$\perp$	V
$\perp$	V	$\perp$
$\perp$	F	$\perp$
$\perp$	$\perp$	$\perp$

Cuantificadores

- La lógica proposicional no alcanza para expresar o describir propiedades que tendran los elementos de un conjunto.
- Formalmente, ese grado de abstracción se alcanzaría introduciendo lógica de primer orden.
- En la LPO existen cuantificadores que permiten predicar sobre algunos o todos los elementos de un conjunto.
- $(\forall x : T)\ P(x)$: Fórmula lógica. Afirma que **todos** los elementos de tipo T cumplen la propiedad P.
  - Se lee "Para todo x de tipo T se cumple $P(x)$"
- $(\exists x : T)\ P(x)$: Fórmula lógica. Afirma que **al menos un** elemento de tipo T cumple la propiedad P.
  - Se lee "Existe al menos un x de tipo T que cumple $P(x)$"

En la expresión $(\forall x : T)\ P(x)$, la variable x está ligada al cuantificador. Una variable es libre cuando no está ligada a ningún cuantificador.

Ejemplo (semiformal)

- **Ejemplo:** Crear un predicado `esPrimo` que sea **Verdadero** si y sólo si el número n es un número primo.

- pred esPrimo(n : $\mathbb{Z}$) {
    n es mayor que 1 y sólo divisible por sí mismo y la unidad
  }

- **Ejemplo:** Especificar el problema de, dado un número mayor a 1, indicar si el número es un número primo (con un mini spoiler/ejemplo de especificación).

- problema *primo*(in n : $\mathbb{Z}$) : Bool {
    requiere $\{n > 1\}$
    asegura $\{res = true \leftrightarrow \text{esPrimo}(n)\}$
  }

Ejemplo

- **Ejemplo:** Crear un predicado `esPrimo` que sea **Verdadero** si y sólo si el número n es un número primo.

- pred esPrimo($n : \mathbb{Z}$) {
  $n > 1 \ \land \ (\forall n' : \mathbb{Z})(1 < n' < n \rightarrow_L n \text{ mod } n' \neq 0)$
  }

- **Observación:** `x mod y` se indefine si $y = 0$.

- **Ejemplo:** Especificar el problema de, dado un número mayor a 1, indicar si el número es un número primo.

- problema *primo*(in $n : \mathbb{Z}$) : Bool {
  requiere $\{n > 1\}$
  asegura $\{res = true \leftrightarrow \text{esPrimo}(n)\}$
  }

Operando con cuantificadores

- **Ejemplo:** Todos los enteros entre 1 y 10 son pares:
  $(\forall n : \mathbb{Z})(1 \leq n \leq 10 \rightarrow n \bmod 2 = 0)$.
- **Ejemplo:** Existe un entero entre 1 y 10 que es par:
  $(\exists n : \mathbb{Z})(1 \leq n \leq 10 \wedge n \bmod 2 = 0)$.
- En general, si queremos decir que todos los enteros x que cumplen $P(x)$ también cumplen $Q(x)$, decimos:
  $(\forall x : \mathbb{Z})(P(x) \rightarrow Q(x))$.
- Para decir que existe un entero que cumple $P(x)$ y que también cumple $Q(x)$, decimos:
  $(\exists x : \mathbb{Z})(P(x) \wedge Q(x))$.

Operando con cuantificadores

- La negación de un cuantificador universal es un cuantificador existencial, y viceversa:
  - $\neg(\forall n : \mathbb{Z})P(n) \;\leftrightarrow\; (\exists n : \mathbb{Z})\neg P(n)$

    No es cierto que todos cumplen P sí y sólo si existe un elemento que no cumple P

    Que no es lo mismo que decir:
    Ningún elemento cumple P sí y sólo si existe un elemento que no cumple P

  - $\neg(\exists n : \mathbb{Z})P(n) \;\leftrightarrow\; (\forall n : \mathbb{Z})\neg P(n)$.

    No existe un elemento que cumple P sí y sólo si todos los elemenos no cumplen P

    Que sí es lo mismo que decir:
    No existe un elemento que cumple P sí y sólo si ningún elemento cumple P

Operando con cuantificadores

- Un cuantificador universal generaliza la conjunción:

$$
\begin{aligned}
& (\forall n : \mathbb{Z})\big(a \leq n \leq b \to P(n)\big) \ \wedge \ P(b+1) \\
\leftrightarrow \ & (\forall n : \mathbb{Z})\big(a \leq n \leq b+1 \to P(n)\big).
\end{aligned}
$$

- Un cuantificador existencial generaliza la disyunción:

$$
\begin{aligned}
& (\exists n : \mathbb{Z})\big(a \leq n \leq b \wedge P(n)\big) \ \vee \ P(b+1) \\
\leftrightarrow \ & (\exists n : \mathbb{Z})\big(a \leq n \leq b+1 \wedge P(n)\big).
\end{aligned}
$$