Introducción a los Algoritmos 2C 2024

Práctico 1: Introducción al formalismo básico. Concepto de validez y satisfactibilidad.

- Expresiones aritméticas y booleanas, tipos.
- Validez y satisfactibilidad

Comenzando con la aritmética

Comenzaremos trabajando con los conceptos matemáticos con los que más familiarizado estás: los números y la aritmética, pero vamos a estudiarlos desde una perspectiva formal. En primer lugar, presentaremos la sintaxis que nos permitirá construir expresiones. La sintaxis se ocupa de definir qué símbolos podremos utilizar y de la forma correcta en que deben estar dispuestos para armar expresiones. Nuestras expresiones podrán tener la siguiente estructura:

- Un número, por ejemplo: 1, 35, 8. A estas expresiones les llamaremos constantes.
- Una letra, por ejemplo: x, y, z. A estas expresiones les llamaremos variables.
- Combinando dos expresiones con un operador, y encerrando la expresión resultante entre paréntesis, podemos construir expresiones más complicadas. Por el momento los operadores que utilizaremos serán +, −, *, /,ˆ (este último símbolo corresponde a la potenciación). Por ejemplo, combinando la variable z con la constante 6 a través del operador − se puede formar la expresión: (z − 6). Podremos eliminar los paréntesis superfluos según las reglas que analizaremos más adelante.
- A su vez, cada una de estas expresiones puede volver a combinarse con otras formando expresiones más complejas, por ejemplo: ((x + y)/(2ˆx)) + (6/3).

Para poder expresar propiedades como 2 + 1 es igual a la expresión 3, o que x es menor a x + 1 agregaremos símbolos de relación.

- Cuando combinamos dos expresiones usando un símbolo de relación decimos que construimos una fórmula. Una **fórmula** es un tipo particular de expresión. Los símbolos de relaciones que consideraremos por el momento serán: =, <, ≤, > y ≥. Por ejemplo, si tomamos las expresiones 2 * 6 y x + 7 y el símbolo de relación =, podemos construir la fórmula 2 + 6 = x + 7.

Precedencia y tipado

La precedencia de los operadores nos permite escribir fórmulas grandes de manera simple y más legible. Cuando un operador tiene mayor precedencia que otro, podemos escribir una fórmula que involucra a ambos sin necesidad de poner paréntesis, y a pesar de esto, la fórmula tiene un sentido único. Un ejemplo conocido por todos es el caso de la suma respecto a la multiplicación. El operador * tiene mayor precedencia que + y por ello es que normalmente interpretamos la expresión 2 + 4 * 3 como 2 + (4 * 3). Esta regla es la que nos permitía en la primaria "separar en términos" una expresión algebraica. Al mismo tiempo, y por la misma regla, sabemos que no es lo mismo la expresión (2 + 4) * 3 que 2 + 4 * 3. No siempre los paréntesis pueden sacarse indiscriminadamente sin cambiar el sentido de la expresión.

De la misma manera, cuando un operador asociativo se aplica múltiples veces es posible eliminar paréntesis, ya que el orden en que se efectúa la operación no altera el resultado. Por ejemplo, (2 + 4) + 7 es igual a 2 + (4 + 7) y por lo tanto podemos escribir simplemente 2 + 4 + 7. **Los operadores + y * son asociativos.**

Por otro lado, la noción de tipado nos permite distinguir expresiones que están "bien escritas", es decir que tienen sentido, que representan algún valor. El tipo de un operador o función nos dice cuál es la clase de valores que toma y cuál es la clase del valor que devuelve. En las expresiones algebraicas en general todos los valores son de tipo número, que denotamos con Num (y los subtipos Nat, Int, . . .). Como ejemplo tomemos el caso de la suma: tanto a la izquierda del símbolo "+" como a la derecha debe haber expresiones que representen números, y el resultado total también es un número.

Además del tipo Num utilizaremos el tipo de los valores booleanos, que denotamos con Bool. Este tipo incluye las constantes True y False que representan las nociones de verdadero y falso respectivamente. Los valores booleanos son importantes porque son el resultado de los operadores de comparación =, ≤ ,< , ≥, >.

Las nociones de precedencia y tipado son importantes para asegurar que escribimos expresiones que tienen un sentido único y bien definido. Los siguientes ejercicios tratan sobre estas nociones en el marco de las expresiones algebraicas.

Ejercicio 1: Sacá todos los paréntesis que sean superfluos según las reglas de precedencia y asociatividad de los operadores aritméticos.

Ejemplo:

(8 − 6) * x = (6 * (x^2)) + 3

≡ { precedencia de * por sobre + }

(8 − 6) * x = 6 * (x^2) + 3

≡ { precedencia de 2 sobre * }

(8 − 6) * x = 6 * x^2 + 3

a) (−(5 + x) + ((3 * 6)/(4 * 5))) * (8 * 5)

b) ((2^2 + 5) − (4 * 2)/4) + 1 = ((5 * x)/8) + (3 * 5) * 3

Ejercicio 2: Introducí paréntesis para hacer explícita la precedencia.

a) 5 * 3 + 4 > 7 − 7 + 3

b) 3 + 4 * x = 4

Ejercicio 3: Escribí el tipo de los siguientes operadores: *, / , >, =.

> **Ejemplo:** Como ejemplo presentamos el caso del operador +. Este operador toma dos números y devuelve otro número. Podemos escribir su tipo en notación funcional listando el tipo de los parámetros y a continuación el tipo resultado:
>
> + : Num → Num → Num
>
> Otra forma de escribir el tipo de este operador, útil para armar el árbol de tipado de una expresión, es en notación de árbol :
>
> $$\frac{\text{Num + Num}}{\text{Num}}$$

Precedencia y tipado con operaciones booleanas

La lógica proposicional introduce dos constantes que representan los valores de verdad (True y False) y diversos operadores booleanos sobre estos valores (∧, ∨,¬, ≡, ⇒). Esta lógica nos permite escribir expresiones que codifican propiedades más interesantes, como por ejemplo que “si un número a es mayor a b y, además, b es mayor a c, entonces a tiene que ser mayor a c”:

$$(a > b) \wedge (b > c) \Rightarrow (a > c)$$

Al complejizarse el lenguaje, es necesario establecer reglas de notación para poder escribir las expresiones con mayor claridad y sin ambigüedades. En particular, es posible eliminar paréntesis cuando los operadores involucrados son asociativos (i.e. no importa el orden en

que asocien los operandos), y según las reglas de precedencia, tal como hemos visto hasta ahora.
Los operadores $\wedge$, $\vee$, $\equiv$ son asociativos. Esto nos permite establecer por ejemplo que las expresiones $p \wedge q \wedge r$, junto con $p \wedge (q \wedge r)$ y $(p \wedge q) \wedge r$ son equivalentes. A continuación se listan los nuevos operadores y aquellos vistos anteriormente, en orden de mayor a menor precedencia. En los siguientes ejercicios comenzamos trabajando principalmente con los aspectos sintácticos del nuevo lenguaje, para asegurar una buena capacidad de lectura y escritura de las nuevas expresiones, y con la semántica de los nuevos operadores, es decir, su significado.

Niveles de Precedencia

√, ^	raíces y potencias
*, /	producto y división
max , min	máximo y mínimo
+, −	suma y resta
=, ≤ ,< , ≥, >	operadores de comparación
¬	negación
∨ ∧	disyunción y conjunción
⇒⇐	implicación y consecuencia
≡ ≢	equivalencia y discrepancia

Ejercicio 4: Sacá todos los paréntesis que sean superfluos según las reglas de precedencia de los operadores booleanos.

a) $((((a = b) \wedge (b = c)) \Rightarrow (a = c)) \equiv True)$

b) $(((p \Rightarrow q) \wedge (q \Rightarrow p)) \Rightarrow (p \equiv q))$

c) $(((p \wedge q) \vee (\neg r)) \Rightarrow (p \wedge (q \vee r)))$

Ejercicio 5: Introducí paréntesis para hacer explícita la precedencia.

a) $p \vee q \Rightarrow r \equiv (p \Rightarrow r) \wedge (q \Rightarrow r)$

b) $p \Rightarrow q \equiv p \vee q \equiv q$

c) $p \Rightarrow q \equiv \neg p \vee q$

Ejercicio 6: ¿Están bien escritas las siguientes expresiones? Para evitar errores, introducí paréntesis de acuerdo a las reglas de precedencia, y en caso de ser posible escribí una tabla declarando el tipo de cada variable.

a) $((\text{True} \wedge \text{False}) \Rightarrow \text{False}) \equiv \text{False}$

b) $2 = 3 \vee 3 = 4 \vee a * a + 2 \leq b + 7$

c) $(x \wedge y \equiv a) \wedge z \leq w$

d) $x + 3 \Rightarrow y$

e) $(x + 3 = y) \wedge \neg z$

f) $a \vee b = 3 + y$

g) $a > b \wedge 3 + 2 < 4 \Rightarrow c \equiv b + 1 = 2$

h) $a + 2 > c \Rightarrow 3 + 2 < b \equiv c \equiv b = 2 * a$

i) $\neg a * b + c = d \vee p \Rightarrow q \equiv r \Leftarrow s \wedge j = k + l * m$

Validez y satisfactibilidad

Dos cualidades asociadas a la semántica de una fórmula, es decir, a la interpretación o significado que le damos, son los conceptos de satisfactibilidad y validez. Estas propiedades ganan relevancia cuando se trata de fórmulas con variables. Decimos que una fórmula es satisfactible cuando es verdadera para algunos valores posibles de las variables. Decimos que es válida cuando es verdadera para todos los valores posibles de las variables.

Nuestro principal interés es distinguir cuándo una fórmula es válida y cuándo no. Como la validez es independiente de los valores de las variables involucradas, si queremos demostrar la validez de una fórmula no podemos hacer ninguna suposición sobre los valores de las variables. Por el contrario, cuando queremos demostrar que una fórmula es no válida es suficiente con encontrar al menos un valor que haga que la fórmula sea falsa: esto es un contraejemplo. Algunas veces además es posible demostrar directamente que la fórmula es falsa para todos los valores de las variables. En este caso decimos que la fórmula es no satisfactible, porque no existe ningún valor posible que haga que sea verdadera.

Importante: Cuando una fórmula es siempre verdadera la llamamos válida, cuando es verdadera para algunos valores la llamamos satisfactible, cuando es falsa para algunos valores la llamamos no válida y cuando es falsa para todos los valores la llamamos no satisfactible.

Ejercicio 7: ¿Qué valor de la variable x satisface las siguientes ecuaciones, es decir, qué valor hace que las fórmulas sean verdaderas?

Ejemplo: para la fórmula $6 * x + 8 = x + 3$:

Veamos cómo resolverlo:

$6 * x + 8 = x + 3$

$\equiv$ { restar x en ambos miembros }

$6 * x + 8 - x = x + 3 - x$

$\equiv$ { aritmética }

$5 * x + 8 = 3$

$\equiv$ { restar 8 en ambos miembros }

$5 * x + 8 - 8 = 3 - 8$

$\equiv$ { aritmética }

$5 * x = -5$

$\equiv$ { dividir por 5 en ambos miembros, aritmética }

$x = -1$

Luego la respuesta es que solamente $x = -1$ hace verdadera la fórmula $6 * x + 8 = x + 3$. En otros casos, puede haber más de un valor para x que satisfaga la fórmula; en ese caso hay que encontrarlos a todos. Notá que cuando realizamos dos pasos a la vez, utilizamos un subrayado diferente para cada subexpresión involucrada.

a) $2 * x + 3 = 5$

b) $x = x + 1$

c) $x + x = 2 * x$

Preguntas para reflexionar:

a) ¿Cómo se interpreta cada resultado?

b) ¿qué relación hay entre estos ejercicios y los conceptos de validez y satisfactibilidad? es decir ¿cómo le llamaríamos a cada fórmula?

Ejercicio 8: Demostrá que las siguientes ecuaciones son válidas, es decir que las fórmulas son verdaderas para cualquier valor que tome la variable x.

Ejemplo:

$4 * x + 14 = 2 * (2 * x + 5) + 4$

$\equiv$ { distributividad de * con +, def. de * }

$4 * x + 14 = 4 * x + 10 + 4$

$\equiv$ { def. de + }

4 * x + 14 = 4 * x + 14

≡ { reflexividad de = (e = e) }

True

a) (x − 1) * (x + 1) = x^2 − 1^2

b) (a + b)^2 = a^2 + 2 * a * b + b^2

Propiedades aritméticas:

Definición de potencia:

x^2 = x*x

Distributiva de * con + :

x * (y + z) = x*y + x*z

Conmutatividad de +:

x + y = y + x

Asociatividad de +

(x + y) + z = x + (y + z)

Conmutatividad de *:

x * y = y * x

Asociatividad de *

(x * y) * z = x * (y * z)

Preguntas para reflexionar

a) ¿identificás estas fórmulas?

b) ¿tienen nombre?

c) ¿qué expresan?

Ejercicio 9: Las siguientes ecuaciones no son válidas, es decir, para al menos algún valor de la variable x cada fórmula es falsa. Justificá dando un contraejemplo. Además, indicá cuáles son satisfactibles y qué valores las satisfacen.

Ejemplo: para la ecuación 3 * x + 1 = 3 * (x + 1).

Contraejemplo: x := 5 falsifica la fórmula ya que

3 * 5 + 1=3 * 5 + 3

≡ { def. de * }

15 + 1 = 15 + 3

≡ { def. de + }

16 = 18

≡ { igualdad de números }

False

a) $x + (y * z) = (x + y) * (x + z)$

b) $x + 100 = x$

Ejercicio 10: Decidí si son válidas o no válidas y satisfactibles o no satisfactibles las siguientes ecuaciones. Justificá en cada caso.

a) $x^2 + y^2 = (x + y)^2$

b) $(a - b) * (a + b) * ((a + b)^2 - 2 * a * b) = a^4 - b^4$

c) $x^2 + 2 * x + 4 = 0$

Preguntas:

a) ¿Qué diferencia existe entre dar un ejemplo y demostrar la validez de una fórmula?

b) ¿Qué diferencia existe entre dar un contraejemplo y demostrar que una fórmula es equivalente a Falso? ¿Cuándo se puede utilizar cada estrategia?

Ejercicio 11: Decidí si cada una de las siguientes fórmulas proposicionales es válida o no. En todos los casos justificá con una tabla de verdad o un contraejemplo, según corresponda.

a) p

b) $p \equiv p$

c) $p \equiv p \equiv p$

d) $p \Rightarrow q \equiv q \Rightarrow p$

e) $p \lor q \Rightarrow p$

f) $p \land q \Rightarrow p$

g) $p \Rightarrow q \land p$

h) $p \Rightarrow q \lor p$

i) $p \Rightarrow q$

j) $p \Rightarrow (q \Rightarrow p)$

k) $p \equiv p \equiv True$

l) $True \lor p$

m) $True \land p$

n) $False \lor p$

ñ) False $\wedge$ p

Tablas de verdad de operadores

p	**¬p** negación
False	True
True	False

p	**q**	**p ∧ q** conjunción	**p ∨ q** disyunción	**p ⇒ q** implicación	**p ≡ q** equivalencia	**p ≢ q** discrepancia
False	False	False	False	True	True	False
False	True	False	True	True	False	True
True	False	False	True	False	False	True
True	True	True	True	True	True	False

Ejercicio 12: Da ejemplos y una justificación apropiada de una fórmula proposicional:

a) válida (y por lo tanto satisfactible).

b) satisfactible pero no válida.

c) no satisfactible (y por lo tanto no válida).