SI1001 Teoría de la Computación
Lógica de Hoare

Andrés Sicard Ramírez

Universidad EAFIT

Semestre 2025-2

Preliminares

- El texto guía para estas diapositivas es (Grassman y Tremblay 1996, cap. 9).
- Los números asignados a las definiciones, ejemplos, ejercicios, figuras, páginas, secciones y teoremas en estas diapositivas corresponden a los números asignados en el texto guía.

Conceptos preliminares

Descripción

Emplearemos la lógica de Hoare para verificar programas de un subconjunto del lenguaje imperativo PASCAL:

(i) operadores aritméticos,

(ii) funciones de la biblioteca estándar,

(iii) bloques `begin-end`,

(iv) instrucciones de asignación (`:=`),

(v) instrucciones de secuenciación (`;`),

(vi) instrucciones `if-then` y `if-then-else`,

(vii) instrucciones `while`.

Conceptos preliminares

Descripción

El **estado de un programa** consiste del valor de la variables utilizadas por el programa.

Conceptos preliminares

Descripción

El **estado de un programa** consiste del valor de la variables utilizadas por el programa.

Descripción

Las **sentencias** (**proposiciones**) de la lógica de Hoare son sentencias (proposiciones) de la lógica de predicados de primer orden en las variables del programa.

Conceptos preliminares

Descripción

El **estado de un programa** consiste del valor de la variables utilizadas por el programa.

Descripción

Las **sentencias** (**proposiciones**) de la lógica de Hoare son sentencias (proposiciones) de la lógica de predicados de primer orden en las variables del programa.

Notación

Para escribir las sentencias emplearemos las siguientes constantes lógicas: $\wedge$ (conjunción), $\vee$ (disyunción), $\Rightarrow$ (implicación), $\neg$ (negación) y $=$ (igualdad).

Conceptos preliminares

Definición 9.1

Una **aserción** (**afirmación**) es una sentencia referente a un estado de programa.

Conceptos preliminares

Definición 9.1

Una **aserción** (**afirmación**) es una sentencia referente a un estado de programa.

Notación

Las aserciones se escribirán entre llaves. Es decir, $\{A\}$ denota que A es una aserción.

Conceptos preliminares

Definición 9.2

*«Si C es una parte de un código, entonces cualquier aserción $\{P\}$ se denomina **precondición** de C si $\{P\}$ sólo implica el estado inicial. Cualquier aserción $\{Q\}$ se denomina **postcondición** [de C] si $\{Q\}$ sólo implica el estado final. Si C tiene como precondición a $\{P\}$ y como postcondición a $\{Q\}$, se escribe $\{P\}\,C\,\{Q\}$. La terna $\{P\}\,C\,\{Q\}$ se denomina **terna de Hoare**.»*

Conceptos preliminares

Ejemplo (terna de Hoare)

Para la terna de Hoare

$$\{y \neq 0\} \texttt{ x := 1/y } \{x = 1/y\},$$

- la precondición $y \neq 0$ afirma que el valor de y es diferente de 0,
- la postcondición $x = 1/y$ afirma que el valor de x es $1/y$,
- y el programa con la precondición y postcondición anteriores es la instrucción `x := 1/y` .

Conceptos preliminares

Ejemplo (precondición vacía)

La terna de Hoare

$$\{\}\ \texttt{a := b}\ \{a = b\},$$

tiene una precondición vacía $\{\}$ la cual se interpreta como «verdadera para todos los estados del programa».

Conceptos preliminares

Pregunta

Las instrucciones `h := a; a := b; b := h` intercambian los valores de las variables `a` y `b`. ¿Cómo escribir una postcondición para estas instrucciones?

Conceptos preliminares

Métodos para establecer la distinción entre los valores de las variables en el estado inicial y en el estado final

(i) Uso de subíndices

El subíndice α se empleará para los valores iniciales de las variables y el subíndice ω se empleará para los valores finales de las variables.

(ii) Uso de variables ocultas

Se utilizan variables que no aparecen en el código para almacenar los valores iniciales de las variables.

Conceptos preliminares

Ejemplos (diferenciando el estado inicial y el estado final)

Ternas de Hoare para las instrucciones que intercambian los valores de dos variables.

1. Empleando los subíndices α y ω

$$\{\}\ \texttt{h := a; a := b; b := h}\ \{(a_\omega = b_\alpha) \wedge (b_\omega = a_\alpha)\}.$$

2. Empleando las variables ocultas A y B

$$\{a = A, b = B\}\ \texttt{h := a; a := b; b := h}\ \{a = B, b = A\}.$$

Conceptos preliminares

Ejemplo

Una terna de Hoare para una instrucción de asignación.

$$\{\}\ \texttt{a := b}\ \{a = b, b = b_\alpha\},$$

Conceptos preliminares

Ejemplo

Una terna de Hoare para una instrucción `if-then-else`.

$\{\}$

```
if x > y then m := x else m := y
```

$\{(x > y \Rightarrow m = x_\alpha) \wedge (\neg(x > y) \Rightarrow m = y_\alpha)\}.$

Conceptos preliminares

Definición 9.3

*«Si C es un código con la precondición $\{P\}$ y la postcondición $\{Q\}$, entonces $\{P\}\,C\,\{Q\}$ se dice que es **parcialmente correcto** si el estado final de C satisface $\{Q\}$, siempre que el estado inicial satisfaga $\{P\}$. [El programa] C también sería parcialmente correcto si no hay estado final debido a que el programa no termina. Si $\{P\}\,C\,\{Q\}$ es parcialmente correcto y C termina, entonces se dice que $\{P\}\,C\,\{Q\}$ es **totalmente correcto**.»*

Conceptos preliminares

Observación

En el subconjunto de PASCAL que estamos considerando el problema de la terminación solo está asociado a programas que tengan instrucciones `while`.

Referencias

Winfried Karl Grassman y Jean-Paul Tremblay (1996). Matemáticas Discretas y Lógica. Una Perspectiva desde la Ciencia de la Computación. Trad. por Rafael García-Bermejo, María Luisa Díez Platas y Vivian de los Ángeles Fernández Vásquez. Prentice Hall, 1996 (vid. pág. 2).