# APENDICE A LÓGICA DE HOARE Y VERIFICACIÓN DE PROGRAMAS

La lógica de Hoare nos brinda artefactos de análisis de software: las pruebas y las demostraciones. Con las notaciones y conceptos de la lógica de Hoare se puede, por ejemplo, hacer pruebas sobre propiedades observables, verificar una ejecución del programa. La utilización de demostraciones permite entre otras cosas, demostrar alguna propiedad del programa, verificar un programa para todas las ejecuciones. Las demostraciones son un artefacto teórico, entonces, por qué aprender acerca de demostraciones si no son algo práctico. Las demostraciones nos dicen como pensar acerca de la corrección de nuestros programas; lo cual es importante para el buen desarrollo y la inspección de nuestros programas. Las demostraciones también constituyen la base para la construcción de herramientas software de análisis estático.

## Especificación y verificación

Una especificación de programa establece formalmente la corrección del programa y relaciona las propiedades de los estados antes y después de la ejecución. Una verificación de programa demuestra formalmente la corrección del programa, en relación a una especificación de programa.

### ¿Por qué especificar programas?

Porque es bueno para la documentación, ya que captura de forma no ambigua lo que el programa debe hacer (y no cómo debe hacerlo). Sin embargo, las especificaciones requieren mucho conocimiento y tiempo.

## ¿Por qué verificar programas?

Porque hacer pruebas solamente puede encontrar errores, pero no puede demostrar su ausencia. Sin embargo, verificar programas tiene un precio, requiere que existan especificaciones formales, y requiere de conocimiento y tiempo.

¿Cómo argumentaría usted que este programa es correcto?

%FontSize=11
%TeXFontSize=11
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
float sum(float *array,int length){
  float sum=0.0;
  int i=0;
  while(i<length){
    sum=sum+array[i];
    i=i+1;
  }
  return sum;
}
\end{verbatim}
\end{document}

## Estados y configuraciones

Un estado (en este contexto) captura los valores de las variables en el programa, formalmente hace un mapeo Var 🡪 Int.

Una configuración describe dónde estamos en la ejecución, consiste de un punto de control y un estado.

## Propiedades de estado

Las propiedades de estado pueden ser expresadas como fórmulas lógicas en cálculo de predicado (sobre las variables del programa). Son llamadas afirmaciones (assertions), verbo y gracia %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x>y
\]
\end{document}, y %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\exists z\,(x=2\ast z)
\]
\end{document} son ejemplos de propiedades expresadas como fórmulas lógicas.

## Especificación de programa

Una especificación de programa puede ser realizada con dos afirmaciones (assertions):

* Pre-condición
* Post-condición

Notación formal: tripleta de Hoare

%FontSize=14
%TeXFontSize=14
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle\Phi\rangle P\langle\Psi\rangle
\]
\end{document}

Que se lee (rústicamente): si la ejecución del programa %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} comienza en un estado donde la precondición %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\Phi
\]
\end{document} se cumple, entonces la ejecución termina en un estado donde la postcondición %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\Psi
\]
\end{document} se cumple. Alternativamente, en la literatura se puede encontrar las tripletas de Hoare escritas con llaves en lugar de paréntesis angulares: %FontSize=14
%TeXFontSize=14
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{\Phi\} P\{\Psi\}
\]
\end{document}.

## Especificación de funciones

Un predicado es una función booleana sobre el estado de un programa, esto es, una expresión que devuelve un booleano. En los predicados frecuentemente se usan símbolos matemáticos, así como texto de programa.

### Ejemplos

* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  x=3
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  y>x
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  (x\neq 0) \Rightarrow (y+z=w)
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  s=\Sigma_{i\in 1 ..\, n}a[i]
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \forall i\in 1..\, n.\,a[i]>a[i-1] 
  \]
  \end{document}
* true

Una especificación de función puede servir como un contrato entre un cliente y la implementación de la función.

### Precondición

Una precondición es un predicado que describe la condición de la que depende una función para su correcta operación.

### Postcondición

Una postcondición es un predicado que describe la condición que la función establece después de ejecutarse correctamente.

### Corrección con respecto a la especificación

Si el cliente de una función cumple con la precondición de la función, la función se ejecutará hasta terminar y cuando la función termina su ejecución, la postcondición será verdadera.

### ¿Qué tiene que cumplir la implementación si el cliente no cumple la precondición?

Nada, absolutamente nada. No hay algo que se pueda hacer si una función se utiliza sin que se cumpla la precondición especificada.

### Ejemplo de especificación de función

%FontSize=11
%TeXFontSize=11
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
/*@requires len>=0&&array.length=len
  @
  @ensures \result ==
  @        (\sum int j; 0<=j&&j<len; array[j])
  @*/
float sum(float array[],int len){
  float sum=0.0;
  int i=0;
  while(i<len){
    sum=sum+array[i];
    i=i+1;
  }
  return sum;
}
\end{verbatim}
\end{document}

## Tripletas de Hoare

Las tripletas de Hoare permiten hacer razonamiento formal acerca de la corrección de programas utilizando pre- y post-condiciones. La sintaxis de una tripleta de Hoare es

%FontSize=14
%TeXFontSize=14
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{P\}S\{Q\}
\]
\end{document}

donde %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} y %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
Q
\]
\end{document} son predicados y %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
S
\]
\end{document} es un programa. Si comenzamos en un estado donde %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} es verdadero y se ejecuta %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
S
\]
\end{document}, entonces %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
S
\]
\end{document} terminará en un estado donde %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
Q
\]
\end{document} es verdadero.

### Ejemplos de tripletas de Hoare

* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{true\}\,x:=5\,\{x=5\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=y\}\,x:=x+3\,\{x=y+3\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x>-1\}\,x:=x*2+3\,\{x>1\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=a\}\,\mbox{\tt if}\,(x<0)\,\mbox{\tt then\ }x:=-x\,\{x=|a|\}
  \]
  \end{document}

### Ejemplo de especificación

El programa factorial %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\mbox{\tt Fact1}
\]
\end{document}

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
y=1;
z=0;
while(z != x){
  z=z+1;
  y=y*z;
}
\end{verbatim}
\end{document}

puede ser especificado con la tripleta de Hoare

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{x\geq 0\}\,\mbox{\tt Fact1}\,\{y=x!\}
\]
\end{document}

## La postcondición más fuerte

Las siguientes son todas ejemplos de tripletas de Hoare válidas

* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=5\}\,x:=x*2\,\{\mbox{true}\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=5\}\,x:=x*2\,\{x>0\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=5\}\,x:=x*2\,\{x=10\|x=5\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=5\}\,x:=x*2\,\{x=10\}
  \]
  \end{document}

todas estas tripletas son verdaderas, pero la última es la más útil. %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x=10
\]
\end{document} es la post-condición más fuerte.

**Definición (Post-condición más fuerte)**

Si %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{P\}S\{Q\}
\]
\end{document} es válida y para toda %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
Q^{\prime}
\]
\end{document} tal que %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{P\}S\{Q^{\prime}\}
\]
\end{document} es válida, %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
Q\Rightarrow Q^{\prime}
\]
\end{document}, entonces %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
Q
\]
\end{document} es la post-condición más fuerte de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
S
\]
\end{document} con respecto a %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document}.

En el ejemplo de arriba, se puede comprobar que

* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  x=10\Rightarrow\mbox{true}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  x=10\Rightarrow x>0
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  x=10\Rightarrow x=10\parallel x=5
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  x=10\Rightarrow x=10
  \]
  \end{document}

es decir, que %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{x=10\}
\]
\end{document} es la post-condición más fuerte de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x:=x*2
\]
\end{document} con respecto a %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{x=5\}
\]
\end{document}.

## La precondición más débil

Se presentan las siguientes tripletas de Hoare válidas

* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x=5\,\&\&\,y=10\}\,z:=x/y\,\{z<1\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{x<y\,\&\&\,y>0\}\,z:=x/y\,\{z<1\}
  \]
  \end{document}
* %FontSize=10
  %TeXFontSize=10
  \documentclass{article}
  \pagestyle{empty}
  \begin{document}
  \[
  \{y\neq 0\,\&\&\,x/y<1\}\,z:=x/y\,\{z<1\}
  \]
  \end{document}

**Definición (Pre-condición más débil)**

Si %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{P\}S\{Q\}
\]
\end{document} y para toda %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P^{\prime}
\]
\end{document} tal que %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{P^{\prime}\}S\{Q\}
\]
\end{document}, %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P^{\prime}\Rightarrow P
\]
\end{document}, entonces %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} es la precondición más débil de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
S
\]
\end{document} con respecto a %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
Q
\]
\end{document}.

Ejemplo de especificación

El programa factorial *Fac1*

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
y=1;
z=0;
while(z!=x){
     z=z+1;
     y=y*z;
}
\end{verbatim}
\end{document}

puede ser especificado con la tripleta de Hoare

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\{x\geq 0\}\,Fac1\,\{y=x!\}
\]
\end{document}

**Corrección parcial**

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\models_{par}\langle\Phi\rangle P\langle \Psi\rangle
\]
\end{document} se cumple si:

si la ejecución de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} comienza en un estado donde la precondición %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\Phi
\]
\end{document} se cumple, y la ejecución termina, entonces la post-condición %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\Psi
\]
\end{document} se cumple en el estado final.

**Corrección total**

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\models_{tot}\langle \Phi\rangle P\langle \Psi\rangle
\]
\end{document} se cumple si:

si la ejecución de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} comienza en un estado donde la pre-condición %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\Phi
\]
\end{document} se cumple, entonces la ejecución termina, y la post-condición %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\Psi
\]
\end{document} se cumple en el estado final.

Ejemplo

¿Cuándo se cumple %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\models_{par}\langle true\rangle P \langle false\rangle
\]
\end{document}? Cuando la ejecución de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
P
\]
\end{document} no termina, independientemente del estado de inicio.

Ejemplo

¿Cuándo se cumple %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\models_{tot}\langle true\rangle P\langle false\rangle
\]
\end{document}? ¡Nunca!

Variables lógicas

Puede el programa factorial *Fac2*

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
y=1;
while(x!=0) {
     y=y*x;
     x=x-1;
}
\end{verbatim}
\end{document}

ser especificado con la tripleta de Hoare

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle x\geq 0\rangle\,Fac2\langle\, y=x!\rangle
\]
\end{document}

Variables lógicas

Necesitamos variables adicionales, las así llamadas variables lógicas, para capturar como las variables se relacionan a los valores iniciales. Esas variables son consideradas universalmente cuantificadas en una tripleta de Hoare. El programa factorial *Fac2*

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
y=1;
while(x!=0){
    y=y*x;
    x=x-1;
}
\end{verbatim}
\end{document}

Puede ser especificado con la tripleta de Hoare

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle x\geq 0\wedge x=x_{0}\rangle\,Fac2\,\langle y=x_{0}!\rangle
\]
\end{document}

De la especificación debe ser claro cómo se puede usar al programa –sin conocer el código del programa--.

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle x\geq 0\wedge x=x_{0}\rangle\,Fac2\,\langle y=x_{0}!\rangle
\]
\end{document}

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x\geq 0
\]
\end{document} entonces el programa termina

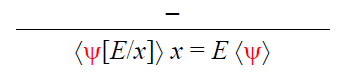
%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x=x_{0}
\]
\end{document} fija el valor inicial de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x
\]
\end{document}

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
y=x_{0}!
\]
\end{document} relaciona el valor final de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
y
\]
\end{document} al valor inicial de %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
x
\]
\end{document}.

Lógica de Hoare

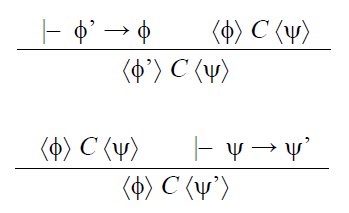
Consiste de un conjunto de reglas para razonar sobre las tripletas de Hoare para verificar programas (corrección parcial). También incluye demostraciones en la forma de árboles de demostración, o las así llamadas “tablas”. Tales reglas reducen la validez de las tripletas de Hoare a la validez de fórmulas lógicas de predicado sobre aritmética.

Regla de asignación

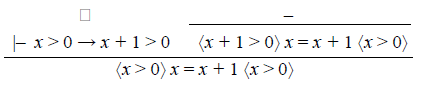


“propaga” la post-condición hacia atrás.

Reglas implicadas

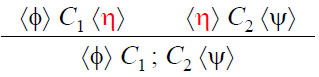


Ejemplo de demostración



Una obligación de demostración: %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\vdash x>0\rightarrow x+1>0
\]
\end{document} a ser demostrada en un sistema de demostración “externo”.

Regla de composición secuencial



Introduce una nueva afirmación %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\eta
\]
\end{document} en el punto de control precedente %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
C_{2}
\]
\end{document}.

Ejemplo

¿Qué hace este programa?

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
z=x;
x=y;
y=z;
\end{verbatim}
\end{document}

y ¿cómo puede ser especificado?

Ejemplo *Swap*

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
z=x;
x=y;
y=z;
\end{verbatim}
\end{document}

Puede ser especificado con %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle x=x_{0}\wedge y=y_{0}\rangle\,Swap\,\langle  x=y_{0}\wedge y=x_{0}\rangle
\]
\end{document}.

Tabla de demostración

Se puede dar una demostración de la corrección del programa en forma de tabla. Las demostraciones de corrección de programas pueden ser presentadas como programas comentados (o anotados) donde los comentarios son afirmaciones asociadas con puntos de control.

La demostración en forma de tabla

|  |  |
| --- | --- |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=x_{0}\wedge y=y_{0}\rangle \] \end{document} | Precondición |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle y=y_{0}\wedge x=x_{0}\rangle \] \end{document} | Implicado |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} z=x; \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle y=y_{0}\wedge z=x_{0}\rangle \] \end{document} | Asignación |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} x=y; \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=y_{0}\wedge z=x_{0}\rangle \] \end{document} | Asignación |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} y=z; \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=y_{0}\wedge y=x_{0}\rangle \] \end{document} | Asignación |

Tablas

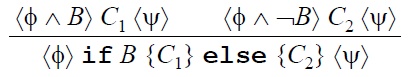
Una tabla es un programa anotado. Una interpretación operacional:

Si la ejecución comienza en un estado donde la primera anotación (pre-condición) se cumple, entonces cada vez que la ejecución alcanza un punto de control, todas las afirmaciones asociadas con este punto de control se cumplen.

Demostraciones en forma de tabla

Una demostración en forma de tabla es un programa anotado con al menos una afirmación en cada punto de control, donde las anotaciones coinciden con las reglas (patrones). El proceso de demostración puede ser visto como una completación de la anotación inicial de “entrada”. Las afirmaciones asociadas con el mismo punto de control dan lugar a obligaciones de demostración.

Regla Si



Ejemplo

¿Qué hace el programa?

%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
if(x>0){
  y=x;
} else {
  y=-x;
}
\end{verbatim}
\end{document}

Y ¿cómo puede este ser especificado?

Programa *Abs*

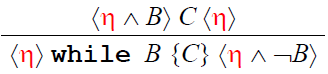
%FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{verbatim}
if(x>0){
  y=x;
} else {
  y=-x;
}
\end{verbatim}
\end{document}

puede ser especificado con %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle x=x_{0}\rangle\,Abs\,\langle y=|x_{0}|\rangle
\]
\end{document}.

Tabla de demostración

|  |  |
| --- | --- |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=x_{0}\rangle \] \end{document} | Pre-condición |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} if(x>0){ \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=x_{0}\wedge x>0\rangle \] \end{document} | If |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=|x_{0}|\rangle \] \end{document} | Implicado |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} y=x; \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle y=|x_{0}|\rangle \] \end{document} | Asignación |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} } else { \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle x=x_{0}\wedge\lnot(x>0)\rangle \] \end{document} | If |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle-x=|x_{0}|\rangle \] \end{document} | Implicado |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} y=-x; \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle y=|x_{0}|\rangle \] \end{document} | Asignación |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \begin{verbatim} } \end{verbatim} \end{document} |  |
| %FontSize=10 %TeXFontSize=10 \documentclass{article} \pagestyle{empty} \begin{document} \[ \langle y=|x_{0}|\rangle \] \end{document} | Post-condición |

Regla while parcial



Ejemplo

Demostrar %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle x<0\rangle\,\mbox{\tt while(x!=0)\ }\,\{\mbox{\tt x=x-1;}\}\langle\mbox{\tt false}\rangle
\]
\end{document}.

Demostrar %FontSize=10
%TeXFontSize=10
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\langle\mbox{\tt true}\rangle\,\mbox{\tt while(x!=0)\ }\,\{\mbox{\tt x=x-1;}\}\langle\mbox{\tt x=0}\rangle
\]
\end{document}.