

Programación Lógica

Programación imperativa → programa es secuencia de instrucciones que una computadora ejecuta para resolver un problema

Programación Lógica (declarativa) → programa especifica conocimiento acerca del problema a resolver; se hace pregunta y se deja que el sistema deduzca la respuesta usando procesos basados en la lógica matemática

Ejemplo: Buscar un elemento x en una lista

x pertenece a [x | _]

x pertenece a [_ | Resto] si x pertenece a Resto

x pertenece a lista?

Ciencias de la Computación II - Filminas de Clase – Mg. Virginia Mauco – Facultad Cs. Exactas – UNCPBA - 2009

Programación Lógica

PROLOG: lenguaje de programación lógica muy usado

- Programa es un conjunto de cláusulas de programa.

- Pregunta se trata como una sentencia a ser demostrada; la sentencia se niega y el motor de Prolog intenta refutar la negación.

- si la refutación tiene éxito, la pregunta se responde positivamente

- si no, la pregunta se responde negativamente

- Al construir la refutación, el motor trata de ver si la sentencia implícita en la pregunta es consistente o no con la información que ya existe en las cláusulas del programa.

Ciencias de la Computación II - Filminas de Clase – Mg. Virginia Mauco – Facultad Cs. Exactas – UNCPBA - 2009

Programación Lógica

CLAUSULAS DE PROGRAMA	Cláusulas de Horn	Sintaxis PROLOG
	Regla $C_1 \wedge C_2 \wedge \dots \wedge C_n \rightarrow C$	$C :- C_1, C_2, \dots, C_n.$
	Hecho C	$C.$
	Objetivo $\neg(C_1 \wedge C_2 \wedge \dots \wedge C_n)$	$?- C_1, C_2, \dots, C_n.$

- Lógica de Programas es un conjunto de cláusulas de programa
- Es satisfacible (tiene Modelos de Herbrand).

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Programación Lógica

Ejemplo:

$\left. \begin{array}{l} \text{ruta(a, b)} \quad \text{ruta(b, a)} \quad \text{ruta(a, d)} \quad \text{ruta(d, a)} \\ \text{ruta(b, c)} \quad \text{ruta(c, b)} \quad \text{ruta(c, d)} \quad \text{ruta(d, c)} \end{array} \right\} \text{Hechos}$	$\exists u \text{ camino}(u, d)$ Pregunta
$\left. \begin{array}{l} \forall x \forall y (\text{ruta}(x, y) \rightarrow \text{camino}(x, y)) \\ \forall w \forall z \forall t (\text{ruta}(w, t) \wedge \text{camino}(t, z) \rightarrow \text{camino}(w, z)) \end{array} \right\} \text{Reglas}$	

Programa Prolog
+
Pregunta

ruta(a, b).
ruta(b, a).
ruta(a, d).
ruta(d, a).
ruta(b, c).
ruta(c, b).
ruta(c, d).
ruta(d, c).
camino(X, Y) :- ruta(X, Y).
camino(W, Z) :- ruta(W, T), camino(T, Z).
?- camino(U, d).

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Enfoques para determinar si un programa es correcto

✓ **Testing**: seleccionar un conjunto de datos de entrada para determinar si los resultados producidos por el programa con esos datos coinciden o no con los valores esperados.

Asegurar que un programa es correcto → analizar con todos los valores posibles de datos de entrada (imposible cuando el c.jto. es infinito)

Testing sólo puede mostrar la presencia de errores y no su ausencia.

✓ **Prueba formal de programas**: Especificar el comportamiento de cada sentencia del programa formalmente y probar que el programa hace lo que su especificación dice.

Asegura que el programa es correcto con respecto a una especificación para todas las entradas

Dos problemas importantes: 1) la manipulación lógica puede ser tediosa y propensa a errores; 2) la prueba solamente muestra que el programa implementa la especificación correctamente. No hay certeza de que la especificación describe lo que el usuario realmente desea.

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

Si un programa usa n variables (x_1, x_2, \dots, x_n) el estado s es una tupla de valores (X_1, X_2, \dots, X_n) donde X_i es el valor de la variable x_i .

Ejemplo:

Dado $s = (x, y) = (7, 8)$ el resultado de ejecutar la sentencia $x := 2*y+1$ es el estado $s' = (x, y) = (17, 8)$

Definición:

Sea U el conjunto de todos los posibles estados del programa y sea $S \subseteq U$. P_S , el predicado característico de S , tal que $S = \{ s' \in S / P_S(s') \}$ (predicado que solamente es satisfecho por los estados del conjunto S)

Ejemplo: Se quiere probar $x \leq 7$ después de ejecutar $x := 2*y+1$. Esto es verdadero si $2*y+1 \leq 7$ antes de ejecutar la sentencia, es decir $y \leq 3$.

$$\{ y \leq 3 \} \quad x := 2*y+1 \quad \{ x \leq 7 \}$$

$\{ y \leq 3 \}$ y $\{ x \leq 7 \}$ se denominan **aserciones**.

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

Una aserción es una fórmula del cálculo de predicados ubicada en un programa, que es verdadera cada vez que el programa pasa por ese punto.

Definición:

$\{P\} S \{Q\}$, P y Q son aserciones llamadas pre-condición y post-condición respectivamente y S es un fragmento de programa.

$\{P\} S \{Q\}$ se denomina **especificación formal de S** y se interpreta como

Si la ejecución de S empieza en un estado caracterizado por P y S termina, entonces terminará en un estado caracterizado por Q.

S se dice **parcialmente correcto** con respecto a **P** y **Q**.

Si además se garantiza que **S termina**, S es **totalmente correcto** con respecto a **P** y **Q**.

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Ejemplos: programas especificados en términos de pre y post condiciones.

1) Dados dos números naturales a y b, $b \neq 0$, encontrar el cociente entre a y b.

P: $\{ a, b \in \mathbb{N} \wedge b > 0 \}$

Q: $\{ a = b*c + r \wedge 0 \leq r < b \wedge c \geq 0 \}$

2) Encontrar el valor máximo de un arreglo de n números enteros.

P: $\{ \forall i: 1 \leq i \leq n: (\text{integer}(a[i]) \wedge a[i] = A[i]) \}$

(A referencia el valor de los elementos del arreglo a antes de iniciar la ejecución)

Q: $\{ \exists i: 1 \leq i \leq n: (\text{max} = a[i] \wedge \forall j: 1 \leq j \leq n: (a[j] \leq \text{max} \wedge a[j] = A[j])) \}$

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Ejemplo:

$\{ y \leq 3 \}$

$x := 2*y+1$

$\{ x \leq 7 \}$

$\{ y \leq 3 \}$ no es la única pre-condición que hace verdadera la post-condición

Otra pre-condición podría ser $\{ y = 1 \vee y = 3 \}$

Se desea determinar como pre-condiciones aquellas fórmulas que describan tantos estados como sea posible. Esto se hace eligiendo el predicado menos restrictivo posible.

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

Una fórmula A es más débil que una fórmula B si $B \rightarrow A$ (B es más fuerte que A). Como A es más débil que B, A describe más estados que B.

T (true) y F(false) son casos especiales de aserciones

T es verdadero en todo estado \rightarrow es la más débil entre las fórmulas

F es más fuerte que toda fórmula.

Ejemplo:

$y \leq 3$ es más débil que $(y = 1 \vee y = 3)$ porque $(y = 1 \vee y = 3) \rightarrow y \leq 3$

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

Para todo fragmento de programa S y fórmula Q, se define la pre-condición más débil $R = wp(S, Q)$ tal que $\{R\} S \{Q\}$ es verdadero.

Es decir, $wp(S, Q)$ representa todos los estados tal que la ejecución de S que comenzó en cualquiera de ellos, si termina, termina en un estado que satisface Q.

La notación $\{P\} S \{Q\}$ es otra notación para $P \rightarrow wp(S, Q)$

wp se denomina **predicado transformador**

(para cualquier fragmento de programa define una transformación de un predicado post-condición en un predicado pre-condición)

Para probar formalmente un programa se comenzará desde la post-condición y se trabajará "hacia atrás" para verificar cada una de las sentencias individuales del programa.

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Predicado wp para un lenguaje de programación con sentencias de asignación, if B then S₁ else S₂, y while B do S

Definición:

$wp(x = e, Q) = Q \{x/e\}$ e es una expresión

Ejemplo

$wp(y = b, y \geq 0) = (y \geq 0) \{y/b\} = b \geq 0$

$\{b \geq 0\}$

$y = b$

$\{y \geq 0\}$

$wp(x = x+1, x = 5) = (x = 5) \{x/x+1\} = (x+1 = 5) = (x = 4)$

$\{x = 4\}$

$x = x+1$

$\{x = 5\}$

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

$$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$$

Ejemplo: Calcular la pre-condición más débil del siguiente fragmento de programa:

```
x = x+1;
y = y+2
{x < y}
```

$$\begin{aligned} wp(x = x+1; y = y+2, x < y) &= wp(x = x+1, wp(y = y+2, x < y)) \\ &= wp(x = x+1, x < y+2) \\ &= x < y+1 \end{aligned}$$

Esto se puede escribir

```
{ x < y+1 }
x = x+1;
{ x < y+2 }
y = y+2
{ x < y }
```

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

$$wp(\text{if } B \text{ } S_1 \text{ else } S_2, Q) = (B \rightarrow wp(S_1, Q)) \wedge (\neg B \rightarrow wp(S_2, Q))$$

Ejemplo: Calcular la pre-condición más débil del siguiente fragmento de programa:

```
if y = 0
  x = 0
else
  x = y - 1
{ x = y }
```

$$\begin{aligned} wp(\text{if } y = 0 \text{ } x = 0 \text{ else } x = y-1, x = y) &= \\ &= (y = 0 \rightarrow wp(x = 0, x = y)) \wedge (\neg(y = 0) \rightarrow wp(x = y-1, x = y)) \\ &= (y = 0 \rightarrow y = 0) \wedge (y \neq 0 \rightarrow y-1 = y) \\ &= \quad T \quad \wedge (y \neq 0 \rightarrow F) \\ &= (y \neq 0 \rightarrow F) \\ &= (\neg(y \neq 0) \vee F) \\ &= \neg(y \neq 0) \\ &= (y = 0) \end{aligned}$$

pre-condición más débil es P: { y = 0 }

Ciencias de la Computación II - Filminas de Clase - Mg. Virginia Mauco - Facultad Cs. Exactas - UNCPBA - 2009

Verificación de Programas

Definición:

Un predicado I se llama invariante de una sentencia S si $wp(S, I) = I$

Definición: Para probar la corrección de una sentencia **while B do S**, con post-condición Q se deberá:

- 1) Calcular el predicado invariante I .
- 2) Probar que $I \wedge B \rightarrow wp(S, I)$
- 3) Probar que I vale antes de iterar
- 4) Probar que $I \wedge \neg B \rightarrow Q$ (Q es la post-condición de la sentencia while B S)