

Universidad Nacional de Jujuy Facultad de Ingeniería

Introducción a la Programación Programación I

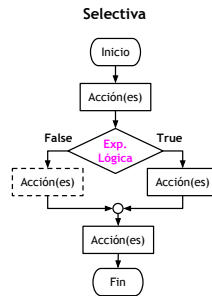
Estructura selectiva

Contenido

- Introducción
- Expresiones y operadores lógicos
- Precedencia combinada de operadores
- Tablas de verdad
- Equivalencia lógica
- Estructura de control selectiva
- Tablas de decisión
- Ejemplos

Introducción

Permite realizar una bifurcación en la secuencia de ejecución del algoritmo evaluando la expresión lógica. Se toma la decisión de ejecutar un conjunto de acciones o no hacerlo.



Estructura selectiva ≡ Estructura condicional ≡ Estructura alternativa

Expresión lógica

Operando1 **operador** Operando2

p **operador** q

Los operandos pueden ser:

- ❑ Constantes lógicas
- ❑ Variables lógicas
- ❑ Expresiones relacionales
- ❑ Funciones que devuelven valores lógicos

Los operadores lógicos pueden ser:

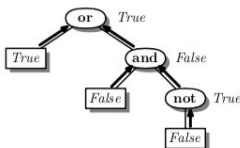
- ❑ Negación
- ❑ Conjunción
- ❑ Disyunción

Operadores lógicos

Operación	Operador	Aridad	Asociatividad	Precedencia
Negación	not	Unario	—	alta
Conjunción	and	Binario	Por la izquierda	media
Disyunción	or	Binario	Por la izquierda	baja

Tabla 2.2: Aridad, asociatividad y precedencia de los operadores lógicos.

(True) or (False) and not (False)



Precedencia combinada de operadores

En Python la precedencia de las operaciones es en este orden: aritméticas, relacionales y lógicas.

Operadores	Descripción
**	Exponenciación
+x, -x	Cambio de signo
*, /, //, %	Multiplicación, división, resto
+, -	Adición y sustracción
<, <=, >, >=, ==, !=, is, is not, in, not in	Operadores relacionales, identidad, pertenencia
not	Negación lógica
and	Conjunción lógica
or	Disyunción lógica

Tablas de verdad

and		
operandos		resultado
izquierdo	derecho	
True	True	True
True	False	False
False	True	False
False	False	False

not	
operando	resultado
True	False
False	True

or		
operandos		resultado
izquierdo	derecho	
True	True	True
True	False	True
False	True	True
False	False	False

xor (disyunción exclusiva)
(True) xor (True)
False

Equivalencia
(p or q) and not (p and q)

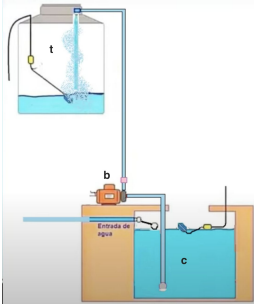
Ejemplo: resolver una exp. lógica

Si $x = 1$; $k = \text{"UNJu"}$; $w = 2$ ¿cuál es el orden de resolución y el resultado de la siguiente expresión lógica?

	True	and	((x==1)	or	not	(k<"UNSa")	and	(w==5)
	6	1	4	3	2	7	5	
1			True					
2					True			
3				False				
4			True					
5							False	
6	True							
7							False	

Ejemplo: plantear una exp. lógica

La bomba b se encenderá cuando haya agua en la cisterna c y el tanque t no esté lleno.



c	t	b
True	True	False
True	False	True
False	True	False
False	False	False

$b = c \text{ and not } t$

Equivalencia lógica

Equivalencia not

not $p != q$	$p == q$
not $p == q$	$p != q$
not $p > q$	$p <= q$
not $p >= q$	$p < q$

Operadores

En lógica	En Python
\neg	not
\wedge	and
\vee	or

Equivalencia lógica

Equivalencia	Ley	Equivalencia	Ley
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identidad	$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Asociación
$p \vee T \equiv T$ $p \wedge F \equiv F$	Dominación	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distribución
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotencia	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan
$\neg(\neg p) \equiv p$	Doble negación	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorción
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Conmutación	$p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$	Tautología Contradicción

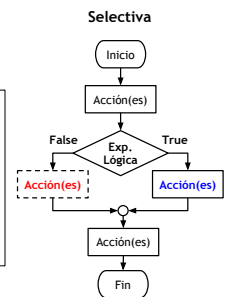
La estructura selectiva

simple

if Exp. lógica:
Acción(es)

compuesta

if Exp. lógica:
Acción(es)
else:
Acción(es)



Selectiva encadenada: if...elif...else...

Es útil cuando se tienen más de dos alternativas

encadenada

```
if Exp. lógica1:
    Acción(es)1
elif Exp. lógica2:
    Acción(es)2
elif Exp. lógica3:
    Acción(es)3
else:
    Acción(es)4
```

compuesta (anidada)

```
if Exp. lógica1:
    Acción(es)1
else:
    if Exp. lógica2:
        Acción(es)2
    else:
        if Exp. lógica3:
            Acción(es)3
        else:
            Acción(es)4
```

Ejemplo

Diseñar tres algoritmos que muestren si x es cero, positivo o negativo y usen respectivamente la sentencia selectiva: simple, compuesta y encadenada ¿Cuál es el mejor diseño?

simple

```
x = int(input('x: '))

if x == 0:
    print('cero')

if x > 0:
    print('positivo')

if x < 0:
    print('negativo')
```

compuesta

```
x = int(input('x: '))

if x == 0:
    print('cero')
else:
    if x > 0:
        print('positivo')
    else:
        print('negativo')
```

encadenada

```
x = int(input('x: '))

if x == 0:
    print('cero')
elif x > 0:
    print('positivo')
else:
    print('negativo')
```

Tablas de decisión

- Es una herramienta que sintetiza procesos complejos en los cuales se dan un conjunto de condiciones y un conjunto de acciones. Es una matriz formada por:

Lista de condiciones Lista de acciones	Reglas de decisión Acciones a realizar
Lista de condiciones (Zona 1)	Cumplimiento de condiciones (Zona 2)
Lista de acciones (Zona 3)	Acciones a realizar (Zona 4)

Ejemplo: Tabla de decisión de la fórmula

$$f = \sqrt{\frac{a-b}{c}}$$

Condiciones/Acciones	R1	R2	R3
C1. $c \neq 0$	True	True	False
C2. $(a-b)/c \geq 0$	True	False	---
A1. Imposible			✓
A2. Solución real	✓		
A3. Solución Imaginaria		✓	

Python: Calcular la fórmula

```
a = float(input('a:'))
b = float(input('b:'))
c = float(input('c:'))

if c != 0 and (a-b)/c >= 0:
    d = (a-b)/c
    f = d**0.5
    print('Real:',f)

if c != 0 and not (a-b)/c >= 0:
    d = (a-b)/c
    f = (-d)**0.5
    print('Imaginaria:',f,'i')

if not c != 0:
    print('Imposible')
```

```
a = float(input('a:'))
b = float(input('b:'))
c = float(input('c:'))

if c != 0:
    d = (a-b)/c
    if d >= 0:
        f = d**0.5
        print('Real:',f)
    else:
        f = (-d)**0.5
        print('Imaginaria:',f,'i')
else:
    print('Imposible')
```

Bibliografía y enlaces

- Introducción a la Programación con Python. Sentencia condicional if (Sección 4.1.2)
- Equivalencia lógica: https://es.wikipedia.org/wiki/Equivalencia_l%C3%B3gica
- Ejercicios: Lógica Equivalente, Tautología, mContradicción: <https://www.zweigmedia.com/MundoReal/logic/logic2.html>
- Tablas de decisión: <http://eve-ingsistemas-u.blogspot.com/2012/05/tablas-de-decision-parte-1.html>