

CMC-Tema-4-Maquinas-de-registros...



Anónimo



Computabilidad y complejidad



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática
Universidad Politécnica de Valencia



**Que no te escriban poemas de amor
cuando terminen la carrera**



*(a nosotros por
suerte nos pasa)*

WUOLAH

WUOLAH

Oh Wuolah wuolithah
Tu que eres tan bonita

El modelo RASP

La *RASP* (*Random Access Stored Program Machine*) es como una RAM pero con el programa a ejecutar almacenado en la propia memoria; así su arquitectura se corresponde con la de Von Neumann.

Es una máquina universal que puede simular a cualquier RAM y en consecuencia realizar todas sus computaciones. Es en este sentido similar a la máquina de Turing Universal (véase el tema 5).

La Máquina Contador

El modelo de la *Máquina Contador* es una restricción de la *RAM*. Sólo dispone de un número finito de registros (que naturalmente pueden variar de una máquina a otra) y no tiene instrucciones con direccionamiento indirecto.

- ➔ El resto de esta sección lo dedicaremos a su estudio exponiendo diversos ejemplos operativos
- Este modelo de máquina, aunque presente restricciones, tiene el poder computacional de la *RAM* y de la *RASP*.

Su conjunto básico de instrucciones es el siguiente:

- **suc(i)**, su semántica es la siguiente: incrementa en una unidad el contenido de **R_i**, es decir:

$$[i] \leftarrow [i] + 1$$

La ejecución continúa en la siguiente instrucción del programa, si no, termina.

- **pre(i, k)**, su semántica es la siguiente:
 - Si $[i] > 0$, entonces $[i] \leftarrow [i] - 1$; la ejecución continúa en la siguiente instrucción del programa si existe, si no termina;
 - En otro caso, la ejecución continúa en la **k-ésima** instrucción si existe, si no termina.

Este par de instrucciones, junto con la existencia de un registro predefinido con contenido nulo son suficientes para realizar cualquier computación.

Seguidamente, veremos algunos fragmentos de programa que ejecutan cálculos sencillos que ilustran la operatividad de este modelo.

En lo inmediato que sigue **R_c** tiene asignado permanentemente el valor **nulo**.

[i] ← 0

n : pre (i, n + 2)

n + 1 : pre (c, n)

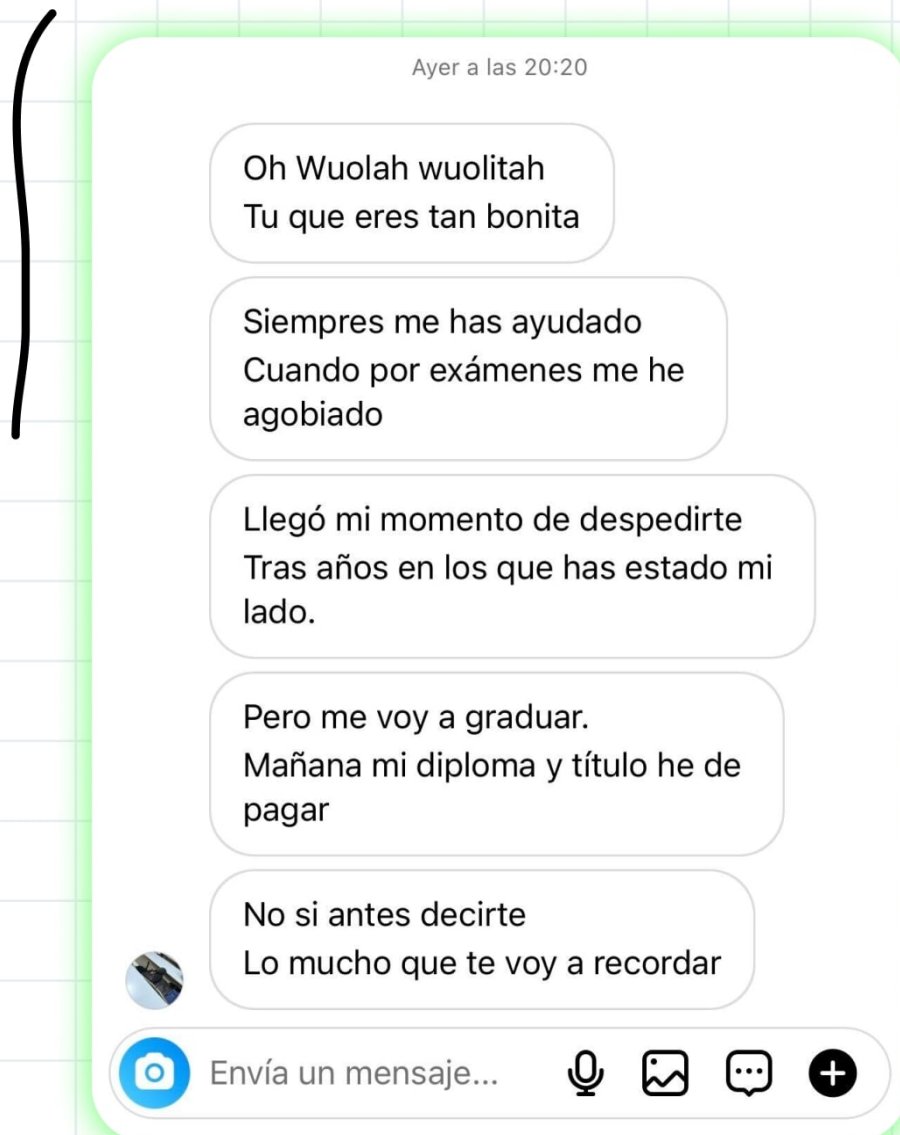
[i] ← [i] + [j] AND [j] ← 0 AND R_i != R_j

n : pre (j, n + 3)

n + 1 : suc (i)

n + 2 : pre (c, n)

**Que no te escriban poemas de amor
cuando terminen la carrera ▶▶▶▶▶▶▶▶**
(a nosotros por suerte nos pasa) 😊

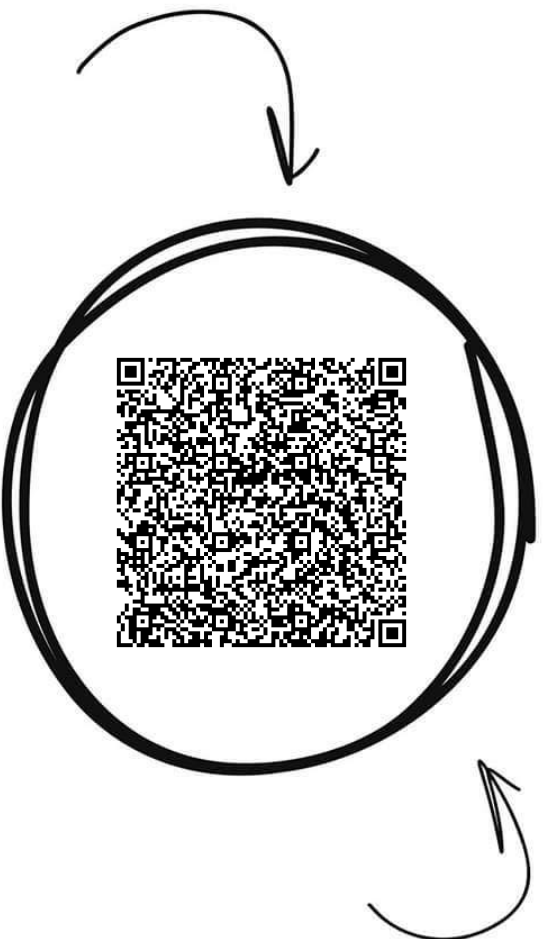


WUOLAH

Computabilidad y complejidad



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Note bank of the UPV

WUOLAH



- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes
- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR

Para evitar el tener que mantener un registro predefinido permanentemente con valor nulo, añadiremos al conjunto de instrucciones la instrucción:

- **goto (n)** : su semántica es la siguiente: ejecutar incondicionalmente la instrucción n-ésima si ésta existe, si no terminar la ejecución.

→ Las instrucciones **suc(i)**, **pre(i, k)** y **goto(n)** son suficientes para realizar cualquier computación.

Los ejemplos anteriores, utilizando la sentencia **goto(n)**, quedan como sigue:

- **[i] ← 0**
n : pre (i, n + 2)
n + 1 : goto (n)

Este Código puede simbólicamente referenciarse mediante la macroinstrucción **cer (i)**

- **[i] ← [i] + [j] AND [j] ← 0 AND Ri != Rj**
n : pre (j, n + 3)
n + 1 : suc (i)
n + 2 : goto (n)
- **[i] ← k**
cer (i)
n + 1 : suc (i)
...
n + k : suc (i)

Este Código puede simbólicamente referenciarse mediante la macroinstrucción **asi (k,i)** (asignar al registro i el valor k).

- **[i] ← j AND Ri != Rj**
Sea Rk un resigtro auxiliar de cáculo distinto de los anteriores: Ri != Rk AND Rj != Rk

```
cer (i)
cer (j)
n: pre (j, n+4)
n+1: suc (i)
n+2: suc (k)
n+3: goto (n)
n+4: pre (k, n+7)
n+5: suc (j)
n+6: goto (n+4)
```

Este Código puede simbólicamente referenciarse mediante la macroinstrucción **cop (j,i)** (copia el contenido del registro j en el registro i).

WUOLAH

Oh Wuolah wuolilah
Tu que eres tan bonita

- $[r] \leftarrow [p] / [q]$ (División entera)

Sean Ri, Rj, Rk y Rm cuatro registros auxiliares de cálculo distintos de los anteriores.

<div style="border: 1px solid black; padding: 2px; display: inline-block;">cer (m)</div>	n+5: goto (n+2)
<div style="border: 1px solid black; padding: 2px; display: inline-block;">cer (k)</div>	n+6: suc (m)
<div style="border: 1px solid black; padding: 2px; display: inline-block;">cop (p, i)</div>	n+7: pre (k, n+2)
<div style="border: 1px solid black; padding: 2px; display: inline-block;">cop (q, j)</div>	n+8: suc (j)
n: pre (j, error)	n+9: goto (n+7)
n+1: suc (j)	n+10: cop (m, r)
n+2: pre (j, n+6)
n+3: pre (i, n+10)	
n+4: suc (k)	error:

Este Código puede simbólicamente referenciarse mediante la macroinstrucción **div (p,q,r)**

- **if ([p] =< [q]) then go to m1
 else go to m2**

Sean Ri y Rj dos registros auxiliares de cálculo distintos de los anteriores.

<div style="border: 1px solid black; padding: 2px; display: inline-block;">cop (p, i)</div>
<div style="border: 1px solid black; padding: 2px; display: inline-block;">cop (q, j)</div>
n: pre (i, m1)
n+1: pre (j, m2)
n+2: goto (n)

Este Código **puede simbólicamente referenciarse** mediante la macroinstrucción **mei (p, q, m1, m2)**

- **if ([i] = [j]) then go to m1
 else go to m2**

Sean Ri y Rj dos registros auxiliares de cálculo distintos de los anteriores.

<div style="border: 1px solid black; padding: 2px; display: inline-block;">cop (p, i)</div>
<div style="border: 1px solid black; padding: 2px; display: inline-block;">cop (q, j)</div>
n: pre (i, n+3)
n+1: pre (j, m2)
n+2: goto (n)
n+3: pre (j, m1)
n+4: goto (m2)

Este Código puede simbólicamente referenciarse mediante la macroinstrucción **igu (p, q, m1, m2)**

- **while ([i] > 0)** **CÓDIGO** (No es una macroinstrucción)

- Si utilizamos *while* --> Tenemos que preocuparnos de que dentro de la parte de **CÓDIGO** modificamos el valor de *i*.

```
n: pre(i,m+1)
n+1: suc(i)
.....
CÓDIGO
.....
m: goto(n)
```

Equivalencias entre modelos

Sea una función (parcial) de la forma:

$$g: N^m \longrightarrow N^k$$

Diremos que la función *g* es:

- *RAM Computable* si y sólo si existe una RAM que la compute.
- *RASP Computable* si y sólo si existe un programa para la máquina RASP con el que esta máquina puede computarla
- *MC Computable* si y sólo si existe una máquina contador que la compute.

Teorema Sea una función (parcial) de la forma:

$$g: N^m \longrightarrow N^k$$

Los siguientes enunciados son equivalentes:

- *g* es *RAM* computable
- *g* es *RASP* computable
- *g* es *MC* computable
- *g* es *Turing* computable

Sea una función (parcial) de la forma:

$$g: N \longrightarrow N$$

Diremos que la función *g* es: *MC2 Computable* si y solo si existe una máquina contador con dos contadores que la compute.

Teorema Sea una función (parcial) de la forma:

$$g: N \longrightarrow N$$

Los siguientes enunciados son equivalentes:

- *g* es *MC* computable
- *g* es *MC2* computable

Observación: El poder computacional del modelo de la máquina contados con dos contadores también equivale al de los modelos:

- RAM
- RASP
- Turing