

# Electrónica Digital

Ingeniería Informática – FICH, UNL  
Leonardo Giovanini



## Máquinas abstractas

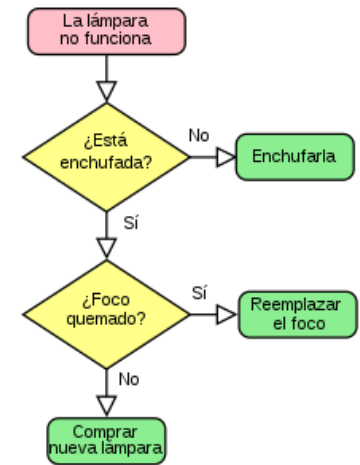
En esta se estudiarán los siguientes temas:

- Algoritmos;
- Máquina abstractas;
- Autómatas
  - *Automatas finito;*
  - *Autómata con pila;*
  - *Autómata con procesador de dato;*
  - *Maquinas de Turing.*
- Máquina con registros
  - *Máquina de conteo;*
  - *Máquina de pila;*
  - *Máquina de acceso aleatorio;*
  - *Máquina de acceso aleatorio con programa almacenado.*

Un **algoritmo** es **una secuencia finita de instrucciones**, o reglas, **no ambiguas, ordenadas, finitas e implementables**, que se usa para resolver problemas, realizar un cálculos y razonamientos automatizados, procesar datos y llevar a cabo cualquier tarea.

En general, no existe ningún consenso definitivo en cuanto a la definición formal de algoritmo. La parte común en todas las definiciones se puede resumir en tres propiedades:

- **Tiempo secuencial** – un algoritmo define una secuencia de estados computacionales por cada entrada válida (la entrada son los datos que se le suministran al algoritmo antes de comenzar);
- **Estado abstracto** – cada estado computacional se describe formalmente utilizando una estructura de primer orden y cada algoritmo es independiente de su implementación de manera que en un algoritmo las estructuras de primer orden son invariantes bajo isomorfismo;
- **Exploración acotada** – la transición de un estado al siguiente queda completamente determinada por una descripción fija y finita; es decir, entre cada estado y el siguiente solamente se puede tomar en cuenta una cantidad fija y limitada de términos del estado actual;
- **Aritmetizabilidad** – solo las operaciones calculables están disponibles en el paso inicial.



La descripción de un algoritmo se hace en tres niveles:

- **Alto nivel** – se explica el algoritmo con ilustraciones y omitiendo detalles;
- **Formal** – se usan formas estructuradas e independientes de las técnicas de implementación para describir la secuencia de tareas;
- **Implementación** – se selecciona las técnicas específicas para implementar el algoritmo.

Los algoritmos pueden ser expresados de muchas maneras, incluyendo al lenguaje natural, pseudocódigo, diagramas de flujo, diagramas de estados, sistemas formales, lenguajes de programación, circuitos electrónicos y dispositivos mecánicos entre otros.

Para estudiar y cuantificar su **complejidad computacional**, es decir los recursos necesarios para resolver algoritmos (operaciones, almacenamiento, comunicaciones y tiempo de ejecución, etc) se introducen los **modelos de computación**.

Hay muchos modelos, que se diferencian en las operaciones admisibles y su costo. Se los puede clasificar en dos categorías:

- **Máquinas abstractas** – usada en pruebas de computabilidad y de los límites superiores en la complejidad computacional de algoritmos, y
- **Árboles de decisión** – usado en las pruebas de los límites más bajos en la complejidad computacional de problemas algorítmicos.

Una máquina abstracta es un **modelo de un sistema informático**, considerado como hardware y/o software, construido para **realizar un análisis** de cómo funciona el sistema. El modelo consiste en entradas, salidas y las operaciones que se pueden realizar.

La abstracción de los procesos informáticos se utiliza en:

- **Informática** para analizar la **complejidad de algoritmos**;
- **Ingeniería de la computación** para **calcular la cantidad de recursos** (tiempo, memoria, etc.) necesarios para implementar un algoritmo particular; e
- **Ingeniería de la computación** para **implementar los sistemas** al definir las posibles formas en que se conectan los recursos físicos (arquitectura).

Las **definiciones más complejas** crean máquinas abstractas con conjuntos de instrucciones completos, registros y modelos de memoria.

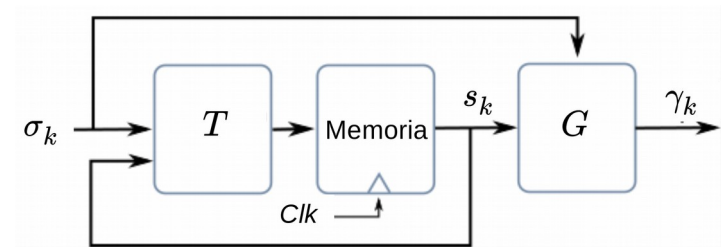
Los modelos de computación se clasifican en tres categorías:

- **Secuenciales** – son modelos en los cuales una tarea se ejecuta después de otra. Su principal ventaja es que es fácil de entender e implementar (Automata finito, Automata con Pila, Automata con Procesador de Datos, Máquinas de Turing);
- **Funcionales** – son modelos en los cuales se usan funciones matemáticas para representar las tareas a realizar; y
- **Concurrentes** – son modelos en los cuales varias tareas ocurren simultáneamente (automata celular, redes de Petri, redes interactivas, redes de procesos de Kahn, modelo actor).

Los automatas finitos también se los denomina **máquina de estado finito** (FSM).

Esta definido por un sextuple  $(\Sigma, \Gamma, \mathcal{S}, s_0, T, G)$  donde:

- $\Sigma$  es el alfabeto de entrada, un conjunto finito y no vacío de símbolos;
- $\Gamma$  es el alfabeto de salida, conjunto finito y no vacío de símbolos;
- $\mathcal{S}$  es un conjunto finito y no vacío de estados, que incluye el estado inicial  $s_0 \in \mathcal{S}$  y al conjunto de estados finales  $\mathcal{F}$ , el cual puede vacío;
- $T: \mathcal{S} \times \Sigma \rightarrow \mathcal{S}$  es la función de transición de estado;
- $G: \mathcal{S} \times \Sigma \rightarrow \Gamma$  es la función de salida.



Una FSM tiene un número finito de estados. Cada estado acepta un número finito de entradas, y cada estado tiene reglas que describen la acción de la FSM para cada entrada, representada en la función  $T$ . Al mismo tiempo, una entrada puede hacer que la máquina cambie de estado. Para cada símbolo de entrada, hay exactamente una transición.

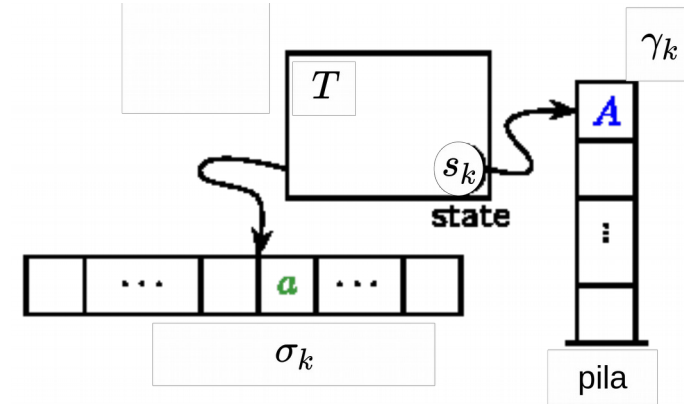
Debido a la **restricción en el número de estados posibles** en una FSM, se puede **solo se pueden implementar** algoritmos que requieran un número finito de estados.

Como los algoritmos pueden ser transformado en un autómata, la mayoría de los dispositivos informáticos pueden modelarse por una FSM, ya que funcionan con recursos finitos.

Los **autómatas con pila** (FSMP) son similares a las FSM pero **tienen una memoria LIFO** (pila) que puede crecer hasta un tamaño arbitrario.

Esta definido por un septupla  $(\Sigma, \Gamma, \mathcal{S}, \mathcal{F}, s_0, \gamma_0, T)$  donde:

- $\Sigma$  es el alfabeto de entrada, un conjunto finito y no vacío de símbolos;
- $\Gamma$  es el alfabeto de la pila, conjunto finito y no vacío de símbolos, que incluye el estado inicial  $\gamma_0 \in \Gamma$ ;
- $\mathcal{S}$  es un conjunto finito y no vacío de estados, que incluye el estado inicial  $s_0 \in \mathcal{S}$  y al conjunto de estados finales  $\mathcal{F}$ ;
- $T: \mathcal{S} \times \Sigma \times \Gamma \rightarrow \mathcal{S}$  es la función de transición de estado;



Al igual que una FSM, una FSMP tiene un número finito de estados y cada estado acepta un número finito de entradas. Las transiciones entre estados están determinadas por la función  $T$ , que depende de la entrada, el estado actual y el símbolo de pila, opcionalmente puede manipular (escribir o leer) la pila.

Para cada combinación de símbolo de entrada, estado actual y símbolo de pila hay una transición u operación sobre la pila.

Si bien tiene la **restricción en el número de estados posibles**, al igual que una FSM, la pila le provee de una memoria que hace que **pueda implementar** algoritmos que las FSM no pueden.



Una **máquina de estado finito con un camino de datos** (FSMD) es una abstracción matemática que se usa para diseñar lógica digital.

Esta compuesto por **una FSM**, que controla el flujo de tareas a ejecutar, y **un camino de datos**, que realiza las operaciones de procesamiento de datos.

Una **ruta de datos** (Datapath) es una colección de unidades funcionales (como unidades aritméticas y lógicas), que realizan operaciones de procesamiento de datos, registros y buses.

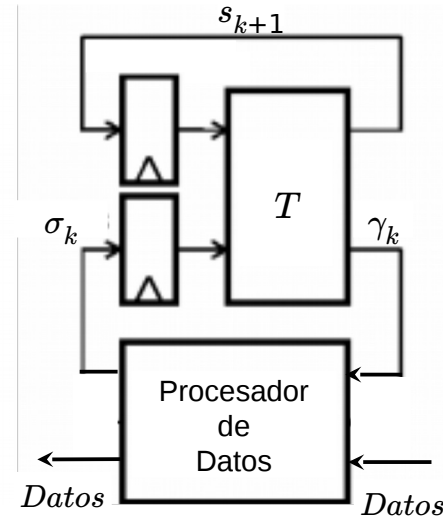
Los FSMD son esencialmente **algoritmos secuenciales** en los que las **declaraciones se han programado en estados**, lo que resulta en diagramas de estado más complejos.

Un programa se convierte en un diagrama de estado complejo en el que **los estados y los arcos pueden incluir expresiones aritméticas**, y esas expresiones pueden usar estados, entradas y salidas como variables.

Los FSM no utilizan variables u operaciones/condiciones aritméticas, por lo tanto, los FSMD son más potentes que los FSM.

Un FSMD es equivalente a una máquina de Turing en el poder computacional.

El nivel de abstracción FSMD a menudo se denomina **transferencia a nivel de registro** (RTL).

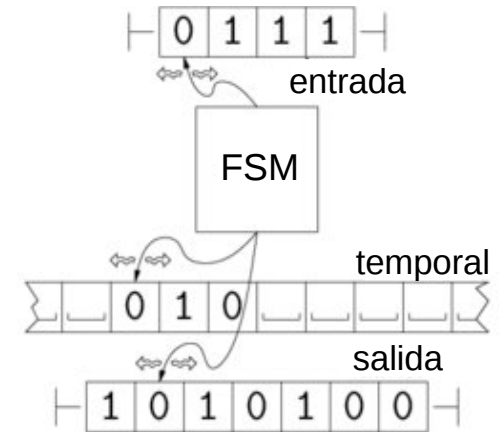




Una **máquina de Turing** es un **modelo computacional** utilizado para **analizar algoritmos**.

Esta definido por un septuple  $(\Sigma, \Gamma, \Xi, \mathcal{S}, s_0, \xi, \mathcal{F}, T)$  donde:

- $\mathcal{S}$  es un conjunto finito de estados, que incluye el estado inicial  $s_0$  y al conjunto de estados finales  $\mathcal{F} \subseteq \mathcal{S}$ ;
- $\Sigma$  es el conjunto finito de símbolos de entrada de la máquina;
- $\Gamma$  es un conjunto finito de símbolos de salida;
- $\Xi$  es un conjunto finito de símbolos de la cinta ( $\Xi \subseteq \Sigma$ ), que incluye el simbolo blanco  $\xi$ ;
- $T: \mathcal{S} \times \Sigma \rightarrow \mathcal{S} \times \Sigma \times \{D, I\}$  es la función de transición de estado.



Este modelo consiste en una **relación de entrada-salida** calculada por la máquina. La entrada se da en la cinta de entrada de la máquina, y la salida consiste en el **contenido** de la cinta de salida cuando la máquina se detiene.

La **FSM dentro de la máquina de Turing**, definida por el número de estados y la función de transición  $T$ , determina el **contenido que se escribe** en la cinta de salida.

En cada paso, el estado actual y el carácter leído en la cinta determinan el siguiente estado en el que se encontrará el FSM, el carácter que la máquina emitirá en la cinta de salida y en qué dirección (derecha o izquierda) se mueve la cinta para leer el próximo carácter.

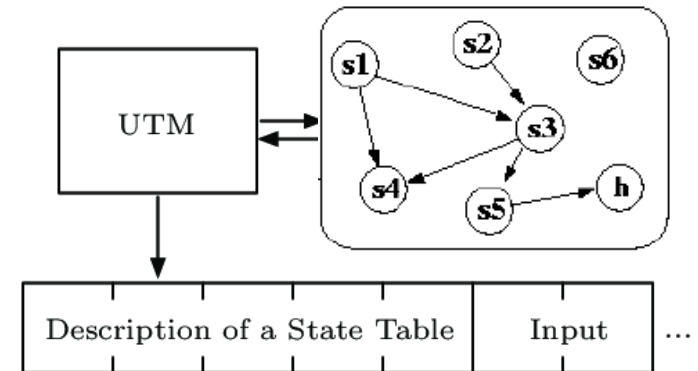
Hay ***muchas versiones*** de la maquina de Turing (con movimiento de espera, con cinta multipista, multicinta y multidimensional, entre otras) modificadas que en parecieran incrementar el poder computacional, pero ***todas ellas son equivalentes***.

***El problema*** con las máquinas de Turing es que se comporta como un computador con un programa fijo, por lo que ***se debe construir uno diferente para cada nuevo cálculo*** a realizar, para cada relación de entrada y salida.

Es por eso que se introduce la noción de una ***máquina universal de Turing***, que no es otra cosa que una máquina de Turing capaz de simular cualquier otra máquina de Turing con entradas arbitrarias.

Sin embargo, como la tabla de transición de cualquier máquina de Turing puede ser codificada en una cadena, la máquina universal se logra leyendo desde la cinta ***la descripción de la máquina*** (tabla de transición de estados) a simular y sus ***entradas***.

***Una máquina universal de Turing puede implementar cualquier algoritmo.***



Una **máquina con registros** es una clase genérica de máquinas abstractas que utiliza **registros con direcciones exclusivas**, cada uno de ellos identificado por un único número entero positivo.

Una máquina de registro consta de:

- **Registros** – un conjunto de registros  $r_0 \dots r_n$  cada uno identificado con un número entero no negativo y de extensión infinita. Puede haber registros especiales que hacen la aritmética;
- **Marcas de conteo** – objetos discretos, indistinguibles de un solo tipo adecuado para el modelo. En el modelo de máquina con contador. De acuerdo con el modelo utilizado, estas marcas se pueden agrupar, o no, para operar con ellas.
- **Conjunto de instrucciones** – las instrucciones se dividen en clases que implementan las operaciones necesarias para implementar cualquier algoritmo, de modo que el modelo sea equivalente a Turing. Las clases son: Aritméticas, Control de flujo de ejecución, Direccionamiento de registros y Operaciones de entrada-salida;
- **Registro de estado** – un registro de instrucción especial, finito y separado de los registros que almacena la instrucción actual a ejecutar y su dirección en la **tabla de instrucciones**. Este registro y la tabla de instrucciones se encuentran en la máquina de estados finitos;
- **Conjunto de instrucciones** – un conjunto mínimo de instrucciones  $I_1 \dots I_m$ . En el caso de las máquinas con contador, puntero y acceso aleatorio (RAM) las instrucciones se encuentran en la tabla de instrucciones, separadas de los datos. En el caso de las máquinas de acceso aleatorio con almacenamiento de programa (RASP), las instrucciones están junto con los datos en los registros.

De acuerdo con su complejidad, se clasifican en cuatro subclases:

- **Máquina de conteo** – comprende un conjunto (al menos uno) de registros y un conjunto, de hasta siete, instrucciones para construir los programas.

El conjunto de instrucciones incluye operaciones aritméticas

CLR( $r$ )    Resetea el registro  $r$ ;

INC( $r$ )    Incrementa el contenido del registro  $r$ ;

DEC( $r$ )    Decrementa el contenido del registro  $r$ ;

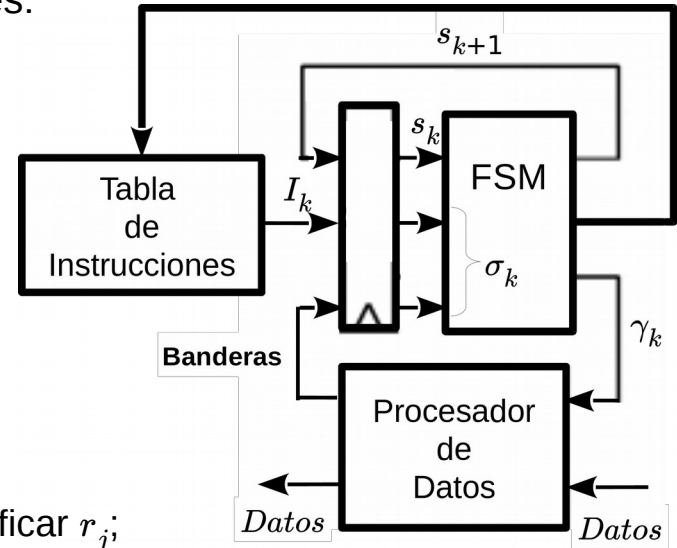
CPY( $r_j, r_k$ ) Copia el contenido del registro  $r_j$  al registro  $r_k$  sin modificar  $r_j$ ;  
al menos una operación condicional

JZ( $r, z$ )    Si el contenido del registro  $r$  es cero entonces ejecuta la instrucción alojada en la dirección  $z$ , sino continua ejecutando la secuencia;

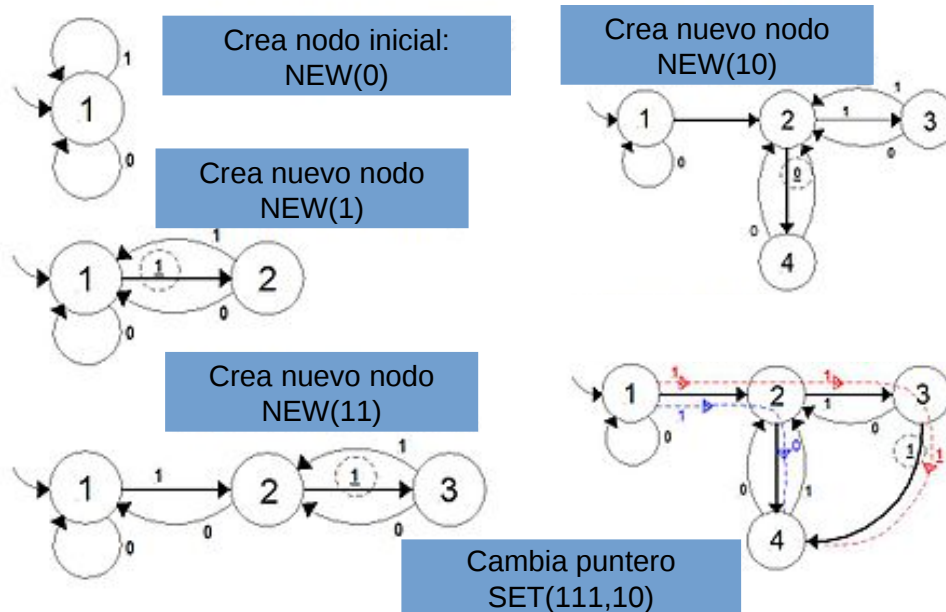
JE( $r_j, r_k, z$ ) Si el contenido del registro  $r_j$  es igual a de  $r_k$  entonces ejecuta la instrucción almacenada en la dirección  $z$ , sino continua ejecutando la secuencia;

y una instrucción que detiene la máquina (HALT).

Una **máquina de estados** que **ejecuta el programa almacenado en la tabla de instrucciones**, que esta separada de los registros. El procesador de datos incluye los registros de datos y los circuitos que implementan las instrucciones aritmeticas.



- **Máquina de puntero** – comprende un conjunto de registros y una FSM que ejecuta el programa. El conjunto de instrucciones **no tiene operaciones aritméticas, solo operaciones de control** que modifican la estructura de almacenamiento. Por ello, la computación procede de leer los símbolos de entrada, haciendo pruebas y modificando su estructura de almacenamiento (el patrón de nodos y punteros) y emitiendo símbolos basados en las pruebas.



La transformación de las secuencias de de entrada en salidas la realiza la FSM al ejecutar el programa almacenado en la tabla de instrucciones.

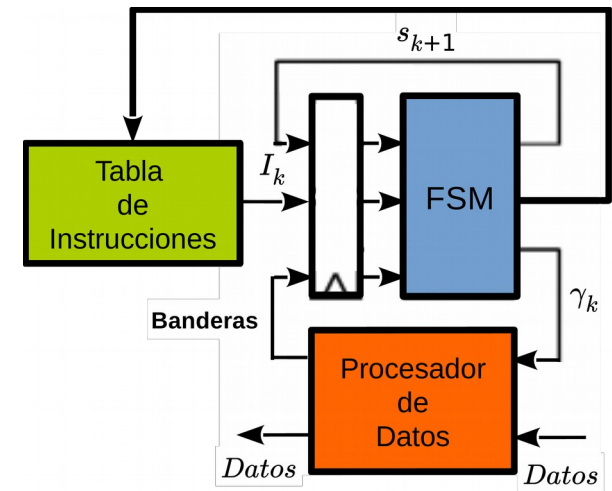
La FSM opera sobre la estructura de almacenamiento la máquina: un conjunto de registros (nodos) interconectados por punteros etiquetados con los símbolos (arcos), al estilo de un grafo orientado. La información está en la estructura de almacenamiento.

Esta máquina es una combinación entre las máquinas de conteo y de acceso aleatorio.

Por su forma de operación, es la menos común y más abstracta de las máquinas con registros.

- **Máquina de acceso aleatorio** – es una máquina de contador con múltiples registros, un conjunto de instrucciones aumentado (operaciones aritméticas extras, lógicas) y direccionamiento indirecto.

Dependiendo de la instrucción la máquina de estados finitos obtiene la dirección de un registro origen/destino (i) **directamente** de la instrucción o (ii) **indirectamente** del contenido de un puntero especificado en la instrucción a partir de un parametro de la misma.



Una **instrucción directa especifica la dirección de los registros** de origen y destino **en la misma instrucción**.

Ejemplo:  $CPY(r_j, r_k)$  copia el contenido del registro  $r_j$  al registro  $r_k$  ( $r_j \rightarrow r_k$ )

Una **instrucción indirecta especifica la dirección de los registros** de origen y/o destino **a través de un registro** (puntero) cuyo **contenido es la dirección del registro** origen y/o destino.

Ejemplo:  $CPY(d, r_j, i, r_k)$  copia el contenido del registro  $r_j$  al registro apuntado por el contenido de  $r_k$  ( $r_j \rightarrow [r_k]$ ).

Esta máquina abstracta (modelo) se **aproxima a las computadoras convencionales** a partir de la adición de registros auxiliares, el más común de los cuales se llama **acumulador**.

- **Máquina de acceso aleatorio con programa almacenado**
  - es una máquina de acceso aleatorio cuyo **programa esta almacenado en sus registros** (memoria) junto con sus entradas (datos).

Al igual que la máquina de acceso aleatorio tiene un conjunto de **instrucciones muy simple y reducido de instrucciones** que incluye operaciones aritméticas, lógicas, movimientos de datos, testeo y control de ejecución.

Pueden incluir registros adicionales con funciones específicas, como un acumulador.

Esta máquina coloca el **programa y los datos en los registros** y **busca las instrucciones en la memoria de manera secuencial**, a menos que produzca un cambio en la secuencia de ejecución.

**Utiliza dos conjuntos** de instrucciones: la tabla de instrucciones de la máquina de estados (**intérprete**) y el **programa** en los registros. Estos conjuntos no tienen que ser iguales o extraerse del mismo conjunto.

Como el algoritmo esta almacenado en memoria, esta máquina tiene **la capacidad de auto-modificar instrucciones** de programa (las instrucciones de la máquina de estado no se puede modificar).

Esto hace posible

- Implementar subrutinas, almacenando la dirección de retorno en el último comando de la subrutina;
- Realizar llamadas tablas JUMP;
- Implementar código auto modificable.

