

ST0244-032

Clase 3

J.F. Cardona

Universidad EAFIT

29 de julio de 2015

Agenda

1 Módulo 1. Introducción a lo lenguajes de programación

- Máquinas abstractas
 - Definición de máquina abstracta
- Interpretador
 - Estructura de las máquina abstractas
 - Ciclo de ejecución
- Implementación de lenguajes
 - Definición

2 Módulo 2. Sintaxis

- Sintaxis
- Backus Naur Form (BNF)
- El lenguaje EWE

Introducción

Máquinas abstractas - Definición

Definición de máquina

Una máquina abstracta no es más que una abstracción del concepto de un computador físico.

Lenguaje

Los algoritmos deben ser representados utilizando las instrucciones de un lenguaje de programación \mathcal{L} .

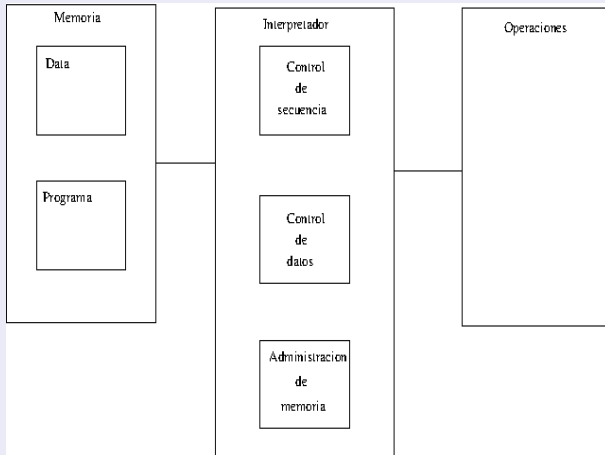
Máquina abstracta

Asuma que dado un lenguaje de programación \mathcal{L} , denotado $\mathcal{M}_{\mathcal{L}}$, es cualquier conjunto de estructuras de datos y algoritmos los cuales pueden ejecutar el almacenamiento y ejecución de programas escritos en \mathcal{L}

Introducción

Interpretador

Estructura de las máquinas abstractas



Introducción

Interpretador - Ciclo de ejecución

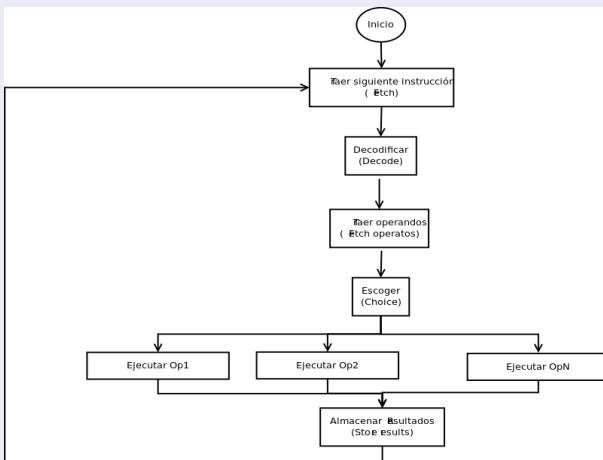
Ejecución de operaciones

- 1 Operaciones para procesar tipos de datos;
- 2 Operaciones y estructuras de datos para controlar la secuencia de la ejecución de operaciones;
- 3 Operaciones y estructuras de datos para controlar la transferencia de memoria;
- 4 Operaciones y estructuras de datos para la gestión de memoria.

Introducción

Interpretador

Ciclo de ejecución del interpretador



Introducción

Interpretador

Lenguajes de máquina

Dada una máquina abstracta $\mathcal{M}_{\mathcal{L}}$, el lenguaje \mathcal{L} , “entendido” por el interpretador de $\mathcal{M}_{\mathcal{L}}$ es llamado el lenguaje de máquina de $\mathcal{M}_{\mathcal{L}}$.

Máquinas abstractas

Lenguajes de bajo nivel, lenguajes de alto nivel

Lenguaje de alto bajo nivel

Los lenguajes de bajo nivel son aquellos lenguajes cuya máquina abstracta es muy cercana a, o coincide con, la máquina física.

Lenguaje de alto alto nivel

Los lenguajes de alto nivel son aquellos que soportan el uso de constructores que utilizan mecanismos apropiados de abstracción para asegurar que ellos son independientes de las características físicas del computador.

Introducción

Implementación de lenguajes

Definición de implementación de un lenguaje

Para implementar un lenguaje de programación \mathcal{L} , significa implementar una máquina abstracta la cual tiene a \mathcal{L} como su lenguaje de máquina.

Sintaxis

Terminología

Sintaxis: La sintaxis de un lenguaje de programación determina si los programas se encuentran bien formados o están sintácticamente correctos.

Semántica: La semántica describe como programa se ejecutará o si este se ejecutará completamente.

- La **sintaxis** es como las cosas lucen.
- La **semántica** es como las cosas trabajan (significado).

- Tiempos:

- ▶ Tiempo de diseño.
- ▶ Tiempo de edición.
- ▶ Tiempo de compilación.
- ▶ Tiempo de edición de enlaces.
- ▶ Tiempo de ejecución.

- Un **terminal** es un símbolo del lenguaje.
- Una **categoría sintáctica** o un **no-terminal** es un conjunto de objetos (cadenas) que pueden ser definidos en terminos de símbolos en el lenguaje (terminal y símbolos no terminales).
- Un **metalenguaje** es un lenguaje de alto nivel utilizado para especificar, discutir, describir o analizar otro lenguaje.

Sintaxis

Backus Naur Form (BNF)

- **Backus Naur Form (BNF)** es un metalenguaje formal para describir la sintaxis.
- La palabra formal significa aquí que la gramática no se presta para ambigüedad o dobles definiciones. Es decir que el lenguaje no es abierto a sus propias interpretaciones.
- BNF fue utilizado por John Backus para describir la sintaxis de Algol en 1963.
- BNF consiste de un conjunto de reglas que tienen la siguiente forma:

$\langle \text{categoría sintáctica} \rangle ::= \text{una cadena de terminales o no-terminales}$

- El carácter $::=$ significa está compuesto de, o produce.

Sintaxis

Backus Naur Form (BNF)

BNF de un lenguaje similar a Java

```
⟨programa java⟩ ::= ...  
    ⟨instrucción⟩ ::= ⟨selección⟩  
                    | ⟨ciclo⟩  
                    | ⟨asignación⟩  
                    | ⟨bloque⟩  
                    | ⟨expresión  
    ⟨selección⟩ ::= if ( ⟨expresión⟩ ) ⟨instrucción⟩  
                  | if ( ⟨expresión⟩ ) ⟨instrucción⟩ else ⟨instrucción⟩
```

Sintaxis

El lenguaje EWE

- EWE es una extensión de un lenguaje *primitivo* llamado RAM diseñado por Sethi, como un lenguaje de enseñanza.
- EWE es un lenguaje que representa el lenguaje de un computador simple.
- EWE es un lenguaje interpretador.

Sintaxis

El lenguaje EWE

- El interpretador del lenguaje EWE reconoce una instrucción por línea.
- Los comentarios comienzan con `#` y se extienden hasta el final de la línea.
- Las instrucciones están seguidas por equ que identifican localizaciones de memoria.
- El modelo de computación EWE consiste de:
 - ▶ las localizaciones de memoria por $M[\dots]$
 - ▶ una instrucción de memoria que contiene instrucciones
- Las instrucciones de EWE son ejecutadas en secuencia a menos se utiliza instrucciones `goto`.
- Las instrucciones terminan cuando un error ocurre o la instrucción `halt` es ejecutada.