Computación y Estructuras Discretas I

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co

Departamento de Computación y Sistemas Inteligentes

2024-2

Agenda del día

1. **Modelos de computación**
   - Introducción
   - Modelo RAM
   - Circuitos combinacionales
   - Autómatas finitos
   - Máquinas de Turing
2. **Análisis de complejidad temporal**
   - Introducción
   - Notación asintótica
   - Combinaciones de funciones
3. **Complejidad espacial**
4. **Ejercicios**

Agenda del día

Modelos de computación

¿Qué es un modelo de computación?

Modelos de computación

¿Qué es un modelo de computación?

- Una definición formal y abstracta de un computador.

Modelos de computación

¿Qué es un modelo de computación?

- Una definición formal y abstracta de un computador.
- Un modelo abstracto que describe una forma de computar.

Modelos de computación

¿Qué es un modelo de computación?

- Una definición formal y abstracta de un computador.
- Un modelo abstracto que describe una forma de computar.
- Es la definición de un conjunto de operaciones permitidas utilizadas en el cómputo y sus respectivos costos.

Modelos de computación

¿Qué es un modelo de computación?

- Una definición formal y abstracta de un computador.
- Un modelo abstracto que describe una forma de computar.
- Es la definición de un conjunto de operaciones permitidas utilizadas en el cómputo y sus respectivos costos.
- Al asumir un cierto modelo de computación es posible analizar los recursos de cómputo requeridos, como el tiempo de ejecución o el espacio de memoria, o discutir las limitaciones de algoritmos o computadores.

Modelos de computación

¿Qué aspectos se deben tener en cuenta al momento de definir un modelo de computación?

Modelos de computación

¿Qué aspectos se deben tener en cuenta al momento de definir un modelo de computación?

- Representación de las entradas y salidas

Modelos de computación

¿Qué aspectos se deben tener en cuenta al momento de definir un modelo de computación?

- Representación de las entradas y salidas
- Operaciones elementales.

Modelos de computación

¿Qué aspectos se deben tener en cuenta al momento de definir un modelo de computación?

- Representación de las entradas y salidas
- Operaciones elementales.
- Combinación de las operaciones para el desarrollo del programa.

Modelos de computación

¿Qué modelos de computación existen?

Modelos de computación

¿Qué modelos de computación existen?

- Existe una amplia variedad de modelos de computación que difieren en el conjunto de operaciones permitidas y su costo de computación.

Modelos de computación

¿Qué modelos de computación existen?

- Existe una amplia variedad de modelos de computación que difieren en el conjunto de operaciones permitidas y su costo de computación.
- Máquinas de acceso aleatorio RAM

Modelos de computación

¿Qué modelos de computación existen?

- Existe una amplia variedad de modelos de computación que difieren en el conjunto de operaciones permitidas y su costo de computación.
- Máquinas de acceso aleatorio RAM
- Circuitos combinacionales

Modelos de computación

¿Qué modelos de computación existen?

- Existe una amplia variedad de modelos de computación que difieren en el conjunto de operaciones permitidas y su costo de computación.
- Máquinas de acceso aleatorio RAM
- Circuitos combinacionales
- Autómatas finitos

Modelos de computación

¿Qué modelos de computación existen?

- Existe una amplia variedad de modelos de computación que difieren en el conjunto de operaciones permitidas y su costo de computación.
- Máquinas de acceso aleatorio RAM
- Circuitos combinacionales
- Autómatas finitos
- Máquinas de Turing

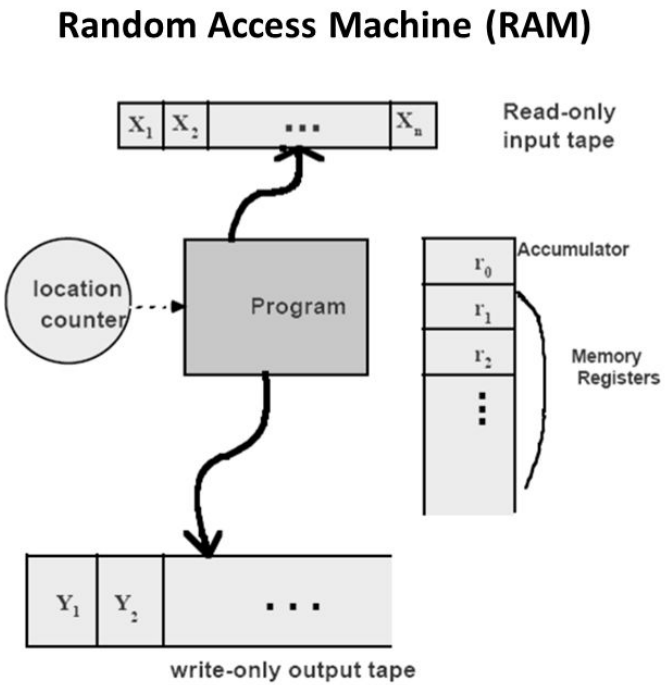
Modelos de computación

¿Qué modelos de computación existen?

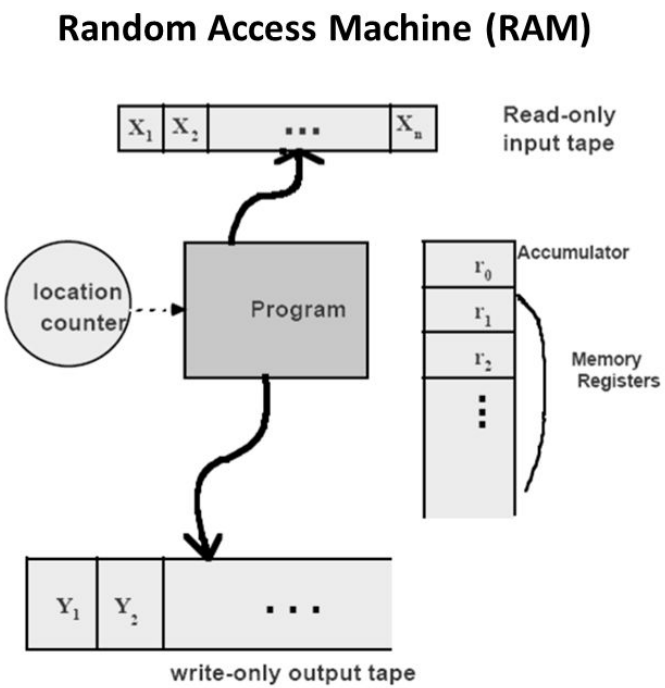
- Existe una amplia variedad de modelos de computación que difieren en el conjunto de operaciones permitidas y su costo de computación.
- Máquinas de acceso aleatorio RAM
- Circuitos combinacionales
- Autómatas finitos
- Máquinas de Turing
- ...

Agenda del día

Modelo RAM

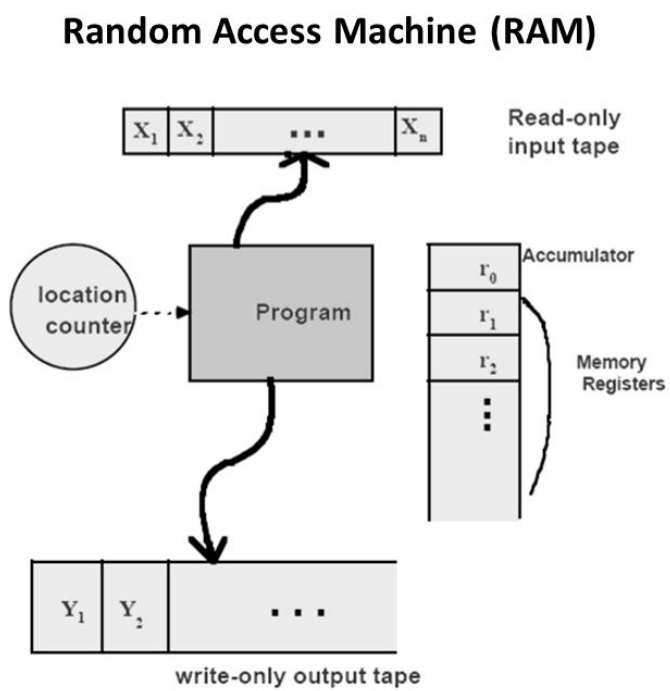

Random Access Machine (RAM)
X1 X2 ... Xn
Read-only
input tape
location
counter
Program
r0
r1
r2
Accumulator
Memory
Registers
Y1 Y2 ...
write-only output tape

Modelo RAM



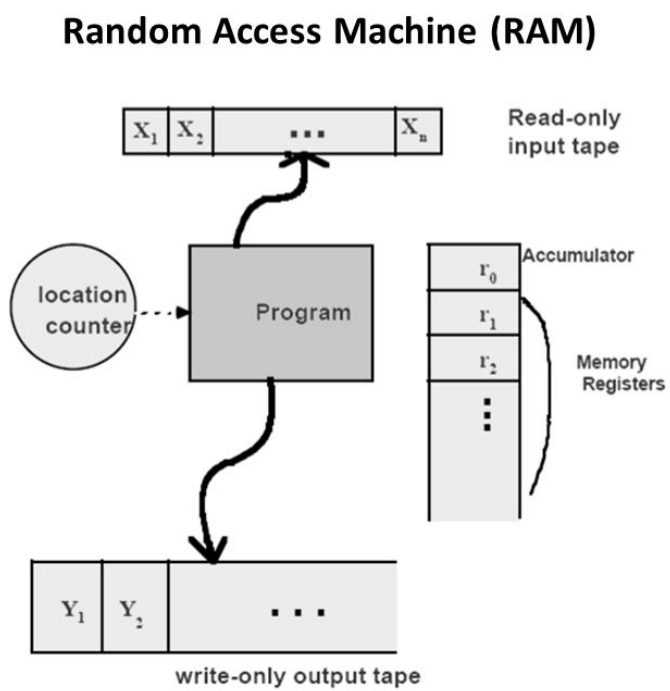
- Es un modelo simple de como los computadores se desempeñan.

Modelo RAM



- Es un modelo simple de como los computadores se desempeñan.
- Bajo el modelo RAM se mide el tiempo de ejecución de un algoritmo al contar la cantidad de pasos que se toma para una instancia de problema dada.

Modelo RAM



- Es un modelo simple de como los computadores se desempeñan.
- Bajo el modelo RAM se mide el tiempo de ejecución de un algoritmo al contar la cantidad de pasos que se toma para una instancia de problema dada.
- Está formado por una cinta de entrada, una de salida, un conjunto de registros y un programa (secuencia de instrucciones).

Modelos RAM

¿Qué se debe tener en cuenta cuando nos encontramos bajo el modelo RAM?

Modelos RAM

¿Qué se debe tener en cuenta cuando nos encontramos bajo el modelo RAM?

- Cada operación simple solo se toma un paso.

Modelos RAM

¿Qué se debe tener en cuenta cuando nos encontramos bajo el modelo RAM?

- Cada operación simple solo se toma un paso.
- Los ciclos y las subrutinas no se consideran operaciones simples.

Modelos RAM

¿Qué se debe tener en cuenta cuando nos encontramos bajo el modelo RAM?

- Cada operación simple solo se toma un paso.
- Los ciclos y las subrutinas no se consideran operaciones simples.
- Este tipo de operaciones se consideran como una composición de operaciones de un solo paso.

Modelos RAM

¿Qué se debe tener en cuenta cuando nos encontramos bajo el modelo RAM?

- Cada operación simple solo se toma un paso.
- Los ciclos y las subrutinas no se consideran operaciones simples.
- Este tipo de operaciones se consideran como una composición de operaciones de un solo paso.
- Cada acceso a memoria toma un solo paso.

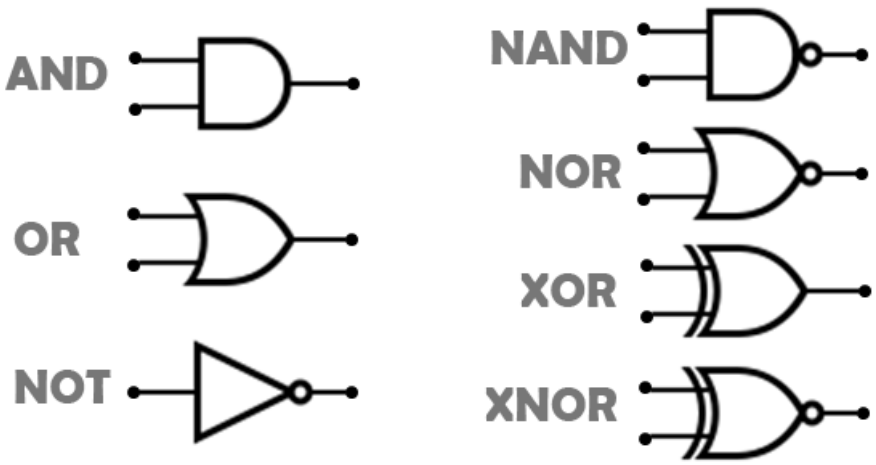
Modelos RAM

¿Qué se debe tener en cuenta cuando nos encontramos bajo el modelo RAM?

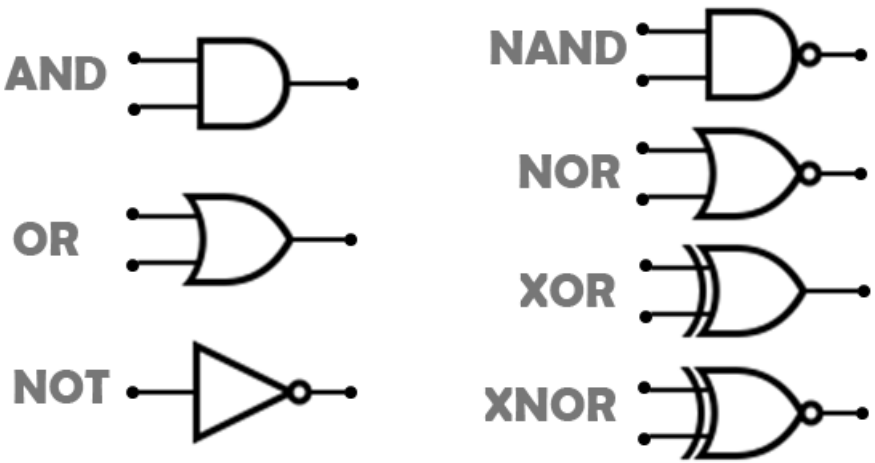
- Cada operación simple solo se toma un paso.
- Los ciclos y las subrutinas no se consideran operaciones simples.
- Este tipo de operaciones se consideran como una composición de operaciones de un solo paso.
- Cada acceso a memoria toma un solo paso.
- El modelo RAM no tiene en cuenta si un elemento esta en caché o en disco, lo cual simplifíca el análisis.

Agenda del día

Circuitos combinacionales

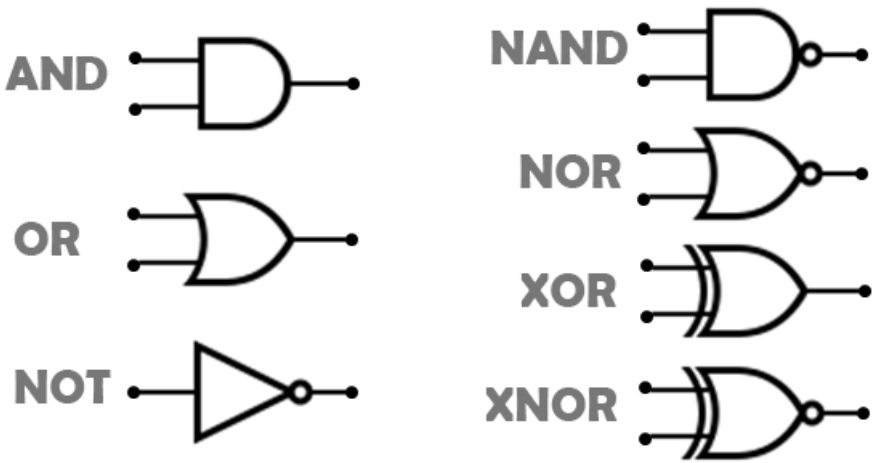


Circuitos combinacionales



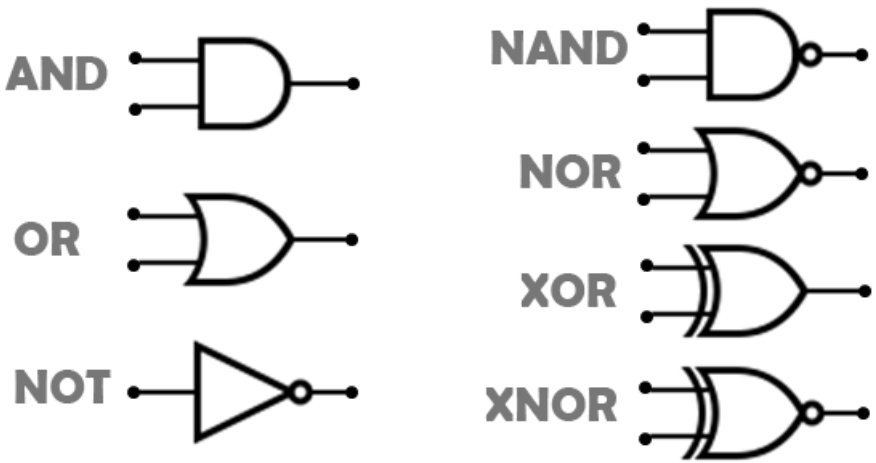
- Entradas: codificación binaria.

Circuitos combinacionales



- Entradas: codificación binaria.
- Salidas: codificación binaria.

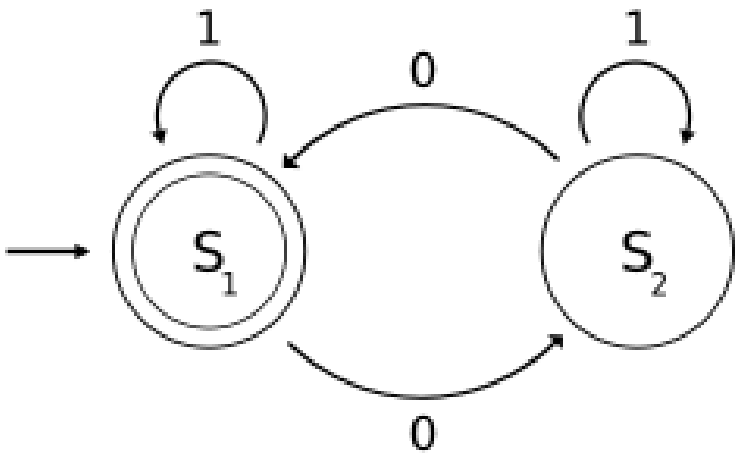
Circuitos combinacionales



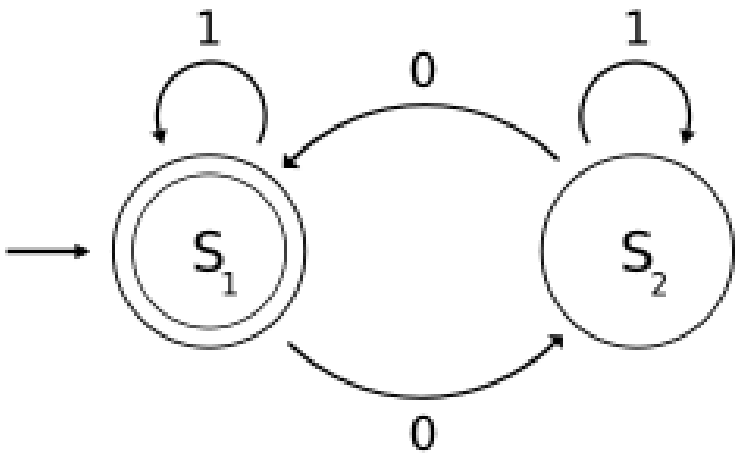
- Entradas: codificación binaria.
- Salidas: codificación binaria.
- Operaciones elementales: compuertas lógicas

Agenda del día

Autómatas finitos

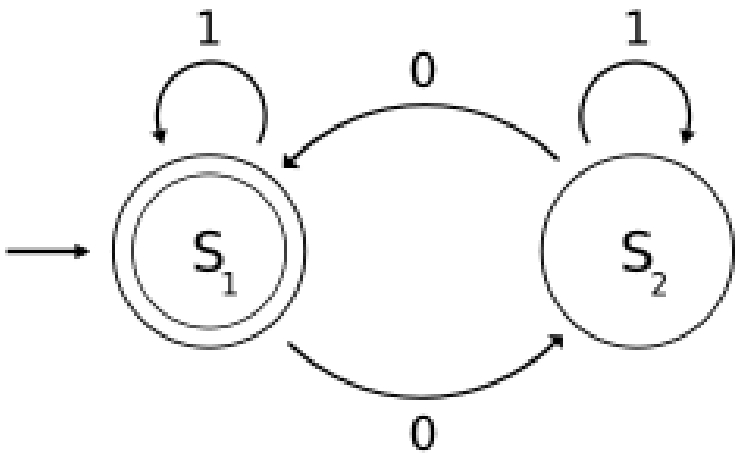


Autómatas finitos



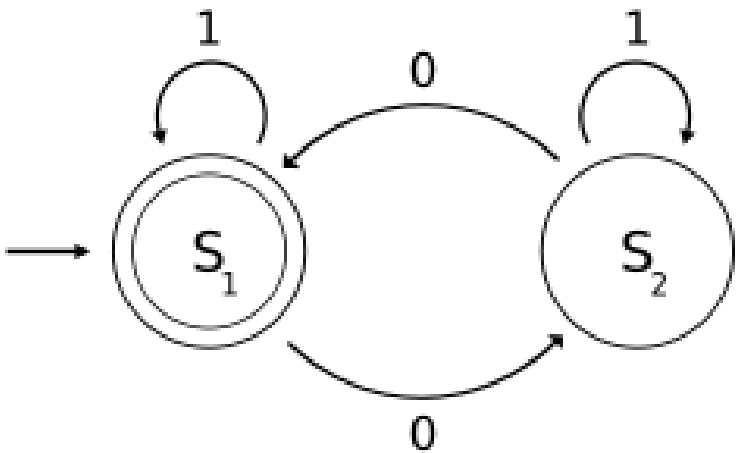
- Procesan cadenas de entrada, las cuales son aceptadas o rechazadas.

Autómatas finitos



- Procesan cadenas de entrada, las cuales son aceptadas o rechazadas.
- Lee símbolos escritos sobre una cinta semi infinita, dividida en celdas, sobre la cual se escribe una cadena de entrada.

Autómatas finitos

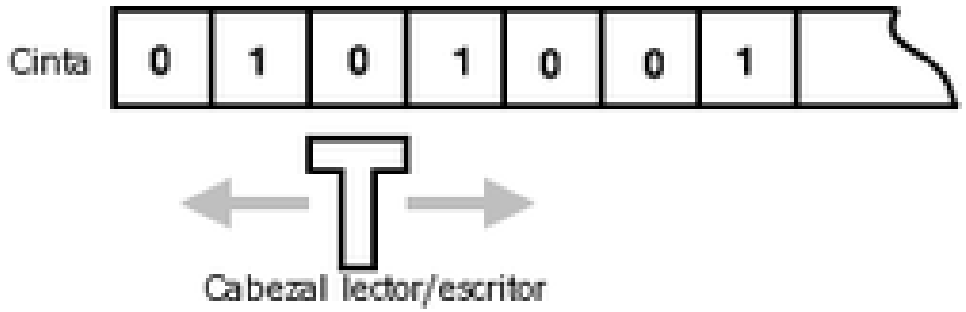


- Procesan cadenas de entrada, las cuales son aceptadas o rechazadas.
- Lee símbolos escritos sobre una cinta semi infinita, dividida en celdas, sobre la cual se escribe una cadena de entrada.
- Posee una cabeza lectora que contiene configuraciones internas llamadas estados.

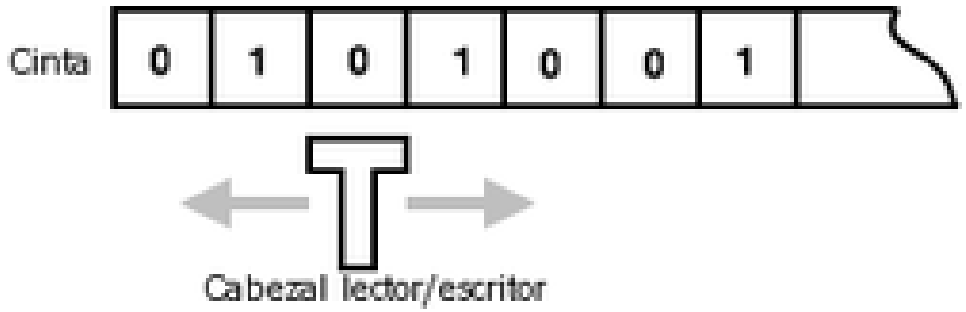
Agenda del día

1. **Modelos de computación**
   - Introducción
   - Modelo RAM
   - Circuitos combinacionales
   - Autómatas finitos
   - Máquinas de Turing
2. **Análisis de complejidad temporal**
   - Introducción
   - Notación asintótica
   - Combinaciones de funciones
3. **Complejidad espacial**
4. **Ejercicios**

Máquinas de Turing

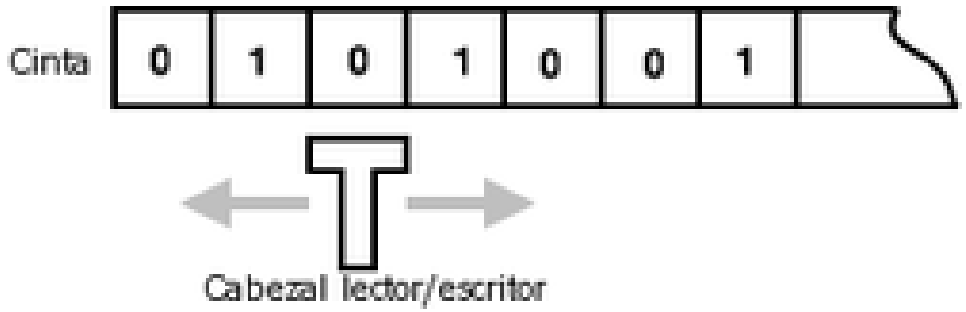


Máquinas de Turing



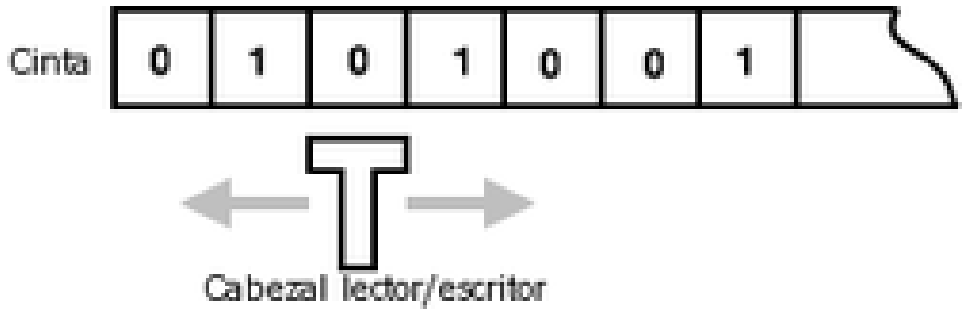
- Es el modelo de autómata con máxima capacidad computacional.

Máquinas de Turing



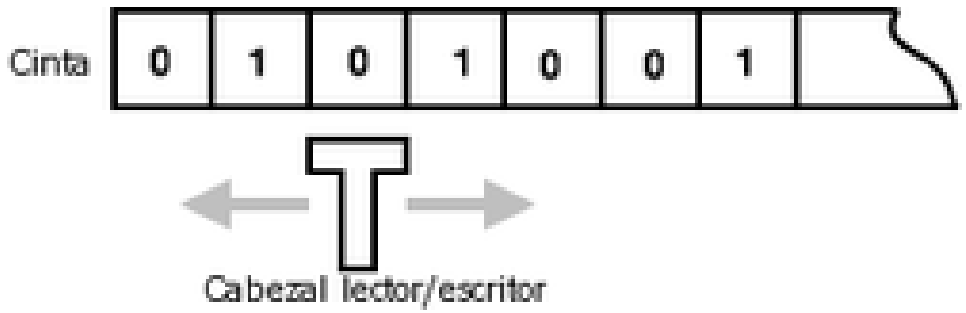
- Es el modelo de autómata con máxima capacidad computacional.
- Entradas: cinta sin fin formada por celdas que almacenan símbolos

Máquinas de Turing



- Es el modelo de autómata con máxima capacidad computacional.
- Entradas: cinta sin fin formada por celdas que almacenan símbolos
- Salidas: contenido final de la cinta.

Máquinas de Turing



- Es el modelo de autómata con máxima capacidad computacional.
- Entradas: cinta sin fin formada por celdas que almacenan símbolos
- Salidas: contenido final de la cinta.
- Operaciones elementales: transición de estado, leyendo un símbolo de la cinta, escribiendo un símbolo en la cinta y realizando un movimiento sobre la cinta (izquierda o derecha).

Agenda del día

Introducción

¿Para qué sirve el orden de crecimiento del tiempo de ejecución de un algoritmo?

Introducción

¿Para qué sirve el orden de crecimiento del tiempo de ejecución de un algoritmo?

- Caracterización simple de la eficiencia de un algoritmo.

Introducción

¿Para qué sirve el orden de crecimiento del tiempo de ejecución de un algoritmo?

- Caracterización simple de la eficiencia de un algoritmo.
- Comparación con el desempeño de algoritmos alternativos para solucionar el problema dado.

Introducción

¿Para qué sirve el orden de crecimiento del tiempo de ejecución de un algoritmo?

- Caracterización simple de la eficiencia de un algoritmo.
- Comparación con el desempeño de algoritmos alternativos para solucionar el problema dado.
- Para evitar esfuerzo innecesario.

Introducción

¿Por qué esfuerzo innecesario?

Introducción

¿Por qué esfuerzo innecesario?

- Las constantes, coeficientes y términos de menor orden de un tiempo de ejecución exacto son dominados por los efectos del tamaño de la entrada cuando este es significativo.

Introducción

¿Por qué esfuerzo innecesario?

- Las constantes, coeficientes y términos de menor orden de un tiempo de ejecución exacto son dominados por los efectos del tamaño de la entrada cuando este es significativo.
- Cuando se estudian entradas de tamaño suficientemente grande para hacer relevante solamente el orden de crecimiento del tiempo de ejecución del algoritmo, estamos trabajando con su eficiencia asintótica.

Ejemplo

Sean tres algoritmos A, B, C tal que

- $T_A(n) = 100$
- $T_B(n) = 2n + 10$
- $T_C(n) = n^2 + 5$

Ejemplo

Sean tres algoritmos A, B, C tal que

- $T_A(n) = 100$
- $T_B(n) = 2n + 10$
- $T_C(n) = n^2 + 5$

Para $n = 1$ $T_A(n) = 100$, $T_B(n) = 12$ *y* $T_C(n) = 6$

Ejemplo

Sean tres algoritmos A, B, C tal que

- $T_A(n) = 100$
- $T_B(n) = 2n + 10$
- $T_C(n) = n^2 + 5$

Para $n = 1$ $T_A(n) = 100$, $T_B(n) = 12$ *y* $T_C(n) = 6$
Para $n = 5$ $T_A(n) = 100$, $T_B(n) = 20$ *y* $T_C(n) = 30$

Ejemplo

Sean tres algoritmos A, B, C tal que

- $T_A(n) = 100$
- $T_B(n) = 2n + 10$
- $T_C(n) = n^2 + 5$

Para $n = 1$ $T_A(n) = 100$, $T_B(n) = 12$ *y* $T_C(n) = 6$
Para $n = 5$ $T_A(n) = 100$, $T_B(n) = 20$ *y* $T_C(n) = 30$
Para $n = 10$ $T_A(n) = 100$, $T_B(n) = 30$ *y* $T_C(n) = 105$

Ejemplo

Sean tres algoritmos A, B, C tal que

- $T_A(n) = 100$
- $T_B(n) = 2n + 10$
- $T_C(n) = n^2 + 5$

Para $n = 1$ $T_A(n) = 100$, $T_B(n) = 12$ *y* $T_C(n) = 6$
Para $n = 5$ $T_A(n) = 100$, $T_B(n) = 20$ *y* $T_C(n) = 30$
Para $n = 10$ $T_A(n) = 100$, $T_B(n) = 30$ *y* $T_C(n) = 105$
Para $n = 100$ $T_A(n) = 100$, $T_B(n) = 210$ *y* $T_C(n) = 10005$

Ejemplo

Sean tres algoritmos A, B, C tal que

- $T_A(n) = 100$
- $T_B(n) = 2n + 10$
- $T_C(n) = n^2 + 5$

Para $n = 1$ $T_A(n) = 100$, $T_B(n) = 12$ *y* $T_C(n) = 6$
Para $n = 5$ $T_A(n) = 100$, $T_B(n) = 20$ *y* $T_C(n) = 30$
Para $n = 10$ $T_A(n) = 100$, $T_B(n) = 30$ *y* $T_C(n) = 105$
Para $n = 100$ $T_A(n) = 100$, $T_B(n) = 210$ *y* $T_C(n) = 10005$
...

Agenda del día

Notación asintótica

¿Qué es la notación asintótica?

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.
- Ellas se definen en términos de funciones cuyo dominio es el conjunto de los números naturales.

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.
- Ellas se definen en términos de funciones cuyo dominio es el conjunto de los números naturales.
- Se aplica sobre funciones.

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.
- Ellas se definen en términos de funciones cuyo dominio es el conjunto de los números naturales.
- Se aplica sobre funciones.
  - Aquellas que caracterizan el tiempo de ejecución de un algoritmo.

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.
- Ellas se definen en términos de funciones cuyo dominio es el conjunto de los números naturales.
- Se aplica sobre funciones.
  - Aquellas que caracterizan el tiempo de ejecución de un algoritmo.
  - Otros aspectos de los algoritmos (espacio)

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.
- Ellas se definen en términos de funciones cuyo dominio es el conjunto de los números naturales.
- Se aplica sobre funciones.
  - Aquellas que caracterizan el tiempo de ejecución de un algoritmo.
  - Otros aspectos de los algoritmos (espacio)
  - Funciones que nada tienen que ver con algoritmos

Notación asintótica

¿Qué es la notación asintótica?

- Son aquellas notaciones utilizadas para describir el tiempo de ejecución asintótico de un algoritmo.
- Ellas se definen en términos de funciones cuyo dominio es el conjunto de los números naturales.
- Se aplica sobre funciones.
  - Aquellas que caracterizan el tiempo de ejecución de un algoritmo.
  - Otros aspectos de los algoritmos (espacio)
  - Funciones que nada tienen que ver con algoritmos
  - ...

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

El tiempo de ejecución sobre el que se va a trabajar.

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

El tiempo de ejecución sobre el que se va a trabajar.

- Peor caso, mejor caso, para cualquier instancia del problema.

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

El tiempo de ejecución sobre el que se va a trabajar.

- Peor caso, mejor caso, para cualquier instancia del problema.

¿Que notaciónes conocemos?

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

El tiempo de ejecución sobre el que se va a trabajar.

- Peor caso, mejor caso, para cualquier instancia del problema.

¿Que notaciónes conocemos?

- Notación Θ.

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

El tiempo de ejecución sobre el que se va a trabajar.

- Peor caso, mejor caso, para cualquier instancia del problema.

¿Que notaciónes conocemos?

- Notación Θ.
- Notación O.

Notación asintótica

¿Qué se debe tener en cuenta previa a la aplicación de la notación asintótica?

El tiempo de ejecución sobre el que se va a trabajar.

- Peor caso, mejor caso, para cualquier instancia del problema.

¿Que notaciónes conocemos?

- Notación Θ.
- Notación O.
- Notación Ω.

Notación asintótica

¿Qué es la notación Θ?

Notación asintótica

¿Qué es la notación Θ?

- Representa una función que sirve de cota tanto superior como inferior de otra función cuando el argumento tiende a infinito.

Notación asintótica

¿Qué es la notación Θ?

- Representa una función que sirve de cota tanto superior como inferior de otra función cuando el argumento tiende a infinito.
- Cota ajustada a una función.

Notación asintótica

¿Qué es la notación Θ?

- Representa una función que sirve de cota tanto superior como inferior de otra función cuando el argumento tiende a infinito.
- Cota ajustada a una función.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Theta(g(n))$ como el conjunto de funciones tal que

Notación asintótica

¿Qué es la notación Θ?

- Representa una función que sirve de cota tanto superior como inferior de otra función cuando el argumento tiende a infinito.
- Cota ajustada a una función.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Theta(g(n))$ como el conjunto de funciones tal que

$$\Theta(g(n)) = \{f(n) \mid \in \exists c_1, c_2, n_0 \ \in \mathbb{Z}^+ \ \wedge \ 0 \leq c_1 \, g(n) \leq f(n) \leq c_2 \, g(n) \, \forall n \geq n_0\}$$

Notación asintótica

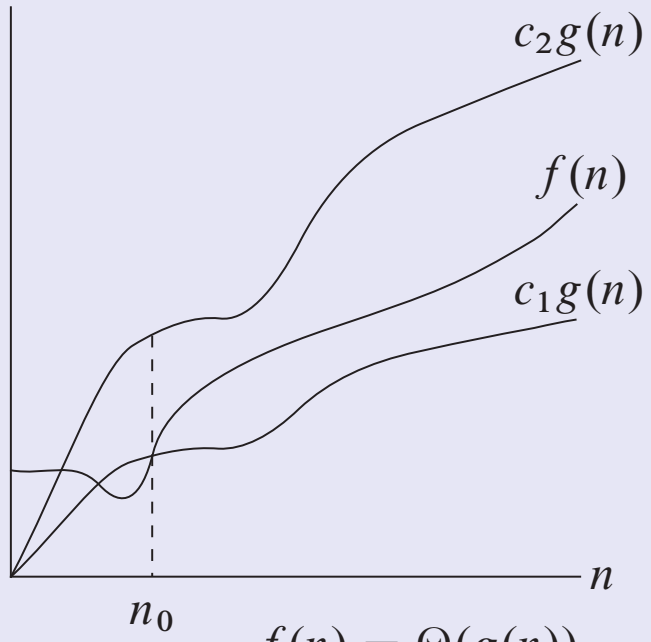
¿Qué es la notación Θ?

- Representa una función que sirve de cota tanto superior como inferior de otra función cuando el argumento tiende a infinito.
- Cota ajustada a una función.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Theta(g(n))$ como el conjunto de funciones tal que

$$\Theta(g(n)) = \{f(n) \mid \in \exists c_1, c_2, n_0 \ \in \mathbb{Z}^+ \ \wedge \ 0 \leq c_1\, g(n) \leq f(n) \leq c_2\, g(n) \, \forall n \geq n_0\}$$

- Se dice que $f(n)$ pertenece a $\Theta(g(n))$ si existen las constantes positivas c_1, c_2, n_0 tal que $f(n)$ pueda ubicarse entre $c_1\, g(n)$ y $c_2\, g(n)$ para un n suficientemente grande.

Notación asintótica



$f(n) = \Theta(g(n))$

Notación asintótica

¿Qué hace Θ?

Notación asintótica

¿Qué hace Θ?

Acota una función dentro de unos factores constantes para un tamaño de entradas suficientemente grande.

Notación asintótica

¿Qué hace Θ?

Acota una función dentro de unos factores constantes para un tamaño de entradas suficientemente grande.

¿Cómo decir que $g(n)$ es una cota ajustada de $f(n)$?

Notación asintótica

¿Qué hace Θ?

Acota una función dentro de unos factores constantes para un tamaño de entradas suficientemente grande.

¿Cómo decir que $g(n)$ es una cota ajustada de $f(n)$?

- $f(n) \in \Theta(g(n))$

Notación asintótica

¿Qué hace Θ?

Acota una función dentro de unos factores constantes para un tamaño de entradas suficientemente grande.

¿Cómo decir que $g(n)$ es una cota ajustada de $f(n)$?

- $f(n) \in \Theta(g(n))$
- $f(n) = \Theta(g(n))$ abusando de la notación

Notación asintótica

Ejemplo 1

Sea $T(n) = \frac{1}{2}n^2 - 3n$, queremos probar que $\Theta(n^2) = \frac{1}{2}n^2 - 3n$

Notación asintótica

Ejemplo 1

Sea $T(n) = \frac{1}{2}n^2 - 3n$, queremos probar que $\Theta(n^2) = \frac{1}{2}n^2 - 3n$

Solución

- Utilizamos la definición de cota ajustada asintótica.

Ejemplo 1

Sea $T(n) = \frac{1}{2}n^2 - 3n$, queremos probar que $\Theta(n^2) = \frac{1}{2}n^2 - 3n$

Solución

- Utilizamos la definición de cota ajustada asintótica.
- Debemos encontrar constantes positivas c_1, c_2, n_0 tal que

$$c_1\, n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2\, n^2 \;\forall n \geq n_0$$

Ejemplo 1

Sea $T(n) = \frac{1}{2}n^2 - 3n$, queremos probar que $\Theta(n^2) = \frac{1}{2}n^2 - 3n$

Solución

- Utilizamos la definición de cota ajustada asintótica.
- Debemos encontrar constantes positivas c_1, c_2, n_0 tal que

$$c_1\, n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2\, n^2 \;\forall n \geq n_0$$

- Se divide por n^2 y se llega a

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

Ejemplo 1

Sea $T(n) = \frac{1}{2}n^2 - 3n$, queremos probar que $\Theta(n^2) = \frac{1}{2}n^2 - 3n$

Solución

- Utilizamos la definición de cota ajustada asintótica.
- Debemos encontrar constantes positivas c_1, c_2, n_0 tal que

$$c_1\, n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2\, n^2 \;\forall n \geq n_0$$

- Se divide por n^2 y se llega a

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

- Buscamos que las desigualdades se mantengan.

Solución

- $\frac{1}{2} - \frac{3}{n} \leq c_2$

Solución

- $\frac{1}{2} - \frac{3}{n} \leq c_2$ se mantiene para cualquier valor de $n \geq 1$ y para $c_2 \geq \frac{1}{2}$

Solución

- $\frac{1}{2} - \frac{3}{n} \leq c_2$ se mantiene para cualquier valor de $n \geq 1$ y para $c_2 \geq \frac{1}{2}$
- Asimismo, $c_1 \leq \frac{1}{2} - \frac{3}{n}$

Solución

- $\frac{1}{2} - \frac{3}{n} \leq c_2$ se mantiene para cualquier valor de $n \geq 1$ y para $c_2 \geq \frac{1}{2}$
- Asimismo, $c_1 \leq \frac{1}{2} - \frac{3}{n}$ se mantiene para cualquier valor de $n \geq 7$ y para $c_1 \leq \frac{1}{14}$.

Solución

- $\frac{1}{2} - \frac{3}{n} \leq c_2$ se mantiene para cualquier valor de $n \geq 1$ y para $c_2 \geq \frac{1}{2}$
- Asimismo, $c_1 \leq \frac{1}{2} - \frac{3}{n}$ se mantiene para cualquier valor de $n \geq 7$ y para $c_1 \leq \frac{1}{14}$.
- Finalmente tenemos las constantes $c_1 = \frac{1}{14}$, $c_2 = \frac{1}{2}$ y $n_0 = 7$ que nos validan la definición, por tal $\Theta(n^2) = \frac{1}{2}n^2 - 3n$.

Notación asintótica

¿Qué se puede hacer cuando se busca una cota asintótica para una función positiva?

Notación asintótica

¿Qué se puede hacer cuando se busca una cota asintótica para una función positiva?

Ignorar los términos de menor orden de aquella función a la que se le busca la cota.

Notación asintótica

¿Qué se puede hacer cuando se busca una cota asintótica para una función positiva?

Ignorar los términos de menor orden de aquella función a la que se le busca la cota.

¿Por qué ignorarlos?

Notación asintótica

¿Qué se puede hacer cuando se busca una cota asintótica para una función positiva?

Ignorar los términos de menor orden de aquella función a la que se le busca la cota.

¿Por qué ignorarlos?

- Para valores de n suficientemente grandes, estos términos resultan insignificantes

Notación asintótica

¿Qué se puede hacer cuando se busca una cota asintótica para una función positiva?

Ignorar los términos de menor orden de aquella función a la que se le busca la cota.

¿Por qué ignorarlos?

- Para valores de n suficientemente grandes, estos términos resultan insignificantes
- El término de mayor grado del polinomio siempre dominará al de menor.

Notación asintótica

En este orden de ideas, ¿cómo encontrar un buen c_1 y c_2?

Notación asintótica

En este orden de ideas, ¿cómo encontrar un buen c_1 y c_2?

- Establecer un c_1 ligeramente inferior que el coeficiente del término de mayor orden

Notación asintótica

En este orden de ideas, ¿cómo encontrar un buen c_1 y c_2?

- Establecer un c_1 ligeramente inferior que el coeficiente del término de mayor orden
- Un c_2 mayor que el coeficiente del término de mayor orden.

Notación asintótica

¿Qué se puede decir con respecto de cualquier polinomio?

Notación asintótica

¿Qué se puede decir con respecto de cualquier polinomio?

Teorema

Para todo polinomio $p(n) = \sum_{i=0}^{d} a_i \cdot n^i$ *donde* a_1 *son constantes y* $a_d > 0$, $p(n) = \Theta(n^d)$

Notación asintótica

¿Qué es la notación O?

Notación asintótica

¿Qué es la notación O?

- Representa una función que sirve de cota superior de otra función cuando el argumento tiende a infinito.

Notación asintótica

¿Qué es la notación O?

- Representa una función que sirve de cota superior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $O(g(n))$ como el conjunto de funciones tal que

Notación asintótica

¿Qué es la notación O?

- Representa una función que sirve de cota superior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $O(g(n))$ como el conjunto de funciones tal que

$$O(g(n)) = \{f(n) \mid \in \exists c, n_0 \in \mathbb{Z}^+ \wedge 0 \leq f(n) \leq c\, g(n) \forall n \geq n_0\}$$

Notación asintótica

¿Qué es la notación O?

- Representa una función que sirve de cota superior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $O(g(n))$ como el conjunto de funciones tal que

$$O(g(n)) = \{f(n) \mid \in \exists c, n_0 \ \in \mathbb{Z}^+ \ \wedge \ 0 \leq f(n) \leq c\, g(n) \, \forall n \geq n_0\}$$

- Se dice que $f(n)$ pertenece a $O(g(n))$ si existen las constantes positivas c, n_0 tal que $f(n)$ pueda ubicarse en o por debajo de $c\, g(n)$ para un n suficientemente grande.

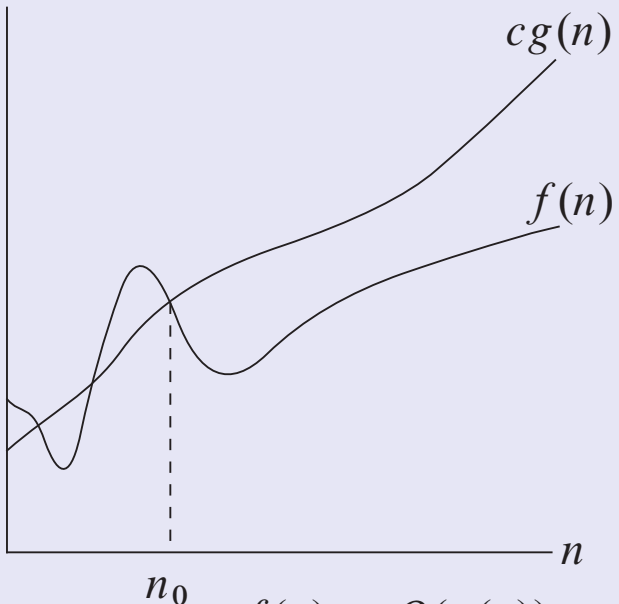
Notación asintótica

- Representa una función que sirve de cota superior dentro de un factor constante.

Notación asintótica

- Representa una función que sirve de cota superior dentro de un factor constante.
- Al usar la notación O se puede describir el tiempo de ejecución de un algoritmo con sólo inspeccionar su estructura general.

Notación asintótica



$f(n) = O(g(n))$

Ejemplo 2

Sea $T(n) = 7n^2$, queremos probar que $7n^2 = O(n^3)$

Ejemplo 2

Sea $T(n) = 7n^2$, queremos probar que $7n^2 = O(n^3)$

Solución

- Utilizamos la definición de cota superior asintótica.

Ejemplo 2

Sea $T(n) = 7n^2$, queremos probar que $7n^2 = O(n^3)$

Solución

- Utilizamos la definición de cota superior asintótica.
- Debemos encontrar constantes positivas c, n_0 tal que

$$7n^2 \leq c\, n^3 \ \forall n \geq n_0$$

Ejemplo 2

Sea $T(n) = 7n^2$, queremos probar que $7n^2 = O(n^3)$

Solución

- Utilizamos la definición de cota superior asintótica.
- Debemos encontrar constantes positivas c, n_0 tal que

$$7n^2 \leq c\, n^3 \ \forall n \geq n_0$$

- Se divide por n^3 y se llega a $\frac{7}{n} \leq c$

Ejemplo 2

Sea $T(n) = 7n^2$, queremos probar que $7n^2 = O(n^3)$

Solución

- Utilizamos la definición de cota superior asintótica.
- Debemos encontrar constantes positivas c, n_0 tal que

$$7n^2 \leq c\, n^3 \ \forall n \geq n_0$$

- Se divide por n^3 y se llega a $\frac{7}{n} \leq c$
- Se cumple para $n \geq 1$ y llegamos a que $c = 7$ y $n_0 = 1$.

Notación asintótica

¿Qué es la notación Ω?

Notación asintótica

¿Qué es la notación Ω?

- Representa una función que sirve de cota inferior de otra función cuando el argumento tiende a infinito.

Notación asintótica

¿Qué es la notación Ω?

- Representa una función que sirve de cota inferior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Omega(g(n))$ como el conjunto de funciones tal que

Notación asintótica

¿Qué es la notación Ω?

- Representa una función que sirve de cota inferior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Omega(g(n))$ como el conjunto de funciones tal que

$$\Omega(g(n)) = \{f(n) \mid \in \exists c, n_0 \in \mathbb{Z}^+ \wedge 0 \leq c\, g(n) \leq f(n)\, \forall n \geq n_0\}$$

Notación asintótica

¿Qué es la notación Ω?

- Representa una función que sirve de cota inferior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Omega(g(n))$ como el conjunto de funciones tal que

$$\Omega(g(n)) = \{f(n) \mid \in \exists c, n_0 \in \mathbb{Z}^+ \wedge 0 \leq c\, g(n) \leq f(n)\, \forall n \geq n_0\}$$

- Se dice que $f(n)$ pertenece a $\Omega(g(n))$ si existen las constantes positivas c, n_0 tal que $f(n)$ pueda ubicarse en o por encima de $c\, g(n)$ para un n suficientemente grande.

Notación asintótica

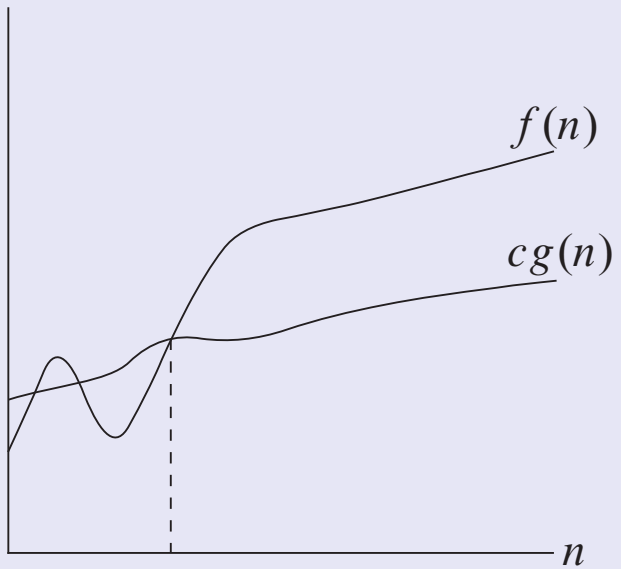
¿Qué es la notación Ω?

- Representa una función que sirve de cota inferior de otra función cuando el argumento tiende a infinito.
- Formalmente, se dice que para una función $g(n)$, se denota a $\Omega(g(n))$ como el conjunto de funciones tal que

$$\Omega(g(n)) = \{f(n) \mid \in \exists c, n_0 \in \mathbb{Z}^+ \wedge 0 \leq c\, g(n) \leq f(n)\, \forall n \geq n_0\}$$

- Se dice que $f(n)$ pertenece a $\Omega(g(n))$ si existen las constantes positivas c, n_0 tal que $f(n)$ pueda ubicarse en o por encima de $c\, g(n)$ para un n suficientemente grande.
- Representa una función que sirve de cota inferior dentro de un factor constante.

Notación asintótica



$f(n) = \Omega(g(n))$

Ejemplo 3

Sea $T(n) = 8n^3 + 5n^2 + 7$, queremos probar que $T(n) = \Omega(n^3)$

Notación asintótica

Ejemplo 3

Sea $T(n) = 8n^3 + 5n^2 + 7$, queremos probar que $T(n) = \Omega(n^3)$

Solución

- Utilizamos la definición de cota inferior asintótica.

Ejemplo 3

Sea $T(n) = 8n^3 + 5n^2 + 7$, queremos probar que $T(n) = \Omega(n^3)$

Solución

- Utilizamos la definición de cota inferior asintótica.
- Debemos encontrar constantes positivas c, n_0 tal que

$$c\, n^3 \leq 8n^3 + 5n^2 + 7 \; \forall n \geq n_0$$

Ejemplo 3

Sea $T(n) = 8n^3 + 5n^2 + 7$, queremos probar que $T(n) = \Omega(n^3)$

Solución

- Utilizamos la definición de cota inferior asintótica.
- Debemos encontrar constantes positivas c, n_0 tal que

$$c\, n^3 \leq 8n^3 + 5n^2 + 7 \; \forall n \geq n_0$$

- Se divide por n^3 y se llega a $c \leq 8 + \frac{5}{n} + \frac{7}{n^2}$

Ejemplo 3

Sea $T(n) = 8n^3 + 5n^2 + 7$, queremos probar que $T(n) = \Omega(n^3)$

Solución

- Utilizamos la definición de cota inferior asintótica.
- Debemos encontrar constantes positivas c, n_0 tal que

$$c\, n^3 \leq 8n^3 + 5n^2 + 7 \; \forall n \geq n_0$$

- Se divide por n^3 y se llega a $c \leq 8 + \frac{5}{n} + \frac{7}{n^2}$
- Se cumple para $n \geq 1$ y llegamos a que $c = 8$ y $n_0 = 1$.

Agenda del día

Notación asintótica

¿De qué otras forma se puede utilizar la notación asintótica?

Notación asintótica

¿De qué otras forma se puede utilizar la notación asintótica?

Dentro de fórmulas matemáticas

Notación asintótica

¿De qué otras forma se puede utilizar la notación asintótica?

Dentro de fórmulas matemáticas

$$2n^2 + 3n + 1 = 2n^2 + \Theta(n)$$

Notación asintótica

¿De qué otras forma se puede utilizar la notación asintótica?

Dentro de fórmulas matemáticas

$$2n^2 + 3n + 1 = 2n^2 + \Theta(n)$$

¿Y que quiere decir esto?

Notación asintótica

¿De qué otras forma se puede utilizar la notación asintótica?

Dentro de fórmulas matemáticas

$$2n^2 + 3n + 1 = 2n^2 + \Theta(n)$$

¿Y que quiere decir esto?

Que $2n^2 + 3n + 1 = 2n^2 + f(n)$ donde $f(n) \in \Theta(n)$ y en este caso $f(n) = 3n + 1$, que es $\Theta(n)$.

Notación asintótica

¿Cuál es la necesidad de incluir una función en notación asintótica adentro de otra?

Notación asintótica

¿Cuál es la necesidad de incluir una función en notación asintótica adentro de otra?

- Se eliminan detalles innecesarios y se depura la ecuación.

Notación asintótica

¿Cuál es la necesidad de incluir una función en notación asintótica adentro de otra?

- Se eliminan detalles innecesarios y se depura la ecuación.
- Muchas algoritmos consisten de dos o más subprocesos separados

Notación asintótica

¿Cuál es la necesidad de incluir una función en notación asintótica adentro de otra?

- Se eliminan detalles innecesarios y se depura la ecuación.
- Muchas algoritmos consisten de dos o más subprocesos separados
- El número de pasos realizados por un computador para solucionar un problema dado, es la suma del número de pasos realizados por todos sus subprocesos.

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

- Instrucciones del programa

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

- Instrucciones del programa
- Valores constantes

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

- Instrucciones del programa
- Valores constantes
- Valores variables

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

- Instrucciones del programa
- Valores constantes
- Valores variables
- ...

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

- Instrucciones del programa
- Valores constantes
- Valores variables
- ...

¿Cómo se define la complejidad espacial?

Complejidad espacial

¿Para que se requiere memoria dentro de un algoritmo?

Para guardar

- Instrucciones del programa
- Valores constantes
- Valores variables
- ...

¿Cómo se define la complejidad espacial?

La cantidad total de memoria computacional necesaria para completar la ejecución de un algoritmo.

Complejidad espacial

¿Cuáles son las razones por las que un programa utiliza memoria computacional?

Complejidad espacial

¿Cuáles son las razones por las que un programa utiliza memoria computacional?

- Espacio de instrucciones (versión compilada de las instrucciones)

Complejidad espacial

¿Cuáles son las razones por las que un programa utiliza memoria computacional?

- Espacio de instrucciones (versión compilada de las instrucciones)
- Espacio de la pila (información de funciones ejecutadas parcialmente)

Complejidad espacial

¿Cuáles son las razones por las que un programa utiliza memoria computacional?

- Espacio de instrucciones (versión compilada de las instrucciones)
- Espacio de la pila (información de funciones ejecutadas parcialmente)
- Espacio de datos (variables y constantes)

Complejidad espacial

¿Cuáles son las razones por las que un programa utiliza memoria computacional?

- Espacio de instrucciones (versión compilada de las instrucciones)
- Espacio de la pila (información de funciones ejecutadas parcialmente)
- Espacio de datos (variables y constantes)

¿Qué se considera al realizar el análisis de complejidad espacial?

Complejidad espacial

¿Cuáles son las razones por las que un programa utiliza memoria computacional?

- Espacio de instrucciones (versión compilada de las instrucciones)
- Espacio de la pila (información de funciones ejecutadas parcialmente)
- Espacio de datos (variables y constantes)

¿Qué se considera al realizar el análisis de complejidad espacial?

Tan sólo se considera el espacio de datos y se ignoran los otros dos.

Complejidad espacial

¿Qué cantidad de memoria se requiere para guardar distintos tipos de datos?

Complejidad espacial

¿Qué cantidad de memoria se requiere para guardar distintos tipos de datos?

- 32 bits (4 bytes) para un entero

Complejidad espacial

¿Qué cantidad de memoria se requiere para guardar distintos tipos de datos?

- 32 bits (4 bytes) para un entero
- 32 bits (4 bytes) para un float

Complejidad espacial

¿Qué cantidad de memoria se requiere para guardar distintos tipos de datos?

- 32 bits (4 bytes) para un entero
- 32 bits (4 bytes) para un float
- 16 bits (2 bytes) para un char

Complejidad espacial

¿Qué cantidad de memoria se requiere para guardar distintos tipos de datos?

- 32 bits (4 bytes) para un entero
- 32 bits (4 bytes) para un float
- 16 bits (2 bytes) para un char
- 64 bits (8 bytes) para un double

Complejidad espacial

¿Qué cantidad de memoria se requiere para guardar distintos tipos de datos?

- 32 bits (4 bytes) para un entero
- 32 bits (4 bytes) para un float
- 16 bits (2 bytes) para un char
- 64 bits (8 bytes) para un double
- ...

Complejidad espacial

¿Cómo se realizaría el análisis de complejidad para este fragmento de código?

Complejidad espacial

¿Cómo se realizaría el análisis de complejidad para este fragmento de código?

Ejemplo

```
int sum(int A[], int n)
{
   int sum = 0, i;
   for(i = 0; i < n; i++)
      sum = sum + A[i];
   return sum;
}
```

Complejidad espacial

Solución

Tipo	*Variable*	*Tamaño de 1 valor atómico*	*Cantidad de valores atómicos*
Entrada	*A*	*32 bits*	*n*
	n	*32 bits*	*1*
Auxiliar	*i*	*32 bits*	*1*
Salida	*sum*	*32 bits*	*1*

Complejidad espacial

Solución

Tipo	*Variable*	*Tamaño de 1 valor atómico*	*Cantidad de valores atómicos*
Entrada	*A*	*32 bits*	*n*
	n	*32 bits*	*1*
Auxiliar	*i*	*32 bits*	*1*
Salida	*sum*	*32 bits*	*1*

- *Complejidad Espacial Total = Entrada + Auxiliar + Salida = n + 3 =* $\theta(n)$
- *Complejidad Espacial Auxiliar = 1 =* $\theta(1)$
- *Complejidad Espacial Auxiliar + Salida = 1 + 1 =* $\theta(1)$

Ejercicios