



IC-5701

Compiladores e Intérpretes

Profesor:

Ing. Allan Rodríguez Dávila, MGP

Aspectos Generales

Especificación de Lenguajes

Procesadores de Lenguajes

Conceptos de Compilación

Máquinas Reales y Abstractas

# Especificación de Lenguajes

# Especificación del Lenguaje

- Sintaxis
- Restricciones Contextuales
- Semántica
- Especificación informal  
vs
- Especificación formal

# Sintaxis

- Define la **forma** de los programas
- Define que **tokens** (símbolos) son permitidos en los programas
- Comandos, expresiones, declaraciones, programas completos
- Incluye las **Gramáticas libres de contexto**

# Gramáticas libres de contexto

Las gramáticas libres de contexto son un **tipo de gramática formal** en el que el *no terminal*  $V$  puede y siempre será sustituido por  $w$  sin tener en cuenta el contexto en el que ocurra.

$$V \rightarrow w$$

# Gramáticas libres de contexto

- Consisten en:
  - Terminales
    - Símbolos básicos
  - No Terminales
    - Variables sintácticas
  - Símbolo Inicial
    - Sus producciones “inician” la gramática
  - Producciones
    - Especifican cómo pueden relacionarse terminales y no terminales

# Gramáticas libres de contexto

programa  $\rightarrow$  bloque

bloque  $\rightarrow$  { decls instr }

...

instr  $\rightarrow$  if (expr) instr else instr

P  
R  
O  
D  
U  
C  
C  
I  
O  
N  
E  
S



# Gramáticas libres de contexto

programa  $\rightarrow$  bloque

bloque  $\rightarrow$  { decls instr }

...

instr  $\rightarrow$  if (expr) instr else instr



Terminales

P  
R  
O  
D  
U  
C  
C  
I  
O  
N  
E  
S

# Gramáticas libres de contexto

*programa*  $\rightarrow$  *bloque*

*bloque*  $\rightarrow$  { *decls instr* }

...

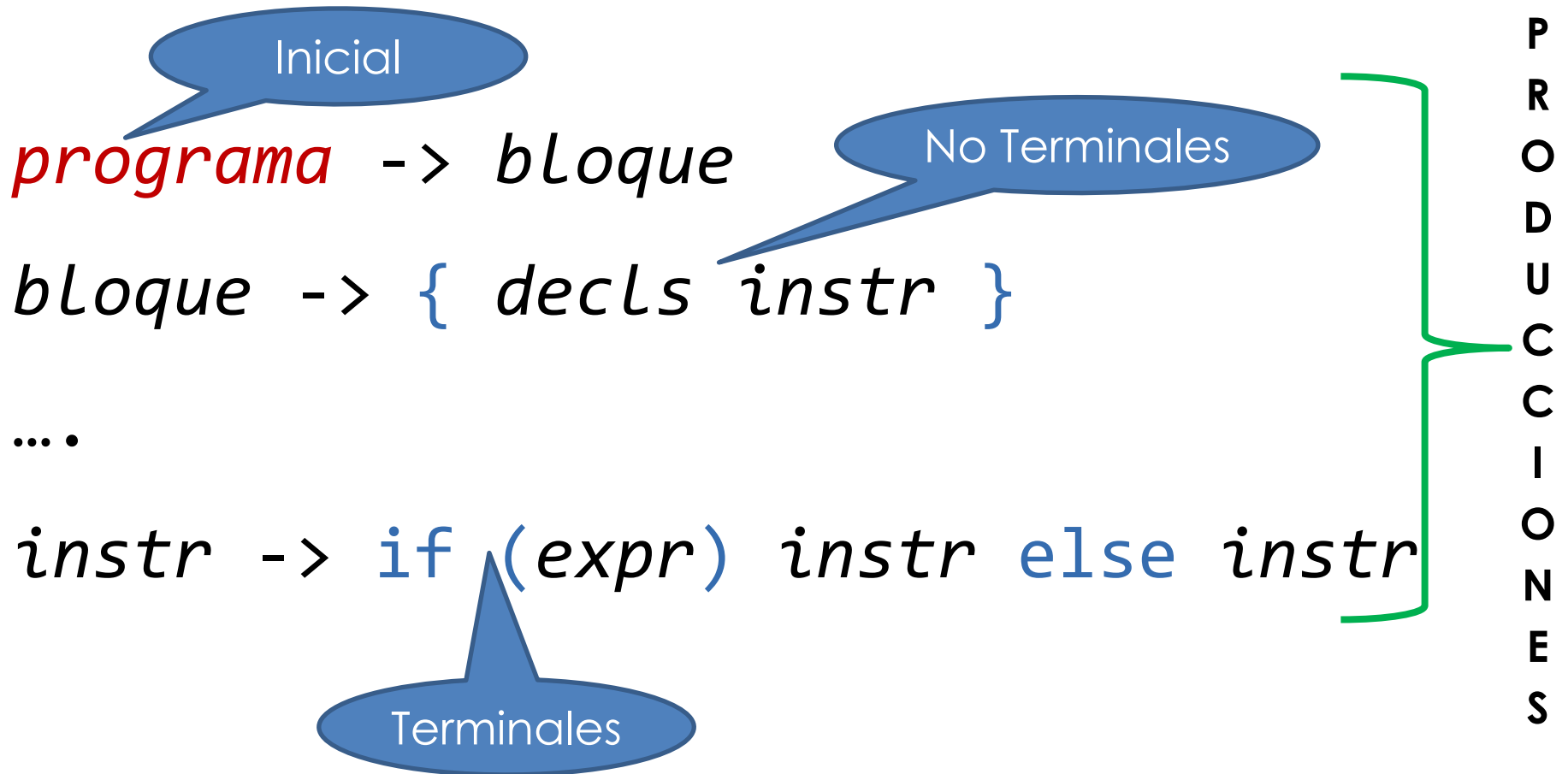
*instr*  $\rightarrow$  *if* (*expr*) *instr* *else* *instr*

No Terminales

Terminales

P  
R  
O  
D  
U  
C  
C  
I  
O  
N  
E  
S

# Gramáticas libres de contexto



# Restricciones Contextuales

- Incluye:
  - Reglas de **Scope** (alcance)
  - Reglas de **Tipos**
- Identifica si una frase está bien formada o no según su contexto

# Restricciones Contextuales

- La declaración de un identificador se le conoce como **ocurrencia de binding**.
- Las siguientes apariciones se les conoce como **ocurrencias aplicadas**.
- Temas de **binding** refieren a reglas de scope.

# Restricciones Contextuales

- Los valores son categorizados en tipos.
- Cada operación en el lenguaje tiene asociado una **regla de tipo**.
- Tipado **Estático** vs Tipado **Dinámico**

# Semántica

- Refiere al **significado** de los programas
  - Comportamiento de corrida
- Semántica de comandos, expresiones y declaraciones
- **Un comando es ejecutado para actualizar una variable**
  - [Tiene efecto secundario de realizar un input-output]

# Semántica

- Una expresión es evaluada para producir un valor.
  - [Tiene el efecto secundario de actualizar variables]
- Una declaración es elaborada para producir un binding (ligado)
  - [Tiene el efecto secundario de asignar (e inicializar) variables]



# Semántica

- Podemos obtener el significado de un programa a partir del comportamiento que tendrá cuando esté en ejecución.

## Especificación informal

- Escrita en algún **lenguaje natural** (Inglés, Español).
- Fácilmente entendida por usuarios de los lenguajes de programación
- Muy **difícil de hacerlas precisas**

# Especificación formal

- Escritas con una notación precisa
- Desambigua, consistente y completa.
- Las personas deben conocer la notación

# Procesadores de Lenguajes

# Procesadores de Lenguajes

- Un procesador de lenguaje es un programa que procesa programas expresados en un lenguaje en particular (Lenguaje fuente). Además es implementado en algún lenguaje.
- Tipos:
  - Traductores
    - Compiladores
  - Intérpretes

# Procesadores de Lenguajes

## Portafolio #1

- Lectura
  - Identificar características de los Traductores y Compiladores
  - Identificar características de los Intérpretes
  - Diferencias Compiladores vs Intérpretes
  - Explicar los diagramas Tombstone