

LENGUAJES DE PROGRAMACIÓN Y PROCESADORES DE LENGUAJES

Construcción de un compilador

MenosC

Parte-I: Análisis léxico sintáctico

El objetivo principal es la **construcción de un compilador** completo,
para un Lenguaje de Programación de alto nivel, sencillo pero no trivial

MenosC

Parte I Construcción del analizador léxico-sintáctico

⇒ límite de entrega **8 de noviembre de 2020**

Parte II Construcción del analizador semántico

⇒ límite de entrega **6 de diciembre de 2020**

Parte III Construcción del generador de código intermedio

⇒ límite de entrega **15 de enero de 2021**

[para la recuperación: **27 de enero de 2021**]

- Los identificadores son cadenas de letras (incluyendo “_”) y dígitos, que comienzan siempre por una letra. Debe distinguirse entre mayúsculas y minúsculas.
- Las palabras reservadas se deben escribir en minúscula.
- En un programa fuente puedan aparecer constantes enteras y reales; por ejemplo: 28 28. .55 28.55
- El signo + (ó –) de las constantes numéricas se tratará como un símbolo léxico independiente.
- Los espacios en blanco, retornos de línea y tabuladores deben ignorarse.
- Los comentarios deben ir precedidos por la doble barra (//) y terminar con el fin de la línea. Los comentarios no se pueden anidar.

programa	→ listaDeclaraciones
listaDeclaraciones	→ declaracion listaDeclaraciones declaracion
declaracion	→ declaracionVariable declaracionFuncion
declaracionVariable	→ tipoSimple id ; tipoSimple id [cte] ;
tipoSimple	→ int bool
declaracionFuncion	→ cabeceraFuncion bloque
cabeceraFuncion	→ tipoSimple id (parametrosFormales)
parametrosFormales	→ ϵ listaParametrosFormales
listaParametrosFormales	→ tipoSimple id tipoSimple id , listaParametrosFormales
bloque	→ { declaracionVariableLocal listaInstrucciones return expresion ; }
declaracionVariableLocal	→ ϵ declaracionVariableLocal declaracionVariable

ESPECIFICACIÓN SINTÁCTICA

listaInstrucciones	$\rightarrow \epsilon \mid \text{listaInstrucciones instrucion}$
instruccion	$\rightarrow \{ \text{listaInstrucciones} \} \mid \text{instruccionAsignacion}$ $\mid \text{instruccionSeleccion} \mid \text{instruccionEntradaSalida}$ $\mid \text{instruccionIteracion}$
instruccionAsignacion	$\rightarrow \mathbf{id} = \text{expresion} ; \mid \mathbf{id} [\text{expresion}] = \text{expresion} ;$
instruccionEntradaSalida	$\rightarrow \mathbf{read} (\mathbf{id}) ; \mid \mathbf{print} (\text{expresion}) ;$
instruccionSeleccion	$\rightarrow \mathbf{if} (\text{expresion}) \text{instruccion} \mathbf{else} \text{instruccion}$
instruccionIteracion	$\rightarrow \mathbf{for} (\text{expresionOpcional} ; \text{expresion} ; \text{expresionOpcional})$ instruccion
expresionOpcional	$\rightarrow \epsilon \mid \text{expresion} \mid \mathbf{id} = \text{expresion}$
expresion	$\rightarrow \text{expresionIgualdad}$ $\mid \text{expresion} \text{operadorLogico} \text{expresionIgualdad}$

expresionIgualdad	→ expresionRelacional expresionIgualdad operadorIgualdad expresionRelacional
expresionRelacional	→ expresionAditiva expresionRelacional operadorRelacional expresionAditiva
expresionAditiva	→ expresionMultiplicativa expresionAditiva operadorAditivo expresionMultiplicativa
expresionMultiplicativa	→ expresionUnaria expresionMultiplicativa operadorMultiplicativo expresionUnaria
expresionUnaria	→ expresionSufija operadorUnario expresionUnaria operadorIncremento id
expresionSufija	→ (expresion) id operadorIncremento id [expresion] id (parametrosActuales) id constante

parametrosActuales	→	ε		listaParametrosActuales
listaParametrosActuales	→	expresion		expresion , listaParametrosActuales
constante	→	cte		true false
operadorLogico	→	&&		
operadorIgualdad	→	==		!=
operadorRelacional	→	>		< >= <=
operadorAditivo	→	+		-
operadorMultiplicativo	→	*		/
operadorUnario	→	+		- !
operadorIncremento	→	++		--