

Clase #18 de 27

Categorías Léxicas y Sintácticas

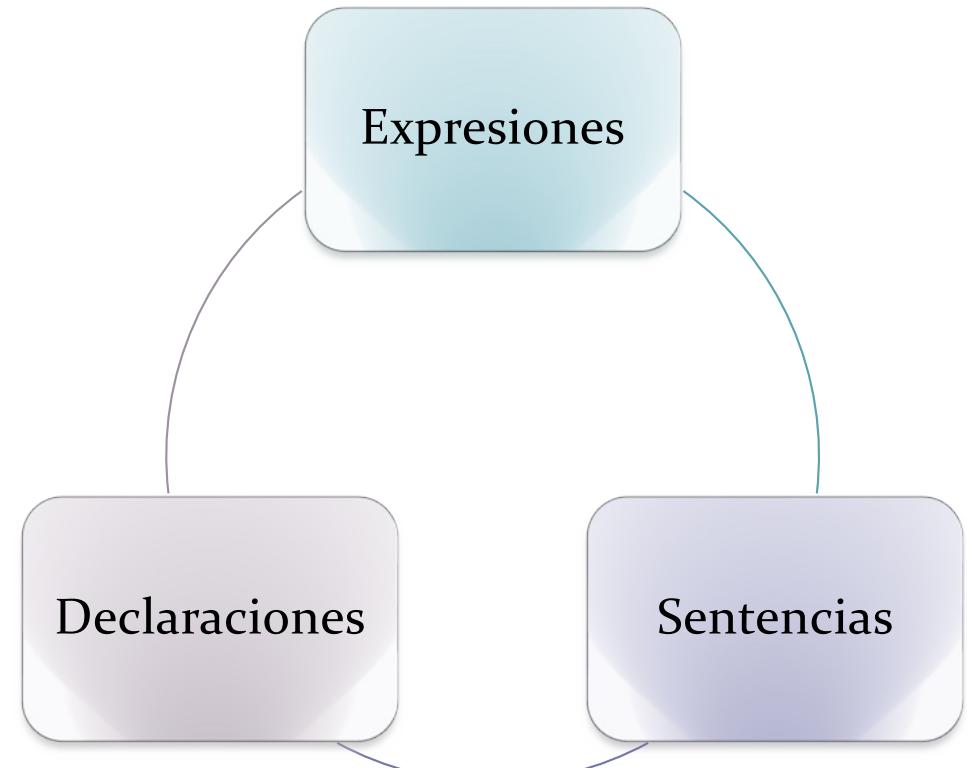
Proceso de Compilación

*Sep 2, Martes
Sep 3, Miércoles*

Categorías Léxicas y Sintácticas

Categorías Sintácticas

- Categorías Léxicas ó Tokens
 - Identificadores
 - Palabras reservadas
 - Literales
 - Enteros
 - Reales
 - Caracteres
 - Cadenas
 - Otros
 - Operadores y puntuación
- Categorías Sintácticas ó Gramaticales ó "constructos sintácticos"
 - Expresiones
 - Declaraciones
 - Sentencias
- ¿Cómo se relacionan?
- ¿Cuál es el "alfabeto de las categorías sintácticas? ¿Es finito?



Categorías

- Categorías Léxicas ó Tokens
 - Identificadores
 - Palabras reservadas
 - Literales
 - Enteros
 - Reales
 - Caracteres
 - Cadenas
 - Operadores y puntuación
 - Otros
- ¿Cuántos hay?
- Categorías Sintácticas ó Gramaticales ó Frases
 - Expresiones
 - Declaraciones
 - Sentencias
- ¿Cuántos hay?
- Frase principal
 - Unidad de Traducción

Categorías Léxicas

Categorías Léxicas

- Categorías Léxicas ó Tokens
 - Identificadores
 - Palabras Clave
 - ¿Qué diferencia con Identificadores?
 - ¿Qué diferencia con Palabras reservadas?
 - Literales
 - Enteros
 - Reales
 - Caracteres
 - Cadenas
 - Operadores y puntuación
 - Otros
- ¿Cuántos hay?
- Leer K&R1988
 - A.1
 - A.2
- Leer MUCH2012
 - V1 s3.6.2.

Keywords

32 en C90, 54 en C23

6.4.1 Keywords

Syntax

keyword: one of

alignas	do	int	struct	while
alignof	double	long	switch	_Atomic
auto	else	nullptr	thread_local	_BitInt
bool	enum	register	true	_Complex
break	extern	restrict	typedef	_Decimal128
case	false	return	typeof	_Decimal32
char	float	short	typeof_unqual	_Decimal64
const	for	signed	union	_Generic
constexpr	goto	sizeof	unsigned	_Imaginary
continue	if	static	void	_Noreturn
default	inline	static_assert	volatile	

Semantics

The previously listed tokens (case sensitive) are reserved (in translation phases 7 and 8) for use as keywords except in an attribute token, and shall not be used otherwise. The keyword **_Imaginary** is reserved for specifying imaginary types.⁶¹⁾

Keywords

32 en C90, 54 en C23

6.1.1 Keywords

Syntax

keyword one of

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

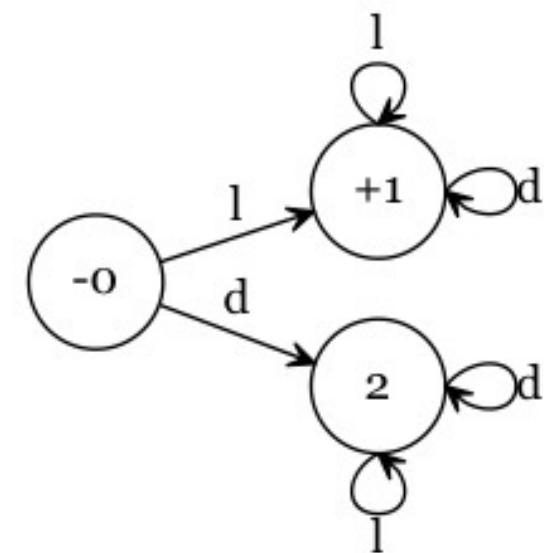
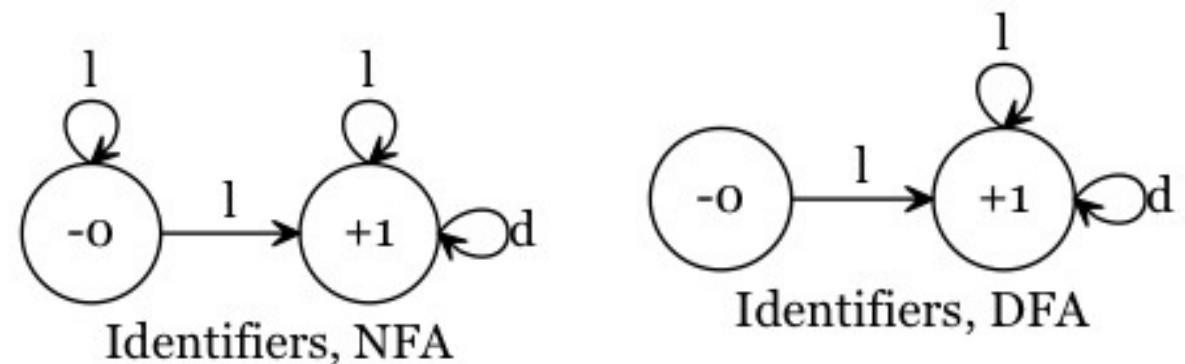
Semantics

The above tokens (entirely in lowercase) are reserved (in translation phases 7 and 8) for use as keywords, and shall not be used otherwise.

Autómata para Identificadores

Componentes de un Autómata Finito

- Estados
 - Inicial
 - Finales
- Transición
 - Origen
 - Evento
 - Destino
 - Transición reflexiva
- Alfabeto
- Ejemplo: Identificadores



Identifiers, Complete DFA

El Proceso de Compilación

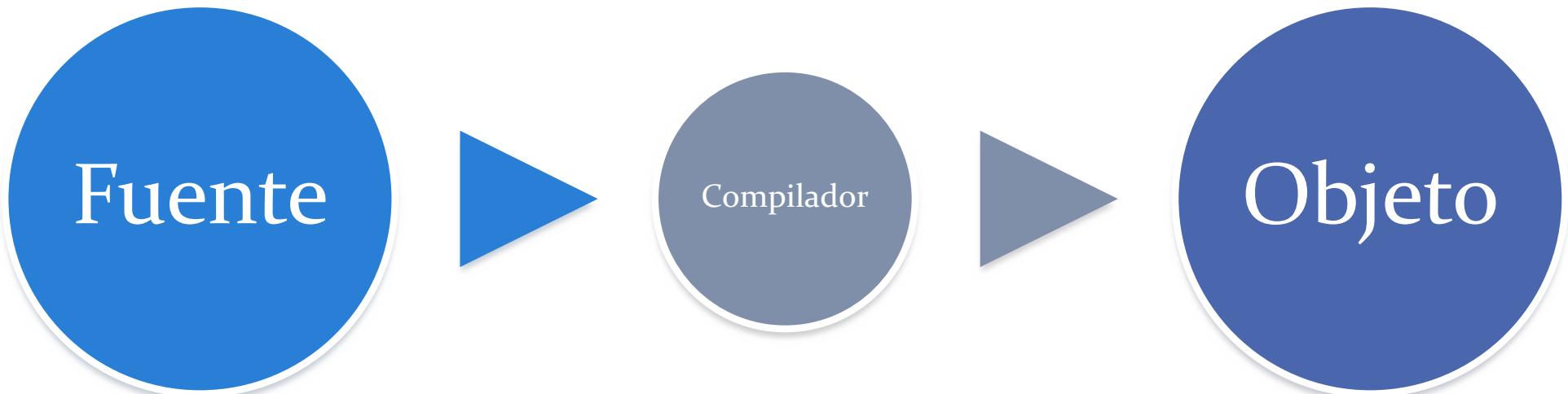
Los Tiempos y Los Ambientes

- Diseño
- Ambiente de Traducción
 1. Preprocesamiento
 2. Compilación
 3. Vinculación (Link)
- Ambiente de Ejecución (Run-Time)
 - Inicio
 - Ejecución: Biblioteca Estándar
 - Finalización
- Lenguajes Interpretados

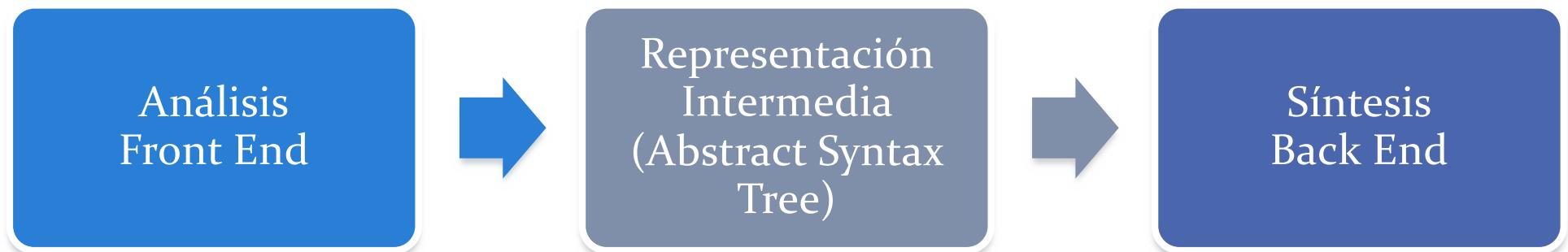
Antes y después del Compilador



El Compilador desde afuera – Compilador como Función



Dentro del Compilador



Front End:

Análisis por Niveles del Lenguaje

Analizador Léxico/ Scanner

Analizador Sintáctico / Parser

Analizador Semántico Estático / Verificador de restricciones

Analizador Pragmático / Analizador de código estático: Warnings (e.g, Dead code)

Productos de cada Analizador



- Secuencia de caracteres



- Secuencia de tokens



- Árbol de Sintaxis Concreta ó
Árbol de Sintaxis Abstracta



- Árbol de Sintaxis Abstracta Anotado o
Representación Intermedia (IR)

Ejemplos de Árboles Sintácticos Concretos y Abstractos, y Representación Intermedia

- Sentencias
 - If
 - If else
 - For
 - While
- Expresión
 - Suma
 - Subindicación
 - Invocación

¿Consultas?



Fin de la clase