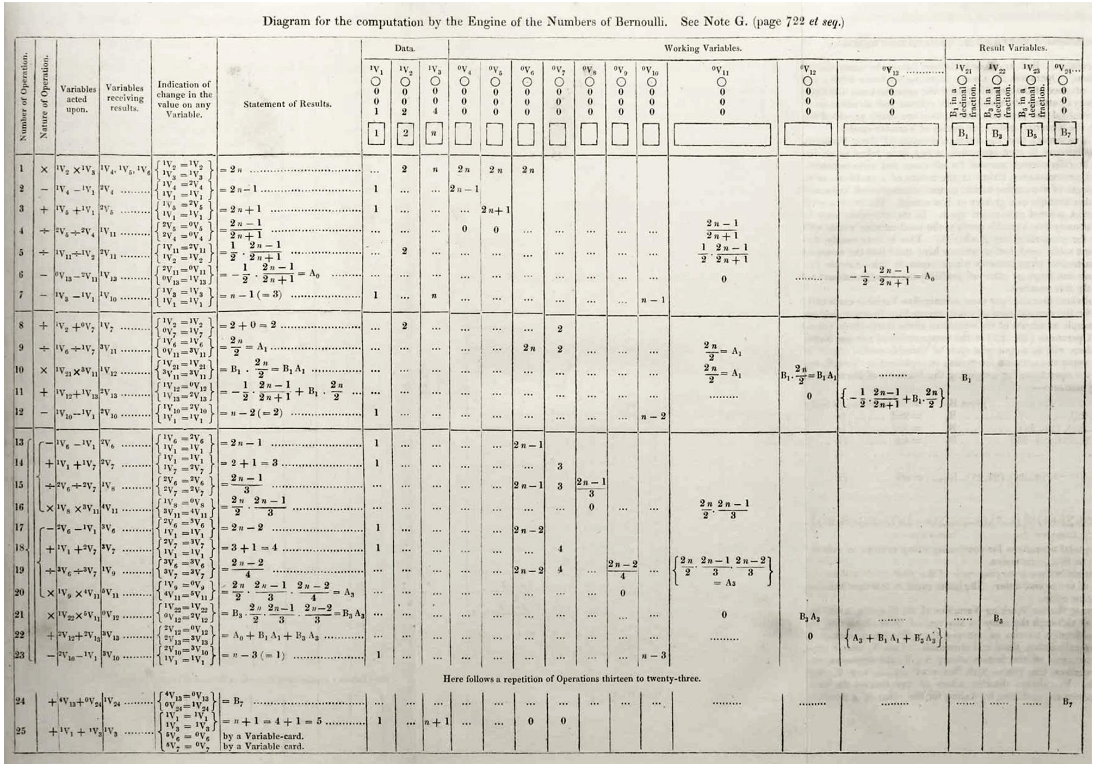
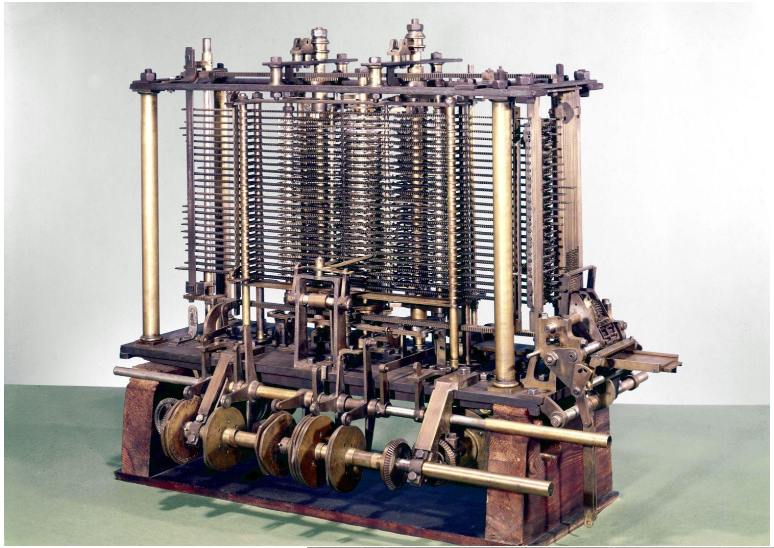
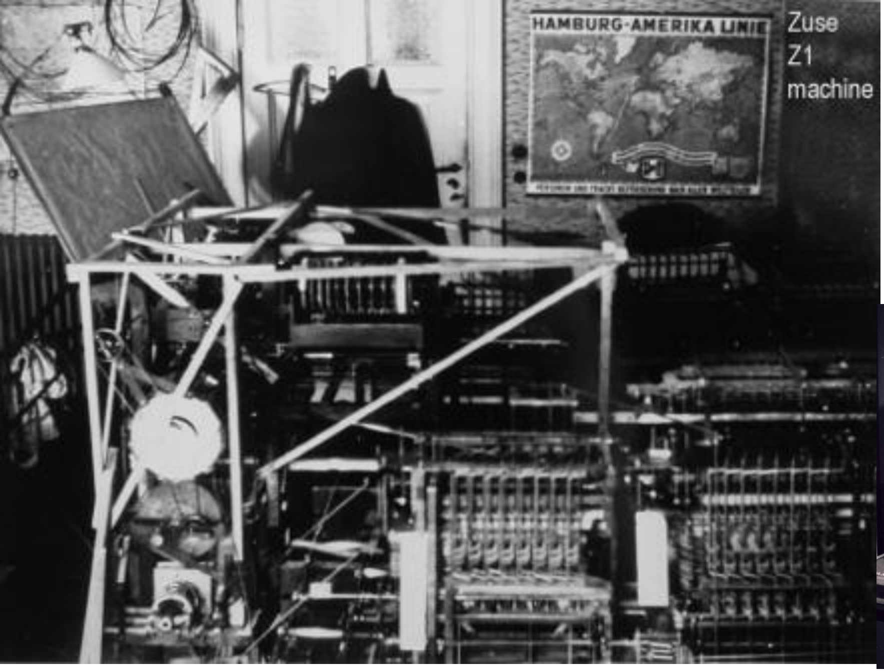
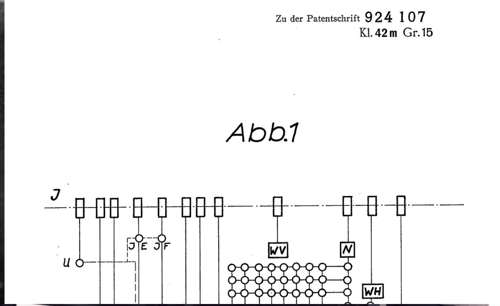
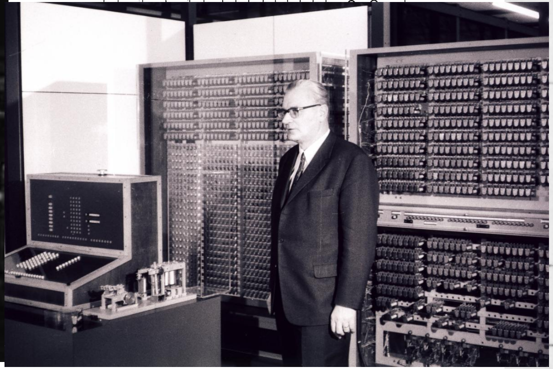


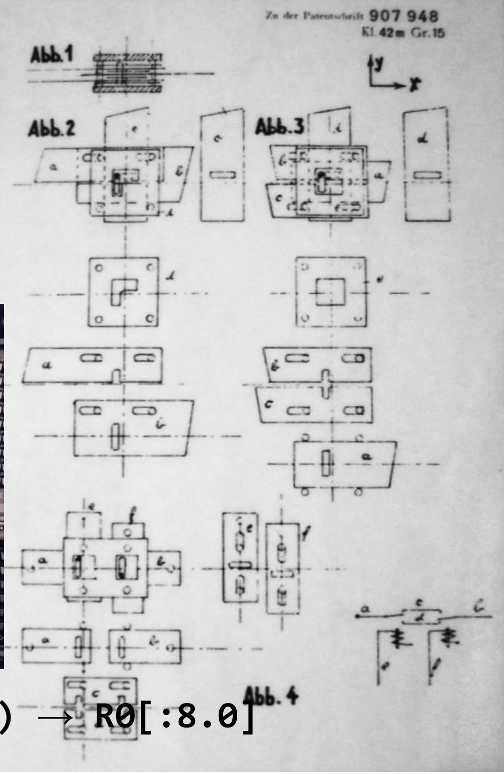


… La nota G estaba dedicada a los números de Bernoulli; en este apartado Ada describe con detalle las operaciones mediante las cuales las tarjetas perforadas "tejerían" una secuencia de números en la máquina analítica. Este código está considerado como el primer algoritmo específicamente diseñado para ser ejecutado por un ordenador, aunque nunca fue probado ya que la máquina nunca llegó a construirse.
(Wikipedia)







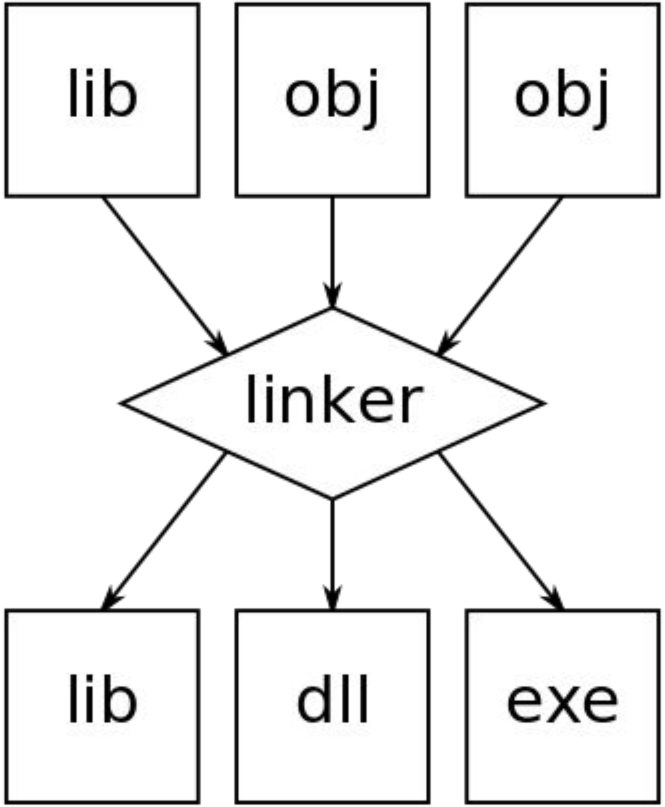


En 1935 Zuse construye la Z1. Leía las instrucciones desde una cinta perforada de 35 mm. No era una máquina Turing completa

```
P2 max (V0[:8.0],V1[:8.0]) → R0[:8.0]
V0[:8.0] → Z1[:8.0]
(Z1[:8.0] < V1[:8.0]) → V1[:8.0] → Z1[:8.0]
Z1[:8.0] → R0[:8.0]
END
```

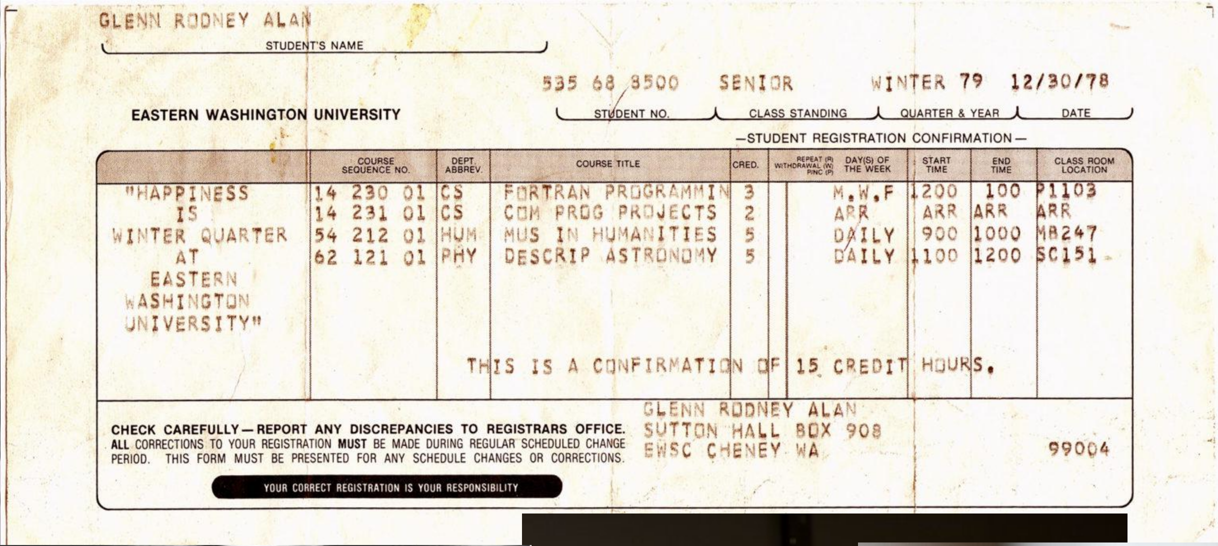

The Henschel Hs 129B ground attack aircraft

La Z3 (1941) era un computador binario de punto flotante de 22 bits con memoria y unidad de cálculo basada en relés telefónicos. No almacenaba el programa en memoria. A pesar de la ausencia de saltos condicionales, el Z3 era un ordenador Turing-completo



Rear Admiral (then Commodore) Grace M. Hopper, 1984. By 1952, Hopper had finished her program linker. The term compiler was coined by Hopper. It was written for the A-0 System.

The Algol 60 people



GLENN RODNEY ALAN
STUDENT'S NAME

EASTERN WASHINGTON UNIVERSITY

535 68 8500	SENIOR	WINTER 79	12/30/78
STUDENT NO.	CLASS STANDING	QUARTER & YEAR	DATE

—STUDENT REGISTRATION CONFIRMATION—

	COURSE SEQUENCE NO.	DEPT. ABBREV.	COURSE TITLE	CRED.	REPEAT (R) WITHDRAWAL (W) P/NC (P)	DAY(S) OF THE WEEK	START TIME	END TIME	CLASS ROOM LOCATION
"HAPPINESS IS WINTER QUARTER AT EASTERN WASHINGTON UNIVERSITY"	14 230 01	CS	FORTRAN PROGRAMMIN	3		M,W,F	1200	100	P1103
	14 231 01	CS	COM PROG PROJECTS	2		ARR	ARR	ARR	ARR
	54 212 01	HUM	MUS IN HUMANITIES	5		DAILY	900	1000	MB247
	62 121 01	PHY	DESCRIP ASTRONOMY	5		DAILY	1100	1200	SC151

THIS IS A CONFIRMATION OF 15 CREDIT HOURS.

CHECK CAREFULLY—REPORT ANY DISCREPANCIES TO REGISTRARS OFFICE.
ALL CORRECTIONS TO YOUR REGISTRATION MUST BE MADE DURING REGULAR SCHEDULED CHANGE PERIOD. THIS FORM MUST BE PRESENTED FOR ANY SCHEDULE CHANGES OR CORRECTIONS.

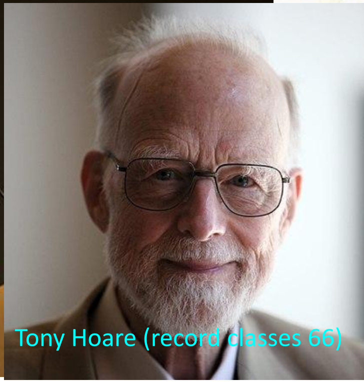
GLENN RODNEY ALAN
SUTTON HALL BOX 908
EWSC CHENEY WA 99004

YOUR CORRECT REGISTRATION IS YOUR RESPONSIBILITY

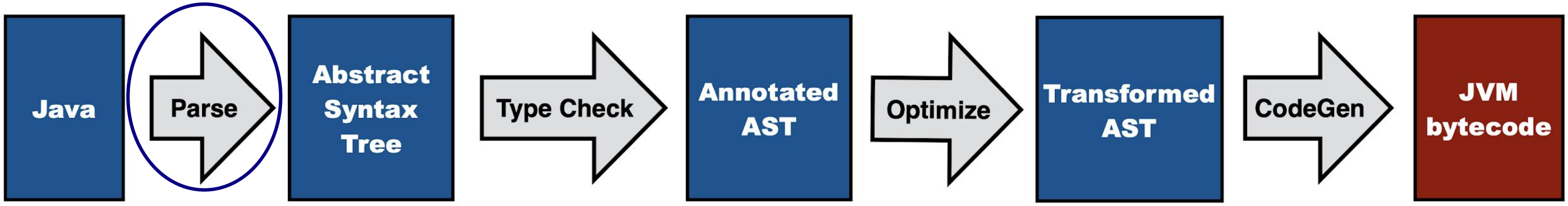
[OO History: Simula and Smallta lk]


Kristin Nygaard and Ole-Johan Dahl at the Norwegian Computing Center. Simula67


Alan Kay (Smalltalk)


Tony Hoare (record classes 66)

The Phases of a Translator



A programming language translator usually consists of a sequence of stages

Lexer:

- Skips the comments and whitespaces and produces the stream of tokens for numbers, identifiers, reserved words, etc

Parser:

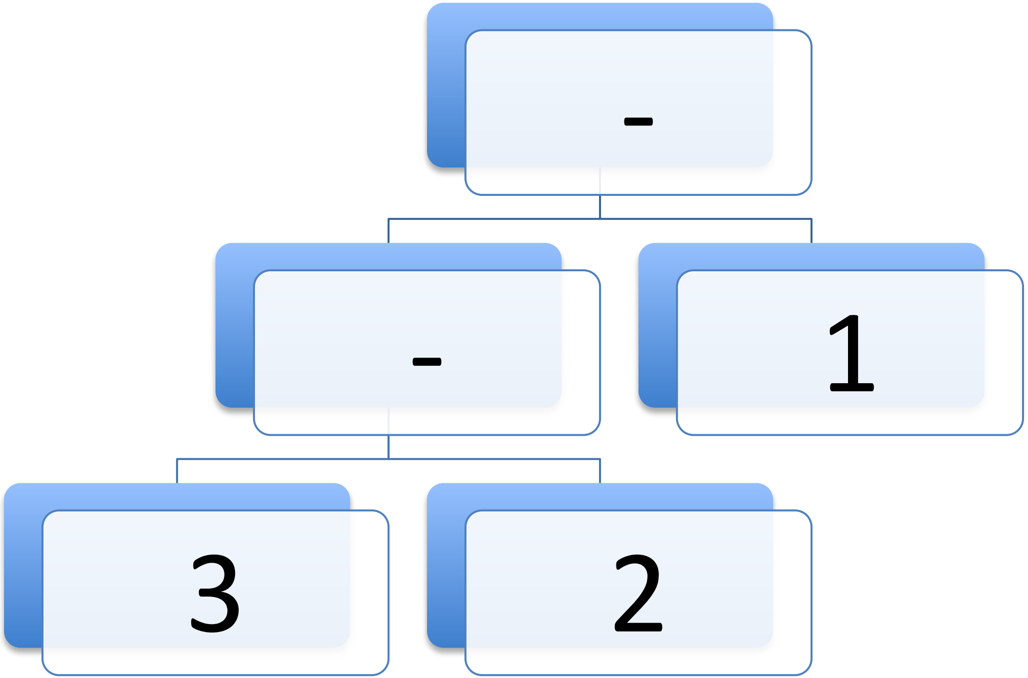
- Reads the stream of tokens, check that it complies with the syntactic rules and produces the *Abstract Syntax Tree*: a data structure representing the underlying syntactic structure of the input program

3 - 2 - 1

Árbol Sintáctico Abstracto

3-2-1

(3-2)-1

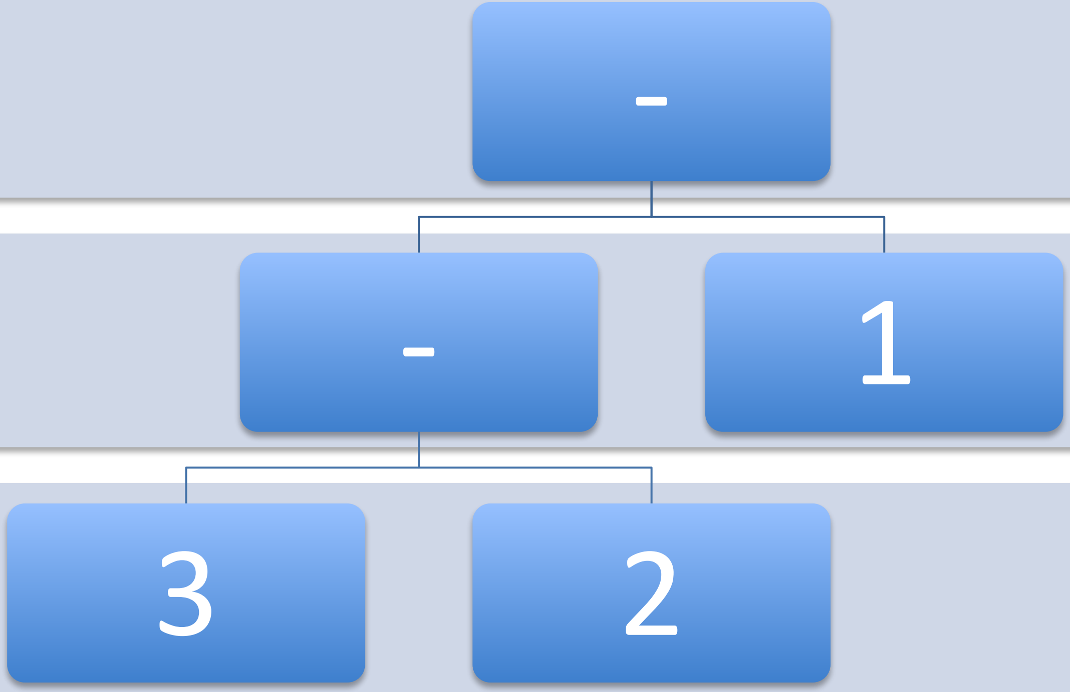


Semántica 3 - 2 - 1

0 = 1 - 1

1 = 3 -2

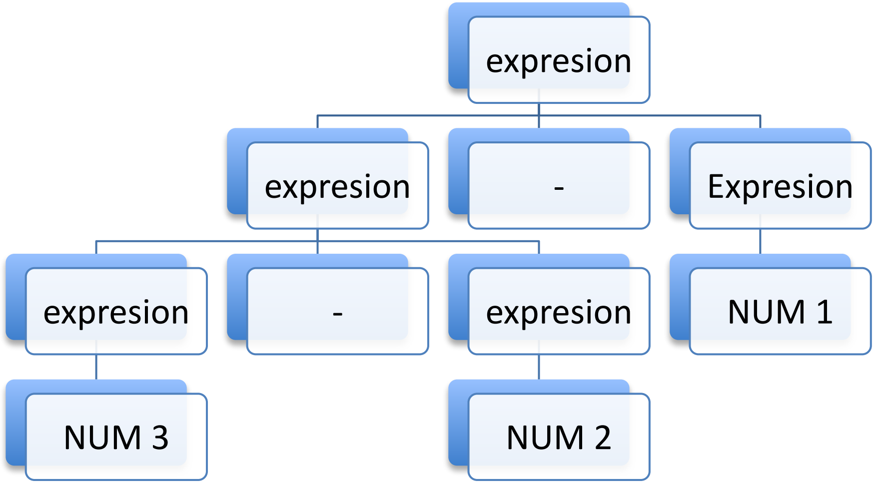
‘3’



Gramática Independiente del Contexto

- expresion ⟶ expresion '-' expresion
- expresion ⟶ NUMERO

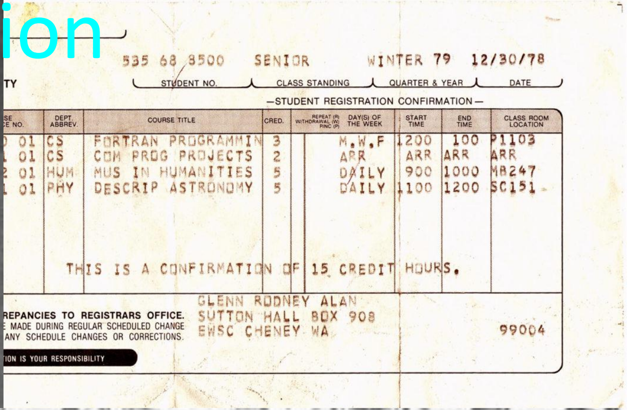
3-2-1



Context Free Grammars

- expresion ⟶ expresion '-' expresion
- expresion ⟶ NUMERO

Noam Chomsky teaching linguistics (1956)



535 68 8500 | SENIOR | WINTER 79 | 12/30/78

STUDENT NO. | CLASS STANDING | QUARTER & YEAR | DATE

—STUDENT REGISTRATION CONFIRMATION—

SE NO.	DEPT ABBREV.	COURSE TITLE	CRED.	REPEAT (R) WITHDRAWAL (W) PINC (P)	DAY(S) OF THE WEEK	START TIME	END TIME	CLASS ROOM LOCATION
01	CS	FORTRAN PROGRAMMIN	3		M,W,F	1200	100	P1103
01	CS	COM PROG PROJECTS	2		ARR	ARR	ARR	ARR
01	HUM	MUS IN HUMANITIES	5		DAILY	900	1000	MB247
01	PHY	DESCRIP ASTRONOMY	5		DAILY	1100	1200	SC151

THIS IS A CONFIRMATION OF 15 CREDIT HOURS.

REPANCIES TO REGISTRARS OFFICE. MADE DURING REGULAR SCHEDULED CHANGE ANY SCHEDULE CHANGES OR CORRECTIONS.

GLENN RODNEY ALAN
SUTTON HALL BOX 908
EWSC CHENEY WA 99004

ION IS YOUR RESPONSIBILITY

Mccarthy

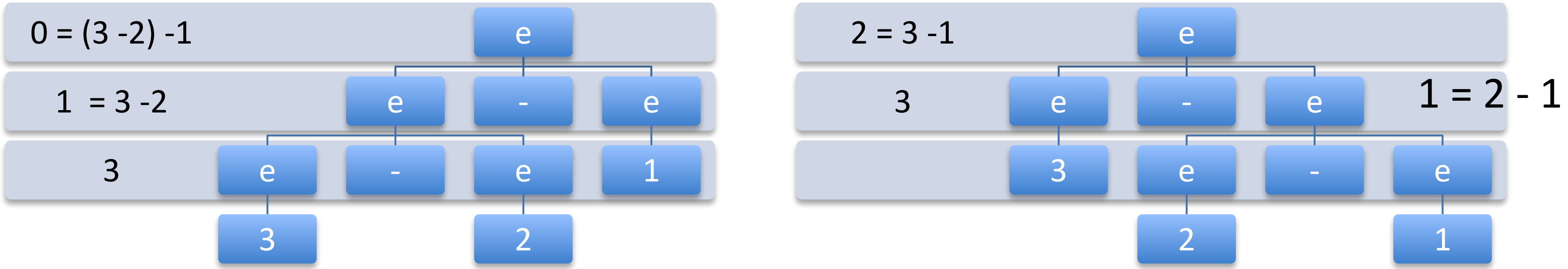
The Algol 60 people

Backus

Naur

Gramática Ambigua

- expresion ⟶ expresion '-' expresion
- expresion ⟶ NUMERO

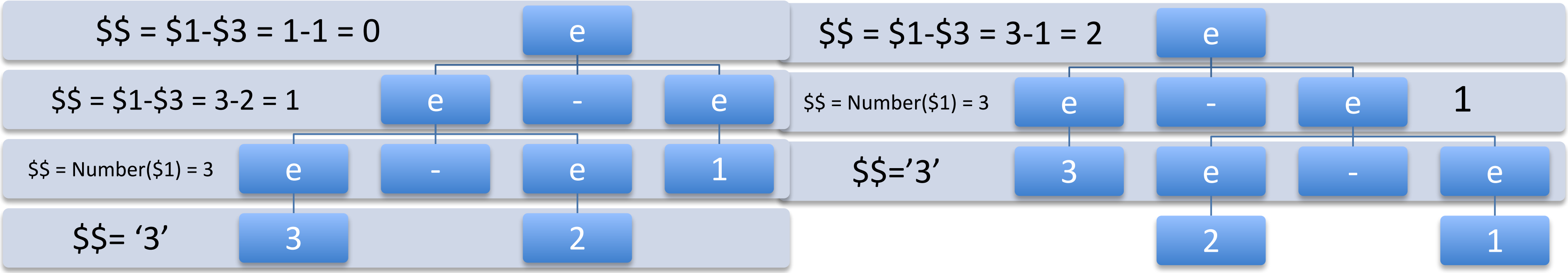


Esquema de Traducción (yacc)

e $\longrightarrow$ e '-' e { $$ = $1 - $3; }

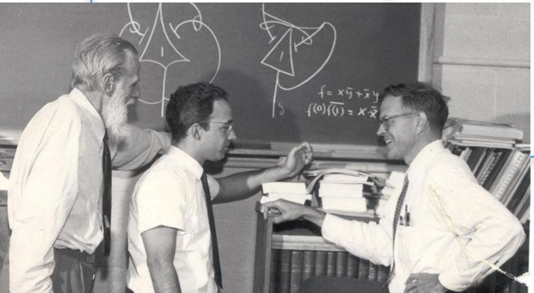
e $\longrightarrow$ NUM { $$ = Number($1); }

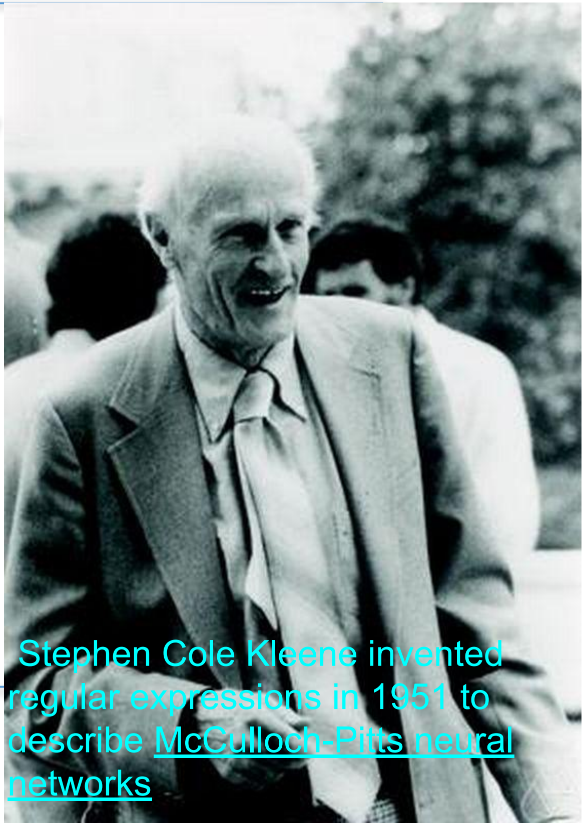
3–2-1



Análisis Léxico y Expresiones Regulares

```
[0-9]+ /* is a Natural Number */
"-"    /* is a '-' */
.      /*Any character but \n*/
```





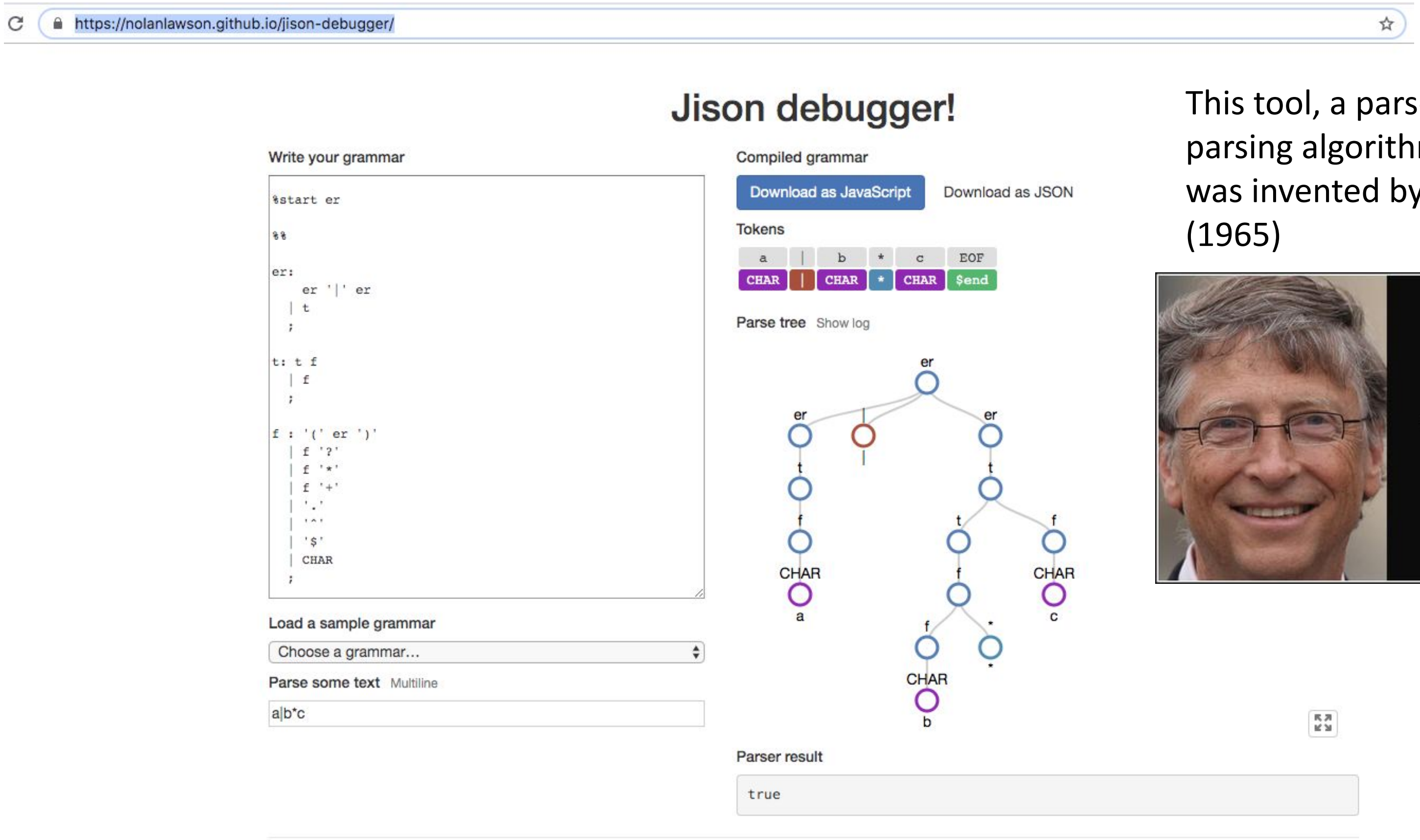
Stephen Cole Kleene invented regular expressions in 1951 to describe McCulloch-Pitts neural networks

Un Programa que Evalúa Expresiones

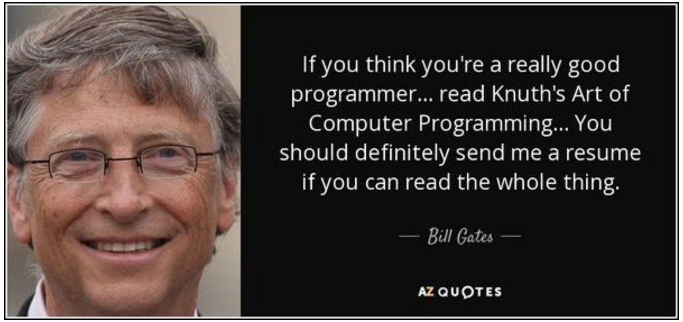
https://nolanlawson.github.io/jison-debugger/

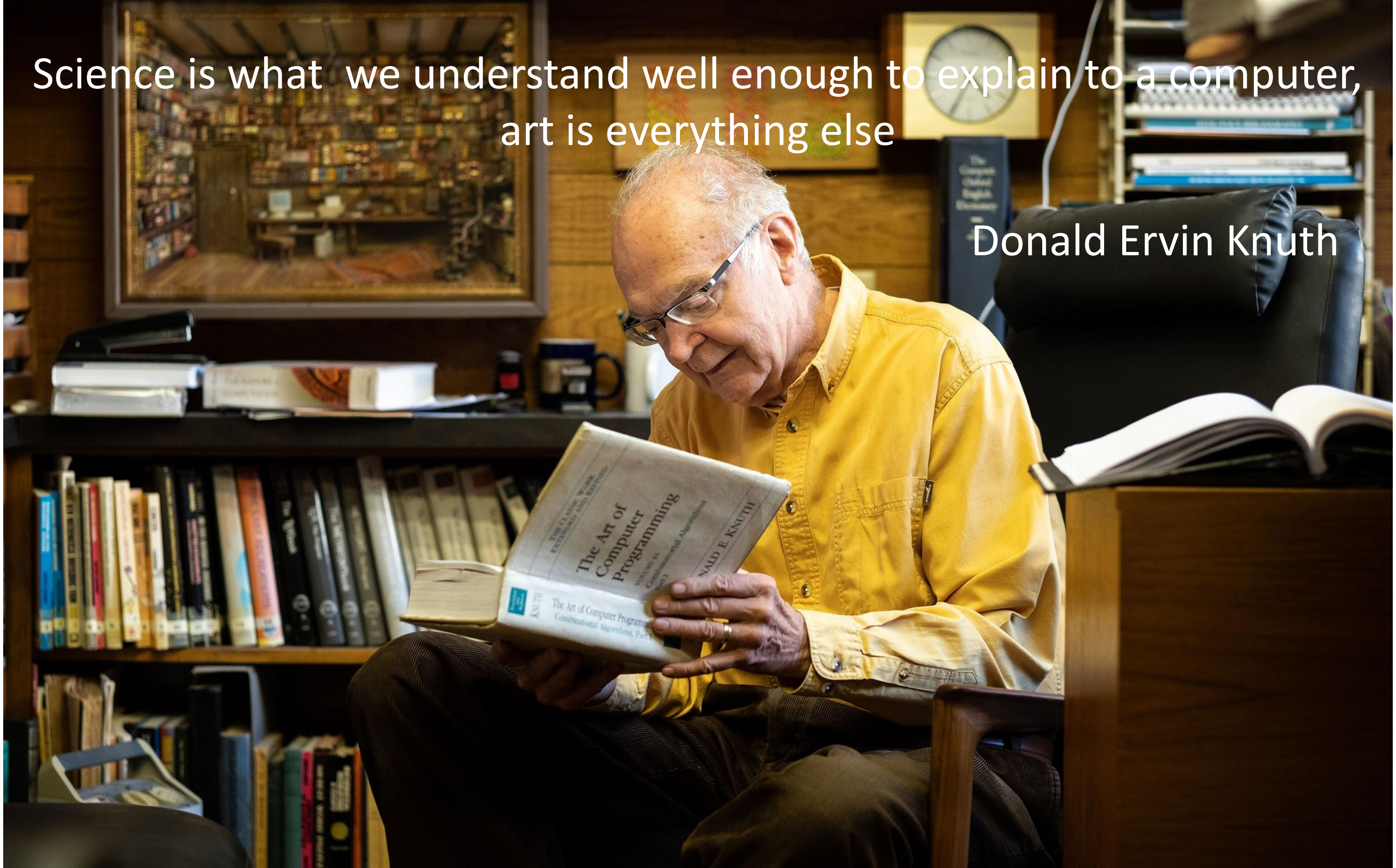
```
%lex
%%
[0-9]+              return 'NUMBER'
"-"                 return '-'
.                   return 'INVALID'
/lex

%%
es: e             {return $1} ;
e : e '-' e       {$$ = $1-$3}
  | NUMBER        {$$ = Number($1)} ;
```

Parser Generators: an example

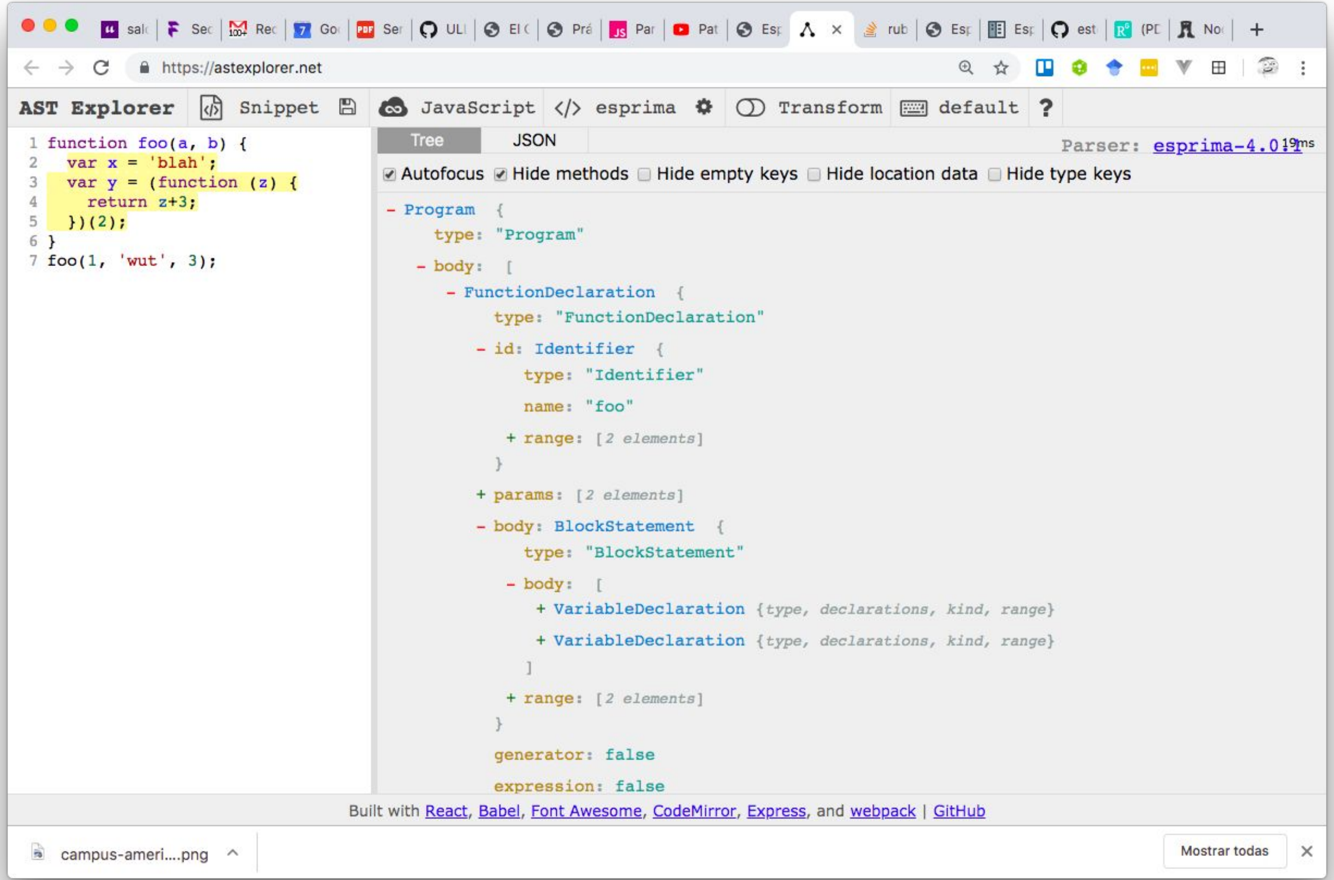


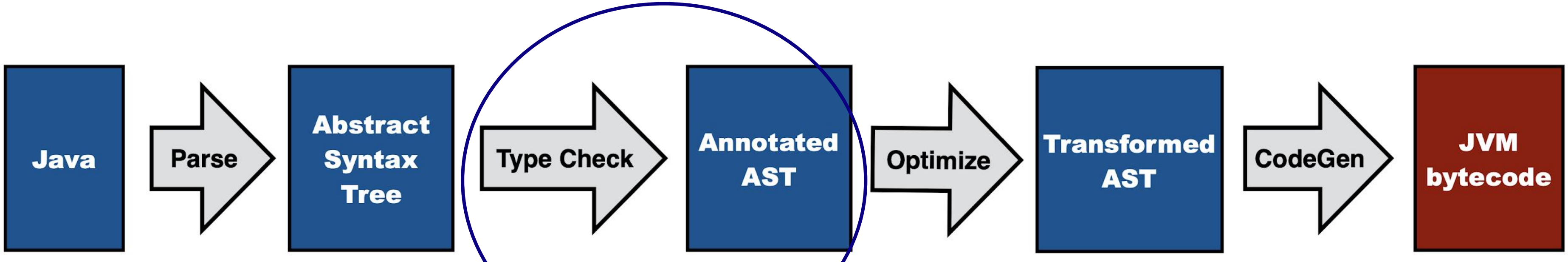
This tool, a parser generator uses a parsing algorithm known as LALR that was invented by Donald Ervin Knuth (1965)




Science is what we understand well enough to explain to a computer,
art is everything else
Donald Ervin Knuth
The Art of Computer Programming

The *Abstract Syntax Tree*: a data structure representing the underlying syntactic structure of the input program: https://astexplorer.net/



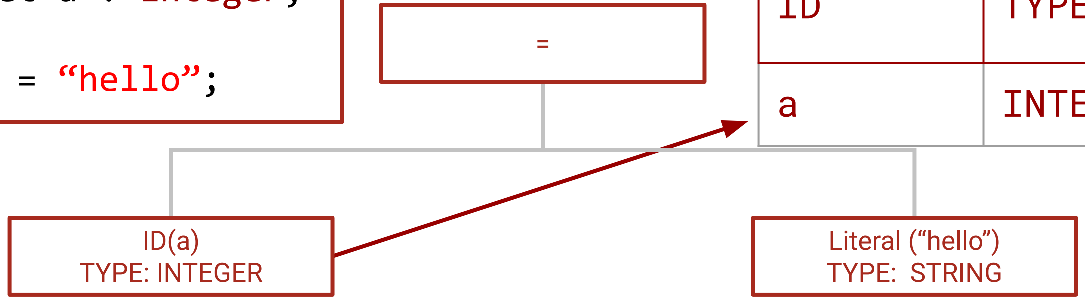


- Receives as input the abstract syntax tree
- Checks that the program complies with the static semantic rules of the language
- Performs name analysis, relating uses of names to declarations of names
- Checks that the types of arguments of operations are consistent with their specification

Input Program

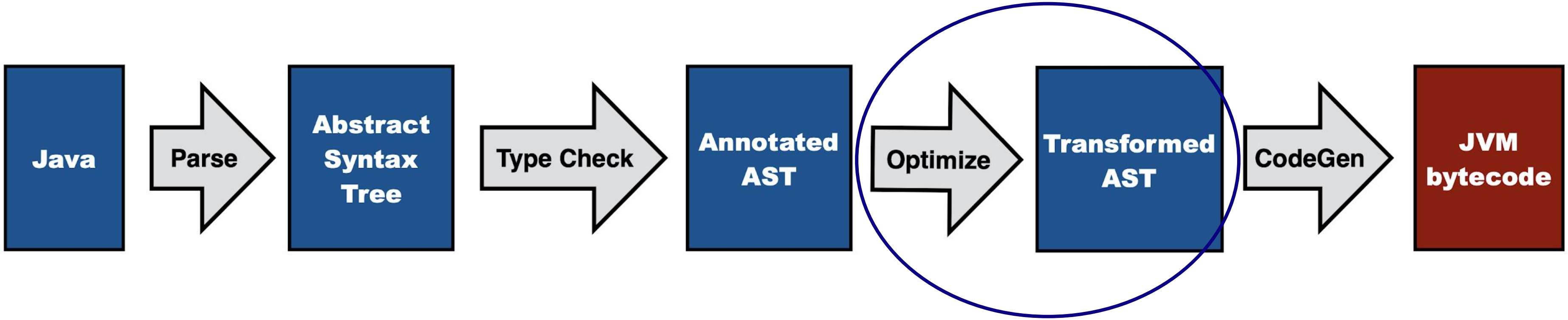
```
let a : integer;

a = "hello";
```

AST



Symbol Table

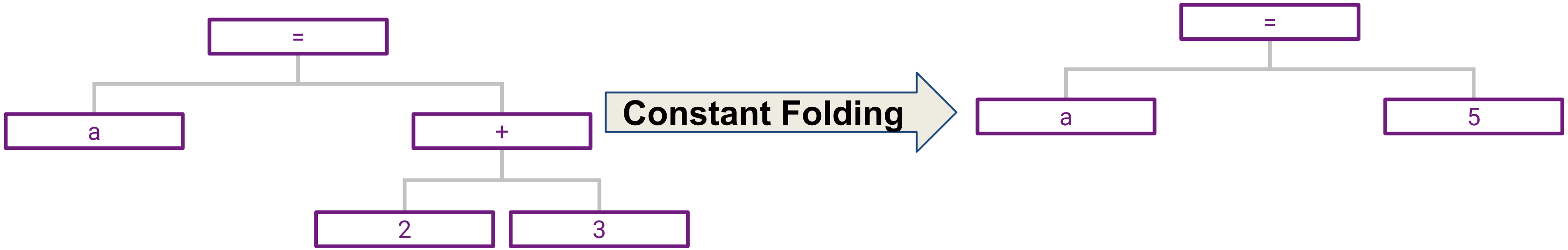
ID	TYPE
a	INTEGER

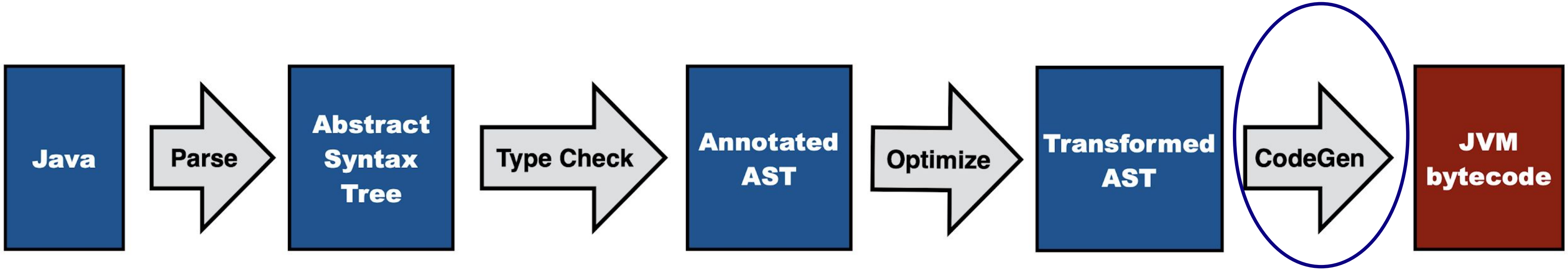


- Applies transformations that improve the program in various goals
- Goals: execution time, memory consumption, energy consumption, etc.
- Examples of transformations: Constant folding, Constant propagation, Loop invariants

Input Program

```
a = 2+3;
```

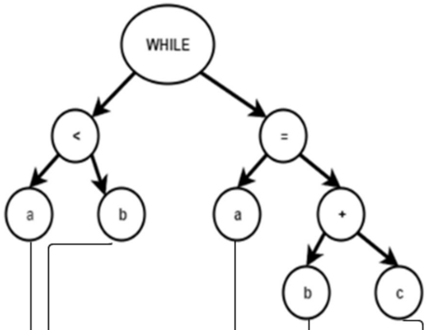




- Transforms abstract syntax tree to instructions for a particular computer architecture

Input Program

```
while (a < b) do
  a = b + c
end while
```



SYMBOL TABLE

VARIABLE NAME	MEMORY ADDRESS
a	4096
b	4097
c	4098

CODE GENERATION

```
//Translating guard
L1:
MOV R0,[4096]
MOV R1,[4097]
LT R0,R1
JZ R0,L2
```

```
//Translating body
MOV R0,[4097]
MOV R1,[4098]
ADD R0,R1
MOV [4096],R0
JMP L1
L2:
```