

Laboratorio 3 - Small20. Sintassi Astratta e Concreta: Scrittura, Rappresentazione e Stampa dei Programmi

Sommario: 09 Aprile, 2020

- Sintassi Astratta: Rivista e Corretta
- Sintassi Concreta: Una CFG per Small20
- Attività: Implementazione AST

Sintassi Astratta: Gli AST di Small20

Type ::= [int] | [bool] | [void]
| [arr] Type Num | [mut] Type | [terr]
| [abs] Type TypeSeq

TypeSeq ::= Type [x] Type | Type

Dcl ::= [const] Type Ide Exp | [var] Type Ide Exp
| [varN] Type Ide
| [array] Type Ide Num

Exp ::= [val] Ide | Num | Bool | Ide [↑] Exp | [-₁]Exp
| Exp [+] Exp | Exp [-] Exp | Exp [*] Exp | Exp [div] Exp
| Exp [==] Exp | Exp [<] Exp | Exp [>] Exp
| [not] Exp | Exp [or] Exp | Exp [and] Exp
| Exp [=] Exp | [emptyE]

Cmd ::= Cmd [seqC] Cmd
| [ifE] Exp Cmd Cmd | [ifT] Exp Cmd
| [for] Stm Exp Stm Cmd | Exp | [emptyC]

Stm ::= Dcl | Exp | Cmd | Stm [seqM] Stm | [emptyS]

Prog ::= [prog] Ide Stm

Sintassi Concreta: Una CFG per Small20

Type ::= Simple | void | Simple [Num]
Simple ::= int | bool

Dcl ::= final Simple ide = Exp | var Simple ide = Exp
| var Simple ide | Simple array ide[num]

Exp ::= Exp or ExpB1 | ExpB1
ExpB1 ::= ExpB1 and ExpB | ExpB
ExpB ::= not ExpB | truth | ExpR
ExpR ::= ExpR == ExpA2 | ExpR < ExpA2 | ExpR > ExpA2 | ExpA2
ExpA2 ::= ExpA2 + ExpA1 | ExpA2 - ExpA1 | ExpA1
ExpA1 ::= ExpA1 * ExpA | ExpA1 / ExpA | ExpA
ExpA ::= num | DExp | DExp = Exp | (Exp) | - ExpA2
DExp ::= ide | ide [ExpA2] | ϵ

Cmd ::= if (ExpB2) Cmd | OtherCmd
OtherCmd ::= if (ExpB2) OtherCmd else Cmd | NonConditionalCmd
NonConditionalCmd ::= for (Stm; Exp; Stm) Cmd | Exp | {cmds}
Cmds ::= Cmd; | Cmd; Cmds

Stm ::= Dcl | Cmd
Stms ::= Stm; | Stm; Stms

Prog ::= Program ide {Stms}

Implementazione AST di Small20: 6 Esercizi

Esercizio (1)

Completare il codice relativo a `toStringTye` e `toStringTseq`.

Esercizio (2)

Completare definizione `toStringDcl`.

Esercizio (3)

Correggere definizione tipo `exp` e fornire relativo `toStringExp`.

Esercizio (4)

Correggere definizione tipo `cmd` e fornire relativo `toStringCmd`.

Esercizio (5)

Correggere definizione tipo `stm` e fornire relativo `toStringStm`.

Esercizio (6)

Correggere definizione tipo `prog` e fornire relativo `toStringPrg`.