12 April 2025
Saturday

Type Expression

It is not mandatory that every Production Rule will have a Semantic Rule attached with it

$T \rightarrow BC \quad \{ T.t = C.t \; ; \; C.b = B.t \; ; \}$

$B \rightarrow int \quad \{ B.t = Integer \; ; \}$

$B \rightarrow float \quad \{ B.t = Float \; ; \}$

$C \rightarrow [num] \; C_1 \quad \{ C.t = array(num.val, C_1.t) \; ;$
$C_1.b = C.b \; ; \}$

(C = parent, C_1 = child)

$C \rightarrow \epsilon \quad \{ C.t = C.b \; ; \}$ নিজের value নিজে নিয়েছে

Basic Types → int, float, double, char, etc.

Composite Type → array, linked list

in the given SDD,
B represents Basic Types
C represents Composite Type

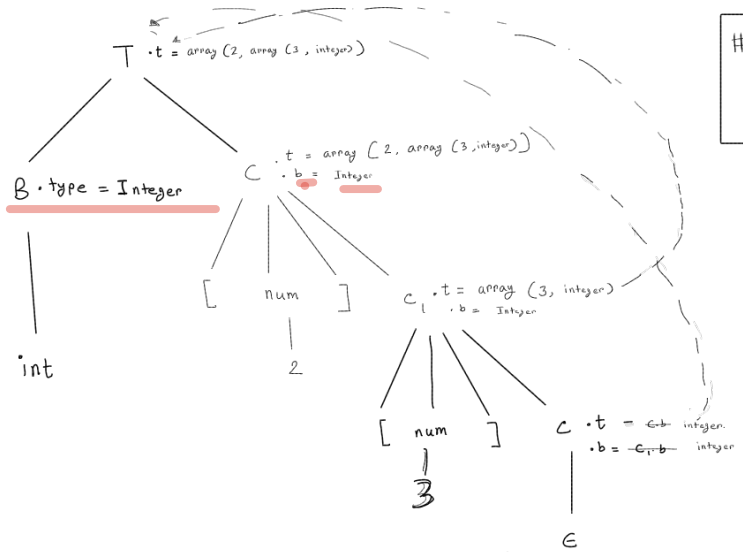
Input String: int [2] [3]
- int → integer array
- [2] → 2 টা row
- [3] → 3 টা column

Generate Type expression for the input string.

$C \rightarrow [num] \; C_1$
এখানে প্রত্যেকের জন্য আলাদা আলাদা branch হবে।

Type expression basically attribute evaluate করা।

ϵ এর জন্য কোনো Type expression consider হবে না



এখানে, C.t = C.b যে
C.b এর value কই থেকে পাবো,
↳ এরজন্য situation root node এ point করছে।
- evaluate attribute value.

যে attribute এর value এখনও পাইনি but যেটা handle করার জন্য এখন root node কে refer করতে বলা হয়,

Parent এর T attribute.

$C_1.b = C.b$
$C.b = C_1.b$
$C_1.t = array(num.val, C.t)$

with the help of Type expression Computer can allocate memory

T এরটা Type expression generate করবে।

এই Tree evaluate করে এই Type expression পাবো।

array (2, array (3, integer)) ⇒ Type expression for input string int [2] [3]

2 টা row
↳ প্রত্যেক column এ integer করছে,
প্রত্যেক row তে 3 টা করে column থাকবে

by default,
integer → 4 bytes
float → 8 bytes

int	[2]	[3]
4 byte	4 byte	4 byte
4 byte	4 byte	4 byte

for each integer 4 bytes are allocated

∴ Total allocated memory = (6 * 4) bytes
= 24 bytes

(3 * 4) = 12 ; for each row

As there are 2 rows, so (2 * 12) = 24 bytes.

Question Type:

- Type expression + full code এর জন্য evaluate.
- Three Address Code Generation

Sequence of Declaration:

. Refers to how declarations are handeled with a grammar's production rules.

$P \rightarrow \{ offset = 0 \}\ D$

$D \rightarrow T\ id\ ;\ \{ top.put(id.lexeme, T.type, offset);\ offset = offset + T.width; \}\ D_1$

$D \rightarrow \in$

input string : int [3] a ;
float b ;

Generate Type expression for the input string.

- P produces { offset = 0 } D
  ↳ এখানে offset = 0 set করছি পরে D বসাতে যাচ্ছি.
- D produces T id ; Top বরাবর stack এ id lexeme, T এর type রাখছে offset রাখছে, offset value calculate করার পরে D_1 বসাতে যাচ্ছি.

এই sequence matter করে কার পরে কে আসছে.

Production Rule কোনো কোন Sequence এ declare হচ্ছে সেটা important.