Steven (2201741132)

Kelas LK01

Tugas Assignment pert 13 (7 Januari 2021)

Quiz 2 persiapan UAS

Soal Kuis 2 Tekom 2020 Ganjil

1. **Botttom Up Parsing 40%**

Consider a grammar as follow:

E → T or E | T

T → Id and T | Id | (E) | not E

Questions:

a) Construct Transition diagram for GOTO operations

b) Construct an SLR table parsing

c) Develop SLR parsing implementation for the input: **not (id or id) and id**

1. **Annotated Parse Tree 15%**

Given a Syntax Directed Definition:

|  |  |
| --- | --- |
| Production Rules | Semantic Rules |
| E 🡪 E + T | E.val = E.val + T.val |
| E 🡪 E – T | E.val = E.val – T.val |
| E 🡪 T | E.val = T.val |
| T 🡪 T \* F | T.val = T.val \* F.val |
| T 🡪 F | T.val = F.val |
| F 🡪 ( E ) | F.val = E.val |
| F 🡪 id | F.val = id.val |

1. Construct annotated parse tree for input: (10 - 3) \* 15 + 12
2. Construct dependency graph of that input
3. **DAG 10%**

Given an expression as follow:

A = (A+B) \* (A+B) + (A-B) \* (A-B) - (A+B) \* (A-B)

1. Create DAG
2. Create three address code from the DAG
3. Create quadruple and triple of the three-address code
4. **Intermediate Code Generator 15%**

Given a simple program as follow:

X = 15;

While ( X % 3 <= 5) {

Result = X \* X;

if (Result >= 10 && X % 2 == 0)

Y = Result /(X+1);

else

Y = Result\*X;

X = X + 1;

}

Create the three-address code of this program

1. **Code Generator 20%**

Create the Generated machine code of the program at exercise 4.

Jawaban:

1.Given Grammar:

E’ → E

E → T or E -> R1

E → T -> R2

T → Id and T -> R3

T → Id -> R4

T → (E) -> R5

T → not E -> R6

a) Transition diagram for GOTO operations

$

Accepted

E’ → E.

E

E → T or E.

E → T or .E

E → .T or E

E → .T

T → .Id and T

T → .Id

T → .(E)

T → .not E

E’ → .E

E → .T or E

E → .T

T →. Id and T

T → .Id

T → .(E)

T → .not E

E

E → T .or E

E → T.

T or

T (

Id Id not

T → Id .and T

T → Id.

T → Id and T.

T → Id and .T

T → .Id and T

T → .Id

T → .(E)

T → .not E

and T

not (

T → (.E)

E → .T or E

E → .T

T → .Id and T

T → .Id

T → .(E)

T → .not E

Id

T → not . E

E → .T or E

E → .T

T → .Id and T

T → .Id

T → .(E)

T → .not E

( not

not E

T → (E).

T → (E.)

( )

T

T ( Id

E not

Id

T → not E.

Follow:

Follow (E) = { $, ) }

Follow (T) = Follow (E) = { or, $ , ) }

b)SLR Parsing table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| State | Action | | | | | | | GoTo | |
| Id | or | and | not | ( | ) | $ | E | T |
| 0 | S3 |  |  | S5 | S4 |  |  | 1 | 2 |
| 1 |  |  |  |  |  |  | accepted |  |  |
| 2 |  | S6 |  |  |  | R2 | R2 |  |  |
| 3 |  | R4 | S7 |  |  | R4 | R4 |  |  |
| 4 | S3 |  |  | S5 | S4 |  |  | 8 | 2 |
| 5 | S3 |  |  | S5 | S4 |  |  | 9 | 2 |
| 6 | S3 |  |  | S5 | S4 |  |  | 10 | 2 |
| 7 | S3 |  |  | S5 | S4 |  |  |  | 11 |
| 8 |  |  |  |  |  | S12 |  |  |  |
| 9 |  | R6 |  |  |  | R6 | R6 |  |  |
| 10 |  |  |  |  |  | R1 | R1 |  |  |
| 11 |  | R3 |  |  |  | R3 | R3 |  |  |
| 12 |  | R5 |  |  |  | R5 | R5 |  |  |

c)SLR Parsing table

Input : **not (id or id) and id**

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Stack | Input | Action |
| 1 | 0 $ | not (id or id) and id $ | S5, push(not) , push(5) |
| 2 | 5not4 $ | (id or id) and id $ | S4, push( ( ) , push(4) |
| 3 | 4(5not4 $ | id or id) and id $ | S3, push(id) , push(3) |
| 4 | 3id4(5not4 $ | or id) and id $ | R4: T->id , pop()x2, push(T), push(goto(T,4)=2) |
| 5 | 2T4(5not4 $ | or id) and id $ | S6, push(or), push(6) |
| 6 | 6or2T4(5not4 $ | id) and id $ | S3, push(id) , push(3) |
| 7 | 3id6or2T4(5not4 $ | ) and id $ | R4: T->id , pop()x2, push(T), push(goto(T,6)=2) |
| 8 | 2T6or2T4(5not4 $ | ) and id $ | R2: E->T , pop()x2, push(E), push(goto(E,6)=10) |
| 9 | 10E6or2T4(5not4 $ | ) and id $ | R1: E->T or E , pop()x6, push(E), push(goto(E,4)=8) |
| 10 | 8E4(5not4 $ | ) and id $ | S12, push( ) ), push(12) |
| 11 | 12)8E4(5not4 $ | and id $ |  |
| 12 | $ | $ |  |
| 13 | $ | $ |  |
| 14 | $ | $ |  |
| 15 | $ | $ |  |