

204433 การแปลภาษาโปรแกรม

การบ้านที่ 5

ให้นัก์ติดตอบคำถามต่อไปนี้

1. พิจารณา grammar ต่อไปนี้

$S \rightarrow AB\$$

$A \rightarrow xB$

$A \rightarrow xyB$

$B \rightarrow zA$

$B \rightarrow w$

1.1. อะไรคือ terminal และ อะไรคือ non-terminal ใน grammar นี้

Non-terminal = {S, A, B}

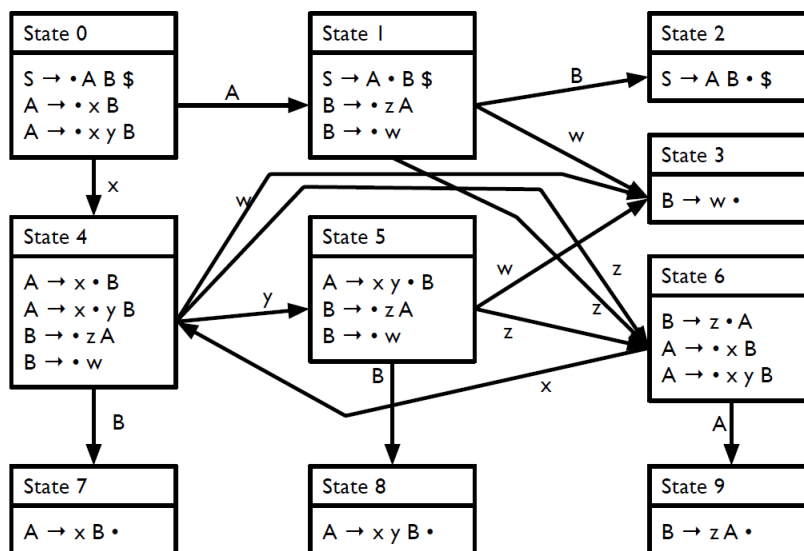
Terminal = {x, y, z, w}

1.2. ให้เหตุผลง่าย ๆ ในหนึ่งบรรทัดว่า grammar นี้เป็น LL(1) หรือไม่ เพราะเหตุใด

เห็นได้ชัดเจนนว่า x ที่เป็นสมาชิกของ FIRST(A) และเมื่อเป็นเช่นนี้ เมื่อ parser กำลังจะขยาย (expand) non-terminal A และเห็น lookahead x เข้ามา มี production ให้เลือกได้มากกว่าหนึ่ง นั่นคือ $A \rightarrow xB$ และ $A \rightarrow xyB$

1.3. พิจารณาว่า grammar นี้เป็น LR(0) หรือไม่

1.3.1. สร้าง DFA เพื่อจะนำไปสู่การสร้าง parsing table ของ LR(0) grammar

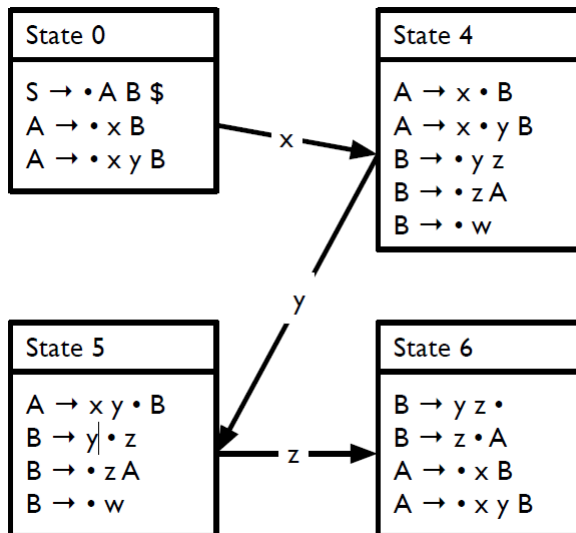


1.3.2. สร้างตาราง parsing ของ LR(0) grammar จาก DFA ที่ได้สร้างมาในข้อที่แล้ว

	x	y	z	w	\$	A	B
0	s4					g1	
1			s6	s3			
2					accepted		
3	B->w	B->w	B->w	B->w	B->w		
4		s5	s6	s3			g7
5			s6	s3			g8
6	s4					g9	
7	A->xB	A->xB	A->xB	A->xB	A->xB		
8	A->xyB	A->xyB	A->xyB	A->xyB	A->xyB		
9	B->zA	B->zA	B->zA	B->zA	B->zA		

1.3.3. ถ้าเราเพิ่ม production ต่อไปนี้เข้าไปใน grammar ข้างต้น $B \rightarrow yz$ ให้อธิบายว่า grammar นี้ยังเป็น LR(0) อยู่หรือไม่เพราะเหตุใด

จะไม่ใช่ LR(0) แล้ว เพราะจะมีจังหวะที่เมื่อ scan z เข้ามาแล้วเกิด shift-reduct conflict ดังเช่นแผนภาพ DFA ที่แสดงด้านล่างนี้



2. จาก EBNF grammar ของภาษา CSubset ให้วิเคราะห์โปรแกรมต่อไปนี้ และบอกว่า parser ของ CSubset จะ parse โปรแกรมต่อไปนี้ผ่านหรือไม่ โดยให้เหตุผลในแต่ละบรรทัดของโปรแกรมว่าแต่ละบรรทัดผ่านหรือไม่ผ่านเพราะเป็นไปตามหรือฝ่าฝืน production rule ตัวใด

```
struct A {
    long x, y;
    struct B {
        long q, r, s;
    } z;
} a, b[3];

void main() {
    long i;
    struct A c;
    struct B qq;
    c.z.r = 987654321;
    b[b[a.x-1].x-1].x = 7;
    WriteLong(a.x); WriteLong(a.y);
    i = 0;
    while (i < 3) {
        WriteLong(b[i].x); WriteLong(b[i].y);
        i = i + 1;
    }
}
```

CFG ของ CSubset สามารถ parse โปรแกรมนี้ได้โดยไม่มี syntactic error ภาษา CSubset อนุญาตให้เราใช้โครงสร้างที่มีในโปรแกรมข้างต้นดังต่อไปนี้

- **struct**
- **nested struct**
- **array**
- **array ของ struct**
- **struct ของ array**
- การ reference เข้าหา member ของ struct โดยใช้ **dot (.) operator**
- การ index เข้าหา array โดยใช้ **member ของ struct** หรือ **member ของ array**
- **function** ที่ไม่มีการ **return** ค่า (นั่นคือ **return void**)
- **while block**