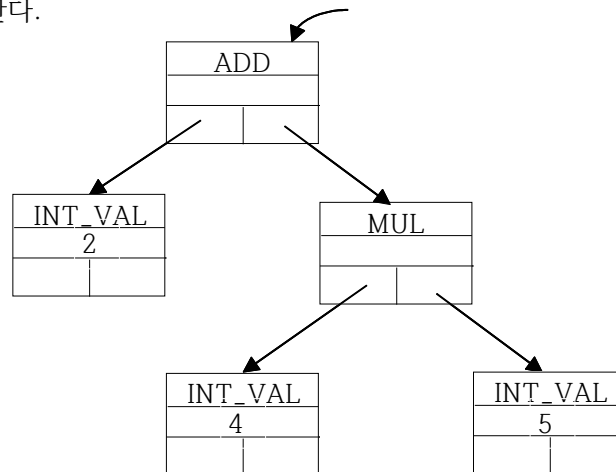


과제-4 신택스 트리 생성

과제-2의 수식계산을 위한 LR 파서를 변형하여, 수식을 LR 파싱하면서 (그 값을 계산하는 것이 아니라) 그 수식에 해당하는 신택스 트리를 만들고자 한다. 예를 들어, $2+4*5$ 란 수식이 입력되면 다음의 그림과 같은 트리를 만들고, accept 될때 `value[1]` 이 그 트리를 가리키도록 한다. `value[]` 배열의 원소들은 수식의 부분적인 값을 보관하는 것이 아니라 트리의 노드를 가리키도록 한다.



노드의 타입, 변수 및 함수는 다음과 같이 정의한다

```

typedef enum {INT_VAL, ADD, MUL} NODE_NAME;
typedef struct node {NODE_NAME name; int val; struct node *llink, rlink;} NODE;
NODE *value[1000];
NODE *makenode(NODE_NAME name,int v,NODE*p,NODE*q)
{
    NODE *n;
    n=malloc(...);
    n->name= ...
    n->val= ...
    n->llink= ...
    n->rlink= ...
    return n;
}

```

트리를 만들기 위한 일들은 모두 LR 파서에서 reduce 동작에서 일어나는데 `reduce()` 함수에서 다음과 같은 방법으로 작성한다.

```

case 1: value[top]= makenode(ADD,0,value[old_top+1],value[old_top+3]); break;
case 2: ...
case 3: value[top]= makenode(MUL,0,value[old_top+1],value[old_top+3]); break;

```

case 4: ...

case 5: value[top]= value[old_top+2];

case 6: value[top]= makenode[INT_VAL,value[old_top+1], NIL,NIL]; break;

본 과제에서는 간단히 하기 위해 실수는 취급하지 말고 정수형 수식만을 취급하도록 한다. 파서가 정상적으로 종료(accept)되면 value[top]은 트리의 루트를 가리키게 되는데 트리가 잘 만들어 졌는가를 확인하기 위하여 적당한 형태 (예로써, 수식의 postfix 표기법)으로 트리의 모양을 출력하도록한다.