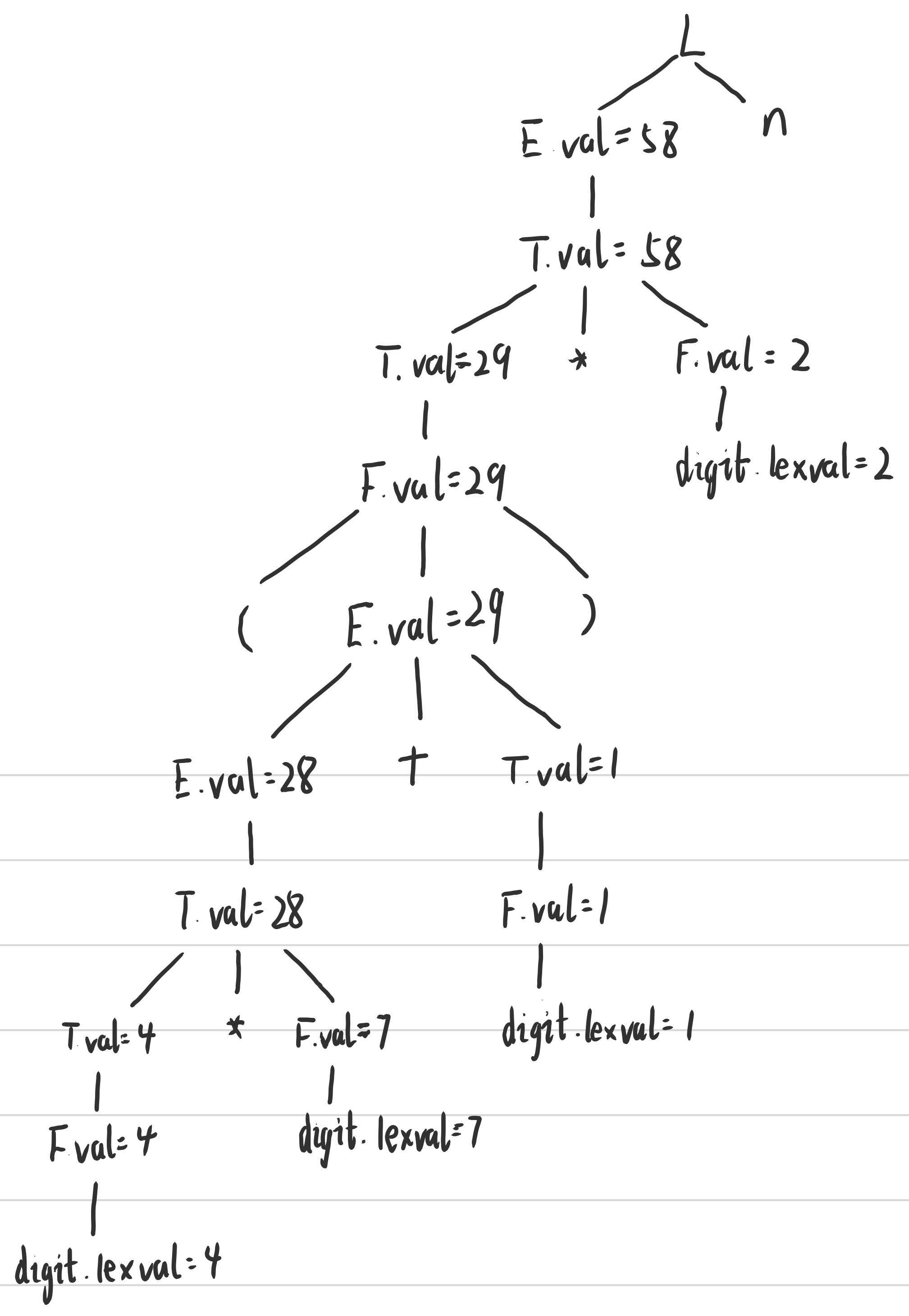
1.



5 (1)

```
E → $E_1$+T   { if ($E_1$.type = int) and (T.type = int)
                  then  E.type = int
                  else  E.type = real }
E → T         { E.type = T.type }
T → num.num   { T.type = real }
T → num       { T.type = int }
```

(2)

```
E → $E_1$+T   { if ($E_1$.type = real  and T.type = int)  then
                    E.type = real
                    print(T.code)
                    print('inttoreal')
                else if ($E_1$.type = int  and  T.type = real)  then
                    E.type = real
                    print('inttoreal')
                    print(T.code)
                else
                    E.type = $E_1$.type
                    print(T.code)
                print('+')
E → T         { E.type = T.type , print(T.code) }
T → num.num   { T.type = real ; T.code = $num_1$.code || '.' || $num_2$.code }
T → num       { T.type = int , T.code = num.code }
```

注：1）只能在T或E+T向E归约时才能输出T.code，因为在归约时可能需要先转换再print，
不能先print再转换

2）其次，E_1+T向E归约时，E已经print过了，所以不需要再print.

9. (1)

```
S → E           { E.side = right }
E → $E_1$:=$E_2$    { if (E.side = left) then error
                  else $E_1$.side = left   $E_2$.side = right
E → $E_1$+$E_2$     { if (E.side = left)  then error
                  else $E_1$.side = $E_2$.side = right    (∵整个表达式只能出现在右侧)
E → ($E_1$)       { if (E.side = left)  then  error
                    $E_1$.side = right
E → id
```

(2)

```
S → E           { E.side = right ; S.code = E.code }
E → $E_1$:=$E_2$    { if (E.side = left)  then  error
                  else $E_1$.side = left , $E_2$.side = right,
                       E.code = $E_1$.code || $E_2$.code || ':=' }
E → $E_1$+$E_2$     { if (E.side = left)  then error
                  else  $E_1$.side = $E_2$.side = right ;
                        E.code = $E_1$.code || $E_2$.code || '+' }
E → ($E_1$)       { if (E.side = left)  then error
                  else   $E_1$.side = right;
                         E.code = $E_1$.code }
E → id          { E.code = id.name }
```