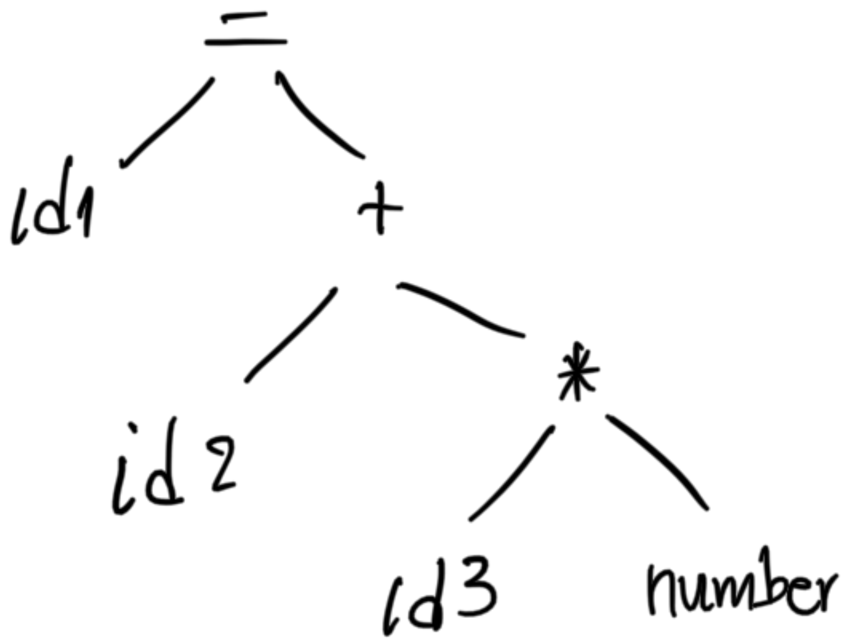
Parser

sum = old+rate * 50;

↳ (LEX) → [id1, =, id2, +, id3, *, number]

↳ (PARSER) →



↳ (INTERM)

↳ temp1 = number (50); /

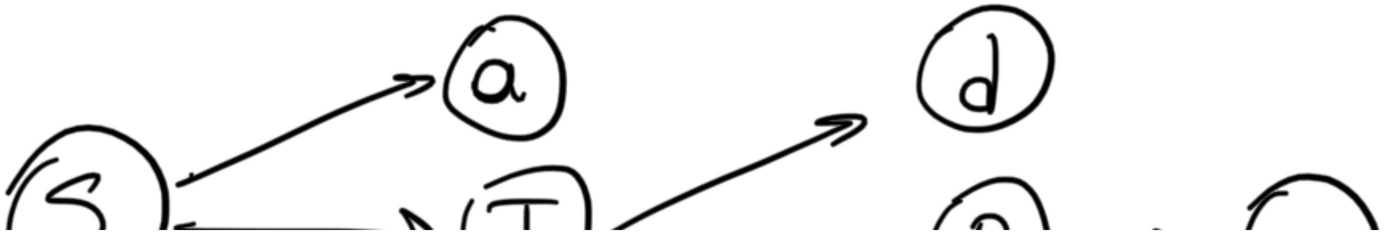
```
temp2 = id3 * temp1
temp3 = id2 + temp2
id1 = temp3;
```

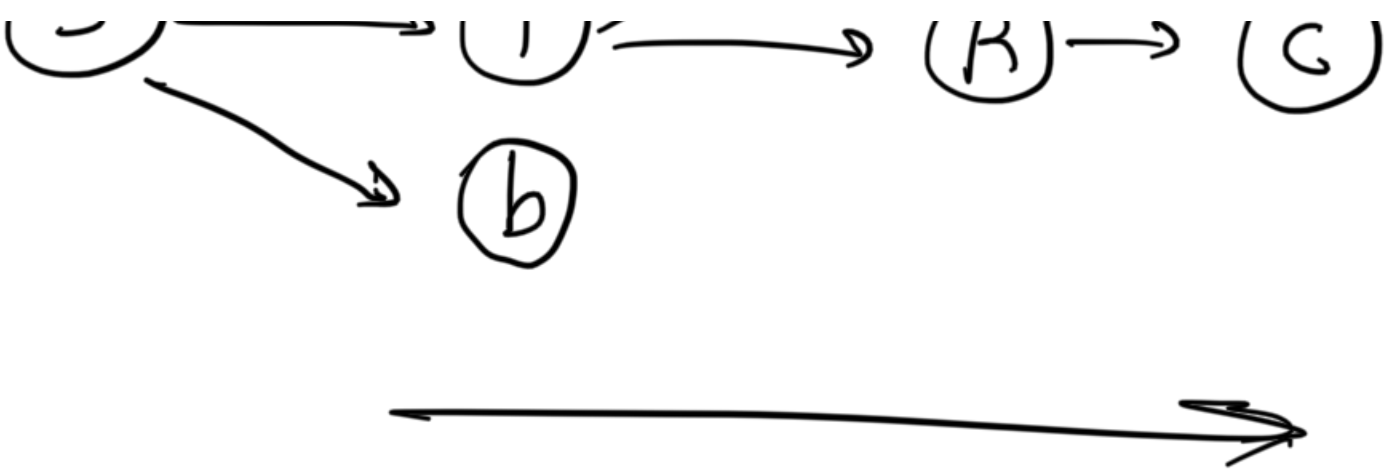
(OPTIM) →

```
temp2 = id3 * 50;
id1 = id2 + temp2;
```

(CODEGEN) →

```
add  $id3, $50
sll . .    . .
add
-    -  -  - -.
```





(Sa), (Sb), (STd), (STRc) . . .

func (p1, p2, p3, p4)

↳ p1, p2, p3 ';' p4

↳ p1, p2 ';' p3 ';' p4

$(\underline{f_1}((f_2(\boxed{f_3()}))))$

$\underline{x1 \text{ and } x2} \text{ and } x3 \dots$

$x1 \text{ 'and' } (x2 \text{ and } x3)$