CS314 Fall 2018 Assignment4 Solution

September 2018

1 Syntax Directed Translation

Use two global variables **addCount** and **assignCount** to store the number of addition and assignment operations. Only main function, assign function and expr function are changed.

```
int addCount = 0;
int assignCount = 0;

void main(){
    token = next_token();
    if (program()) {
        print("accept");
        print("%d assignments, %d addition operation\n",
            assignCount, addCount);
    } else {
        print("error");
    }
}

bool assign() {
    switch(token) {
        case a:
        case b:
        case c:
            if (!variable())
                return false;
            if (token != =)
                return false;
            token = next_token();
            if (!expr())
                return false;
            assignCount++;
            return true;
        default:
            return false;
    }
}
```

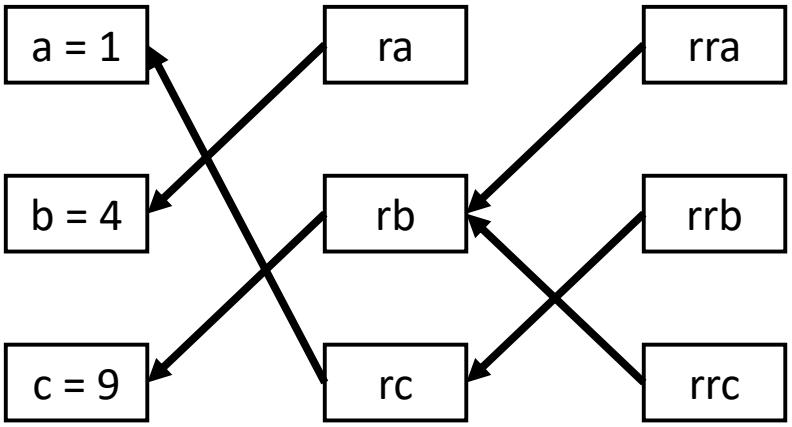
```
bool expr() {
    switch(token) {
        case a:
        case b:
        case c:
        case 0:
        case 1:
        case 2:
            if (!term())
                return false;
            if (token != +)
                return false;
            addCount++;
            token = next_token();
            return term();
        default:
            return false;
    }
}
```

2 Problem — Pointers

2.1 (a)

int *ra, int *rb, int *rc, int **rra, int **rrb, int **rrc

2.2 (b)



2.3 (c)

1 4 9
4 9
9 1 9

2.4 (d)

The answer is flexible, such as $ra = *a$

3 Problem — Freeing Memory

The answer is flexible, here are two example solutions

```
for (current_cell = head; current_cell != NULL;;) {
    list_cell *temp = current_cell->next;
    free(current_cell);
    current_cell = temp;
}
```

```
for (current_cell = head; current_cell != NULL;;) {
    list_cell *temp = current_cell;
    current_cell = current_cell->next;
    free(temp);
}
```

The idea is the original code tries to access current_cell→next after the memory of current_cell has been released, which is an incorrect way.