Syntaxanalyse

Beispiel für Top-Down-Analyse:

S->ABC
A->aa
A->b
B->bbB
B->ε
C->c


```
#include <stdio.h>

char token;
void S(); void A(); void B(); void C();
char scanner()
{
        return getchar();
}

void check(char e_token)
{
        if (token==e_token)
                token=scanner();
        else
        {
                printf("Syntaxfehler: %c erwartet\n",e_token);
                while (token!='\n') token=scanner(); // Lesen bis Eingabeende
        }
}
```

```
void parser()
{
        token=scanner();
        S();
        while (token!='\n')
        {
                printf("Syntaxfehler am Eingabeende\n");
                token=scanner();
        }
}

void main()
{
        while (!feof(stdin)) parser();
}
```

```
void S()
{
        if ((token == 'a') ||  (token == 'b'))
        {
                printf("regel S->ABC angewandt\n");
                A(); B(); C();
        }
        else
        {
                printf("Syntaxfehler, a oder b erwartet\n");
        };

}
```

```
void A()
{
        if (token == 'a')
        {
                printf("regel A->aa angewandt\n");
                check('a'); check('a');
        }
        else if (token == 'b')
        {
                printf("regel A->b angewandt\n");
                check('b');
        }
        else
        {
                printf("Syntaxfehler, a oder b erwartet\n");
        };

}
```

```
void B()
{
        if (token =='b')
        {
                printf("regel B->bbB angewandt\n");
                check('b'); check('b'); B(); // Rekursion, da B->bbB rechtsrekursive Produktion
        }
        else if (token == 'c')
        {
                printf("regel B->epsilon angewandt\n");
        }
        else
        {
                printf("Syntaxfehler b oder c erwartet\n");
        }
}
```

```
void C()
{
        if (token == 'c')
        {
                printf("regel C->c angewandt\n");
                check('c');
        }
        else
        {
                printf("Fehler c erwartet\n");
        }
}
```