

THÉORIES ET MÉTHODES DE COMPILATION

Cédric van Eetvelde
Département d'Informatique

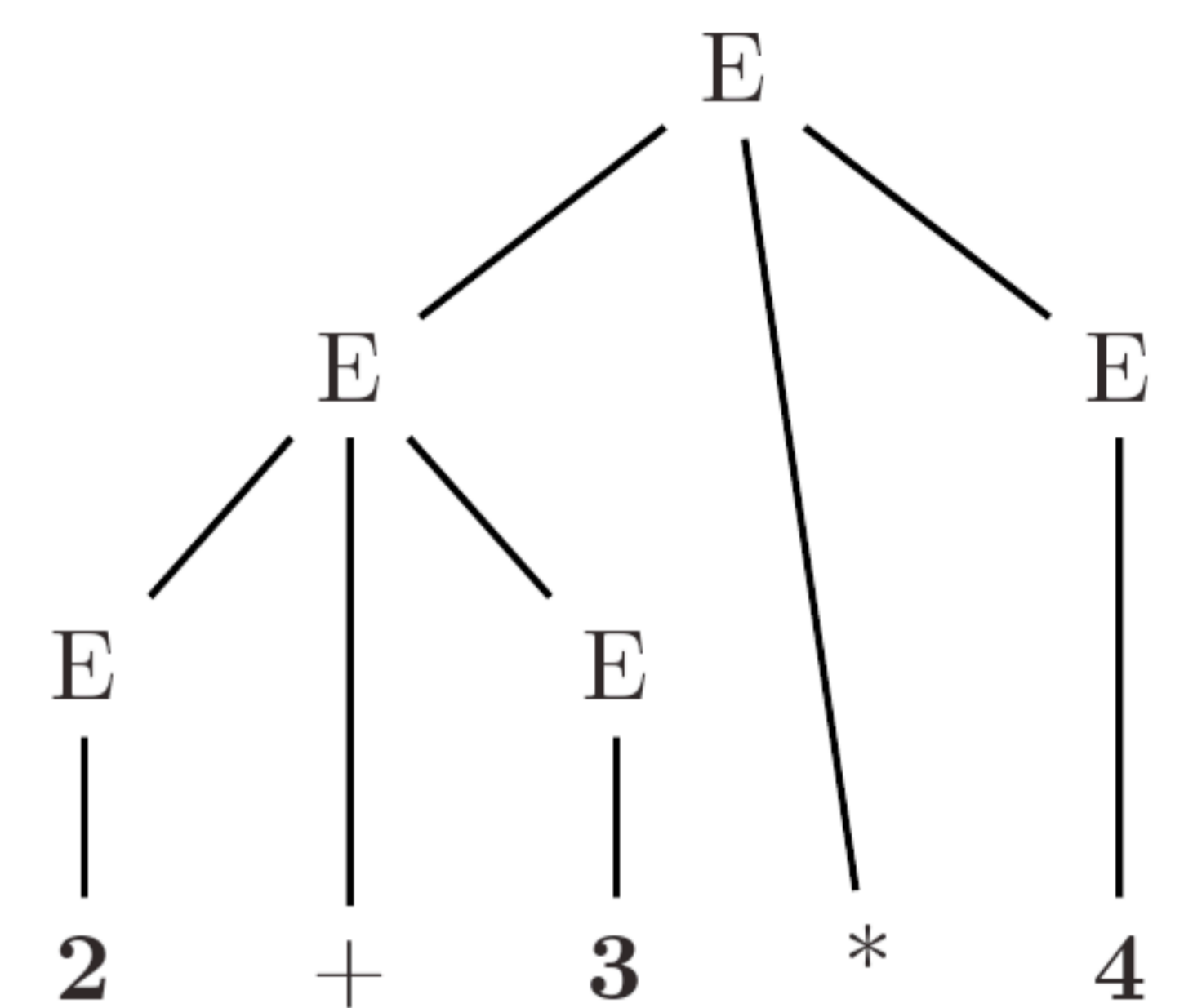
Un micro-processeur exécute des instructions, qui doivent lui être données dans un langage précis mais peu compréhensible pour un humain. Pour faciliter le travail du programmeur, il écrit ses programmes dans un langage plus lisible, qu'il faut ensuite traduire efficacement pour le processeur. C'est le rôle des compilateurs.

$$E \rightarrow E + F \mid E - F \mid F$$

$$F \rightarrow F * T \mid F / T \mid T$$

$$T \rightarrow (E) \mid \text{nombre}$$

À partir d'un ensemble de règles (la grammaire du langage), le compilateur peut "comprendre" le code et lui donner une signification valide.



Un compilateur commence par grouper les caractères en mots. Puis il regroupe les mots en phrases. Il peut ensuite en vérifier le sens. Enfin, il pourra les traduire de la manière qui convient.

On distingue donc les différentes phases :

- Analyse Lexicale
- Analyse Syntaxique
- Analyse Sémantique
- Optimisations
- Production de code

```
char * md5(const void * datas, const unsigned int length, char hash[33]){
    const char digit[] = "0123456789abcdef";
    unsigned char hashed[17];

    gcry_md_hash_buffer(GCRY_MD_MD5, hashed, datas, length);

    for (int i=0;i<16;++i){
        hash[i<<1] = digit[(hashed[i] & 0xF0)>>4];
        hash[(i<<1)+1] = digit[hashed[i] & 0x0F];
    }
    hash[32] = 0;

    return hash;
}
```



```
0000:0000 7f 45 4e 01 01 01 00 00 00 00 00 00 00 00 00 | ELF.....
0000:0010 02 00 03 00 01 00 00 00 50 87 04 08 34 00 00 00 | .....P...4...
0000:0020 30 15 00 00 00 00 00 00 34 00 20 00 08 00 28 00 | .....4...(.
0000:0030 25 00 22 00 06 00 00 00 34 00 00 00 34 80 04 08 | .....4...4...
0000:0040 34 80 04 08 00 01 00 00 00 01 00 00 05 00 00 00 | .....4...4...
0000:0050 04 00 00 03 00 00 00 34 01 00 00 34 81 04 08 | .....4...4...
0000:0060 34 81 04 08 13 00 00 00 13 00 00 04 00 00 00 00 | .....4...
0000:0070 01 00 00 01 00 00 00 00 00 00 00 00 80 04 08 | .....
0000:0080 00 80 04 08 b4 0c 00 00 b4 0c 00 05 00 00 00 00 | .....X...
0000:0090 00 10 00 01 00 00 b4 0c 00 00 b4 9c 04 08 | .....X...
0000:00a0 b4 9c 04 08 58 01 00 00 02 00 00 06 00 00 00 | .....X...
0000:00b0 00 10 00 02 00 00 00 cc 0c 00 00 cc 9c 04 08 | .....X...
0000:00c0 cc 9c 04 08 e8 00 00 e8 00 00 06 00 00 00 00 | .....X...
0000:00d0 04 00 00 04 00 00 48 01 00 00 48 81 04 08 | .....H...H...
0000:00e0 48 81 04 08 20 00 00 20 00 00 04 00 00 00 00 | .....H...H...
0000:00f0 04 00 00 50 e5 74 64 90 0b 00 90 8b 04 08 | .....Ptd...
0000:0100 90 8b 04 08 34 00 00 34 00 00 04 00 00 00 00 | .....4...4...
0000:0110 04 00 00 51 e5 74 64 00 00 00 00 00 00 00 00 | .....Qtd...
0000:0120 00 00 00 00 00 00 00 00 00 00 06 00 00 00 00 | .....Qtd...
```