ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

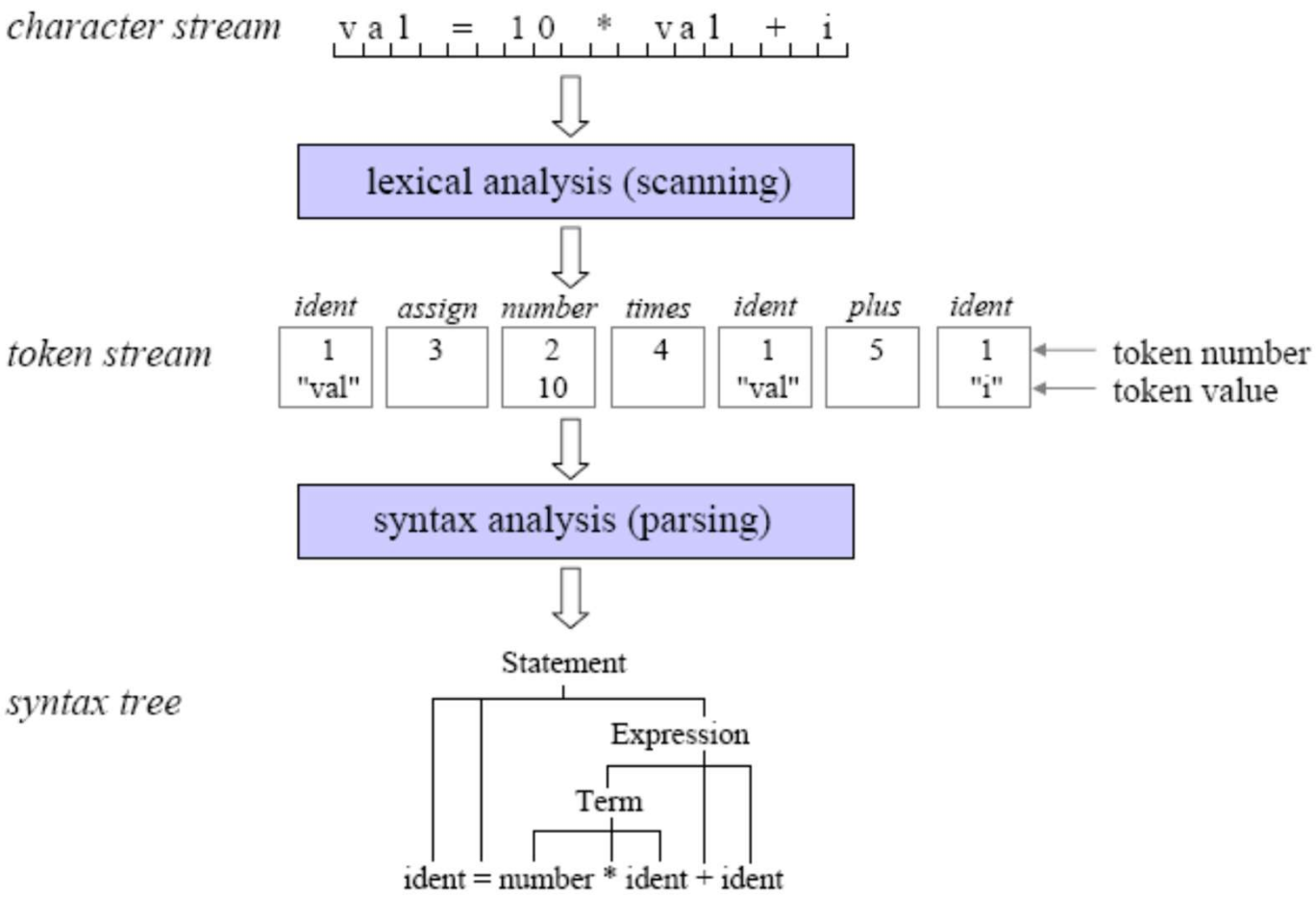
Bài tập
NGÔN NGỮ VÀ PHƯƠNG PHÁP DỊCH
Nhóm 2

Phân tích từ vựng

Scanner là gì?

- Trong một chương trình dịch, thành phần thực hiện chức năng phân tích từ vựng gọi là scanner.

Bộ phân tích từ vựng (scanner) là gì?



Nhiệm vụ của một scanner

- Bỏ qua các ký tự vô nghĩa như: dấu trống, tab, ký tự xuống dòng, chú thích.
- Phát hiện các ký tự không hợp lệ
- Phát hiện token
  - định danh (identifier)
  - từ khóa (keyword)
  - số (number)
  - Hằng ký tự
  - special symbol
  - ...

Nhiệm vụ của một scanner

- Khi tích hợp với bộ phân tích cú pháp, scanner chuyển lần lượt các token cho bộ phân tích cú pháp (parser)
- Trong bài tập này toàn bộ các từ tố của chương trình được in ra.

Bảng chữ cái của KPL

- Chữ cái (letter): a-z, A-Z, '_'
- Chữ số (digit): 0-9
- Các ký hiệu đặc biệt
  - +, -, *, /, >, <,!, =, [space], [comma], ., :, ;, ', (,)
- Các ký hiệu không được liệt kê ở các dòng trên được xếp vào ký tự lỗi, trư khi chúng xuất hiện trong hằng ký tự.

Các token của ngôn ngữ KPL

- Từ khóa

  PROGRAM, CONST, TYPE, VAR, PROCEDURE, FUNCTION, BEGIN, END, ARRAY, OF, INTEGER, CHAR, CALL, IF, ELSE, WHILE, DO, FOR, TO

- Toán tử

  := (assign), + (addition), - (subtraction), * (multiplication), / (division), = (comparison of equality), != (comparison of difference), > (comparison of greaterness), < (comparison of lessness), >= (comparison of greaterness or equality), <= (comparison of lessness or equality)

Các từ tố của KPL

- Ký hiệu đặc biệt

  ; (semicolon), . (period), : (colon), , (comma), ((left parenthesis),) (right parenthesis), ‘ (singlequote)

- Và

  (. và .) để đánh dấu chỉ mục của mảng

  (* và *) để đánh dấu điểm bắt đầu và kết thúc của chú thích

- Ngoài ra

  định danh, số, hằng ký tự

Xây dựng scanner – Cấu trúc

STT	Tên tệp	Nội dung
1	Makefile	Project
2	scanner.c	Tệp chính
3	reader.h, reader.c	Đọc mã nguồn
4	charcode.h, charcode.c	Phân loại ký tự
5	token.h, token.c	Phân loại và nhận dạng token, từ khóa
6	error.h, error.c	Thông báo lỗi

Xây dựng scanner – reader

```
// Đọc một ký tự từ kênh vào
int readChar(void);
// Mở kênh vào
int openInputStream(char *fileName);
// Đóng kênh vào
void closeInputStream(void);

// Chỉ số dòng, cột hiện tại
int lineNo, colNo;
// Ký tự hiện tại
int currentChar;
```

Xây dựng scanner – charcode

```
typedef enum {
  CHAR_SPACE,              // Khoảng trống
  CHAR_LETTER,             // Chữ cái
  CHAR_DIGIT,              // Chữ số
  CHAR_PLUS,               // '+'
  CHAR_MINUS,              // '-'
  CHAR_TIMES,              // '*'
  CHAR_SLASH,              // '/'
  CHAR_LT,                 // '<'
  CHAR_GT,                 // '<'
  CHAR_EXCLAIMATION,       // '!'
  CHAR_EQ,                 // '='
  CHAR_COMMA,              // ','
  CHAR_PERIOD,             // '.'
  CHAR_COLON,              // ':'
  CHAR_SEMICOLON,          // ';'
  CHAR_SINGLEQUOTE,        // '\''
  CHAR_LPAR,               // '('
  CHAR_RPAR,               // ')'
  CHAR_UNKNOWN             // Ký tự ngoài bảng chữ cái
} CharCode;
```

Xây dựng scanner – charcode

- charcode.c định nghĩa một bảng `charCodes` ánh xạ từng ký tự trong bảng mã ASCII vào một trong các CharCode được định nghĩa
- Lưu ý: Lệnh đọc ký tự `getc` có thể trả về mã EOF có giá trị nguyên là -1, nằm ngoài bảng mã ASCII

Xây dựng scanner – token

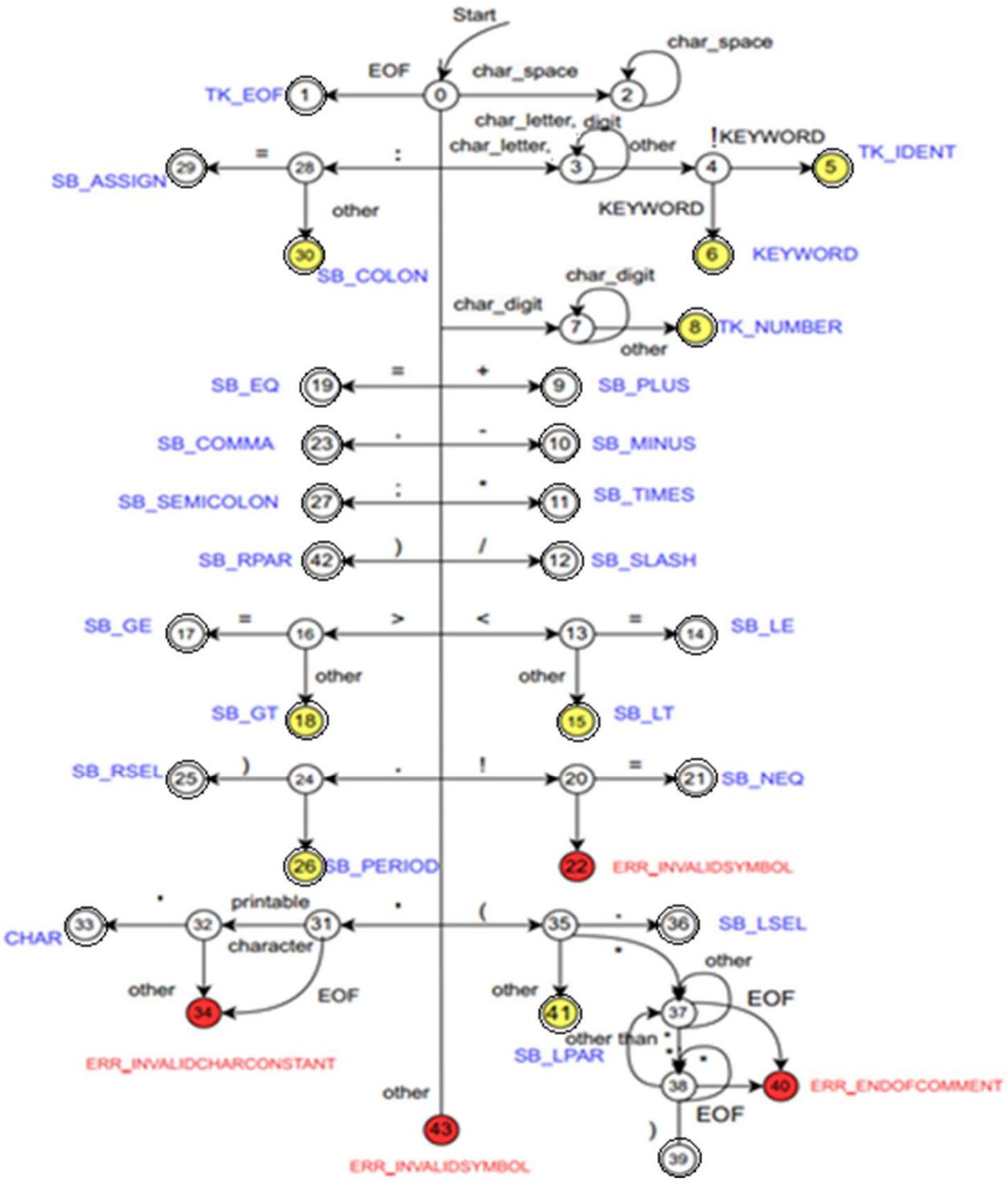
```
typedef enum {
  TK_NONE,       // Đại diện cho một lỗi
  TK_IDENT,      // Định danh
  TK_NUMBER,     // Số
  TK_CHAR,       // Hằng ký tự
  TK_EOF,        // Kết thúc chương trình
  // Các từ khóa
  KW_PROGRAM, KW_CONST, KW_TYPE, KW_VAR,
  KW_INTEGER, KW_CHAR, KW_ARRAY, KW_OF,
  KW_FUNCTION, KW_PROCEDURE,
  KW_BEGIN, KW_END, KW_CALL,
  KW_IF, KW_THEN, KW_ELSE,
  KW_WHILE, KW_DO, KW_FOR, KW_TO,
  // Các ký hiệu đặc biệt
  SB_SEMICOLON, SB_COLON, SB_PERIOD, SB_COMMA,
  SB_ASSIGN, SB_EQ, SB_NEQ, SB_LT, SB_LE, SB_GT, SB_GE,
  SB_PLUS, SB_MINUS, SB_TIMES, SB_SLASH,
  SB_LPAR, SB_RPAR, SB_LSEL, SB_RSEL
} TokenType;
```

Xây dựng scanner – token

```
// Cấu trúc lưu trữ của một token
typedef struct {
  char string[MAX_IDENT_LEN + 1];
  int lineNo, colNo;
  TokenType tokenType;
  int value;
} Token;

// Kiểm tra một xâu có là từ khóa không
TokenType checkKeyword(char *string);
// Tạo một token mới với kiểu và vị trí
Token* makeToken(TokenType tokenType, int lineNo, int colNo);
```

Xây dựng scanner – error

```
// Danh sách các lỗi trong quá trình phân tích từ vựng
typedef enum {
  ERR_ENDOFCOMMENT,
  ERR_IDENTTOOLONG,
  ERR_INVALIDCHARCONSTANT,
  ERR_INVALIDSYMBOL
} ErrorCode;

// Các thông báo lỗi
#define ERM_ENDOFCOMMENT "End of comment expected!"
#define ERM_IDENTTOOLONG "Identification too long!"
#define ERM_INVALIDCHARCONSTANT "Invalid const char!"
#define ERM_INVALIDSYMBOL "Invalid symbol!"

// Hàm thông báo lỗi
void error(ErrorCode err, int lineNo, int colNo);
```



Chương trình giao cho nhóm là một project chứa bộ phân tích từ vựng KPL chưa hoàn chỉnh.

Dựa vào sơ đồ trạng thái của ôtômat, hoàn thiện project

Yêu cầu trình bày

- Luật từ vựng của KPL
  - Ngôn ngữ cho phép sử dụng những ký tự nào trong chương trình?
  - Các loại hằng
  - Các kiểu dữ liệu: dữ liệu chuẩn, dữ liệu có cấu trúc
  - Các toán tử: số học, so sánh, gán
  - Các từ khóa. Vấn đề phân biệt chữ hoa/thường cho từ khóa
  - Quy định về định danh. Vấn đề chữ hoa/thường
  - Lưu trữ thông tin cho số. Giá trị số nhập vào được kiểm soát như thế nào?

Yêu cầu trình bày

1/Hoàn thiện các hàm đánh dấu TODO

2/Chạy chương trình với ví dụ 1.Sửa ví dụ 4 để gây lỗi . In màn hình kết quả

3/Sửa để chương trình khi gặp lỗi không dừng

4/Trình bày hoạt động của các hàm sau:

- getToken
- printToken. Chú ý các trường hợp gặp lỗi