

Apêndice A

Reconhecedor Lexical

De seguida apresenta-se o código *C* de um reconhecedor para a linguagem definida pela seguinte expressão regular “*c*ab[abc]**”.

```
#define ERRO -1
#define n_estados 7
#define n_simb 4
#define INICIAL 0

int tt[n_estados][n_simb] = {{1,-1,2,-1},
    {-1,3,-1,-1},
    {1,-1,2,-1},
    {4,5,6,-1},
    {4,5,6,-1},
    {4,5,6,-1},
    {4,5,6,-1}};

int Z[] = {3,4,5,6};

main(){
    int i,j;
    char *reconhece();

    printf("\n\n%s\n",reconhece());
}

char *reconhece(){
    int estado=INICIAL;
    char *r;
    char simbolo;
```

¹(Versão 1.2)

```
int simb2int(char);
int e_terminais(int,int []);

simbolo=getchar();
while ((simbolo!='\n') && (estado!=ERRO)) {
    estado=tt[estado][simb2int(simbolo)];
    simbolo=getchar();
}
if (e_terminais(estado,Z) && (simbolo=='\n'))
    r="aceita";
else
    r="erro";
return r;
}

int e_terminais(int estado,int Z[]) {
    int i,res;

    i=res=0;
    while ((i<4) && (res==0)) {
        if (estado==Z[i])
            res= 1;
        i=i+1;
    }
    return res;
}

int simb2int(char simbolo){
    switch (simbolo) {
    case 'a' :
        return 0;
        break;
    case 'b':
        return 1;
        break;
    case 'c':
        return 2;
        break;
    default :
        return 3;
    }
}
```