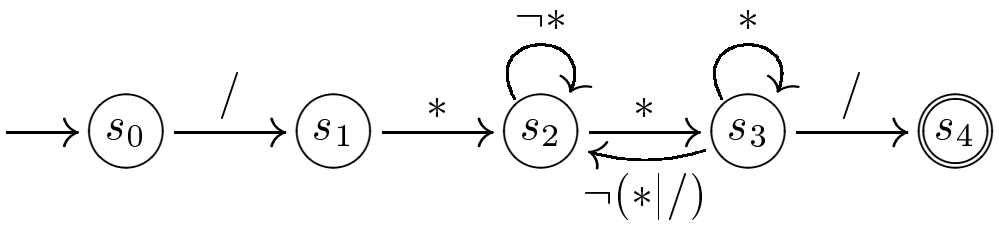
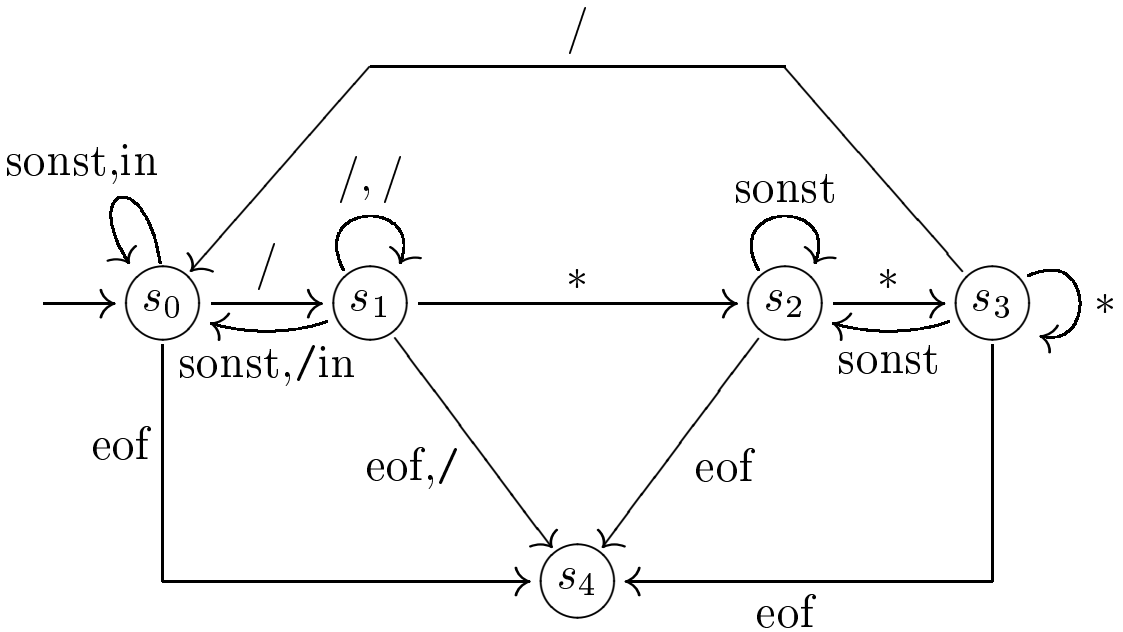
Erkennung von /*....*/



Elimination von Kommentaren



Elimination von Kommentaren

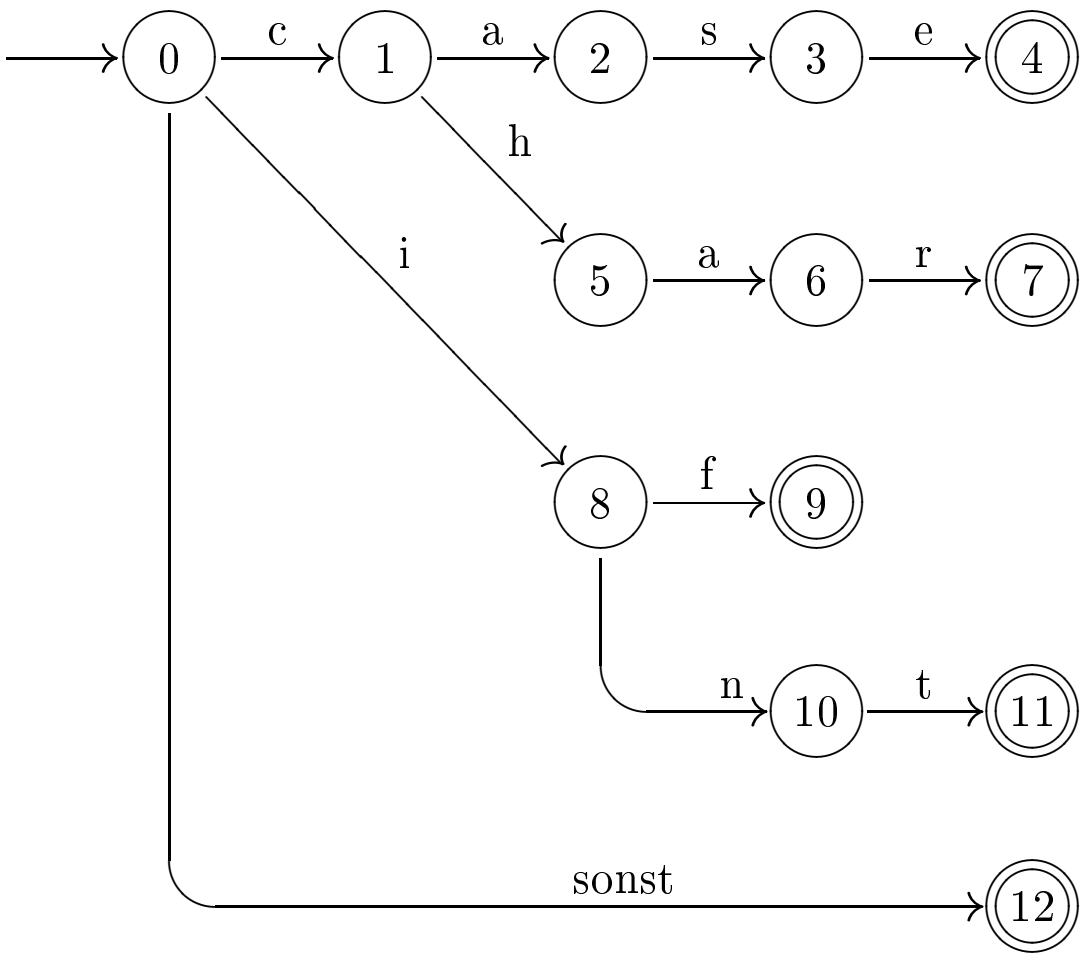
```
/*
   Entfernung aller Kommentare aus C-Programm.

   Interpretation der Zustaende:
     S0   Startzustand
     S1   moegliches erstes '/' gelesen
     S2   innerhalb des Kommentars
     S3   moegliches Kommentarende
*/

#include <stdio.h>

void
error(void)
{
  printf("Dateiende innerhalb eines Kommentars!\n");
  exit(0);
}

int
main(void)
{
  int state;
  char c;

  state = 0;
```

```
  while(1) {

   c = getchar();

   switch(state) {

   case 0:
     if (c=='/')       state = 1;   /* -> Zustand S1 */
     else if (c==EOF) exit(0);     /* Dateiende     */
     else             putchar(c);  /* kopieren      */
     break;

   case 1:
     if (c=='*')       state = 2;   /* -> Zustand S2 */
     else if (c=='/') putchar('/');/* altes '/'     */
     else if (c==EOF) {            /* Dateiende     */
       putchar('/');               /* altes '/'     */
       exit(0);
     }
     else {
       putchar('/');               /* altes '/'     */
       putchar(c);                 /* kopieren      */
       state = 0;                  /* -> Zustand S0 */
     }
     break;

   case 2:
     if (c=='*')       state = 3;   /* -> Zustand S3 */
     else if (c==EOF) error();     /* vorz. Ende    */
     break;

   case 3:
     if (c=='/')       state = 0;   /* -> Zustand S0 */
     else if (c==EOF) error();     /* vorz. Ende    */
     else if (c!='*') state = 2;   /* -> Zustand S2 */
     break;

   }/*switch*/

  }/*while*/

}/*main*/
```

Beispiel für Scanner-Implementierung

Einfacher Scanner für:

Schlüsselwörter **case**, **char**, **if**, **int**
Bezeichner, die nur aus Buchstaben bestehen



Beispiel für Scanner-Implementierung

```
/*
   Einfacher Scanner fuer:
     Schluesselwoerter case, char, if, int
     Bezeichner, die nur aus Buchstaben bestehen
*/

#include <stdio.h>

#define MAX_ID     31
#define MAX_STATE  13
#define MAX_BRANCH  2

typedef enum {
  TOKEN_ID,  TOKEN_CASE,  TOKEN_CHAR, TOKEN_IF,
  TOKEN_INT, TOKEN_ERROR, TOKEN_EOF
} token;

char id_name[MAX_ID+1];

unsigned char scan_tab[MAX_STATE][1+2*MAX_BRANCH] =
{
  { 2, 'c', 1, 'i', 8 },
  { 2, 'a', 2, 'h', 5 },
  { 1, 's', 3         },
  { 1, 'e', 4         },
  { 0, TOKEN_CASE     },
  { 1, 'a', 6         },
  { 1, 'r', 7         },
  { 0, TOKEN_CHAR     },
  { 2, 'f', 9, 'n',10 },
  { 0, TOKEN_IF       },
  { 1, 't',11         },
  { 0, TOKEN_INT      },
  { 0, TOKEN_ID       }
};

#define ID_STATE (MAX_STATE-1);
```

```
token
get_token(void)
{
  int c, i, idlen, state, num;
  char *idP, *tableP;

  c = getchar();
  while (c==' ' || c=='\t' || c=='\n') c = getchar();

  if (c==EOF) return TOKEN_EOF;

  if (!isalpha(c)) return TOKEN_ERROR;

  c = tolower(c);
  state = 0;
  idlen = 0;
  idP = id_name;

  while (1) {

         :
         :
         :

  }

} /* get_token */
```

```
tableP = scan_tab[state];
num = tableP[0];

if (isalpha(c)) {

  idlen++;
  if (idlen <= MAX_ID ) {
    *idP = c;
    idP++;
  }

  if (num==0) {
    state = ID_STATE;
    *idP = '\0';
  }
  else {
    tableP++;
    for (i=1; i<=num; i++) {
      if (c==*tableP) {
        tableP++;
        state = *tableP;
        break;
      }
      else
        tableP+=2;
    }
    if (i==num+1) {
      state = ID_STATE;
      *idP = '\0';
    }
  }
}
else {

  ungetc(c,stdin);
  if (num==0) return tableP[1];
  *idP = '\0';
  return TOKEN_ID;
}

c = tolower(getchar());
```

```
int
main(void)
{

  token t;
  static char *token_name[] =
  { NULL, "CASE", "CHAR", "IF", "INT", "ERROR", "EOF"};

  do {

    t = get_token();

    if (t!=TOKEN_ID)
      printf("<%s>\n", token_name[t]);
    else
      printf("Identifier %s\n", id_name);

  } while (t!=TOKEN_EOF);

  printf("\n\n");

}
```

```
f
case
bez bezwo
char then iff
if
22
int zzzz
intzz
```

```
Identifier f
<CASE>
Identifier bez
Identifier bezwo
<CHAR>
Identifier then
Identifier iff
<IF>
<ERROR>
<ERROR>
<INT>
Identifier zzzz
Identifier intzz
<EOF>
```