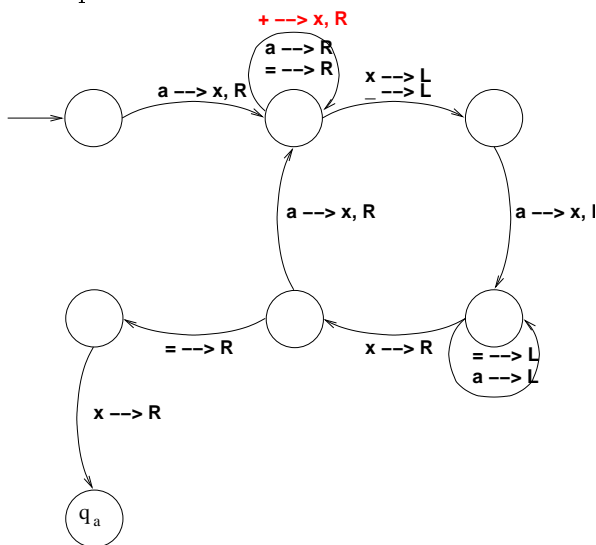


1. (1)  $S \rightarrow aSa$   
 $S \rightarrow +X$   
 $X \rightarrow aXa$   
 $X \rightarrow =$

(2) Machine de Turing : on suppose que l'on vérifie d'abord que la chaîne d'entrée est un mot de  $a^+a^+=a^+$  et que l'on traite les cas extrêmes.



2. Version minimale : l'automate est représenté par sa table de transition (table à double entrée); on a un vecteur de booléens pour le caractère terminal ou non des états. Par convention, les lettres commencent à 0 pour a, les états commencent à 1 (on garde 0 pour la complétion). Les cases « vides » sont initialisées à -1.

```

void completion(){
int i,j ;
for (i=0 ; i<col_max ; i++)
    automate[0][i] = 0 ;
for (i=0 ; i<lgn_max ; i++)
    for (j=0 ; j<col_max ; j++)
        if (automate[i][j] == -1)
            automate[i][j]=0 ;
}

#define MAX_TABLE 20
int automate[MAX_TABLE][MAX_TABLE] ;
int col_max, lgn_max ;
int term_table[MAX_TABLE] ;

int terminal(etat q) {
    return term_table[q] ;
}

int existe_t(etat q, char c) {
    return
        automate[q][c-'a'] == -1 ;
}

int delta(etat q, char c) {
    return automate[q][c-'a'] ;
}

int reconnaissance(char * m) {
    int l = strlen(mot) ;
    int i ;
    etat q = 1 ;
    for (i=0 ; i<l ; i++)
        q = delta(q,mot[i]) ;
    return terminal(q) ;
}
    
```

Voir sur le web une version légèrement plus complète.