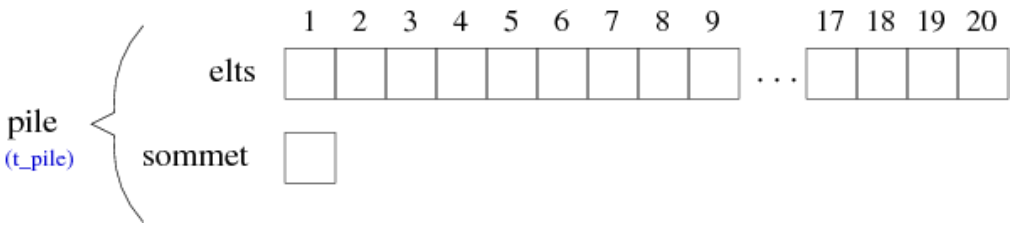
1 Implémentation statique / contiguë : dans un tableau

1.1 Le type

Nous avons besoin d'un tableau (à une dimension = vecteur) pour ranger les éléments et d'un entier `sommet` qui nous permette de savoir en permanence où se situe celui-ci. Pour avoir quelque chose de systémique, nous allons déclarer une pile statique comme la composée d'un ensemble de valeurs empilées (un vecteur d'éléments) et d'un indicateur de sommet (entier), ce qui donne :



Exemple de déclaration algorithmique :

```
constantes
   Nbmax = 20                          /* Nombre maximum d'éléments */

types
   t_element = ...                     /* Un élément */
   t_elements = Nbmax t_element        /* Vecteur d'éléments */
   t_pile = enregistrement             /* La pile */
      t_elements elts
      entier sommet
   fin enregistrement t_pile

variables
   t_pile pile
```

1.2 Implémentation des opérations

(abstrait) p : Pile	(implém) `t_pile pile`
p ← *pilevide*	`pile.sommet ← 0`
estvide(p)	`pile.sommet = 0`
sommet(p)	`pile.elts[pile.sommet]`
p ← *empiler(e,p)*	`pile.sommet ← pile.sommet + 1` `pile.elts[pile.sommet] ← e`
p ← *depiler*(p)	`pile.sommet ←pile.sommet - 1`

Toutes les opérations s'implémentent en *temps constant* : le nombre d'opérations exécutées ne dépend pas de la taille de la pile.

2 Implémentation dynamique / chaînée

2.1 Le type

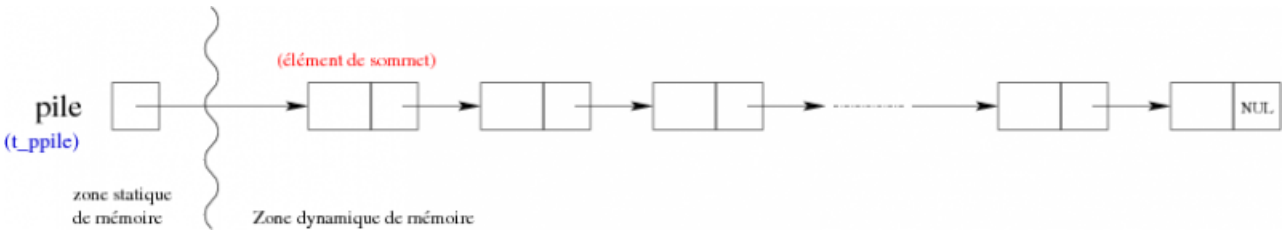
Dans ce cas, les éléments de la pile sont chaînés entre eux, et le pointeur représente le sommet de celle-ci. On peut noter que le lien sur les éléments est un lien de précédence qui permet lorsque l'on dépile de savoir quel élément avait été précédemment empilé.

Exemple de déclaration algorithmique :

```
types
   t_element = ...                  /* un élément */
   t_ppile = ↑ t_pile              /* la pile */
   t_pile = enregistrement          /* un enregistement */
      t_element elt
      t_ppile   precedent
   fin enregistrement t_pile

variables
   t_ppile  pile
```

Ce qui correspondrait à la structure suivante :



2.2 Implémentation des opérations

Le **pile vide** sera représentée par le pointeur NUL.

Pour récupérer la valeur du **sommet**, il suffit, sur une pile non vide, d'accéder à l'élément pointé [1] par `pile`.

Empiler un élément se fera de la manière suivante :
— On crée l'enregistrement pour le nouvel élément ;
— on le "chaîne" au reste : on l'insère en tête de la structure chaînée ;
— il est le nouveau sommet : on déplace le pointeur `pile` dessus.

Pour **dépiler** :
— On garde un pointeur sur le sommet ;
— on déplace le pointeur `pile` sur l'élément qui suit : il est le nouveau sommet ;
— enfin, on libère la mémoire occupée par l'ancien sommet.

Ici aussi, toutes les opérations s'implémentent en temps constant.

1. Par abus de langage on parle de *l'élément pointé* pour *la valeur du champ elt de l'enregistrement pointé.*