

2. LA TABLE DES IDENTIFICATEURS.

L'objectif est de choisir une représentation pour la table des identificateurs et de définir les opérations d'ajout et de recherche d'un identificateur dans cette table.

⇒ Au fur et à mesure de l'analyse du programme source, le compilateur rencontre des identificateurs qui ne sont pas des mots-clés réservés du mini-Pascal. Un identificateur peut désigner une variable, une constante, un type, une procédure, une fonction, etc. Un certain nombre d'informations sont associées à un identificateur:

- le nom,
- le type,
- la valeur si l'identificateur est une constante,
- la dimension si l'identificateur est un tableau,
- l'adresse si l'identificateur est une variable,
- le mode de passage si l'identificateur est un paramètre,
- le nombre de paramètres si l'identificateur est une procédure ou une fonction,
- le type de la valeur de retour si l'identificateur est une fonction, etc.

Ces informations sont conservées dans **la table des identificateurs**. Les identificateurs sont entrés dans la table des identificateurs au moment de l'analyse lexicale, mais des informations sur ces identificateurs sont ajoutées lors de l'analyse syntaxique et lors de l'analyse sémantique. La table des identificateurs est construite et utilisée par le compilateur lors de l'analyse, elle sera encore utilisée lors de la traduction.

⇒ Chaque fois qu'un nom est rencontré dans le texte source, une recherche s'effectue dans la table des identificateurs:

- si le nom recherché n'est pas dans la table des identificateurs, il faut créer une entrée pour ce nom dans la table, enregistrer toutes les informations connues et détecter les erreurs éventuelles;
- si le nom recherché dans la table des identificateurs est présent dans la table, il faut prendre en compte d'éventuelles informations et/ou vérifier qu'il n'y a aucun conflit avec les informations déjà présentes pour ce nom.

La table est modifiée si l'on a rencontré un nouveau nom, ou une nouvelle information sur un nom existant. Chaque entrée de la table des identificateurs correspond à la déclaration d'un nom.

Les opérations que l'on peut effectuer sur la table des identificateurs sont:

- **créer une entrée pour un nom**,
- **rechercher un nom**.

Ce sont des opérations indépendantes que l'on peut vouloir effectuer l'une sans l'autre.

⇒ Le mécanisme de la table des identificateurs doit permettre d'ajouter et de retrouver des identificateurs de façon efficace. Il existe **différentes représentations** possibles de la table des identificateurs.

■ Entrées de la table des identificateurs

Une entrée dans la table des identificateurs correspond à un identificateur et aux informations qui lui sont associées. Le nombre et le type des informations associées à chaque identificateur dépend du genre (variable, constante, fonction, etc.) de l'identificateur. L'utilisation des enregistrements à partie variable permet d'associer à chaque identificateur, en plus du nom et du genre de cet identificateur, des informations supplémentaires spécifiques à son genre:

```
type T_IDENT = ( variable, constante );
      T_ENREG_IDENT = record
                                nom: array[1..LONG_MAX_IDENT] of char;
                                case typ : T_IDENT of
                                    variable :    ( typv : integer; { type de la variable }
                                                    adrv : integer ); { adresse de la variable }
                                    *
                                    constante:    ( typc : integer; { type de la constante }
                                                    val : integer ) { valeur de la constante }
                                end;
```

■ Représentation de la table des identificateurs

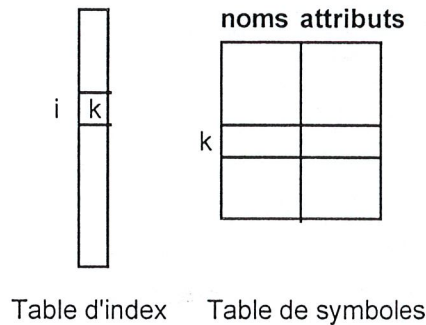
① LISTE LINEAIRE TRIÉE

La table des identificateurs peut être déclarée comme un tableau de T_ENREG_IDENT. Afin de minimiser la place occupée en mémoire par la table des identificateurs, il est préférable de définir la table des identificateurs comme un tableau de pointeurs sur T_ENREG_IDENT:

```
const NB_IDENT_MAX = 100;
type P_T_ENREG_IDENT = ^T_ENREG_IDENT;
var TABLE_IDENT : array[1.. NB_IDENT_MAX] of P_T_ENREG_IDENT;
```

Ainsi, la table des identificateurs nécessite relativement peu de place au départ de la compilation et la mémoire nécessaire est allouée dynamiquement au fur et à mesure des besoins.

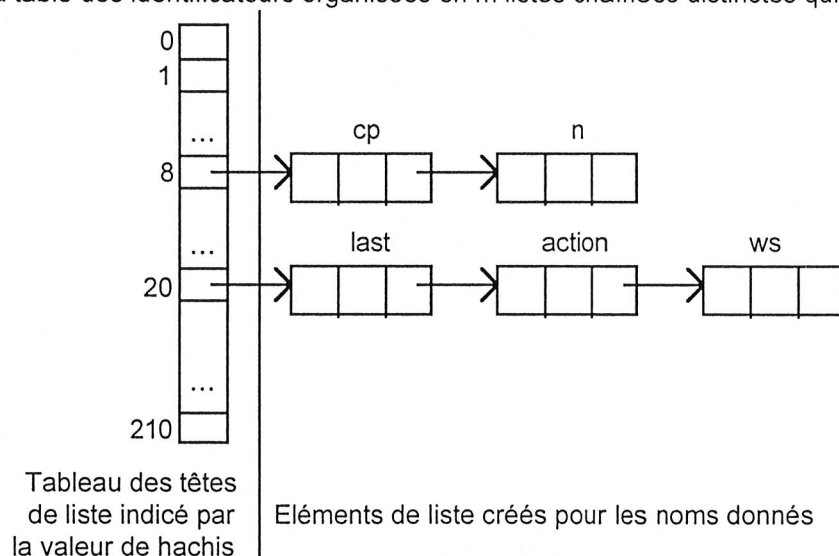
De plus, afin de diminuer le temps de recherche d'un identificateur, la table des identificateurs peut être triée. La technique de recherche classique est alors la **recherche dichotomique**, qui est plus rapide que la recherche linéaire. La technique de tri la plus utilisée est le **tri par insertion**, compte tenu de l'ordre d'arrivée des éléments de la table. Le problème est que les informations doivent constamment être déplacées dans la table pour rester ordonnées. En effet, la création d'une entrée peut nécessiter l'insertion d'information entre des entrées existantes. Cette opération est très pénalisante en temps. Par conséquent, on utilise, en plus de la table des identificateurs, une **table d'index** qui permet d'assigner à chaque enregistrement dans la table des identificateurs un index dans cette table qui correspond à un tri sur une clé. Le plus souvent, la clé utilisée est le nom et l'ordre utilisé est l'ordre alphabétique. Ainsi le k-ième nom, par ordre d'arrivée, est associé au k-ième enregistrement dans la table des identificateurs et est le i-ème par ordre alphabétique. Seuls des déplacements dans la table d'index sont à effectuer en cas d'insertion.



② TABLE DE HACHAGE

La structure de données comporte deux parties:

- la table de hachage qui est un tableau de m pointeurs vers des entrées dans la table des identificateurs,
- les entrées dans la table des identificateurs organisées en m listes chaînées distinctes qui peuvent être vides.



Une table de hachage de taille 211

Pour déterminer si une entrée existe pour la chaîne x dans la table des identificateurs, nous appliquons à x une **fonction de hachage** (hashcode) h , telle que $h(x)$ retourne un entier compris entre 0 et $m-1$. Si x est présent dans la table des identificateurs, il est alors dans la liste numérotée $h(x)$. Si x n'est pas encore présent dans la table des identificateurs, on l'y entre en créant un enregistrement pour x qui est chaîné en tête de la liste numérotée $h(x)$.

Fonction de hachage

En général, la fonction de hachage h n'est pas injective. On parle de collision (primaire) lorsque $h(x_1) = h(x_2)$ pour deux chaînes différentes x_1 et x_2 . Une bonne fonction de hachage doit, par conséquent:

- donner au moins (et pas plus) toutes les valeurs possibles d'indices dans la table,
- donner aussi peu de collisions que possible.

Une fonction donnée n'est meilleure que les autres que pour certaines distributions des entrées.

Il faut aussi prendre en compte le temps de calcul de la fonction de hachage. Ce temps est, de toute façon, inférieur au temps de comparaison de chaînes de caractères. Le choix d'une fonction, son évaluation sont des

problèmes difficiles à résoudre. Le comportement de la fonction dépend de la taille de la table, du taux d'occupation, de la distribution des entrées dans l'espace et dans le temps.

Dans le contexte de la table des identificateurs, la fonction de hachage doit être appliquée à une chaîne et doit fournir un indice dans la table. De nombreux travaux ont été consacrés à la conception d'une fonction de hachage qui soit facile à calculer sur les chaînes de caractères et qui répartisse uniformément les chaînes parmi les m listes. Une approche convenable est la suivante:

- déterminer un entier positif p à partir des caractères de la chaîne x . La conversion d'un caractère en un entier est habituellement fournie par le langage d'implantation (fonction `ord` en Pascal);
- convertir l'entier p en un numéro de liste, i.e. un entier compris entre 0 et $m-1$. Une pratique raisonnable consiste à prendre simplement le reste de la division de p par m .

Les fonctions de hachage qui prennent en compte tous les caractères de la chaîne x sont meilleures que celles qui ne considèrent que quelques caractères, par exemple aux extrémités ou vers le milieu de la chaîne x .

Exemples de fonction de hachage simples:

- une technique simple pour calculer p consiste à additionner les valeurs entières des caractères de la chaîne.
- une meilleure solution, pour le calcul de p , est de multiplier la valeur de p par une constante α avant d'y ajouter le caractère suivant: $p_0 = 0$, $p_i = \alpha p_{i-1} + c_i$ pour $1 \leq i \leq k$ et $p = p_k$ où k est la longueur de la chaîne. L'addition simple de tous les caractères est obtenue avec $\alpha = 1$.
- Une stratégie similaire, toujours pour calculer p , consiste à effectuer un "ou exclusif" de c_i avec $\alpha h_{i-1} + c_i$ au lieu de les additionner.

Recherche dans la table

Tous les noms de même valeur de hachis sont dans la même liste chaînée. Si la fonction a été correctement choisie pour l'ensemble des noms, si la table est correctement dimensionnée, aucune liste ne devrait être longue. Chaque liste peut néanmoins être triée si nécessaire, ou être elle-même représentée par une table de hachage.

La liste linéaire est le procédé le plus simple à mettre en oeuvre, mais ses performances sont mauvaises lorsque le nombre d'identificateurs et le nombre de recherches effectuées dans la table deviennent grands. Les tables de hachage procurent de meilleures performances pour un effort de programmation plus important et un surcoût en occupation de la mémoire.

■ La fonction CHERCHER

La fonction CHERCHER recherche un identificateur dans la table des identificateurs. Elle admet un nom d'identificateur en argument et renvoie l'indice trouvé si l'identificateur est dans la table, ou un indice invalide sinon.

■ La fonction INSERER

La fonction INSERER insère un identificateur dans la table des identificateurs. Elle admet deux arguments, un nom d'identificateur et un genre (constante, variable, fonction, etc.) d'identificateur, et renvoie l'indice de la nouvelle entrée dans la table.

■ La procédure AFFICHE_TABLE_IDENT

La procédure AFFICHE_TABLE_IDENT permet de visualiser la table des identificateurs à l'écran.

■ Le programme principal

De façon à tester les opérations sur la table des identificateurs, le programme principal de l'analyseur lexical doit:

- appeler la procédure INITIALISER,
- reconnaître toutes les unités lexicales du fichier SOURCE,
- pour chaque identificateur reconnu, rechercher son nom dans la table des identificateurs et l'insérer le cas échéant,
- appeler la procédure TERMINER,
- appeler la procédure AFFICHE_TABLE_IDENT.

Remarque: On supposera qu'un nom d'identificateur figure une seule fois dans la table des identificateurs.