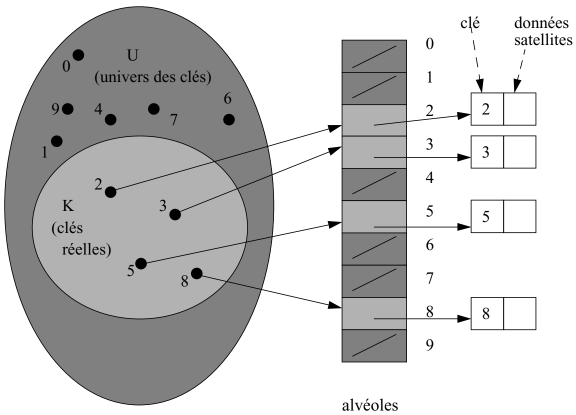
Techniques d'adressage dispersé

Les techniques d'**adressage dispersé** consistent à stocker et à retrouver des enregistrements à partir d'une **clé primaire**.

Tables à adressage direct



Exemple: Dans un compilateur on utilise une **Table des symboles** pour stocker les identificateurs d'un programme en prenant leurs noms comme clés.

Problème: Dans le cas de Fortran, on a 1,617,038,306 possibilités de noms d'identificateurs. Quelle devrait être la taille de cette table?

Solutions:

Utiliser un tableau avec autant d'entrées que d'identificateurs possibles.

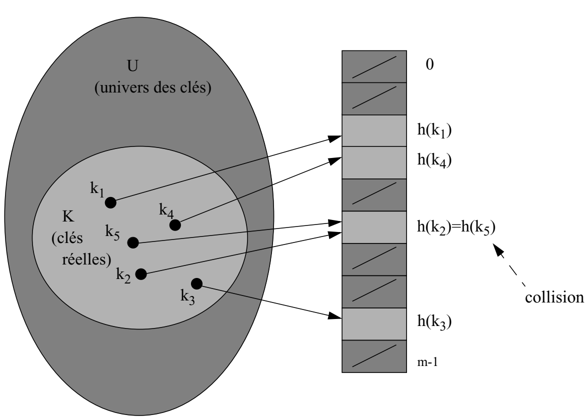
Cette solution permet de faire la recherche et les modifications en un temps constant. Cependant, il y a généralement peu d'identificateurs dans un programme par rapport au nombre d'identificateurs possibles.

Réserver une table au nombre restreint d'entrées (ex: 200 entrées).

Il faut donc une méthode pour faire correspondre chaque identificateur à une entrée de la table. Cette méthode s'appelle: **fonction de hachage** ou **fonction d'adressage dispersé**.

Tables de hachage

H : U -> { 0, 1, . . . , m-1 } /* On dit qu'un élément de clé k est haché dans alvéole h(k), ou h(k) est la valeur de hachage de la clé k */



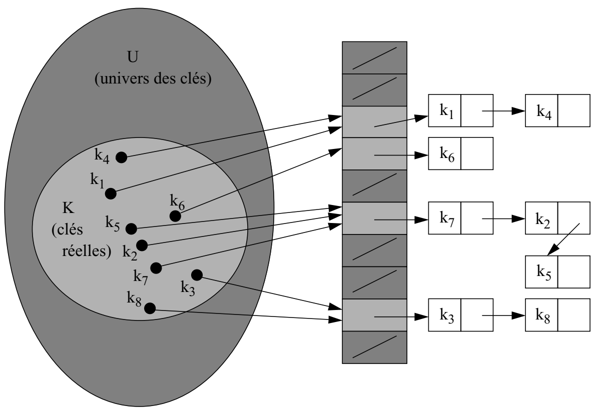
Problème: Plusieurs clés peuvent correspondre à une même entrée (on appelle ces clés des **synonymes**). On dit alors qu'on a des **collisions**.

Donc, il faut une méthode de correspondance qui minimise les collisions (répartir les clés uniformément dans la table).

- Il faut lui ajouter des procédures de **traitement des collisions.**

- Il faut que cette méthode soit rapide.

Résolution des collisions par chaînage



Fonctions de hachage (fonctions d'adressage dispersé)

Les **fonctions de hachage** doivent:
Être déterministes:
Donner le même résultat lorsqu'elles sont appliquées à la même clé.
Assurer une dispersion uniforme des clés sur la zone de stockage.

La fonction de hachage prend une clé k et calcule une adresse H(k).

$$H : U \to \{ 0, 1, \ldots, m-1 \}$$

Le résultat est une adresse dans le tableau de hachage appelée: **Adresse primaire**.

Numérisation des clés

Certaines **techniques de numérisation** consistent à faire correspondre un nombre à chaque caractère de la clé, et puis à concaténer ces nombres pour obtenir une clé numérique.

Par exemple, N('ABCD')=11121314. Si la fonction d'adressage primaire est obtenue en prenant le modulo de cette numérisation, on obtient H(k)=N(k) mod Q. Or N(k) est généralement un nombre trop grand pour être contenu dans un mot machine (même en double précision).

Les exemples de fonctions de hachage

Méthode de la division

H(k) = k mod m

Cett fonction possède les propriétés suivantes:

Facile à implanter.
Rapide à calculer.
Donne une distribution uniforme des clés.
Par contre, elle donne des résultats moyens car la participation de la clé est restreinte.

Pliage logique

Consiste à:

1. Séparer la clé en portions de même longueur
2. Combiner les portions (somme, XOR)
3. Prendre le modulo du résultat

Cette fonction possède les propriétés suivantes:

Est rapide à calculer
Donne de bons résultats car toute la clé peut participer.

Méthodes de résolution des collisions

- adressage ouvert

- méthode de chaînage

Adressage ouvert (méthodes de sondage)

Méthode linéaire séquentielle

$$P_{i+1} = (P_i + 1) \bmod m$$

Cette méthode possède les propriétés suivantes:

Facile à implanter et rapide.
La taille des regroupements augmente de plus en plus rapidement
Plutôt moyenne.

Méthode linéraire avec incrément fixe:

$$P_{i+1} = (P_i + \text{incrément}) \bmod m \quad (\text{incrément} > 1)$$

Méthode linéraire avec incrément variable:

$$P_{i+1} = (P_i + f(k)) \bmod m \quad (\text{double hashing})$$

Cette méthode donne de bons résultats.

Méthodes de chaînage

Chaînage externe des synonymes

Cette méthode consiste à faire des listes chainées des synonymes dans une zone séparée

Chaînage interne des synonymes

Cette méthode consiste à faire des listes chainées des synonymes dans la table elle-même

Il y a 3 possibilités lors d'une insertion d'une clé k:

H(k) est libre
H(k) est occupée par une clé k_1 t.q. $H(k) = H(k_1)$
H(k) est occupée par une clé k_1 t.q. $H(k) \neq H(k_1)$

Cette méthode possède les propriétés suivantes:

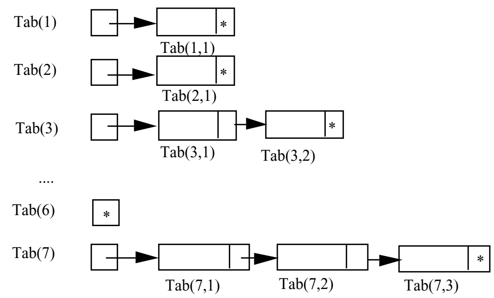
Elle est excellente.
Elle demande peu d'espace.

INF4393 - Jurek Czyzowicz

Applications de ces techniques

Tables de symboles (identificateurs d'un programme)

Matrices creuses (i.e. à dimensions flexibles):



Implantation des dictionnaires (de définitions de mots)
Accès direct dans les fichiers.