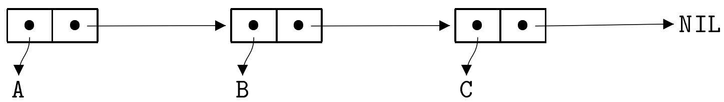
8 Représentation des listes, garbage collection

8.1 Doublets

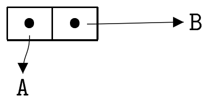
Dans 1.2 nous avons déjà vu comment une liste peut être représentée en doublets.

Un doublet consiste en un pointeur vers le `car` de la liste et un pointeur vers le `cdr` de la liste. Ces deux peuvent être des atomes ou à nouveau des doublets. (Dans 1.2 nous avons simplifié la représentation un peu en mettant directement l'atome au lieu d'un pointeur vers cet atome.)

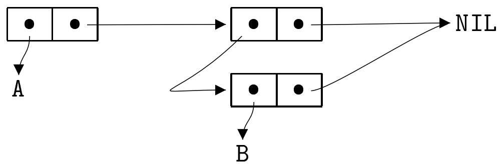
Liste `(a b c)`



Le doublet `(a . b)`:



La liste `(a (b))`:



8.2 append versus nconc

`append` prend deux listes et en forme une nouvelle sans modifier la première liste. C'est-à-dire `append` ne remplace pas le dernier pointeur vers `nil` dans la première liste par un pointeur vers la deuxième. Mais, au lieu de cela, `append` copie la première liste et dans la copie le dernier pointeur est modifié.

```
> (setq list1 '(a b))
(A B)
> (setq list2 '(c))
(C)
> (setq result (append list1 list2))
(A B C)
> list1
(A B)
```

Par contre, la fonction `nconc` produit la même liste que `append` en modifiant le dernier pointeur dans la première liste.

```
> (setq result (nconc list1 list2))
(A B C)
> list1
(A B C)
```

`append` a alors l'avantage de ne pas modifier des listes déjà existantes, mais `append` utilise beaucoup de mémoire en faisant des copies de ses premiers arguments.

8.3 Garbage collection

LISP gère une liste des doublets libres, et chaque fois qu'on a besoin d'un nouveau doublet, on prend un doublet de cette liste. Pour remplir cette liste de temps en temps, LISP collecte tous les doublets utilisés dont on n'a plus besoin et les remet dans la listes des doublets libres. Ce procès s'appelle *garbage collection*.

Garbage collection marque d'abord tous les doublets encore utilisés : Commençant par les définitions de fonctions et variables actuelles et par les fonctions appelées, le système suit les pointeurs et marque tous les doublets accessibles à partir des variables actuelles. Ensuite, tous les doublets non-marqués sont rendus á la listes des doublets libres.