

Próximo:[argolas](#), Anterior:[Tabelas de char](#), Acima:[Vetores de matrizes de sequências](#) [[Conteúdo](#)][[Índice](#)]

## 6.7 Bool-vetores

Um vetor bool é muito parecido com um vetor, exceto que ele armazena apenas os valores `t` e `nil`. Se você tentar armazenar qualquer não `nil` valor em um elemento do vetor bool, o efeito é armazenar `tlá`. Como em todos os arrays, os índices bool-vector começam em 0 e o comprimento não pode ser alterado depois que o bool-vector é criado. Bool-vetores são constantes quando avaliados.

Várias funções trabalham especificamente com vetores bool; além disso, você os manipula com as mesmas funções usadas para outros tipos de arrays.

### Função: *inicial do comprimento do vetor make-bool*

Retorna um novo vetor bool de elementos de *comprimento*, cada um inicializado com *inicial*.

### Função: *bool-vetor &rest objetos*

Esta função cria e retorna um bool-vetor cujos elementos são os argumentos, *objetos*.

### Função: *objeto bool-vetor-p*

Isso retorna `t` se o *objeto* for um vetor bool e `nil` caso contrário.

Existem também algumas funções de operação do conjunto bool-vetor, descritas abaixo:

### Função: *bool-vetor-exclusivo-ou a b &opcional c*

Retorna *bit a bit exclusivo ou* de vetores bool *a* e *b*. Se o argumento opcional *c* for fornecido, o resultado desta operação será armazenado em *c*. Todos os argumentos devem ser vetores booleanos do mesmo tamanho.

### Função: *bool-vector-union a b &opcional c*

Retorno *bit a bit ou* de vetores bool *a* e *b*. Se o argumento opcional *c* for fornecido, o resultado desta operação será armazenado em *c*. Todos os argumentos devem ser vetores booleanos do mesmo tamanho.

### Função: *bool-vetor-intersecção a b &opcional c*

Retorno *bit a bit e* de vetores bool *a* e *b*. Se o argumento opcional *c* for fornecido, o resultado desta operação será armazenado em *c*. Todos os argumentos devem ser vetores booleanos do mesmo tamanho.

### Função: *bool-vector-set-difference a b &opcional c*

Retorna *a diferença* do conjunto de vetores bool *a* e *b*. Se o argumento opcional *c* for fornecido, o resultado desta operação será armazenado em *c*. Todos os argumentos devem ser vetores booleanos do mesmo tamanho.

### Função: *bool-vector-not a &opcional b*

Retorna o *complemento* do conjunto do vetor bool *a*. Se o argumento opcional *b* for fornecido, o resultado desta operação é armazenado em *b*. Todos os argumentos devem ser vetores booleanos do mesmo tamanho.

### Função: **bool-vetor-subsetp a b**

Retorna tse todo tvalor em *a* também estiver tem *b* , nilcaso contrário. Todos os argumentos devem ser vetores booleanos do mesmo tamanho.

### Função: **bool-vector-count-consecutive a bi**

Retorna o número de elementos consecutivos em *um b* igual começando em *i* . aé um vetor bool, *b* é t ou nil, e *i* é um índice em *a*.

### Função: **bool-vector-count-population a**

Retorna o número de elementos que estão tno vetor bool *a* .

O formulário impresso representa até 8 valores booleanos como um único caractere:

```
(vetor bool t nil t nil)
  ⇒ #&4"E"
(bool-vetor)
  ⇒ #&0""
```

Você pode usar vconcatpara imprimir um vetor bool como outros vetores:

```
(vconcat (vetor bool nil t nil t))
  ⇒ [nil t nil t]
```

Aqui está outro exemplo de criação, exame e atualização de um vetor bool:

```
(setq bv (fazer-bool-vetor 5 t))
  ⇒ #&5"^-"
(aref bv 1)
  ⇒ t
(ativo por 3 nulo)
  ⇒ nada
bv
  ⇒ #&5"W"
```

Esses resultados fazem sentido porque os códigos binários para control-\_ e control-W são 11111 e 10111, respectivamente.

Próximo:[argolas](#), Anterior:[Tabelas de char](#), Acima:[Vetores de matrizes de sequências](#) [[Conteúdo](#)][[Índice](#)]