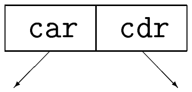
DATENSTRUKTUR LISTE

Interne Struktur

- Wird vom LISP-Reader bei Eingabe einer Liste automatisch erzeugt.
- variable Länge von Listen
  ⟶ Realisiert durch Paare von Verweisen

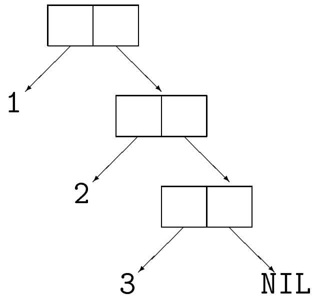
`cons`-Element:



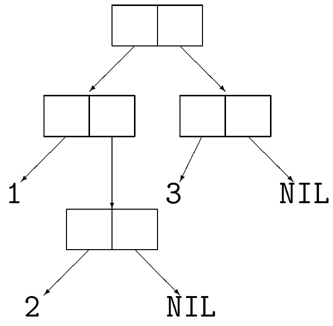
- Die letzte Restliste ist **NIL**.
- `car`-Komponenten der Paare heißen Elemente der Liste.

Beispiel:

`(1 2 3)`



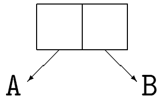
`((1 2) 3)`



DATENSTRUKTUR LISTE

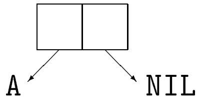
Externe Darstellung

- Wird vom LISP-Printer bei der Ausgabe von `cons`-Elementen erzeugt.



extern:

`(A . B)`



extern:

`(A)`

Struktur einer Parameterliste

LISP stellt einen überaus mächtigen Parameterübergabemechanismus zur Verfügung. Eine Parameterliste (auch *Lambdaliste* genannt) hat allgemein folgende Struktur:

({*var*}* **1**
  [**&optional** {*var* | (*var* [*initform* [*svar*]])}*] **2**
  [**&rest** *var*] **3**
  [**&key** {*var* | ({*var* | (*keyword var*)} [*initform* [*svar*]])}*
    [**&allow-other-keys**]] **4**
  [**&aux** {*var* | (*var* [*initform*])}*] **5**
)

var und *svar* sind Symbole, *initform* ein beliebiger symbolischer Ausdruck.

Jeder einzelne oder sogar alle Teile dürfen leer sein.

Struktur einer Parameterliste

Jedes Element der Parameterliste ist entweder eine Parameterspezifikation oder ein spezielles Schlüsselwort. Diese Lambdalisten-Schlüsselwörter sind daran zu erkennen, daß das erste Zeichen **&** ist.

Beim Funktionsaufruf werden die formalen Parameter der Reihe nach durch die entsprechenden Funktionsargumente ersetzt. Die Abarbeitung erfolgt dabei immer von links nach rechts. Wichtig hierbei ist, daß Aufrufmuster und Parameterliste miteinander vereinbar sein müssen.

1. Obligatorische Parameter

Alle formalen Parameter bis zum ersten Schlüsselwort, bzw. bis zum Ende der Parameterliste, sind notwendig und müssen daher beim Aufruf einer Funktion angegeben werden. Die Parameterspezifikation eines obligatorischen Parameters besteht aus einem Symbol, das als lexikalische Variable behandelt wird.

2. Optionale Parameter

Die durch **&optional** ausgezeichneten Parameter können, müssen aber nicht beim Aufruf spezifiziert werden. Falls kein entsprechendes Argument vorhanden ist, wird *initform* ausgewertet und dieser Wert an den Parameter gebunden. Wurde *initform* nicht angegeben, so ist in diesem Fall der Wert des Parameters gleich **nil**. Die Variable *svar* erhält den Wert **t**, falls der optionale Parameter mit einem Argument gebunden wurde und **nil**, falls *initform* ausgewertet werden mußte. Die Angabe von *svar* ist optional.

3. Restparameter

Der Restparameter wird gebunden an eine (möglicherweise leere) Liste noch verbleibender Argumente. Durch Ausnutzung des Restparameters lassen sich somit Funktionen definieren, die mit beliebig vielen Argumenten aufgerufen werden können. Wird der Restparameter mit Schlüsselwortparametern kombiniert, so werden dieselben verbleibenden Argumente bei der Verarbeitung der Schlüsselwortparameter *nocheinmal* berücksichtigt!

4. Schlüsselwortparameter

Schlüsselwortparameter stellen ein anderes Konzept zur Behandlung optionaler Argumente dar. Im Gegensatz zu den anderen Parametertypen basiert die Parameterzuweisung dabei nur bedingt auf der Stellung der Argumente in der Aufrufstruktur. Stattdessen wird dem entsprechenden Argument beim Aufruf ein Schlüsselwort vorangestellt. Dieses Schlüsselwort ergibt sich aus einem **:** unmittelbar gefolgt von dem Namen des formalen Parameters, oder durch explizite Angabe eines Namens.

5. Hilfsvariablen

Als letztes Argument einer Parameterliste kann, eingeleitet durch das Schlüsselwort **&aux**, eine Spezifikation lokaler Hilfsvariablen erfolgen (→ **let***).

PARAMETERÜBERGABE

Beispiel:

```
(a b)                    3 15      a=3 b=15
(a &optional (b 2))      3 15      a=3 b=15
(a &optional (b 2))      3         a=3 b=2
(a &rest b)              3 15      a=3 b=(15)
(a &key b)               3 :b 15   a=3 b=15
(a &key (b 2))           3         a=3 b=2
(a &key (b 2))           3 :b 15   a=3 b=15
(a &key ((:a b) 2))      3 :a 15   a=3 b=15

(a &optional (b 3) &rest x)     3 4 5
                          a=3 b=4 x=(5)

(a &optional (b 3) &rest x)       3 4 5 6 7
                          a=3 b=4 x=(5 6 7)

(a &optional (b 3) &rest x
   &key c (d 6) (e 7))            'a 'a :d 2 :c 1
            a=A b=A c=1 d=2 e=7 x=(:D 2 :C 1)
```