リスプオブジェクトの実装

リスプオブジェクトとはアトムまたはリストである。

Chaitin のリスプでは、すべてのリスプオブジェクトは、二つの long 型配列 car[SIZE]とcdr[SIZE]によって実現されるノード上に配置される。ノード番号=配列添字は、

$$0 \ \leq \ \text{long型の数値} \ < \ \text{SIZE}$$

を取る。以後これを「(ノード)アドレス」と呼ぶことにする。cdr[]はノードアドレスを値とし、 car[]はノードアドレスまたは文字の ascii コード番号を値とする。

次の図はアドレス $z0$ に配置されたリストの例である。（nil＝0 に注意）

リストは今後、右側のような簡略した図で表現する。使用されたノードは使いっぱなしでゴミ集めの機構はない。未使用ノードのうち、最初のもののアドレスを大域変数 next に保持する。(→ 関数 cons を参照)

48行目：

next . long 型の大域変数。未使用ノードの先頭アドレス。0 ≤ long型の数値 < SIZE

next ≥ SIZEになった時点でオーバーフローとなる。

■ ノードアドレスzに存在するリスプオブジェクトの属性

アドレス z にあるリスプオブジェクトの属性を記述するために次に示す幾つかの配列が使われる。

car[z] .（z から始まる）リストの場合、その car 。値はノードアドレス、または文字のコード番号

cdr[z] .（z から始まる）リストの場合、その cdr、即ち、次のノードのアドレス。

atom[z] . リスプオブジェクト z がアトムであるかどうかのフラグ。

アトムである場合 1、そうでない場合、0 の値を取る。

numb[z] . リスプオブジェクト z が数値（従って同時にアトム）であるかどうかのフラグ。

数値である場合 1、同でない場合 0 の値を取る。

以下の配列はリスプオブジェクト z がアトムの場合に意味を持つ。リストの場合、これらの z での値は 0 に初期化される。(→ 261-264 行目、関数 cons の定義の中で)

pname[z] . アトムであって数値でな場合、 mk_string を使ってその語（=アトムの名前）を構成する文字列を逆順に並べたリストを作る。数値の場合には、in_word2 による式の読み込みの際に先頭部に続く余分な 0 を取り去りって、数字の逆順のリストを作る。(382 行目)
pname[z] は、それらのリストの先頭アドレスを保持する。

pf_numb[z]. アトム z が組み込みの基本関数を表す語である場合、その識別番号を保持する。
（481 行目に始まる関数 eval の中で switch 構文(514 行目) の分岐に使用する。）
car, cdr から始まり、read-exp まで、1 から 23 までの番号が振られている。(169-191行目)

その他、演算子、'(=quote)、lambda, if などの特殊形式をあらわす語、および関数 in による読み込みの際にマクロ展開される define, let, cadr, caddr などの語の識別番号は 0 としている。

pf_args[z] . アトム z が関数を表す語である場合、[引数の個数]+1>0。pf_args[z] が 0 のときは、関数ではないことを表す。

vlst[z] . アトム z に関する、束縛値のリストの先頭アドレス。

プログラム中に現れるアトムのアドレスを記録するために一つのリスト="オブジェクトリスト"が用意される。

obj_lst . オブジェクトリストの先頭アドレス。数値以外のアトムのアドレスはこのオブジェクト

リストに追加されていく。(→ 216 行目、322 行目)

■ リスプオブジェクトのアドレスの確保/リストの生成

リスプオブジェクトのアドレスの確保は、すべて関数 cons が行う。

- リストの生成

  $z \leftarrow \mathbf{cons}(x, y)$

  y . リストのアドレス

  x . リスト y の先頭に追加するリスプオブジェクトのアドレス

大域変数 next には、未使用ノードの内で一番若いアドレスが保持されている。cons は、新たなオブジェクトためのアドレスとして next の値を返すとともに、next が次の未使用ノードを指すよう next の値を 1 だけ増す。大域変数 next の値を変えるのは、cons だけである。

```
z = next++      (←256行目)
```

ただし、cons の呼び出し時点で next ≥ SIZE であるならば、既にノードアドレスが尽きているので

"Storage overflow!"

のメッセージとともにプログラムを停止する。(→253 行目)

引数 y はリストであるのが正常である。y がリストでない（即ち、nil でなく、かつアトムである (atom[y]==1) とき ）ような異常な呼び出しに対して cons(x,y) は x を返す。(→249 行目)

返り値 z はリスト（のアドレス）なので、

atom[z], numb[z], pname[z], pf_numb[z], pf_args[z], vlst[z]

は全て 0 に初期化される。（ pname[z], vlst[z] については nil(=0) と言うべきかもしれない。）

アトムの実装と生成

リスプにおいてアトムは数値か、記号列である。例えばリスプ式

```
(abc  27)
```

において記号列 abc と数値 27 がアトムである。

アトムはリスプオブジェクトなので固有のアドレスを持つ。アトムは、数値である場合と数値でないアトムとで扱いが若干異なる。

先ず数値以外のアトムについて見ておこう。

■ 数値以外のアトム

■ 数値アトム