ФУНКЦИИ И ПРЕДИКАТЫ ПЛЭННЕРА

```
[ELEM n l]      - взять n-ый эл-т списка l (сокр. [n l]),
                  n>0 - слева направо, n<0 - справа налево
                  (пр. [.v l] ,  [:c l] - с выч. перем.)
[INDEX l n1 n2 ... nk] = [ELEM nk [...[ELEM n1 l]...]]
[REST n l]      - N первых эл-тов отбр. (N>=0), или |N| (N<0).
[HEAD n l]      - N первых эл-тов ост. (N>=0), или |N| (N<0).
[LENGTH {a|l}] - длина
[EVAL e]        - насильств. вычисление выражения
[FORM << e1 ... ek >> ] -> << E1 ... Ek >> ("<<",">>" - любые скобки)
[MAX n1 ... nk], [MIN n1 ... nk], [RANDOM], синусы и т.п.
[NOT e], [OR e1 ... ek], [AND e1 ... ek]
[COND (p1 e11...e1m1) ... (pk e11 ... ekmk)]
[NUM e]         - если E = N (число), то T.
[EQ e1 e2]      - для произв. выражений
[MEMB e l]      - входит ли E в список L - возвр. первое вхождение
[PROG (v1 ... vk) e1 ... em] - возвр. Em, если нет RETURN,
                  vj = {ij | (ij E) }, где E - иниц. значение
[SET i e]       - присваивание, возвр. E
[RETURN e]      - возвр. E
[GO i]
[DO op1 ... opk] - сост. оператор, внутрь вх. нельзя, вых. можно по-разн.
[COND (p1 op11 ... op1m1) ...] - можно опустить (T ...) - пойдет дальше
[FIN I1 I2]     - перемещает в I1 первый элемент списка I2.
                  (пр.: [PROG ((R ()) E) A [COND([FIN E L] [RETURN .R])]
                                           [SET R (.E !.R)] [GO A]]
[LOOP x l e1 ... ek] - для кажд. элем. L (котор. в x) вып. e1 ... ek
[FOR x n e1 ... ek], [WHILE p e1 ... ek], [UNTIL e1 ... ek p]
[CSET i e]      - присваивание константе
[PLIST i pl], [PUT i ind v], [GET i ind], [GET i] - работа со свойствами
                  pl - список свойств - (IND1 V1 IND2 V2 ...)
[OPEN f t]      - открыть файл f (SCREEN,<filename>) для t (GET,PUT)
[ACTIVE f t]    - активизировать (выбрать текущим)
[CLOSE f t]     - закрыть (в лекциях формат не рассматривался)
[DEFINE fn (LAMBDA (v1 ... vk) e)] - опр. функции, возвр. имя ф-ии FN
[TIME], [CLOCK], [TRACK] - служебные
[ATL a] -> (lit1 A ... litk A) - Atom To Litera(s)
[LTA l]         - наоборот, Litera(s) To Atom
[EXIT e fn n], [EXIT e fn] - выход из n-го рекурс. вызова ф-ии fn с вып. E.
                  [EXIT () ()] - выход из программы
[CATCH e1 e2]   - вычисляется E1, если ошибка, то переходим к E2
  [ERRINF] -> (номер_ошибки контекст имя)



[IS pat e] -> T или ():
                1) pat=A    - res := E=A
                2) pat=*v   - res := T, v:=E
                3) pat=.v   - res := если v не им. знач.,то (2), иначе V=E
                4) pat=L    - res := послед. сопост. элементов (закон min)
                5) pat=обр. к ф-ии - res := рез. ф-ии = E
                6) pat=обр. к сопост. - res := (???)
                7) pat=сегм.элементы  - res := (по правилам десегментации)
     (пр.: [IS (!*x+!*y)(1+2+3+4)]->T  (правило min): x->(1), y->(2+3+4) )
  [LIST n],<LIST n> - сопоставитель - список из n элементов
  [ET s1 ... sk], [AUT s1 ... sk] - или/и для сопоставлений
     (пр.: [DEFINE F1 (KAPPA () [AUT () [LIST 1] [SAME (x) (*x <F1> .x)]])]
      - определение сопоставления, здесь пров. на палиндром)
```

```
[FAIL], [FAIL e] - возвр. неуспех к ближ. разв. со зн. E (читать - [MESS])
[FP i]           - возвр. к именованной развилке (не ясно, где д.б. метка)
[AMONG l]        - если () - FAIL, иначе развилка по всем эл-там L,
                   значение ф-ии - каждый элемент L (по порядку до успеха).
[ALT e1 ... ek]  - развилка, по порядку с 1 до k, если Ei - успешно, конец
     (пр.: [DEFINE SUM (LAMBDA(L N) [PROG(K (M ()) (S 0))
            A [SET K [AMONG .L]] [SET M (.K !.M)] [SET S [+ .S .K]]
              [COND ([EQ .S .N] [RETURN .M]) ([LT .S .N] [GO A])
                    (T [FAIL])])])   - [SUM (6 3 2 1) 5] = (2 3)
      все рез-ты: [PROG() [ALT() [RETURN T]] ...
                   [PRINT [SUM (6 3 2 1) 5]] ... [FAIL] ]        )

[PSET i e]       - неотменяемое (в развилках) присваивание
[ASSERT astr {with} {rec} {else}] - операция записи в БД
    (astr - утверждение, with - список св-в, rec - рекомендации по выбору
те-
     орем, else - реакция, если запись не удалась)
    (пр.: [ASSERT (BOX A) (WITH COL .X) (ELSE)] - COL=цвет, (ELSE) = [FAIL])
[ERASE astr {with} ..???] - операция вычеркивания из БД
[SEARCH pat test] - опреация поиска в БД (pat - образец, test - спис. св-в)
                    ставится развилка на все найденные
    (пр.: [SEARCH (BOX []) (TEST COL [NON RES])]   )
[SEARCH1 pat test] - ищется только один объект
типы теорем CONSEQ (целевая), ANTEC (при записи), ERASING (при вычеркивании)
    (пишутся вместо LAMBDA...)
пример целевой теоремы:
    [DEFINE TR-R (CONSEQ (x y) (AT R *y) [SEARCH1 (AT R *x)]
     [ERASE (AT R .x)] [ASSERT (AT R y)])] - перемещ. робота из X в Y
[ACHIEVE pat {rec}] - вызов целевой теоремы
[GOAL oat {test} {rec}] - SEARCH+ACHIEVE - сначала ищет такое утв.,
                          если нашли, то не надо доказывать.
```