

Zusammenfassung LISP

Ausdrücke:

(+ A B C)

+ = Operator: +, -, *, /, GCD, MAX, EXPT

A B C = Parameter. Jeder Parameter kann auch wieder eine Liste sein.

(**SETQ** L. '(A B) M Mwert N Nwert): weist L den Wert (A B) zu., M den Wert Mwert und N Nwert. Die letzte Zuweisung wird ausgegeben!

': wird das weggelassen, wird die Liste ausgewertet, sonst nicht.

GENERELL:

ATOM = - A, TEST, „a b c d“ =

LISTE = (A B (C D))

EVAL: Ausdruck auswerten, d.h. in den Atomen nachschauen, was da „druntersteckt“ (s. Folie 15)

CONS: 1. Parameter an Liste (2. Parameter) VORNE anhängen

(**CAR** Liste) : nur erstes Element

(**CDR** Liste) : Rest der Liste außer erstes Element

C und R stehen immer; alles dazwischen wird von rechts nach links ausgewertet d.h.:

(**CAADR** Liste) : zuerst Rest davon erstes davon erstes

(**APPEND** Liste1 Liste2): nimmt die Elemente beider Listen und fügt sie zu einer neuen Liste zusammen

(**LIST** Liste1 Liste2 Atom3): packt alle Elemente (auch Listen) eine Liste drum

(**REVERSE** Liste): Elemente der Liste umdrehen (F27)

LAST Liste: letztes Element

Es gilt:

T oder irgendein Wert: wahr

NIL: Falsch: (beachte: **NIL** = '() !)

(**ATOM** Ausdruck): T wenn Ausdruck ein **Atom** (also keine Liste) ist.

(**CONSP** Ausdruck): T wenn Ausdruck eine **Liste** (also kein Atom) ist. (P = Prädikat)

(**NULL** ...): T wenn leere Liste

(**LISTP** ...): **CONSP** oder **NULL**

(**NUMBERP** Ausdruck): → T wenn Ausdruck eine Zahl ist.

(**SYMBOLP** Ausdruck): → T wenn Ausdruck ein „Symbol“ ist (z.B: Variable oder Variablenkette)

(**STRINGP** Ausdruck): → T wenn Ausdruck ein String (in „“) ist.

Generell:

(**TYPEP** Ausdruck **Typbezeichnung**): Typbezeichnung z.B. NUMBER; SYMBOL; INTEGER; SINGLE-FLOAT

beachte: als einziges Element ist NIL Liste UND Atom zugleich !

Vergleiche:

(**EQAL** Ausdruck1 Ausdruck2): *T wenn in beiden dasselbe steht*
(**EQ** Ausdruck1 Ausdruck2): *T wenn in beiden dasselbe steht UND am gleichen Ort gespeichert ist !!*

(**MEMBER** Ausdruck Liste): *→ Ausgabe der Restliste (inkl. Ausdruck) wenn Ausdruck in der Liste ist. (nur auf 1. Ebene wird geprüft!)*

(**MEMBER** Ausdruck Liste : **Test EQUAL**): *testet mit EQUAL statt mit EQ*

(**LENGTH** Liste) *→ Ausgabe der Anzahl der Listenelemente*

(**=** Ausdruck1 Ausdruck2 Ausdruck3) *T wenn alle Zahlen gleich sind. (genauso: /= für ungleich)*

(**<** Ausdruck1 Ausdruck2 Ausdruck3) *T wenn alle Zahlen aufsteigend sortiert. (genauso: >, <=, >=)*

(**AND** A1 A2 A3): *T wenn alle T sind*

(**OR** A1 A2 A3): *Ausgabe des **ersten** Wertes, der nicht NIL ist („irgendwas“ ist ja auch „T“ (s.o.))*

(**NOT** A): *klar.*

(**COND** (Test1 Befehl1) (Test2 Befehl2) (Test3 Befehl3)): *der erste Befehl, beim dem „Test“ wahr ist, wird ausgeführt.*

(**DEFUN** Funktionsname (ParameterListenName) (Ausgabe))

(in der Ausgabe kann der Parameter verwendet werden!)

definiert eine Funktion.

(**LAMBDA** Funktionsname (ParameterListenName) (Ausgabe)) *(Die Parameter Liste selbst)*

(**LET** (X 2) (LIST X)):

alle Wertzuweisungen innerhalb des LET sind hier gültig! (im Gegensatz zu SETQ)

(**SETQ** FN1 (COND (..) (..))) *möglich: **Variablen können Funktionen als Wert haben !** mit FUNCALL oder APPLY auswerten.*

(**APPLY** + 1 2 3 ,(4 5 6)): *wertet den gesamten Ausdruck bis zum letzten aus,*

(**SYMBOL-FUNCTION** 'FN): *gibt die Funktion „FN“ aus*