

Grundlagen der Informatik

1. Semester, 1999

Kapitel 4: Funktionale Programmierung

4.1 Modell

4.1.1 Arbeitszyklus

4.2 Ausdrücke

4.2.1 Atomare Ausdrücke

4.2.2 Listen

4.2.3 Auswertung von Ausdrücken

4.2.4 Grundoperationen

4.2.5 Funktionsdefinitionen

4.2.6 Selektionen

4.2.7 Rekursion

4.1 Modell

Der **Programmierer** muß

- Daten strukturieren
- Operatoren definieren
- Operatoren auf Daten anwenden

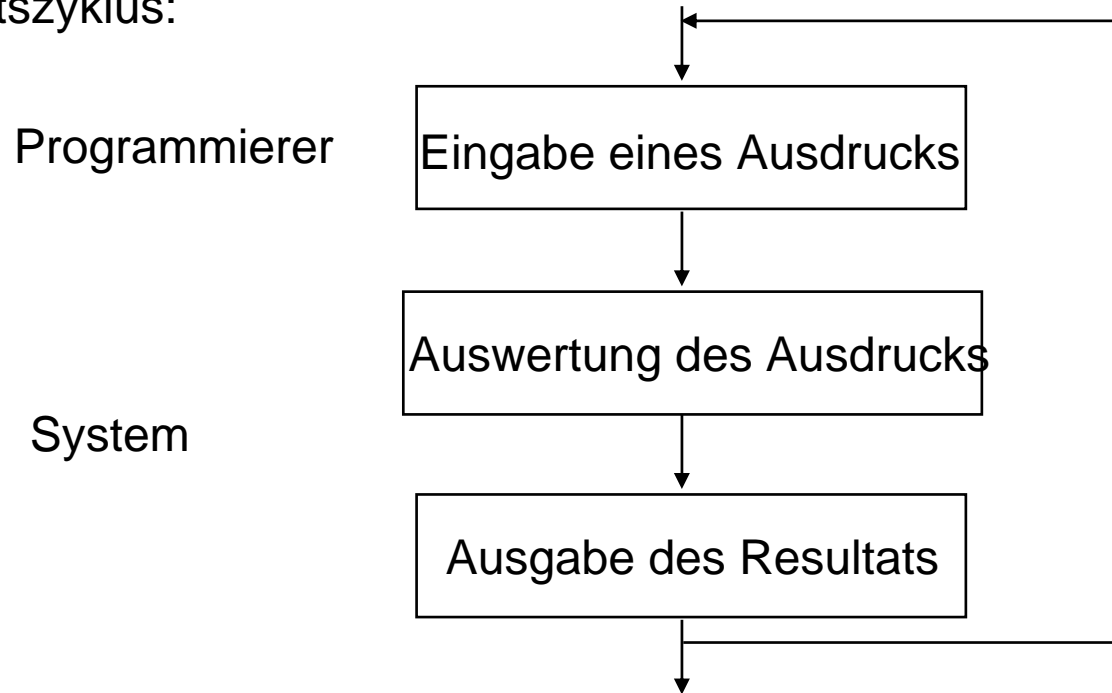
LISP:

- die bekannteste funktionale Programmiersprache
- steht für: **LISt Processing**
oder: **Lots of Idiotic Silly Paratheses**
- Listen werden zur Darstellung von Daten
wie auch von Operationen verwendet.
- hat (zu!) viele Dialekte
z.B. MACLISP, INTERLISP, FRANZLISP, ZETALISP, QLISP, XLISP,
COMMON LISP

LISP: Arbeitszyklus

COMMON LISP:

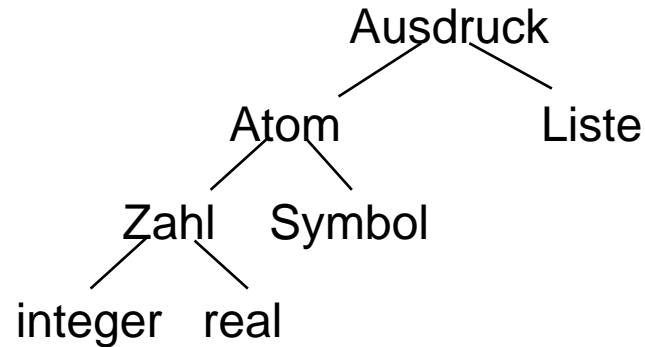
- interaktive Sprache
- Arbeitszyklus:



- Auswertung eines Ausdrucks (Warnung: Ausnahmen!)
 1. Auswertung der Operanden
 2. Anwendung des Operators

4.2 Ausdrücke

Klassifikation:



Zahlen:

gewohnte Darstellung z.B. 27 -36 27.6 3.14e-7

Symbole:

- atomare Bestandteile, die keine Zahlen sind
- z.B. Bezeichner boris BECKER H_Kohl

Ausdrücke

Listen:

- (Element1 Element2 Elementn)

Die Listenelemente sind Ausdrücke (eventuell auch Listen)

- Die Darstellung der leeren Liste ist () oder NIL

- Beispiele:**
- (1 2 3 vier)
 - (apfel pflaume birne)
 - (+ siebzehn vier)
 - ((GRAF HUBER) (HINGIS SCHNYDER) (TAUZIAT PIERCE))
 - Listenform eines binären Baumes

Prinzip: Zwischenordnung

(..Liste für linken Teilbaum.. Wurzel ..Liste für rechten Teilbaum..)

z.B. (((NIL d NIL) b NIL) a (NIL c NIL))

Auswertung von Ausdrücken

- Resultat einer Zahl ist ihr numerischer Wert

z.B. Eingabe: 34

Ausgabe: 34

- Resultat eines Symbols (speziell: Bezeichners) ist der Wert, der an dieses Symbol gebunden ist. Bindungen werden in einer Symboltabelle festgehalten.

Symbol	Value
VFB	10

=> Eingabe: VFB

KSC

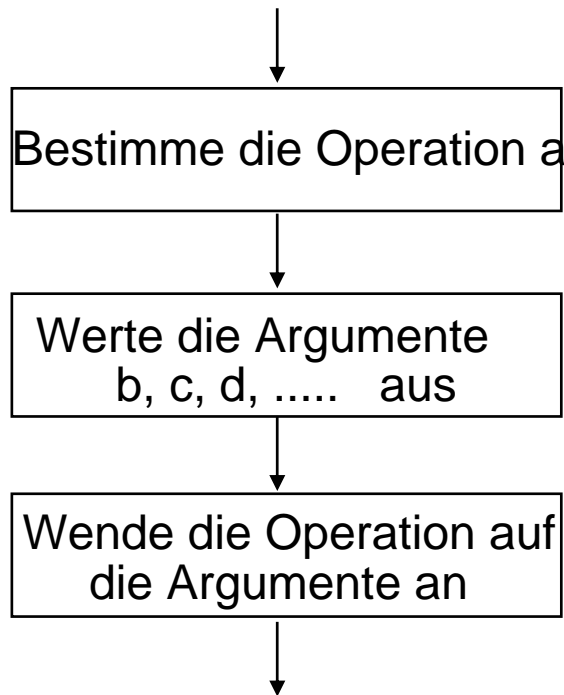
Ausgabe: 6

?

Die folgenden Seiten beschreiben, wie eine Bindung für ein Symbol festgelegt wird.

Auswertung von Ausdrücken

- Die Auswertung einer Liste (a b c d) erfolgt nach dem Muster:
 - * betrachte a als Operator
 - * betrachte b, c, d, als Operanden



Beispiele:

Eingabe: (+ 17 4)

Ausgabe: 21

Eingabe: (+ 17 4 9)

Ausgabe: 30

Eingabe: (MIN 2 4 3)

Ausgabe: 2

Eingabe: (* (MIN 7 4) (MAX 13 11))

Ausgabe: 52

Eingabe: (/ (FLOAT 14) (FLOAT 4))

Ausgabe: 3.5

Grundoperation: QUOTE

QUOTE:

QUOTE ist der Identitätsoperator. (QUOTE (.....)) liefert (.....)

Eingabe: (QUOTE (Tic Tac Toe))

Ausgabe: (Tic Tac Toe)

QUOTE wertet seine Argumente nicht aus! Sonst wäre das Resultat von (QUOTE (Tic Tac Toe)) das Ergebnis der Anwendung des Operators Tic auf die Argumente Tac und Toe.

Anwendung: Der QUOTE Operator wird benutzt, um Daten als Daten zu erhalten

Eingabe: (QUOTE A)

Ausgabe: A

Abkürzung: 'A

Grundoperationen: CAR, CDR, CONS

- zum Aufteilen und Zusammensetzen von Listen

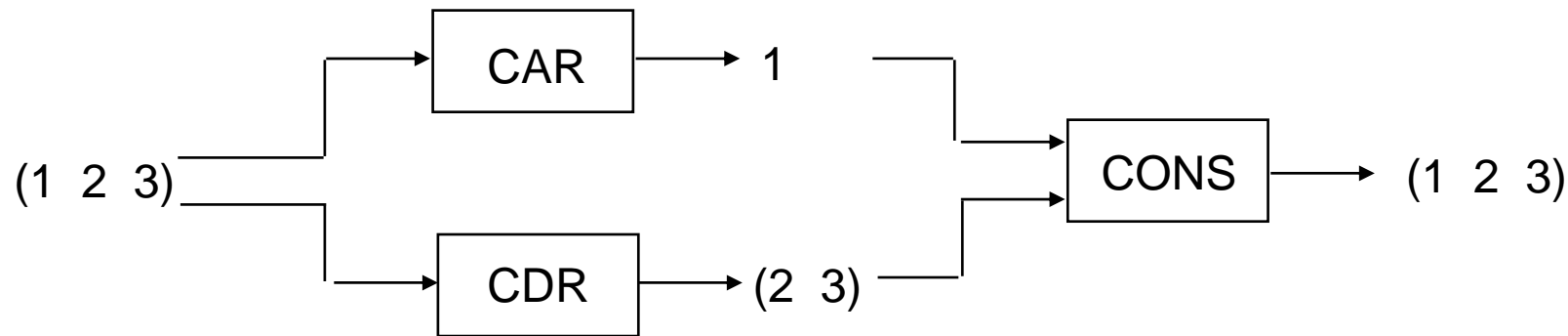
Anwendung: (CAR L)

(CDR L)

(CONS E L)

wobei L eine Liste und E ein Ausdruck ist

Semantik:



Beispiele: (CAR '(ONE TWO THREE FOUR))

==> ONE

(CAR (ONE TWO THREE FOUR)) würde den Operator ONE auf die (ausgewerteten) Argumente TWO, THREE und FOUR anwenden. Falls das Ergebnis keine Liste wäre, könnte CAR nicht angewendet werden.

Grundoperationen: CAR, CDR, CONS

(CDR '(ONE TWO THREE FOUR))

==> (TWO THREE FOUR)

(CAR (CDR '(ONE TWO THREE FOUR)))

==> TWO

(CONS 'ZERO '(ONE TWO THREE FOUR))

==> (ZERO ONE TWO THREE FOUR)

(CONS 'SINGLE NIL)

==> (SINGLE)

(CONS 'SINGLE 'NIL)

==> Fehler, da 'NIL keine Liste sondern die Zeichenkette "NIL" darstellt

Weitere Operationen

(APPEND Liste1 Liste2)

fügt alle Elemente der Argument-Listen in die Ergebnis-Liste ein

(LIST Ausdruck1 Ausdruck2)

liefert eine Liste mit den Elementen: (A1 A2)

wobei A_i das Ergebnis der Auswertung von Ausdruck $_i$ ist.

(REVERSE L)

liefert eine Liste, die die Elemente von L in umgekehrter Reihenfolge enthält.

(LENGTH L)

bestimmt die Anzahl der in L enthaltenen Elemente

Funktionsdefinitionen, Prädikate

Form: (DEFUN Name (Parameter) Ausdruck)

Beispiele: (DEFUN Euro (DM Kurs) (* DM Kurs))

Anwendung: (Euro DM 0.48)

(DEFUN Abschluss (L E) (APPEND L (LIST E)))

Anwendung: (Abschluss BRIEF 'Ade)

Prädikate: sind Operationen die testen, ob bestimmte Bedingungen erfüllt sind

FALSCH wird durch NIL dargestellt

TRUE wird durch T dargestellt

Einige Standardprädikate:

(ATOM A) testet ob A ein Atom ist

(LISTP A) testet ob A eine Liste ist

(NULL A) testet ob A die leere Liste ist

(EQUAL A B) testet ob A und B gleich sind

Verzweigungen (COND, IF)

COND: (COND (Test1 Ausdruck1)
 (Test2 Ausdruck2)

 (Testn Ausdruckn))

Semantik: Die Tests werden von oben nach unten ausgewertet. Beim ersten Test der positiv ausfällt wird der zugehörige Ausdruck ausgewertet

Beispiel: (COND ((EQUAL E (CAR L)) 'Gold)
 ((EQUAL E (CADR L)) 'Silber)
 ((EQUAL E (CADDR L)) 'Bronze))

(COND ((NULL L) NIL) (T (CAR L))

IF: (IF Test AusdruckT AusdruckF)

Beispiel: (IF (NULL L) NIL (CAR L))

Rekursion

Form: (DEFUN F (.....)
 (COND (..... ..))

 (..... (F))
 ))

Beispiel: (DEFUN Atom_zahl (L)
 (COND ((NULL L) 0)
 ((ATOM L) 1)
 (T (+ (Atom_zahl (CAR L)) (Atom_zahl (CDR L))))))

Anwendung: (Atom_zahl '(Na (so was)))
 ==> 3