SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Prof. Dr. Marlon Schumacher

Institut für Musikwissenschaft und Musikinformatik
Hochschule für Musik Karlsruhe

HOCHSCHULE
FÜR MUSIK
KARLSRUHE

Vorlesung 1 - Einführung, Organisation

IMWI Institut für Musikinformatik und Musikwissenschaft

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Moodle

URL https://moodle.hfm-karlsruhe.de/course/view.php?id=1151

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Domain-Specific Languages (DSLs) für Musik

Music N (M. Matthews, Bell Labs, 1960s)
Csound (B.Vercoe, MIT, 1980s)
...
Faust (Orlarey, Grame, 2002)
Max/MSP / PureData (M. Puckette, IRCAM 1991, UCSD, 1998)
Artic, Nyquist (R. B. Dannenberg, CMU, 1989, 1997)
SuperCollider (McCartney, 1998)
Formes (Rodet et al., IRCAM, 1982–85)
PatchWork (M. Laurson et al., IRCAM, 1989)
Common Music (H.Taube, CCRMA, 1991)
OpenMusic (C.Agon et al., IRCAM, 1998)
PWGL (M. Laurson et al., Sybelius Ac., 2002)
Elody (Y. Orlarey et al., Grame, 1997)

- BASED ON LISP
- BASED ON FUNCTIONAL PARADIGM

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Warum LISP?

- eine der ältesten Sprachen in Verwendung [1]
- beliebte Sprache für Künstliche Intelligenz Forschung
- “programmierbare” Programmiersprache - Kontext-sensitiv
- Einfache Syntax: Rapid Prototyping (>700 vordefinierte Funktionen)
- Interpretiert (interaktiv, dynamisch) vereinfacht Experimentation während ein Programm entwickelt wird (*explorative programming*)
- “Hochsprache” (garbage collection, etc.), hohe Flexibilität
- Äquivalenz von Daten und Programmen (“Code as Data”)
- Spaß!

[1] Steele, G. L., Jr, & Gabriel, R. P. (1993). The Evolution of Lisp. *ACM Sigplan Notices*, *28*(3), 231–270.

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

SPRACHE

ANSI Common LISP/
CLOS (Common LISP Object System)

Entwicklungsumgebung

LISPWORKS > 8
OpenMusic > 7

Begleitende Literatur

s. Webseite

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Übersicht für diese Sitzung

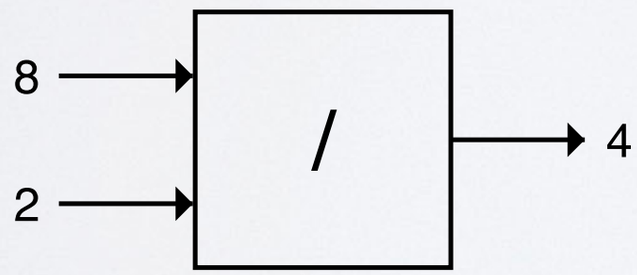
S-Expressions, Formen
Evaluierung
Operatoren
Funktionsdefinition

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

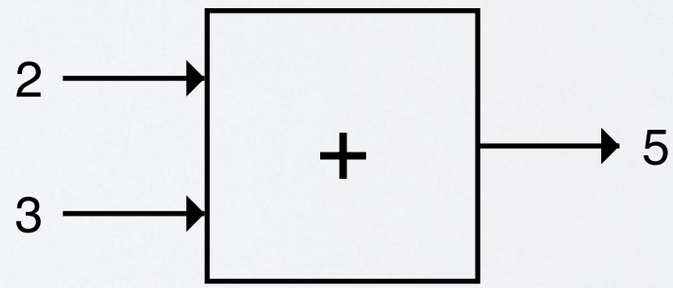
Funktionen und Daten

"The term data means information, such as numbers, words, or lists of things. You can think of a function as a box through which data flows. The function operates on the data in some way, and the result is what flows out." [1]

(/ 8 2)



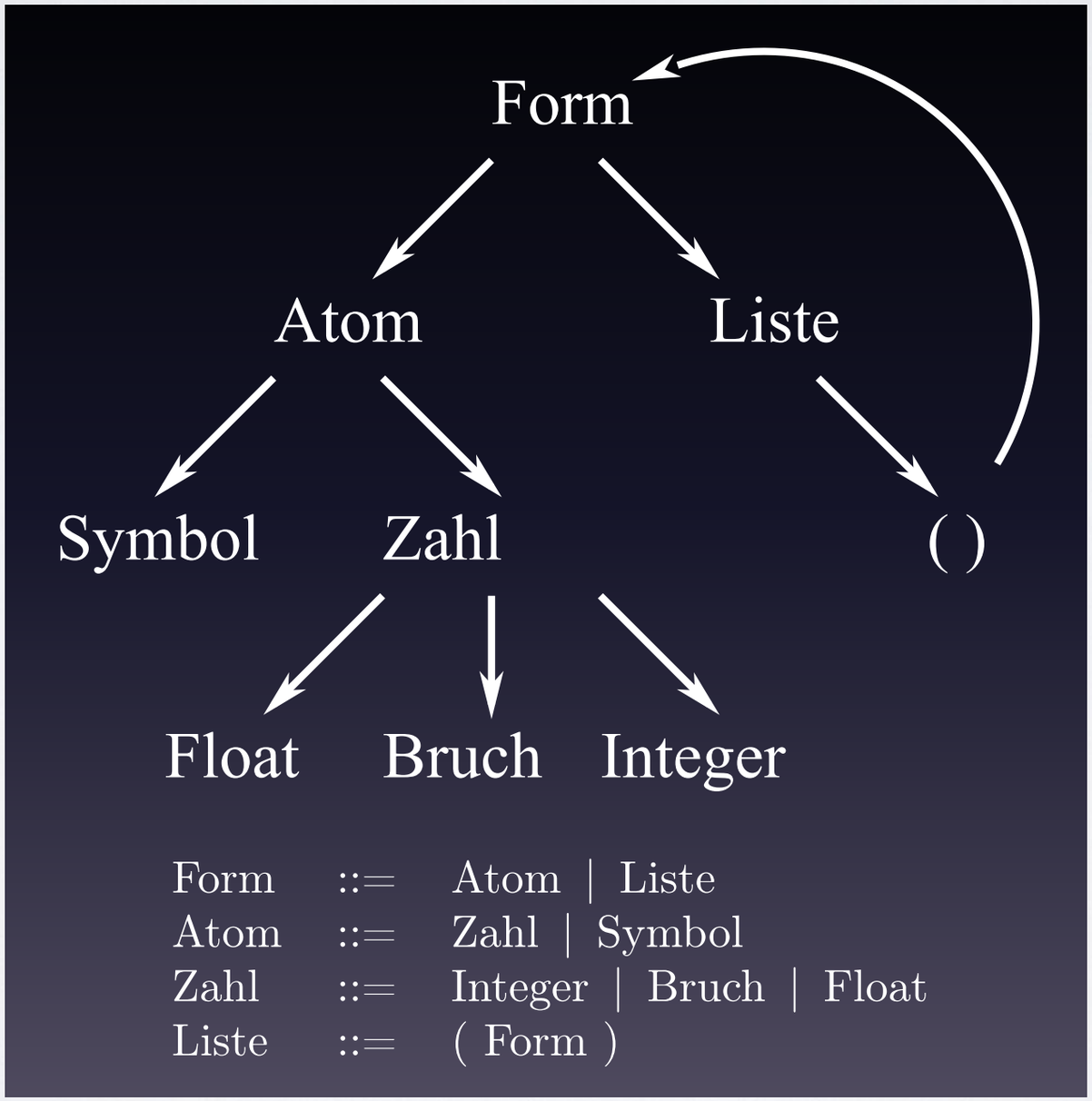
(+ 3 2)



[1] Touretzky, D. (1990). Common LISP: a gentle introduction to symbolic computation. Carnegie Mellon University.

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Rekursive Syntax



übernommen von F. Zalkow / D. Lorrain

NB:

- Symbole evaluieren zu ihrem Variablen-Wert
- Alle anderen Atome sind selbstevaluierend

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Beispiel für S-Expressions

(* 2 (+ 3 4))

form

atom: *

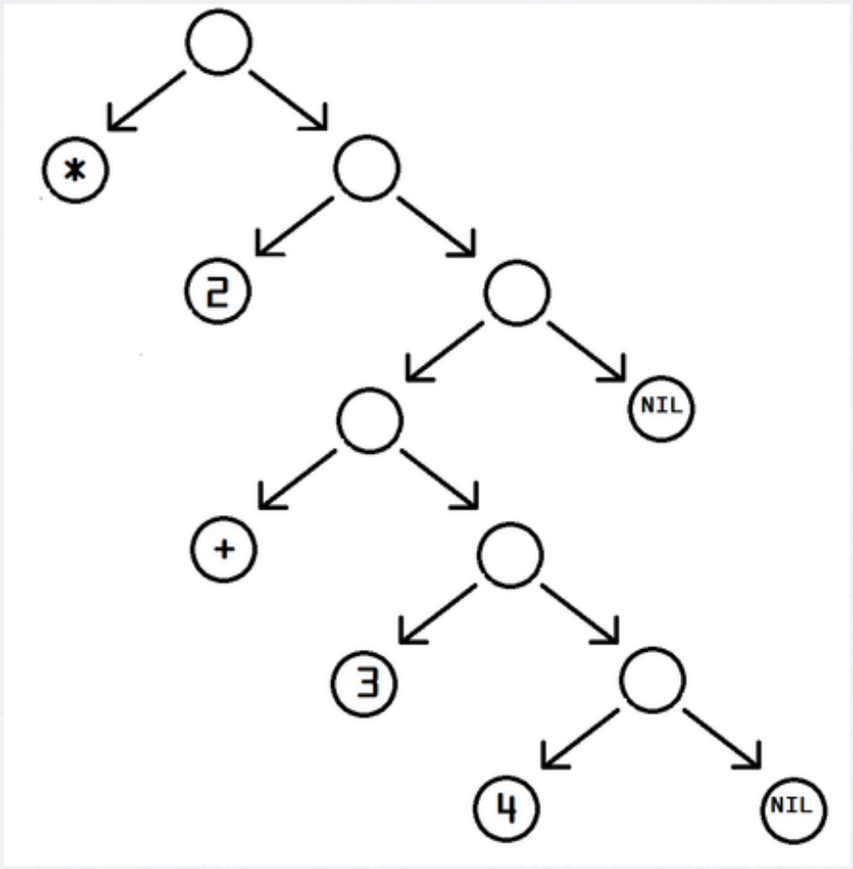
atom: 2

form

atom: +

atom: 3

atom: 4



SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Wie wird eine LISP Form evaluiert?

(+ 33 (* 2.3 4) 9)
(+ 33 (* 2.3 4) 9)
(+ 33 (* 2.3 4) 9)
(+ 33 (* 2.3 4) 9)
(* 2.3 4)
(* 2.3 4)
(* 2.3 4)
(+ 33 9.2 9)
(+ 33 9.2 9)
(+ 42.2 9)

=> 51.2

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Funktionen können für eine fixierte Anzahl von Argumenten definiert werden, oder für eine variable Anzahl von Argumenten. Ein Beispiel für variable Anzahl von Argumenten sind die primitiven arithmetischen Funktionen in Common LISP.

Die Reihenfolge der Evaluierung zu beachten ist wichtig für mathematische Funktionen, die nicht kommutativ sind, z.B. Division

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Übung

Schreiben Sie die lineare Gleichung

$$f(x) = m*x + b$$

als LISP Form.

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Spezielle Funktionen: Prädikate

returns two special symbols
representing boolean values:
t (= true)
nil (= false)

numberp
symbolp
zerop
equalp
…

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Funktionen definieren in LISP

Macro Funktion “defun” (define function)

```
(defun function-name-symbol
            (param1 param2 param3 ...)
    expr1
    expr2
    expr3
    ... )
```

SYMBOLISCHE PROGRAMMIERUNG MIT COMMON LISP I

Übung: Würfelspiel