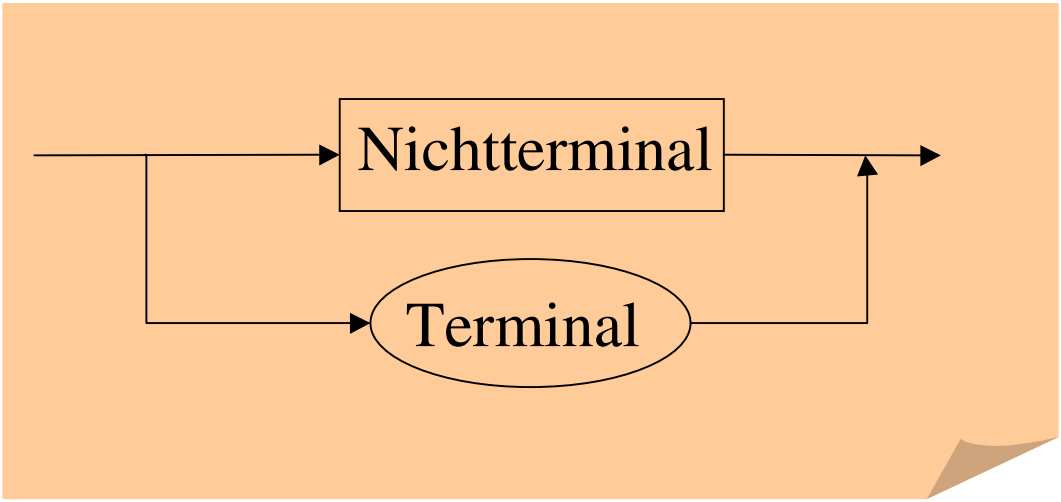
Die Syntax von Prolog

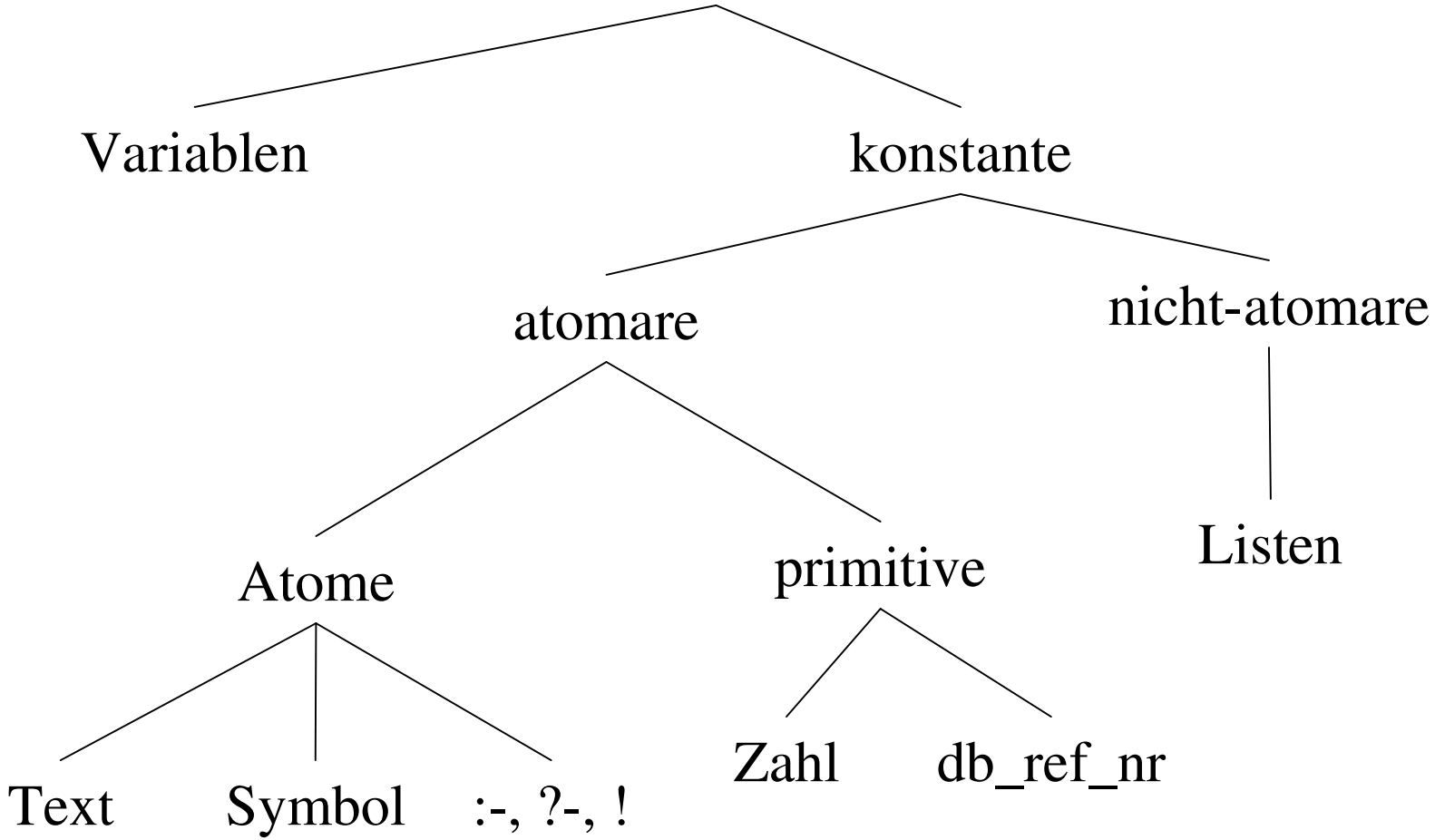


Wie sieht ein Prolog-Programm aus?

Begriffe und Notationen

in der log. Programmierung		**in Prolog**		
Konstante	a, b, c, ...	Atom	a, b, c, ...	Objekte
		Zahl	0, 1, 2, ...	
Term	f(...)	Struktur	$f(t_i...)$	
		Komponente	t_i	
Variable	X, Y, ...	Variable	X, Y, ...	
Klausel	=> A	Fakt	A.	
Klausel	$A_1,...,A_n$=>A	Regel	A :- $A_1,...,A_n$	
Anfrage	$A_1,...,A_n$=>	Anfrage	?- $A_1,...,A_n$	
Programm		Datenbank		

Prolog-Datentypen



atomare Datentypen

Buchstabe	::= a...z | A...Z
Ziffer	::= 0 ... 9
Sonderzeichen	::= !|#|$|%|^|&|*|(|)| _|+|{|}|:|“|<|>|?|-|=|'|[|]|;|,| | |.|/
Zahl	::= <integer>|<real>
integer	::= (+| -)$^{0|1}$ <Ziffer>$^{+}$
real	::= (+| -)$^{0|1}$ <Ziffer>$^{+}$. <Ziffer>$^{+}$(E<integer>)$^{0|1}$
primitive	::= <Zahl>|<db_ref_nr> (*)
Atom	::= <Text>|<Symbol>| :- | ?- | !
Text	::= (a...z)$^{0|1}$(<Buchstabe> | _ | <Ziffer>)* | '(<Buchstabe> | <Ziffer> | <Sonderzeichen>)* '
Symbol	::= <Sonderzeichen>$^{+}$

(*) <db_ref_nr> ist im Zusammenhang mit der Manipulation der internen Datenbank von Interesse

Strukturen

Struktur ::= <funktor> | <funktor>(<argument>(,<argument>)*)

argument ::= <Struktur> | <atomarer Datentyp> | <Variable> |
<operation> | <Liste>

funktor ::= <Text> | <Symbol> | <Operator>

operator ::= + | - | * | / | // | mod | \/ | << | abs | exp | log | ... |
<benutzerdefinierter Operator>

- Arithmetische Operatoren sind üblicherweise als linksassoziativ vereinbart.
- Operatoren mit kleineren Präferenzen bilden Unterstrukturen von Operatoren mit größeren Präferenzen.

Listen

list ::= [ ] | [(<term>(,<term>)*)] % Objektnotation

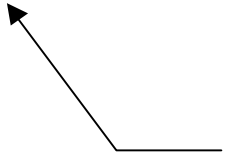
list ::= [ ] | .(<head>,<tail>) % Listenkonstruktor **. /2**
head ::= <term>
tail ::= <list>

list ::= [ ] | [<head>, <tail>] % Listennotation
head ::= <term>
tail ::= <head>, <tail> | <list> |<term>

term ::= (<Regel)|<Faktum>|(<Frage>)|<Variable>|<atomarer Datentyp>|
<Liste>|<Struktur>|<Operation>|(<Programm>)| (<term>(,<term>)*)

Variablen

Variable ::= _ | (A...Z)$^{0|1}$(<Buchstabe> | _ | <Ziffer>)* |
_(<Buchstabe> | _ | <Ziffer>)$^{+}$



anonyme Variable

Bsp. Suchen von Elementen in einer Liste

- Ist das gegebene Objekt gleich dem ersten Element der Liste, dann ist es in dieser enthalten.

  enthaelt([Element| _],Element]).
- Wenn das gegene Objekt im Rest der Liste enthalten ist, dann ist es auch in der gesamten Liste enthalten.

  enthaelt([_|Rest], Element) :- enthaelt(Rest,Element].

Prolog-Programm

Programm ::= (<Fakt>. o | <Regel>.o | <Frage>.o)$^+$

Fakt ::= <Struktur>

Regel ::= <Struktur> :- <regelrumpf>

regelrumpf ::= (<Struktur> | <Variable> | (<regelrumpf>)) (,<regelrumpf>)*

Frage ::= ?- (<Struktur> | <()-struktur>)(,<Struktur> | <()-struktur>)*

()-struktur ::= ((<Struktur> | <()-struktur>)(,<Struktur> | <()-struktur>)*)

Zur „Gleichheit von Termen“

$5 \neq 2 + 3$

- Zahlen und Atome sind nur zu sich selbst gleich.
  Bsp. 1903 = 1903 fritz ≠ hugo
- Eine Variable ist gleich einer anderen, wenn die Namen gleich sind.
  Bsp. _13 = _13
- Zwei Strukturen sind gleich, wenn sie den gleichen Funktor und die gleiche Anzahl von Komponenten haben und wenn die entsprechenden Komponenten paarweise gleich sind.
  Bsp. weiblich(maria) = weiblich(maria)
  istMutterVon(elfi,maria) ≠ istMutterVon(elfi,hugo)
  +(2,3) = 2+3
  verheiratet(M,anton) = verheiratet(maria,anton) % sub={M/maria}