Rekursion! Rekursion! Rekursion!

1. Rekursionsgleichungen:
   $x^0 = 1$
   $x^n = x \cdot x^{n-1}$ für $n > 0$

2. Verbinden mit Hilfe eines Konditionals:
   $x^n =$ if $n > 0$ then $x \cdot x^{n-1}$ else 1

3. Codierung in ML:

```
fun potenz (x:int, n:int) :  int =
          if n>0 then x*potenz(x,n-1) else 1
```

Selbstanwendung

Wiederholung: Endrekursion

Eine Prozedur, die das Ergebnis
unmittelbar durch eine rekursive Anwendung liefert
ist eine **endrekursive Prozedur**.

```
fun p(a:int,x:int,n:int):int =
        if n=0 then a else p(a * x, x, n-1)

fun power(x:int,n:int):int=p(1,x,n)
```

a: "Akku" (Akkumulator, "akkumuliert" das Ergebnis)

Verarbeitungsphasen eines Interpreters

1. Lexikalische Analyse:
   kann die eingegebene Zeichenfolge
   als Sequenz von Wörtern aufgefasst werden?
2. Syntaktische Analyse (Parsing):
   beschreibt die Wortfolge ein Programm?

} Syntax

3. Semantische Analyse (Elaboration):
   ist das Programm zulässig?
4. Ausführung (Auswertung, Evaluation):
   führe das Programm aus.

} Semantik

Darstellung und Aufbau von Phrasen

Zeichendarstellung:

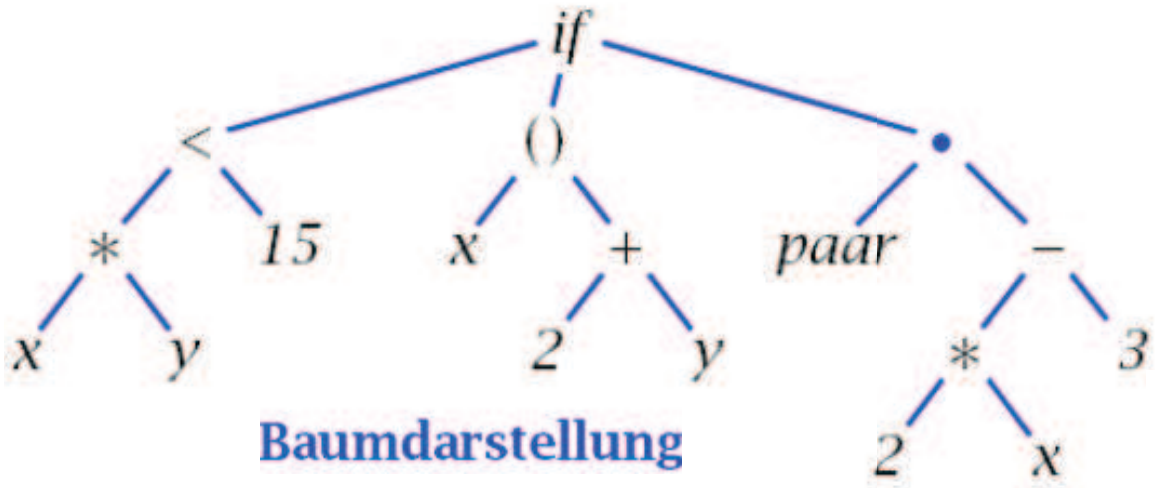
```
if x*y<15then(x,2+y)else paar(2*x-3)
```

Darstellung als Buchstabenfolge

Wortdarstellung:

```
if x * y < 15 then ( x , 2 + y ) else paar ( 2 * x - 3 )
```

Darstellung als Wortfolge



Baumdarstellung

Darstellung der hierarchischen Struktur

Freie und gebundene Bezeichner

- `val p = if x>0 then 3 else 4` (x tritt frei auf)
- `fun p (x:int) = if x>0 then 3 else 4` (x gebunden)
- Phrasen ohne freie Bezeichner: geschlossen
- Phrasen mit freien Bezeichnern: offen

- Ausführbar?

```
fun h(x:int):int = h x
fun p(x:int) = x
val x = x
fun q (x:int) = 3 + (p x)
```