$\{0, 1\}^7$ set of binary strings of length 7.

$(-7 * 8 \wedge 2)$ $(-7 * 8 \wedge 2))$

$-7 \cdot 8^2$ $(-7 - * 8 \wedge 2)$

$((())())$

$B = \{0, 1\}$ B^{17} binary strings of length 17.

B^* the set of all finite length binary strings.

λ = empty string.

length of λ is 0 ""

$B^* = B^0 \cup B^1 \cup B^2 \cup B^3 \cup \cdots$

$B^0 = \{\lambda\}$

Recursive def of B^*

1) Base case: $\lambda \in B^*$
2) If $x \in B^*$
then $x0 \in B^*$ $x1 \in B^*$

concat

101.

$\lambda \longrightarrow \lambda 1 = 1 \longrightarrow 10 \longrightarrow 101$

$\in B^*$ $\quad \in B^*$ $\quad \in B^*$.

Length of string x $\quad |x|$ # chars.

$|1101| = 4$.

Base case

Length of a string in B^* $\quad$ 1) $|\lambda| = 0$.

2) $x \in B^*$ $\quad |x0| = |x| + 1$

(Recursive rule.

$|x1| = |x| + 1$.

$\lambda \to 1 \to 10 \to 101$

0 $\quad$ 1 $\quad$ 2 $\quad$ 3.

(()()) (()())))()(

Set of properly nested strings of ( )

Base: ()

Recursive rule : A) If x is properly nested
then (x) prop nested.

B) If x + y are properly nested
then. xy is properly nested.

$$((())())$$

→ 1) $()$ base case.
→ 2) $(())$ RR A.
3) $(())()$ RR B.
4) $((())())$

$$n! = n(n-1)(n-2)\cdots 2\cdot 1$$

Base case : $0! = 1$.

Inductive Rule : $n! = n\cdot(n-1)!$

Re

$$1! = 1\cdot 0! = 1\cdot 1 = 1.$$

$n=1$

Recursively defined sequences: initial conditions (base case)
recurrence relation (recursive rule)

Fibonacci: $\left.\begin{array}{l} f_0 = 1. \\ f_1 = 1 \end{array}\right\}$ initial conditions.

For $n \geq 2$ $\quad f_n = \underline{\underline{f_{n-1} + f_{n-2}}}.$

$1, \underline{1}, \underline{2}, \underline{3}, 5, 8, \ldots$

$g_0 = 2 \qquad g_n = 3 \cdot g_{n-1} + n. \Leftarrow n=1.$

$g_0 = 2$

$g_{\underline{1}} = 3 \cdot g_{\underline{0}} + \underline{1} = 3 \cdot 2 + 1 = 7.$

$g_2 = 3 \cdot g_1 + 2 = 3 \cdot 7 + 2 = 23$

$t_n = 3 \cdot t_{\underline{n-1}} + 2 \cdot t_{\underline{n-3}}$

$t_1 = 1 \qquad t_4 = 3 \cdot t_3 + 2 \cdot t_1$

$t_2 = 2 \qquad = 3 \cdot 1 + 2 \cdot 1 = 5.$

$t_3 = 1$

$t_n =$

Factorial (n)   non-neg int. $0! = 1$.

If (n == 0) Return (1).

Return(n · Factorial (n-1))

End

Power(a, n) // a^n a real #
n non-neg integer.

If (n=0) Return 1.

Return (a · Power(a , n-1)) a^{n-1}

End;

$a^{18} = (a^9)^2$

Power (a, n)

$\left(a^{n/2}\right)^2 =$

If (n=0) return 1.

If n is even

Return (Power(a, n/2)^2);

$a^{n/2 \cdot 2} = a^n$

If n is odd

Return(a · (Power(a, ⌊n/2⌋))^2);

$a^{2k+1} = a^{2k} \cdot a$

End;

a^{19} $\lfloor 19/2 \rfloor =$
9.