LECT. 15  09/29/99

RECURSIVE ALGORITHMS

BY THE EUCLIDEAN ALGORITHM
\[
gcd(a, b) = gcd(b, r_a)
\]
CIRCULAR PROEDURE!

THIS GUARANTEES THE CONVERGENCE:
\[
\min(a, b) \leq \min(b, r_a)
\]

AN ALGORITHM IS CALLED RECURSIVE IF IT WORKS BY REDUCING TO ITS OWN VALUE ON SMALLER INPUTS (USUALLY PERFORMED BACKWARD)

EXAMPLE:
\[
gcd(a, b) = gcd(b, a \mod b)
\]
\[
gcd(a, b) = d \text{ when } a > b
\]

SPECIFIES A SEQUENCE OF REDUCTIONS THAT ENDS BY
\[
gcd(a, d) = d
\]

- 98 -

ITERATIVE PROCEDURE:

INPUT a
\[
P_n(a) = P(P_{n-1}(a))
\]
USUALLY PERFORMED STRAIGHT
a, P(a), P^2(a), ..., P^n(a)

RECURSIVE

PROCEDURE factorial(n)
IF n = 1 THEN
factorial(1) = 1
ELSE
factorial(n) = n \times factorial(n-1)
END

PROCEDURE fibonacci(n)
IF n = 0 THEN fibonacci(0) = 0
ELSE IF n = 1 THEN fibonacci(1) = 1
ELSE fibonacci(n) = fibonacci(n-1) + fibonacci(n-2)
END

RECURSIVE COMPUTING OF a^n
\[
\begin{align*}
a^0 &= 1 \\
a^n &= a \cdot a^{n-1}
\end{align*}
\]
THE SAME PROCEDURE, SMALLER INPUT

PROCEDURE power(a, n : INTEGER)
IF n = 0 THEN power(a, 0) = 1
ELSE power(a, n) = a \times power(a, n-1)
END

RECURSIVE SEQUENTIAL SEARCH ALGORITHM: FIND x IN a_1, a_2, ..., a_n
SEARCHES FOR x IN a_1, a_2, ..., a_n
IF a_i = x THEN LOCATION = i
ELSE IF i = n THEN LOCATION = 0
ELSE SEARCH(a_{i+1}, a_n)
END

TH: IF f(x) IS DEFINED RECURSIVELY
\[
f(x) = g(x)
\]
AND g(x) IS COMPUTABLE, THEN f(x) IS COMPUTABLE.
PROOF: COMPUTE f(x), f(x), f(x), ..., f(x)

RECURSIVE FIBONACCI COMPUTATION

PROBLEM: FIND f_n

EXERCISE!

ITERATIVE FIBONACCI COMPUTATION
\[
f_0, f_1, f_2, ..., f_{n-1}, f_n
\]
1 ADDITIONS / LINEAR

HW 3.4: 4, 14