Session 51: Recursion, Induction and Iteration

- Recursion and induction
- Recursion and iteration

Recursion and Induction

Induction and recursion are different approaches to proving results and solving problems

- They have in common that they both in the first place rely on the ability to achieve the desired result for the smallest possible version of the problem at hand
- Induction **extends** this ability to problems of any size
- Recursion **reduces** a problem of any size to the smallest possible ones

Induction can be used to prove correctness of recursive algorithms

Proving Recursive Algorithms Correct

Prove that the algorithm for computing the powers of real numbers is correct

power(a, n) :=

if n <= 0 **then return** 1 **else return** $a \cdot power(a, n-1)$

Proof: (assume $a \neq 0$)

Basis Step: $power(a,0) = 1$, algorithm is correct, since $a^0 = 1$

Inductive Step: inductive hypothesis: $power(a,k) = a^k$, $k > 0$

$$power(a,k+1) = a \cdot power(a,k) = a \cdot a^k = a^{k+1}$$

($k+1 > 0$ ↑; IH ↑)

algorithm is correct ∎

Recursion and Iteration

A recursively defined function can be evaluated in two different ways

- **Recursively**: for a value apply directly the recursive definition, till a base case is reached, a recursive algorithm
- **Iteratively**: start with base cases, and apply the recursive definition to compute the function for larger values

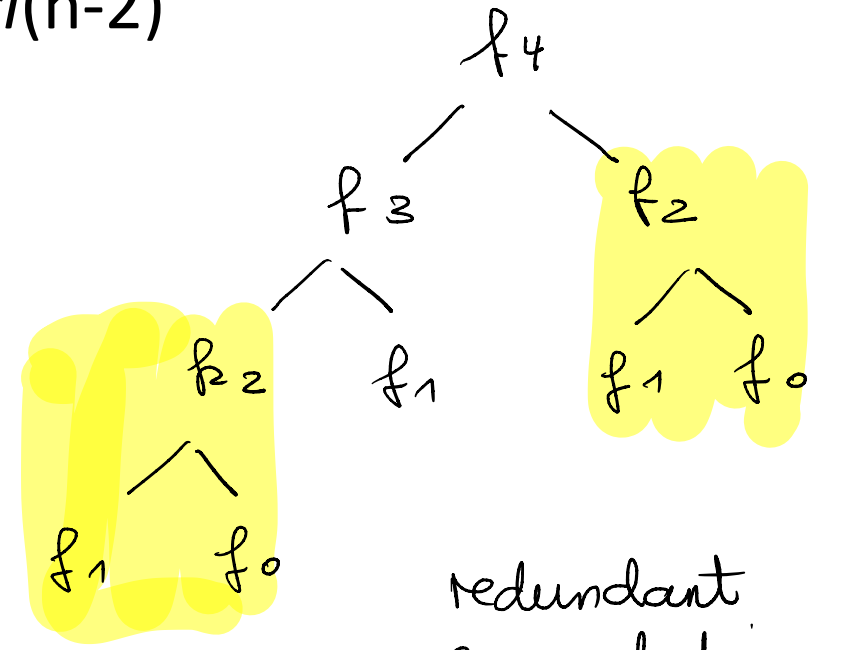
Fibonacci Function

fibonacci(n) :=

if n ≤ 1 **then return** n

else return *fibonacci*(n-1) + *fibonacci*(n-2)

Example : computing fibonacci (4)



redundant computations → inefficient

Iterative Algorithm

```
iterative_fibonacci(n) :=
    if n = 0 then return 0
    else
        previous := 0
        current := 1
        for i = 1 to n - 1
                next := previous  + current
                previous := current
                current := next
        return current
```

iterative-fibonacci (4) :

remember the last 2 fib. numbers

i	next	previous	current
–	–	0	1
1	1	1	1
2	2	1	2
3	3	2	3

result

Complexity is $\Theta(n)$

Summary

- Proving correctness of recursive algorithms using induction
- Iterative algorithm for Fibonacci numbers