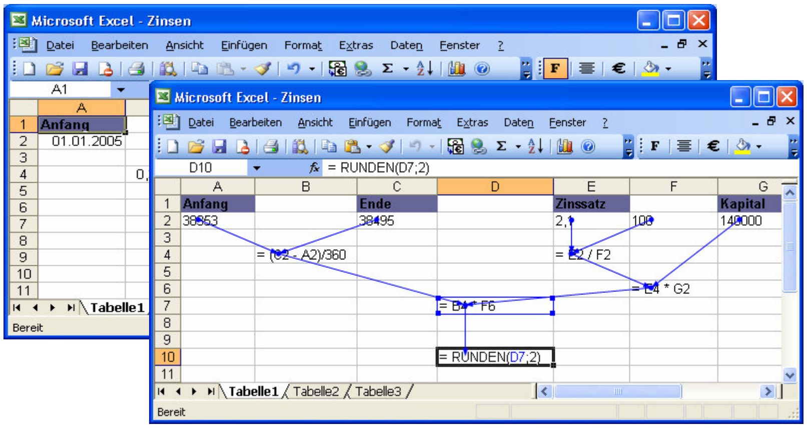
# A strategy to introduce functional modeling at school informatics

**Markus Schneider**
**Institut für Informatik**
**Technische Universität München**
**markus.schneider@in.tum.de**

## Overview

- **Why to use functional modeling ?**
- **The concept of functional modeling ?**
  - *An introductive example*
  - *Graphic Elements*
  - *Transformation to Spreadsheet Applications*
- **The concept of function**
  - *Didactical differences to mathematics*
- **An algorithm to transform graphical models to terms used in spreadsheets**
- **Standard expressions: Boolean and conditional expressions**
- **Iterative structures**

## Why to use functional modeling ?



Microsoft Excel - Zinsen

A1

|   | A |
|---|---|
| 1 | Anfang |
| 2 | 01.01.2005 |
| 4 | 0, |

Microsoft Excel - Zinsen

D10 = RUNDEN(D7;2)

|    | A | B | C | D | E | F | G |
|----|---|---|---|---|---|---|---|
| 1 | Anfang | | Ende | | Zinssatz | | Kapital |
| 2 | 38353 | | 38495 | | 2,1 | 100 | 140000 |
| 3 | | | | | | | |
| 4 | | = (C2 - A2)/360 | | | = E2 / F2 | | |
| 5 | | | | | | | |
| 6 | | | | | | = E4 * G2 | |
| 7 | | | | = B4 * F6 | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | = RUNDEN(D7;2) | | | |
| 11 | | | | | | | |

## Why to use functional modeling

**Spreadsheet applications are often the entry theme to Informatics (for example in Austria, Bavaria, …)**

**Functional modeling is the theoretical basis for spreadsheet calculations**

**The use of spreadsheets without a theoretical concept**

- *blocks abstraction from specific products (Excel, StarOffice, …),*
- *blocks the insight into the mathematical background of spreadsheet calculations,*
- *blocks the development of a structured strategy for problem solving; i.e. first, a model of the solution ought to be designed and then the implementation can be made*

# The concept of functional modeling: An introductive example

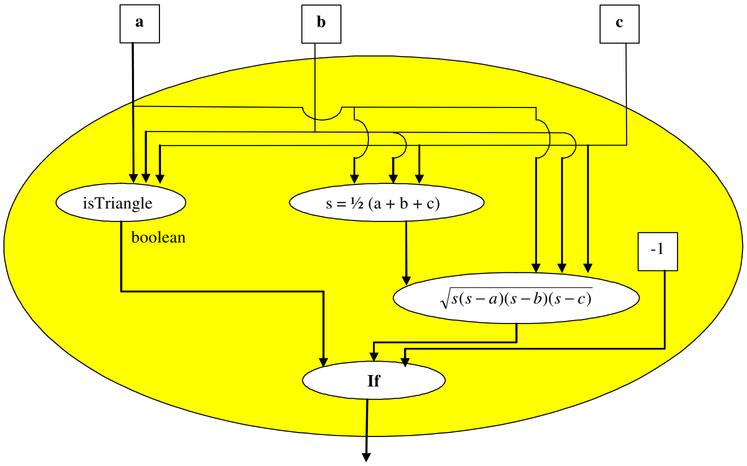
**Heron’s formula to calculate the area of a triangle:**

- *Given the length of the three edges of an triangle: a, b, c,*
  - If a + b > c, a + c > b, b + c > a,
  - and a > 0, b > 0, c > 0,
- *then the area of the triangle is given by:*

$$\sqrt{s(s-a)(s-b)(s-c)}$$

- *Where s is calculated by:* $\dfrac{a \;+\; b \;+\; c}{2}$

# An introductive example: Heron’s formula

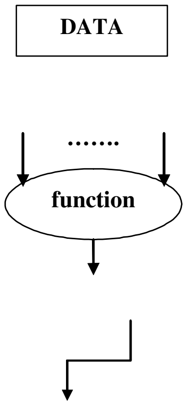


## Functional modeling: Graphical Elements

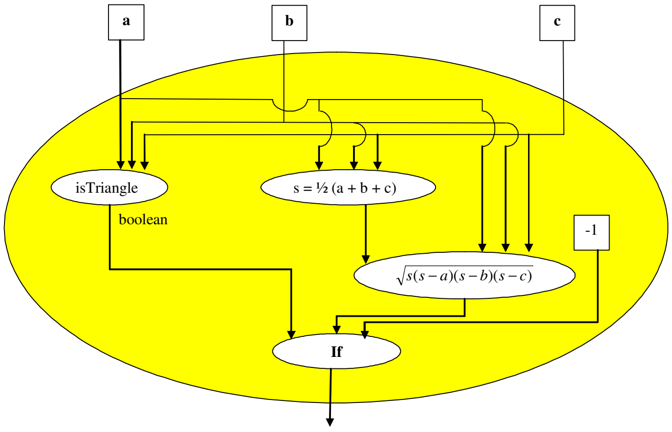
Data

Functions

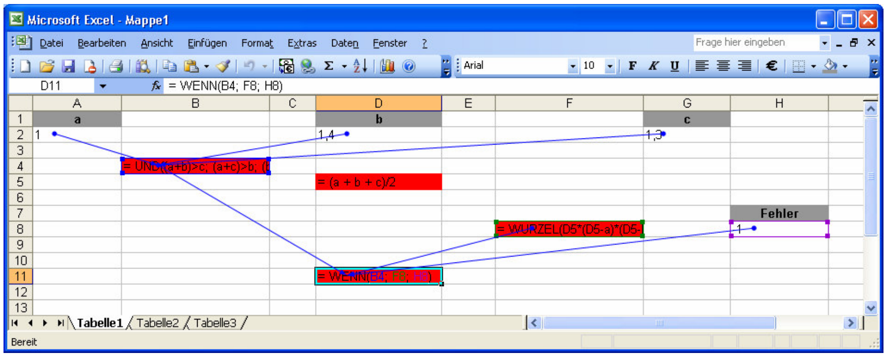
References



## The concept of functional modeling: Heron’s formula



## The concept of functional modeling: Heron’s formula



D11 = WENN(B4; F8; H8)

|  | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | a |  |  | b |  |  | c |  |
| 2 | 1 |  |  | 1,4 |  |  | 1,3 |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  | = UND((a+b)>c; (a+c)>b; ( |  |  |  |  |  |  |
| 5 |  |  |  | = (a + b + c)/2 |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  | Fehler |
| 8 |  |  |  |  |  | = WURZEL(D5*(D5-a)*(D5- |  | 1 |
| 9 |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |
| 11 |  |  |  | = WENN(B4; F8; H8) |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |

## The concept of function

**The concept of function in didactic of mathematics:**

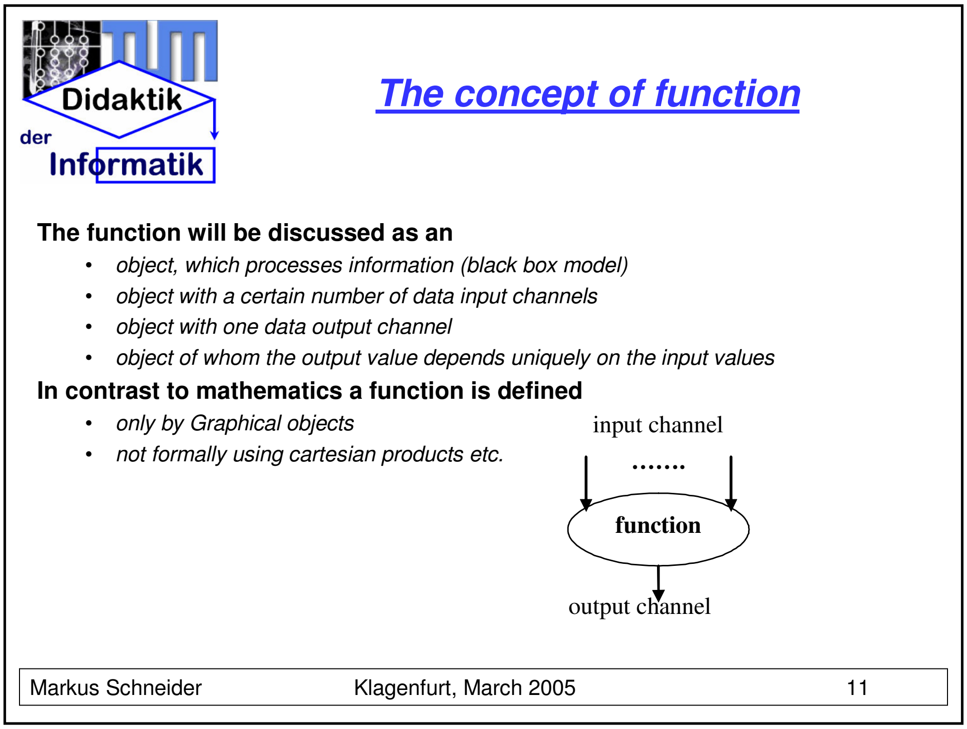
- *Functions of arity 1*
- *Functions over (real, rational, integer, … ) numbers*

**The requirements of functional modeling:**

- *Functions over general data–types (strings, boolean-values, currencies, …)*
- *Functions having arbitrary arity*

**The problem:**

- *The concept of function is not yet discussed in mathematics*
- *A rigorous mathematical introduction of the function (using cartesian product) is not possible (the pupils are about 14 years old!)*



Didaktik der Informatik

## The concept of function

**The function will be discussed as an**

- *object, which processes information (black box model)*
- *object with a certain number of data input channels*
- *object with one data output channel*
- *object of whom the output value depends uniquely on the input values*

**In contrast to mathematics a function is defined**

- *only by Graphical objects*
- *not formally using cartesian products etc.*

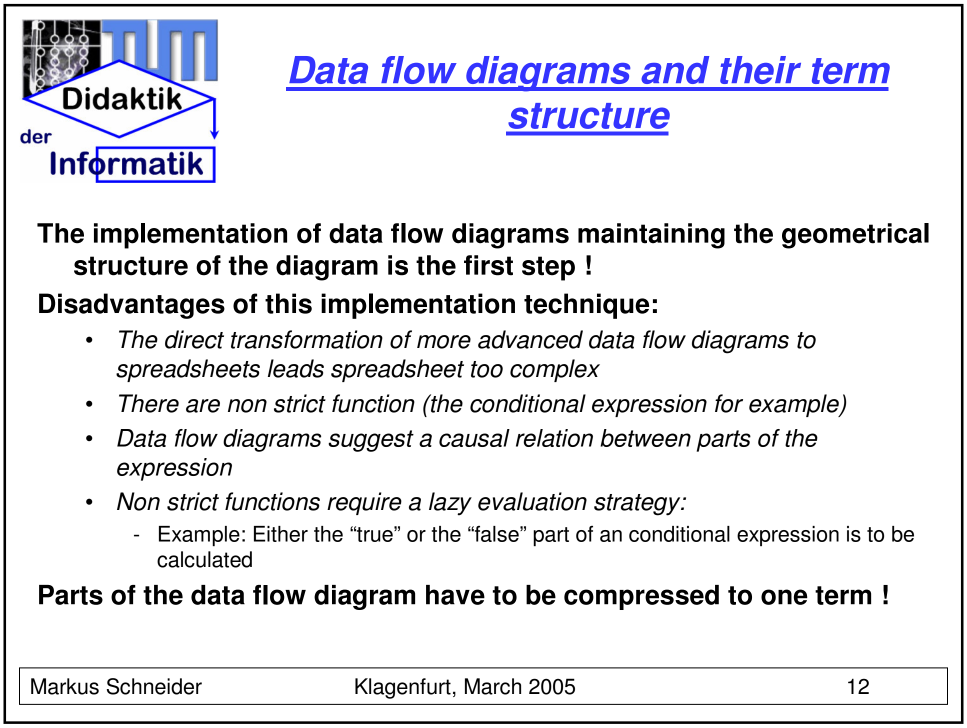
input channel

.......

**function**

output channel

Markus Schneider Klagenfurt, March 2005 11



Didaktik der Informatik

## Data flow diagrams and their term structure

**The implementation of data flow diagrams maintaining the geometrical structure of the diagram is the first step !**

**Disadvantages of this implementation technique:**

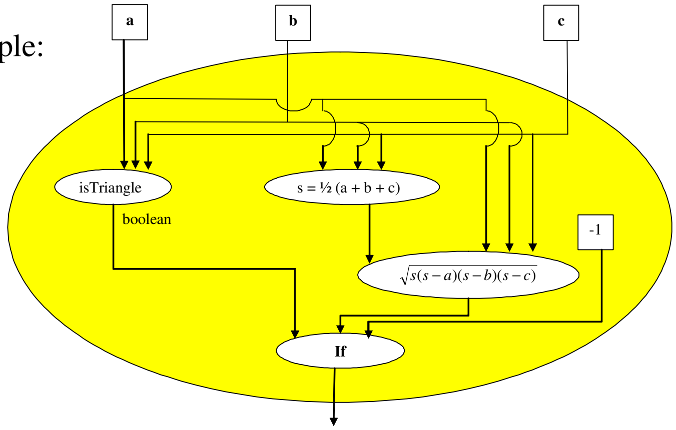
- *The direct transformation of more advanced data flow diagrams to spreadsheets leads spreadsheet too complex*
- *There are non strict function (the conditional expression for example)*
- *Data flow diagrams suggest a causal relation between parts of the expression*
- *Non strict functions require a lazy evaluation strategy:*
  - Example: Either the “true” or the “false” part of an conditional expression is to be calculated

**Parts of the data flow diagram have to be compressed to one term !**

Markus Schneider Klagenfurt, March 2005 12

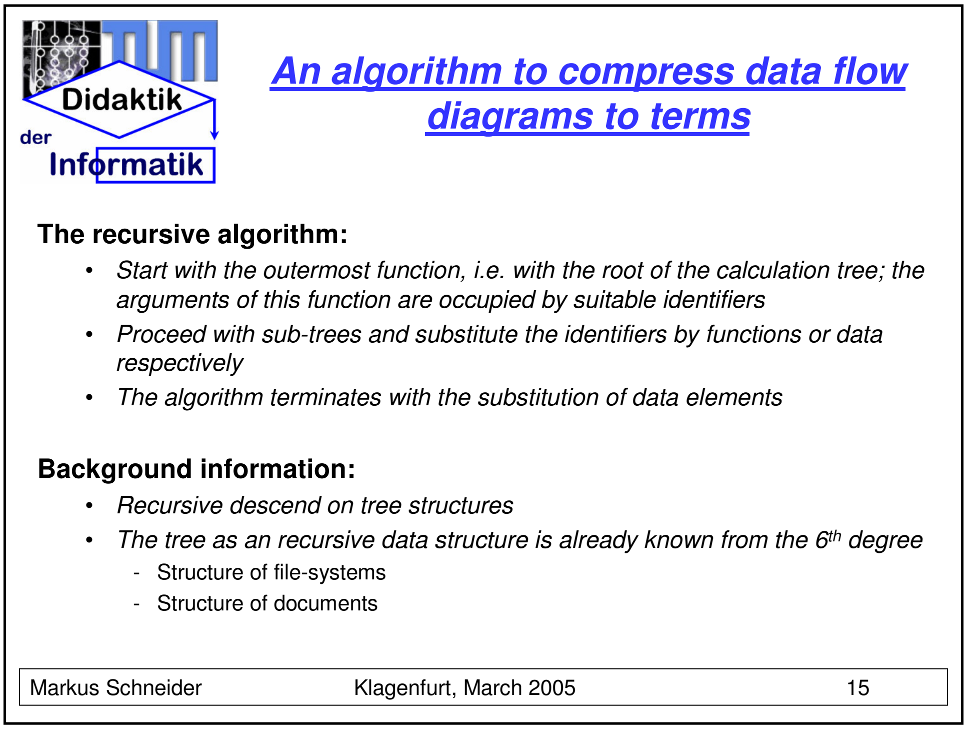
## An algorithm to compress data flow diagrams to terms

Example:



## An algorithm to compress data flow diagrams to terms

```
If( ?; ?; ?)
If(isTriangle(?; ?; ?); SQRT(?); -1)
If(isTriangle(a; b; c); SQRT(?*(?-a)*(?-b)*(?-c)); -1)
If(isTriangle(a; b; c); SQRT((0.5*(a+b+c))
                              *((0.5*(a+b+c))-a)
                              *((0.5*(a+b+c))-b)
                              *((0.5*(a+b+c))-c)); -1)
If(And(a>0; b>0; c>0; a<b+c; b<a+c; c<a+b);
                         SQRT((0.5*(a+b+c))
                              *((0.5*(a+b+c))-a)
                              *((0.5*(a+b+c))-b)
                              *((0.5*(a+b+c))-c)); -1)
```



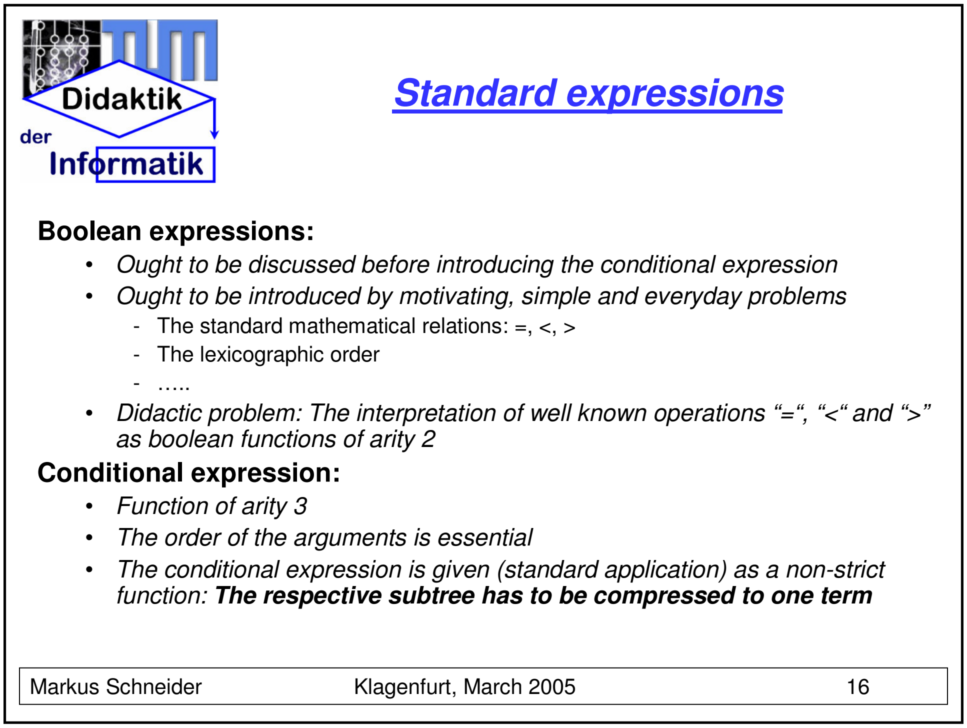
## An algorithm to compress data flow diagrams to terms

**The recursive algorithm:**

- *Start with the outermost function, i.e. with the root of the calculation tree; the arguments of this function are occupied by suitable identifiers*
- *Proceed with sub-trees and substitute the identifiers by functions or data respectively*
- *The algorithm terminates with the substitution of data elements*

**Background information:**

- *Recursive descend on tree structures*
- *The tree as an recursive data structure is already known from the 6th degree*
  - Structure of file-systems
  - Structure of documents



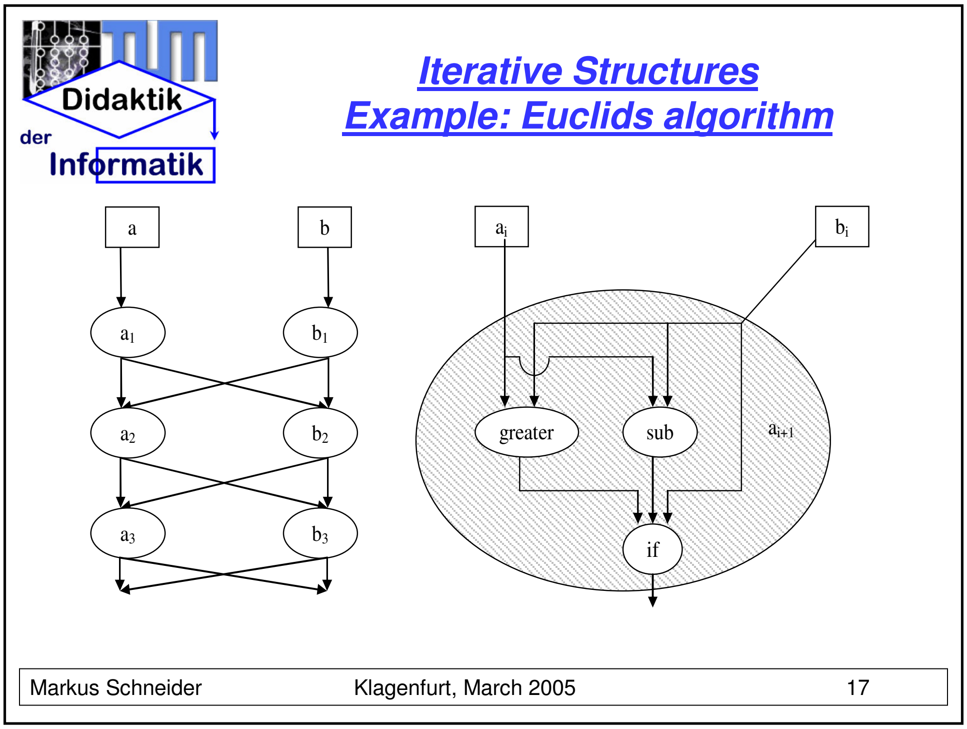
## Standard expressions

**Boolean expressions:**

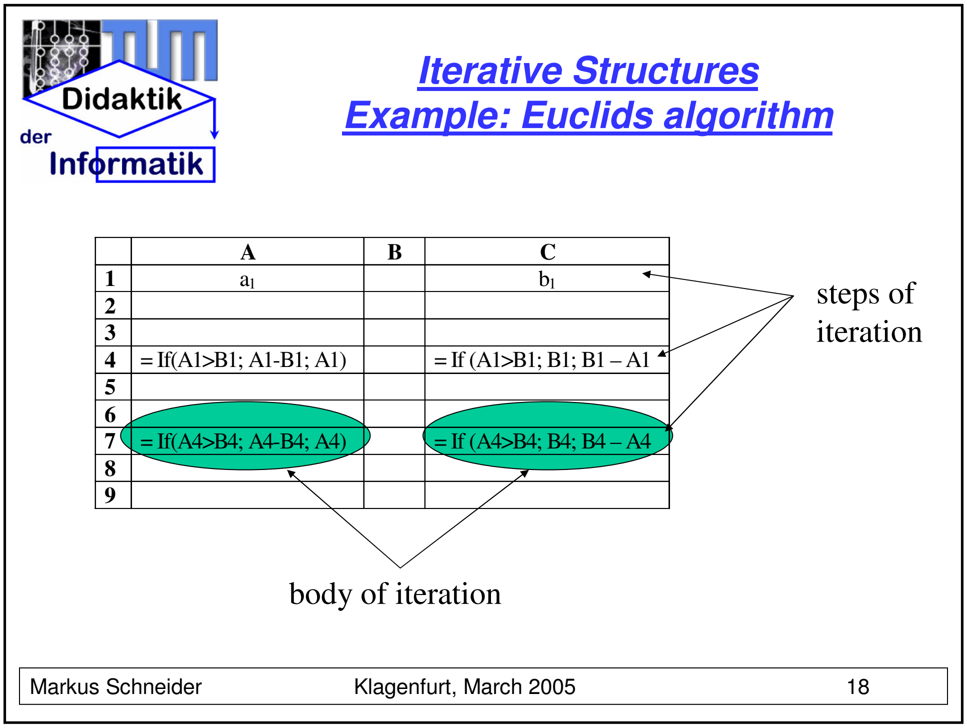
- *Ought to be discussed before introducing the conditional expression*
- *Ought to be introduced by motivating, simple and everyday problems*
  - The standard mathematical relations: =, <, >
  - The lexicographic order
  - …..
- *Didactic problem: The interpretation of well known operations "=", "<" and ">" as boolean functions of arity 2*

**Conditional expression:**

- *Function of arity 3*
- *The order of the arguments is essential*
- *The conditional expression is given (standard application) as a non-strict function:* ***The respective subtree has to be compressed to one term***



## Iterative Structures
## Example: Euclids algorithm



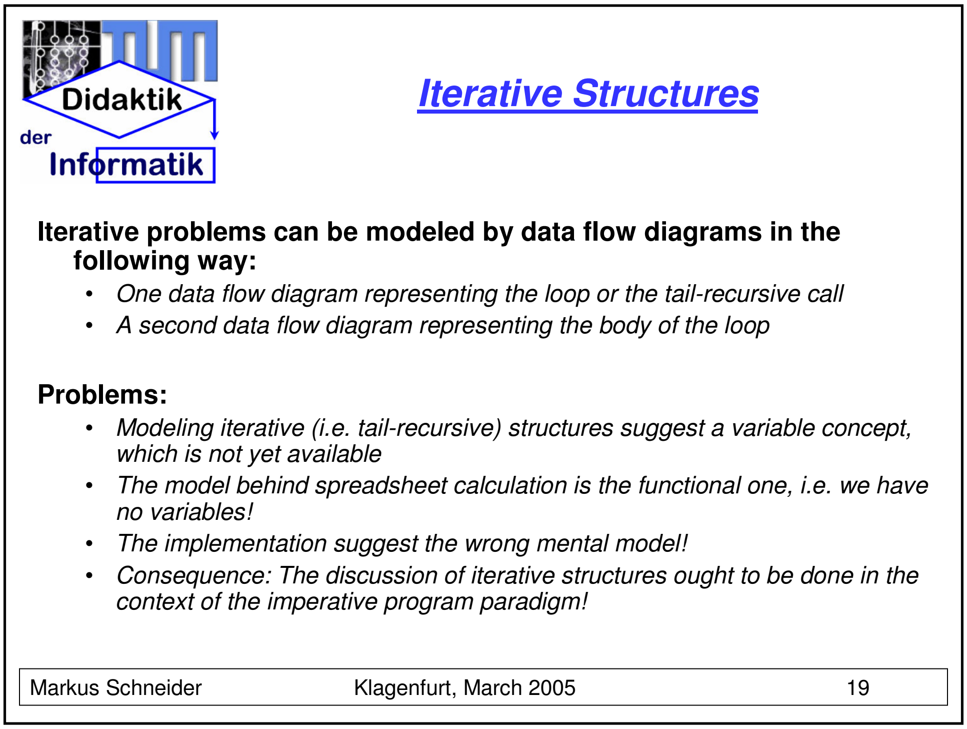
Didaktik
der
Informatik

## Iterative Structures
## Example: Euclids algorithm

| | A | B | C |
|---|---|---|---|
| 1 | $a_1$ | | $b_1$ |
| 2 | | | |
| 3 | | | |
| 4 | = If(A1>B1; A1-B1; A1) | | = If (A1>B1; B1; B1 – A1 |
| 5 | | | |
| 6 | | | |
| 7 | = If(A4>B4; A4-B4; A4) | | = If (A4>B4; B4; B4 – A4 |
| 8 | | | |
| 9 | | | |

steps of iteration

body of iteration



# Iterative Structures

**Iterative problems can be modeled by data flow diagrams in the following way:**

- *One data flow diagram representing the loop or the tail-recursive call*
- *A second data flow diagram representing the body of the loop*

**Problems:**

- *Modeling iterative (i.e. tail-recursive) structures suggest a variable concept, which is not yet available*
- *The model behind spreadsheet calculation is the functional one, i.e. we have no variables!*
- *The implementation suggest the wrong mental model!*
- *Consequence: The discussion of iterative structures ought to be done in the context of the imperative program paradigm!*