Algorithms

Introduction

Ming-Hsien Tsai

Academia Sinica
National Taiwan University

Based on: U. Manber. Introduction to Algorithms - A Creative Approach. Addison-Wesley, 1989.
Also based on slides made by Professor Yih-Kuen Tsay.

Algorithms

- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end

Algorithms

- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end

Baby is crying

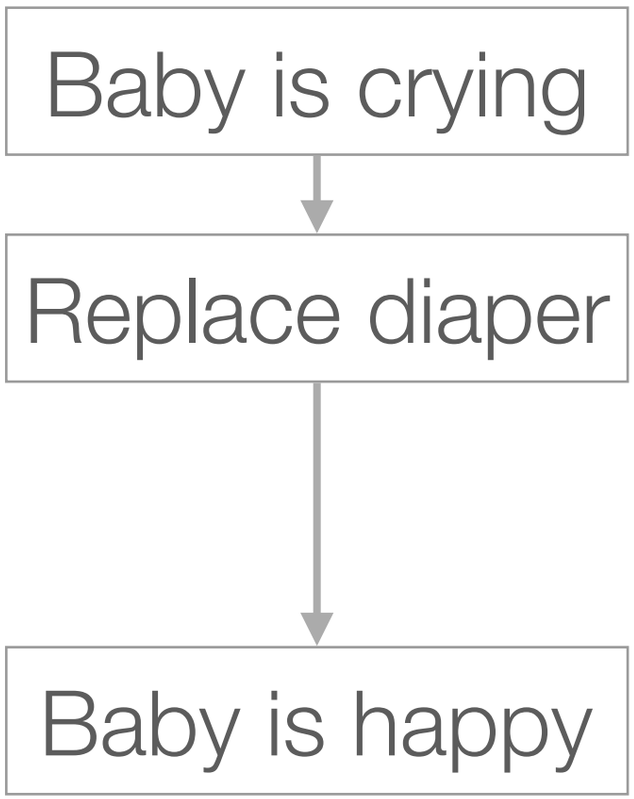
BabySoothing

Algorithms

- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end

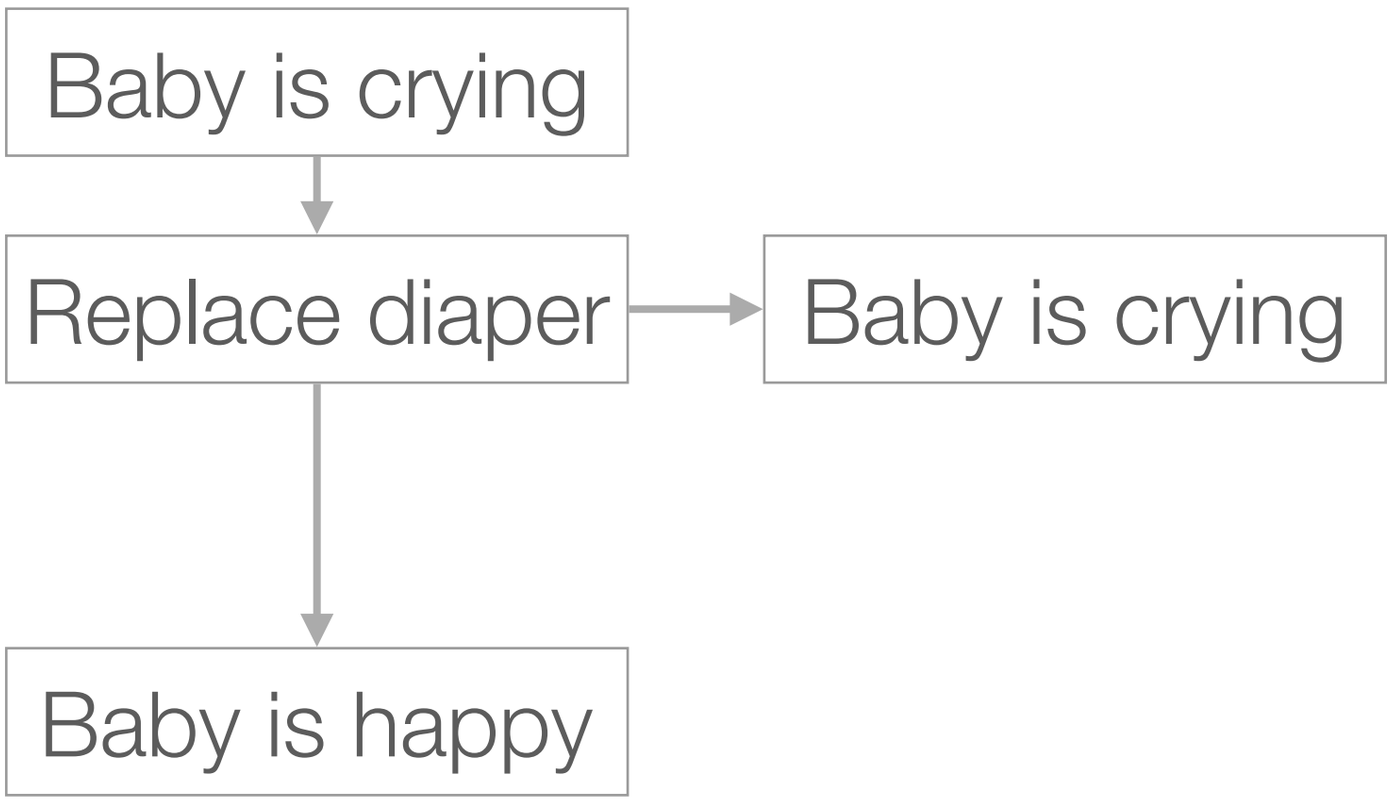
Algorithms

- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end



Algorithms

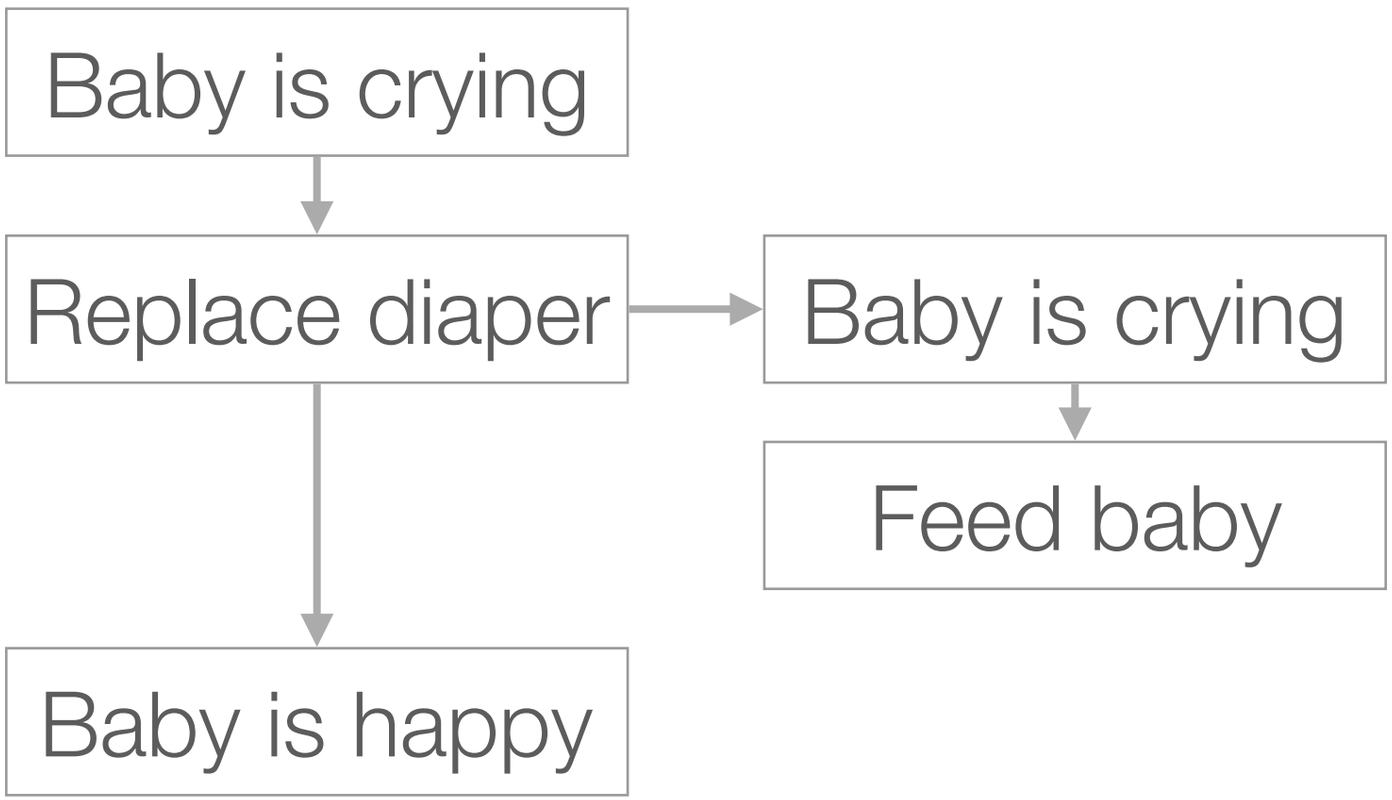
- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end



BabySoothing

Algorithms

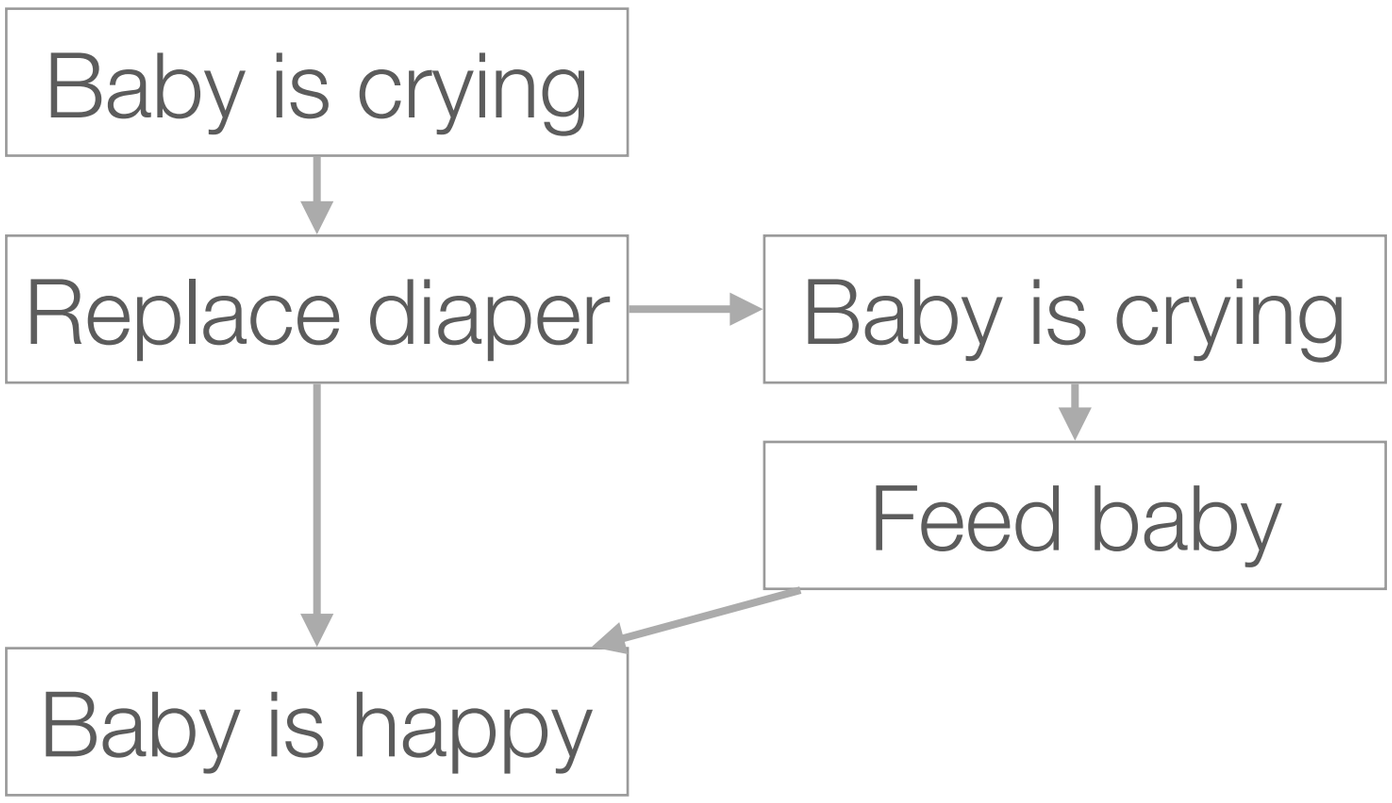
- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end



BabySoothing

Algorithms

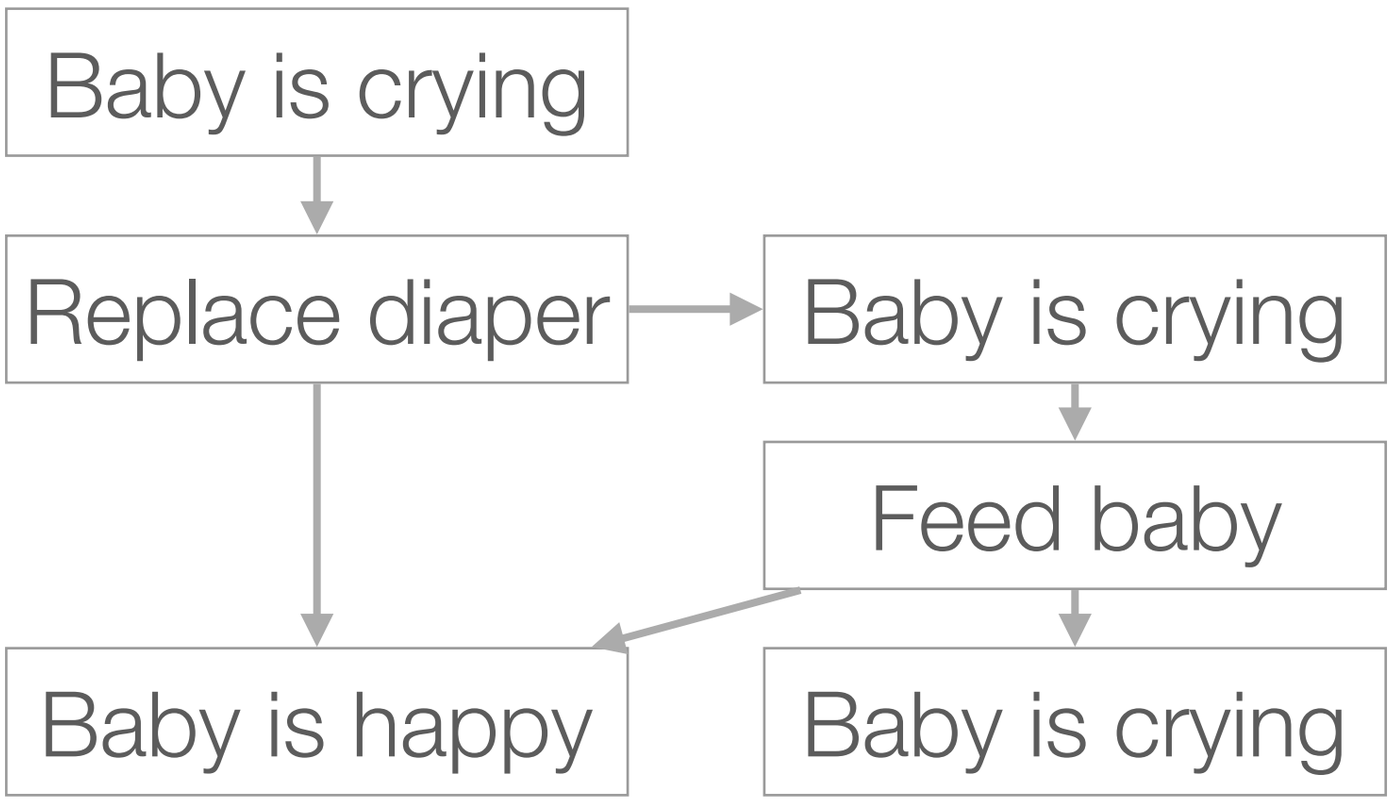
- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end



BabySoothing

Algorithms

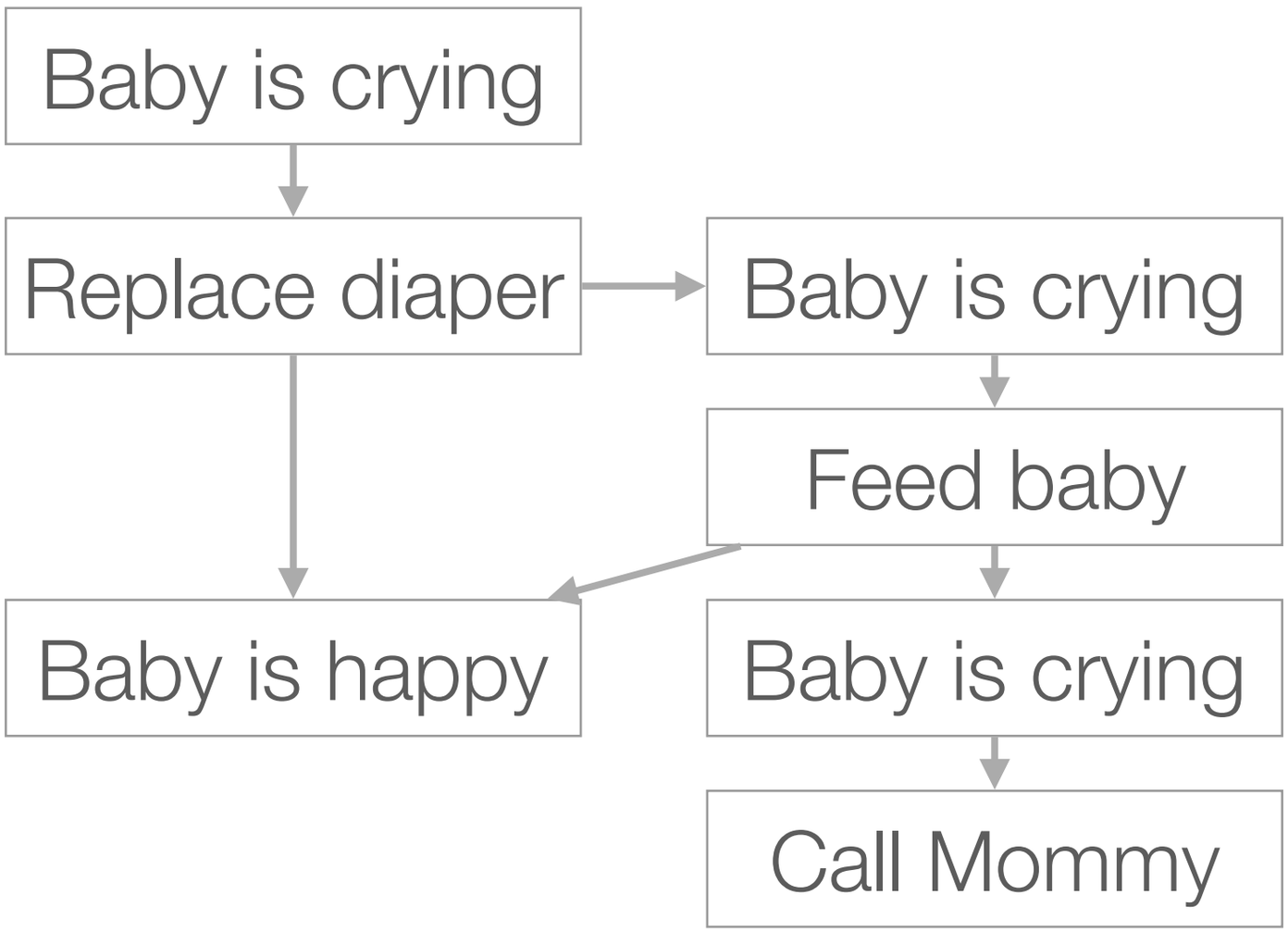
- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end



BabySoothing

Algorithms

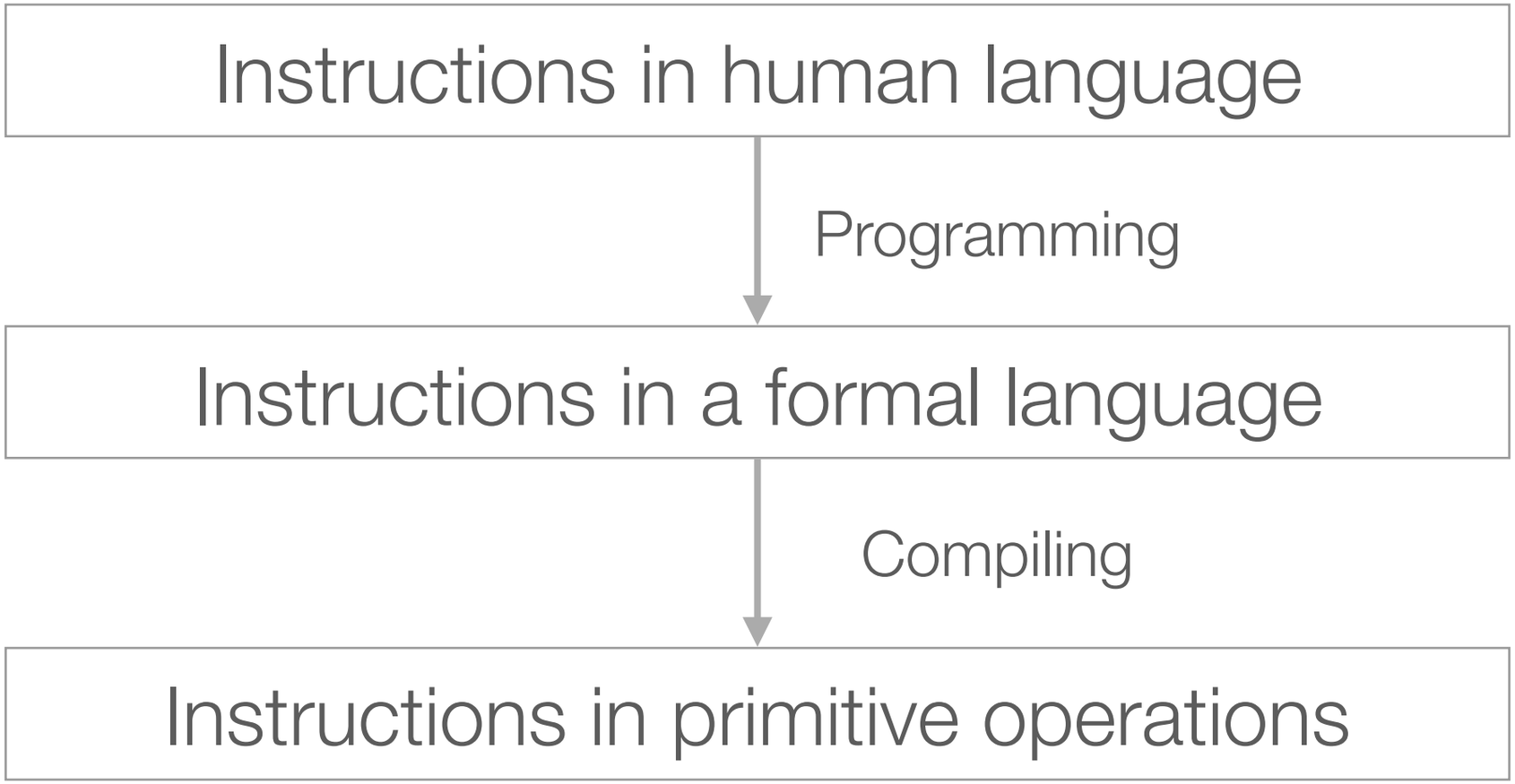
- An ***algorithm*** is a step-by-step procedure for solving a problem or accomplishing some end



BabySoothing

Computer Algorithms

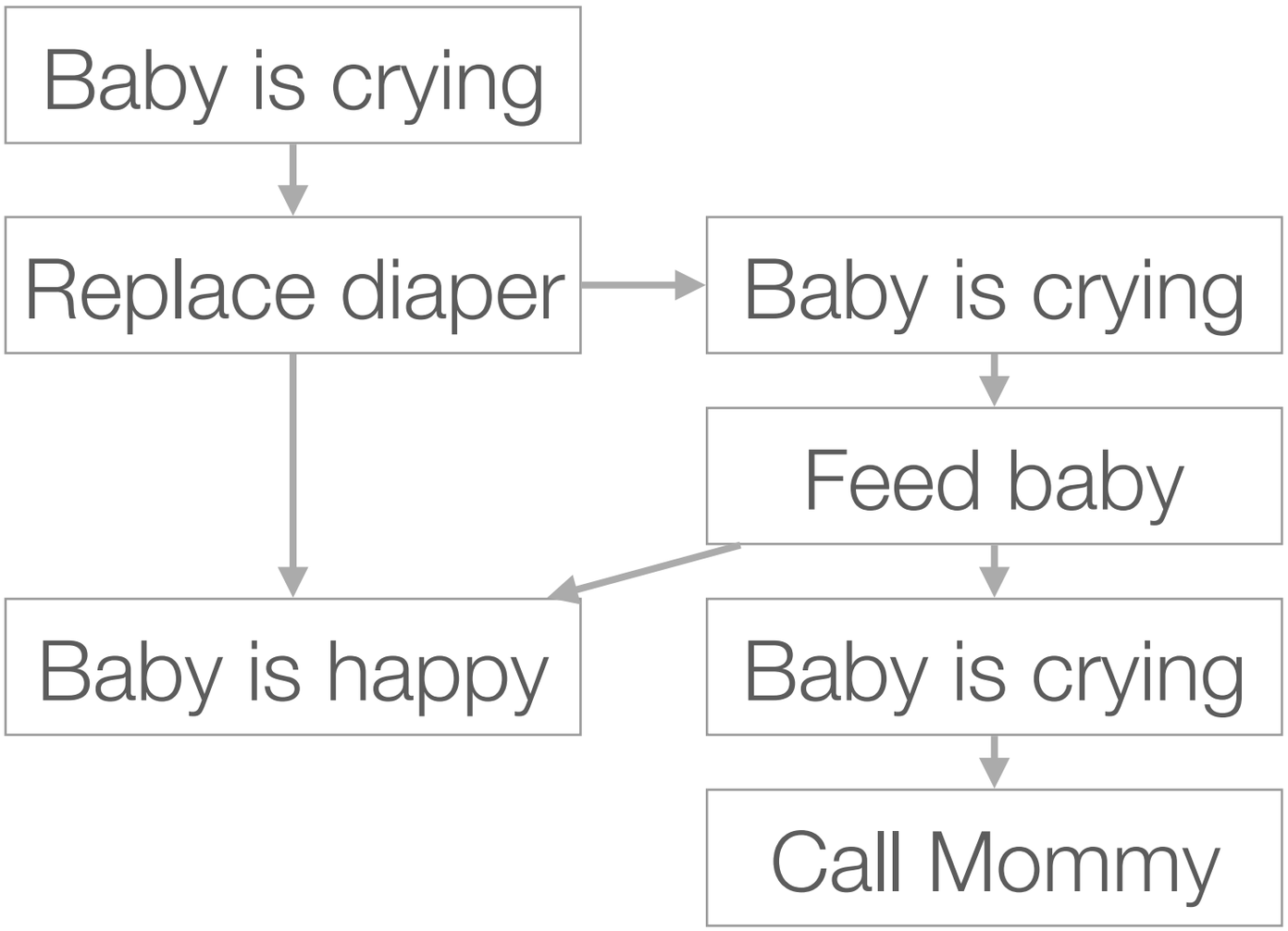
- A ***computer algorithm*** is an algorithm executed by a computer
- Assume a set of well-defined, limited primitive operations executable by a computer



Running Algorithms

- To run the BabySoothing algorithm, the computer has to
  - detect that the baby is crying,
  - replace the diaper,
  - feed the baby,
  - call the Mommy, and
  - follow the flow

Performance of Algorithms



BabySoothing

BabySoothing’

Speed of Computers

- Computers may not be able to replace a diaper quickly
- Computers can perform millions of instructions per second
- Computers can deal with billions, or even trillions, of bits of information

Large-Scale Computations

- Computers can deal with very complicated tasks
- Computers may have to perform those tasks many times
- Algorithms work well for small problems may be terrible for large problems
  - Example: https://www.toptal.com/developers/sorting-algorithms
- It is worthwhile to spend time designing better algorithms (designed once, used forever)

Intuition vs. Complexity

- An intuitive algorithm is usually easy to understand but is inefficient
- An efficient algorithm is usually complicated and hard to understand
- Various algorithms can be developed to solve a problem
- Algorithm design is an important and usually the hardest part of programming

Development of Algorithms

- We are typically given a problem statement, including input and output requirements
- The development of an algorithm involves the following tasks:
  - Design (main subject of this course)
  - Proof of correctness (or verification)
  - Analysis
  - Implementation
- May need to iterate

Creative Approach to Algorithm Design

- Emphasis of the creative side
  - not only memorizing solutions
  - but also learning to create by trying to create
- ***Induction*** as one central design method
  - to explain/understand the principles behind a design
  - to systematically guide the creation process

Design by Induction

- Draw analogies from proving theorems by mathematical induction
- In a proof by induction, we do not prove a statement from scratch, but rather we show
  - the correctness of the statement follows from that of the same statement for smaller instances and
  - the correctness of the statement for a small base case
- Concentrate on extending solutions for smaller problem instances to solutions for larger ones
- Induction may not solve every problem, but is very helpful