The model of algorithmic thinking – dimensions and levels

Ján GUNIŠ – Ľubomír ŠNAJDER

11th -13th of September 2012, **Information & Communication Technology in Education 2012**
Rožnov pod Radhoštěm, Czech Republic

Content

- Algorithmic thinking
- Three-dimensional model of algorithmic thinking
- Ideas for the development of algorithmic thinking

Algorithmic thinking

- Algorithmic thinking is a part of some competencies defined in the State Educational Program:
  - competencies to apply mathematical thinking and learning in science and technology,
  - competencies in the field of information and communication technologies,
  - competencies to solve problems.

Algorithmic thinking

A set of capabilities (1/2):

- to **analyse** given **problems** and specify a problem precisely
- to **find** a **way** to transform the **input data to output data**
- to **identify** the **basic actions** (instructions) of the executor
- to **construct** an **algorithm** consisting of basic actions
- to **perform** the **algorithm**, or at least simulate the performance of an algorithm
- to **think** about **all cases** of the problem

Algorithmic thinking

A set of capabilities (2/2):

- to **verify** (show, prove) the **correctness** of the algorithm
- to **increase** the **efficiency** of the algorithm
- to **compare** different **algorithms** according to various criteria
- to **understand** the **algorithm** within the meaning, **identify** the **problems** it solves
- to **recognize** a **situation** in which the algorithm can be used

Algorithmic thinking

Other ways of thinking connected with solving of algorithmic problems:

- abstract,
- logical,
- creative,
- critical,
- mathematical,
- analytical,
- ...

Computational thinking

Algorithmic thinking is a part of the **computational thinking**, which is often referred to as a key capability of 21st century.

- **thinking recursively**
- **parallel processing**
- **thinking** at multiple **levels of abstraction**
- thinking algorithmically and with the ability to apply mathematical concepts such as induction to develop more efficient, fair and secure solutions.

Computational thinking

To solve the problem we are considering:

- urgency of the solution,
- resources (tools),
- our skills and abilities,
- reusing solutions,
- different variations of the problem
- …

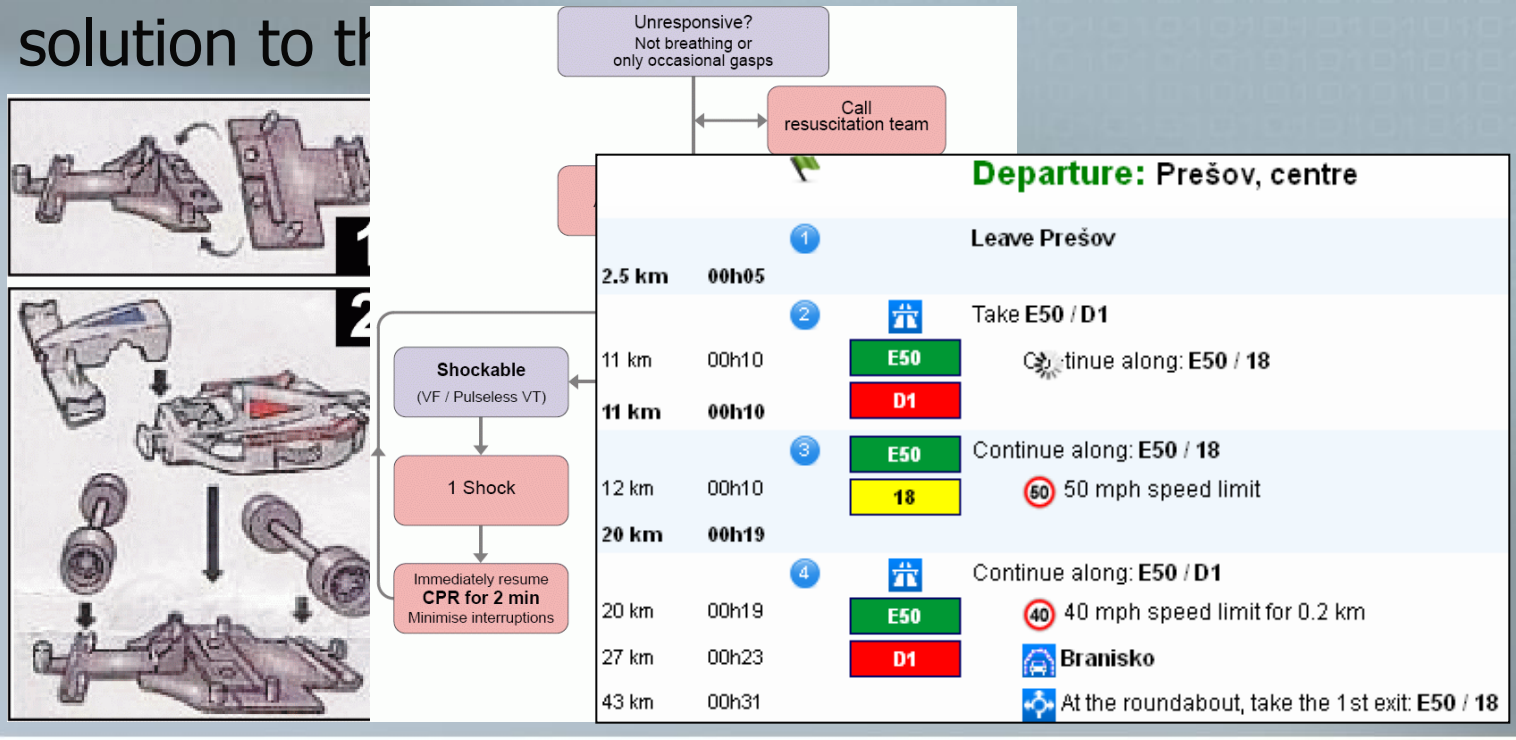
Computational thinking

Example - rendering the target:

- use paper and markers
- draw it in a raster graphics editor
- draw it in a vector graphics editor
- create a program to draw it (size, number of annulus, colour etc.) in a raster format
- create a program to draw it (size, number of annulus, colour etc.) in a vector format
- publish program code for others (not only to use, but also to edit and improve the code to solve other problems)

Algorithmic thinking

- is not just the domain of computer engineers and programmers, is part of education of each pupil,
- not only to solve a problem but also to find a solution to th



Three-dimensional model of algorithmic thinking

Two cases:

- The first pupil wrote binary search program in a programming language Pascal in the written exam.
- The second pupil used the binary search algorithm in the guess number game.

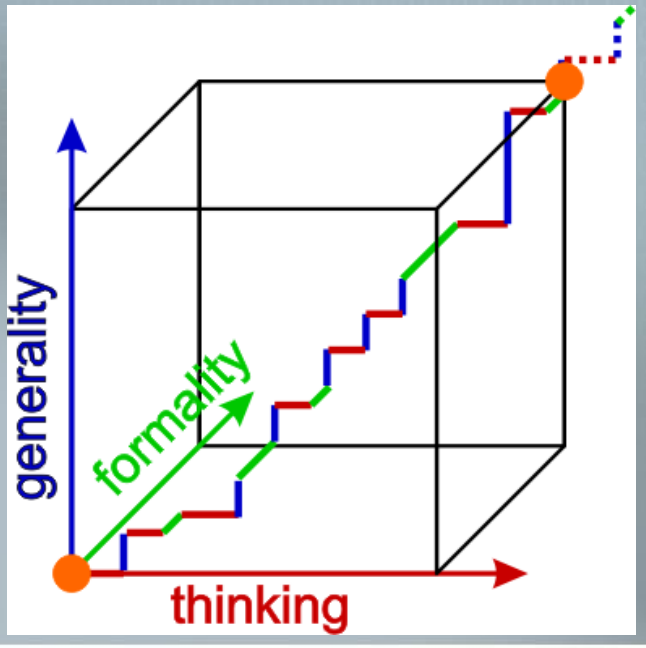
How can we compare the level of algorithmic thinking in these two specific cases?

Three-dimensional model of algorithmic thinking

During process of finding solution of problems:

- use various cognitive functions,
- find solutions with various levels of generality
- express found solutions in various formal levels

For describing the level
of algorithmic thinking
to solve problems
we consider 3 dimensions:



Three-dimensional model of algorithmic thinking

Proposed dimensions are demonstrated on a problem of a mole that is finding food hidden somewhere in its maze of corridors.

Dimension of generality of the solution

levels of dimension of generality:

- a solution covering a particular case, e.g. finding food in the particular maze of corridors,
- a solution covering several particular cases, e.g. finding food in one of several not intersecting corridors,
- a solution covering several cases, e.g. finding food in one of the several branching (mutually intersecting) corridors,
- almost general solution not covering some particular cases, e.g. finding food on the place where a mole starts to look for or the case where there is no food,
- a general solution covering all cases.

Dimension of formal level of expressing the solution

levels of dimension of formality (1/3):

- **nonverbal expression** of an algorithm by a **direct action**, activity, e.g. by marking particular route leading to food or passing corridors drawn on the ground by physical way to reaching food,
- expression of solving algorithm in **natural language without any restrictions** – a pupil not only performs, but also describes her/his solution, e.g. "put your right hand on the wall and walk along the wall all the time",

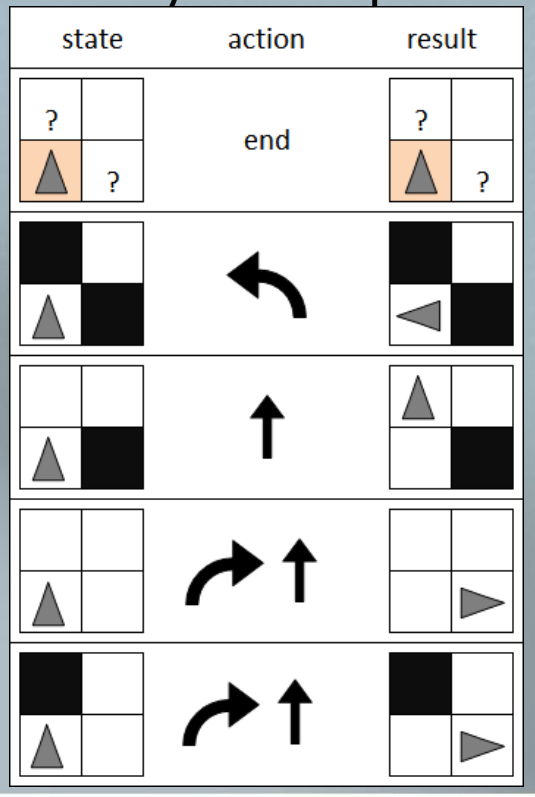
Dimension of formal level of expressing the solution

levels of dimension of formality (2/3):

- expression of solving algorithm in **natural language with elements of specific domain notations** or by the help means of executor of algorithms, e.g.

```
put right hand to the wall
while not find the food
  walk along the wall
```

- expression of solving algorithm by the help of formal **iconic (pictorial) language**



Dimension of formal level of expressing the solution

levels of dimension of formality (3/3):

- expression of solving algorithm using **formal language**, e.g. a programming language

```
to go
  right 90
  while [dotColor <> "black][
    forward 1
 ]
  back 1
  left 90

  while [dotColor<> "red] [
    ifElse isRightWall [
      ifElse isFrontWall [
        left 90
      ][
        forward 1
      ]
    ][
      right 90
```

```
to isFrontWall
  forward 1
  ifElse dotColor = "black [
    back 1
    output "true
  ][
    back 1
    output "false
  ]
end

to isRightWall
  right 90
  forward 1
  ifElse dotColor = "black [
    back 1
    left 90
```

Dimension of cognitive functions involvement during solution finding

levels of dimension of thinking (1/2):

- **remembering** – reproduction of remembered and already existing algorithm without understanding, e.g. a pupil wrote the procedure go,
- **understanding** – reproduction of remembered and already existing algorithm with understanding, e.g. a pupil can explain in her/his own words the significance of internal condition,
- **applying** – application of a known algorithm in a particular situation, e.g. a pupil can modify the algorithm (program) so that a mole will colour its corridor walls yellow,

Dimension of cognitive functions involvement during solution finding

levels of dimension of thinking (2/2):

- **analysing** – analysis of existing algorithm, identification, location and correction an error, e.g. a pupil found that the algorithm does not work properly (it is infinite) in case of the absence of food in the corridors, and fix the error,
- **evaluating** – rationale of the correctness of the algorithm, e.g. a pupil reveals that the mole visits all corridors along walls following right side, ... proves that the algorithm leads to the right solution,
- **creating** – e.g. a pupil creates an algorithm for finding all the pieces of food in the corridors that branch and intersect each other.

Three-dimensional model of algorithmic thinking

- Proposed 3D model:
  - **does not describe** the level of pupil's algorithmic thinking **in general**,
  - **describes** the level of her/his algorithmic thinking **considering the problem**.
  - if a pupil during solving **various problems** achieves **similar levels** of algorithmic thinking, we can **estimate** the level of algorithmic thinking in **solving further problems**.

Ideas for the development of algorithmic thinking

- It is natural to assume that algorithmic thinking is **best** developed in solving **programming tasks**.
- But we believe that algorithmic thinking can be developed in the **whole population** of pupils and in **different school subjects**. It doesn't matter if it is not achieved imaginary point [1, 1, 1] in the above 3D model.
- It is important to realize that **we cannot** make **big jumps** in one dimension at the expense of others.

Examples of development of algorithmic thinking

- Programming of algorithmic problems in programming language
- Playing computer games aimed at developing of algorithmic thinking
- Solving algorithmic problems in environments of office suite, multimedia editors, specialized modelling programs
- Solving of algorithmic tasks without programming

Conclusion

- The proposed 3D model allows us to better understand the algorithmic thinking and realize that algorithmic thinking can be developed in different ways with/without computer, within programming/non-programming school subjects.
- We believe that teachers realize the importance of this ability of their pupils and will contribute to its targeted development.

These results described in the article have achieved in the frame of the project LPP-APVV 0057-09 Developing youth talents via correspondence seminars and competitions

Contact

Mgr. Ján Guniš
jan.gunis@upjs.sk
RNDr. Ľubomír ŠNAJDER, PhD.
lubomir.snajder@upjs.sk
Pavol Jozef Šafárik University in Košice
Faculty of Science
Institute of Computer Science
Jesenná 5, 041 54 Košice, Slovak Republic

GPS: 48° 43' 44.78" N, 21° 14' 50.83" E

Tel: +421 55 234 2539 (our office),
+421 55 234 2502 (our secretary)