GÖTEBORGS UNIVERSITET

UPPSALA UNIVERSITET

Researching the implementation of programming in school mathematics

NORMA 2024, Copenhagen

Cecilia Kilhamn

NCM Nationellt centrum för matematikutbildning

GÖTEBORGS UNIVERSITET

1

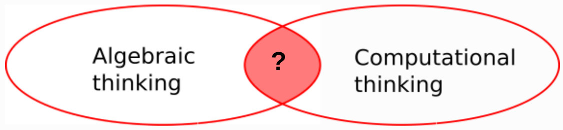
In memory of Kajsa Bråting

* 1975-02-07 † 2024-02-03

Professor i didaktik
med inriktning mot matematik
vid Uppsala universitet

GÖTEBORGS UNIVERSITET

2



inPAC

Integrating Programming in School Mathematics

Exploring the intersection of Algebraic and Computational thinking

Kajsa Bråting

Cecilia Kilhamn

Lennart Rolandsson

Co-authored papers with:
Rimma Nyman
Ola Helenius & John Mason
Raimundo Elicer & Andreas Lindenskov Tamborg

Special thanks to
IFOUS: Eva Norén & Jalal Nouri
The Swedish Research Council
[grant number 2018-03865]

GÖTEBORGS UNIVERSITET

2019 – 2023 Funded by the Swedish Research Council (VR)

3

Setting the scene around 2018: Implementation of programming in school curricula

England and Denmark.
Programming as parts of a separate subject: *Computing* / *Teknologiförståelse*.

Sweden, Finland, Norway:
Programming across the curriculum and/or integrated with mathematics.

Sweden:
Programming included in the ”core content of algebra”.

New curriculum presented 2017 (early adopters)
Compulsory for all math teachers aug 2018
Insufficient professional development offered
No pre-produced teaching materials
No recommended tasks, platforms or progams

GÖTEBORGS UNIVERSITET

Bråting, K., Kilhamn, C., & Rolandsson, L. (red.) (2021). *Programmering i skolmatematiken – möjligheter och utmaningar.*

4

Programming in the core content of algebra
Swedish curriculum 2018

Grades 1-3:

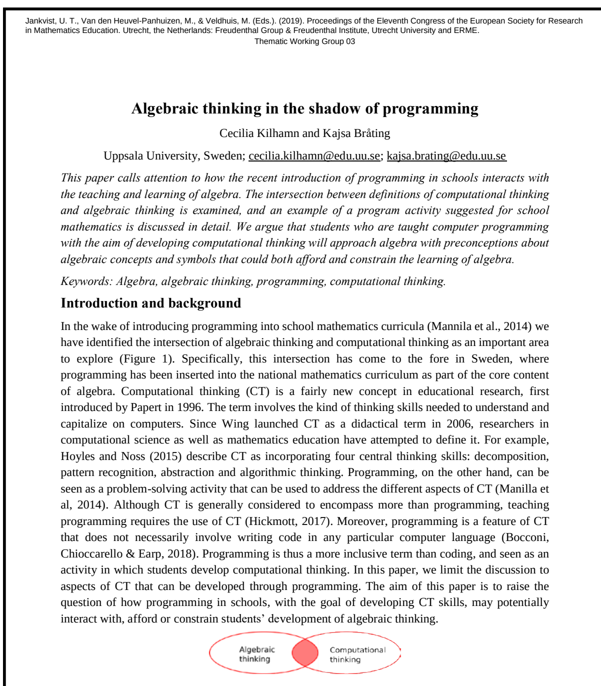
- How unique stepwise instructions can be constructed, described and followed as a basis for programming.
- The usage of symbols in connection with stepwise instructions.

Grades 4-6 and 7-9:

- How algorithms can be created and used in programming.
- Programming in visual/different programming environments.

GÖTEBORGS UNIVERSITET

5


Jankvist, U. T., Van den Heuvel-Panhuizen, M., & Veldhuis, M. (Eds.). (2019). Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education. Utrecht, the Netherlands: Freudenthal Group & Freudenthal Institute, Utrecht University and ERME.
Thematic Working Group 03

Algebraic thinking in the shadow of programming

Cecilia Kilhamn and Kajsa Bråting

Uppsala University, Sweden; cecilia.kilhamn@edu.uu.se; kajsa.brating@edu.uu.se

This paper calls attention to how the recent introduction of programming in schools interacts with the teaching and learning of algebra. The intersection between definitions of computational thinking and algebraic thinking is examined, and an example of a program activity suggested for school mathematics is discussed in detail. We argue that students who are taught computer programming with the aim of developing computational thinking will approach algebra with preconceptions about algebraic concepts and symbols that could both afford and constrain the learning of algebra.

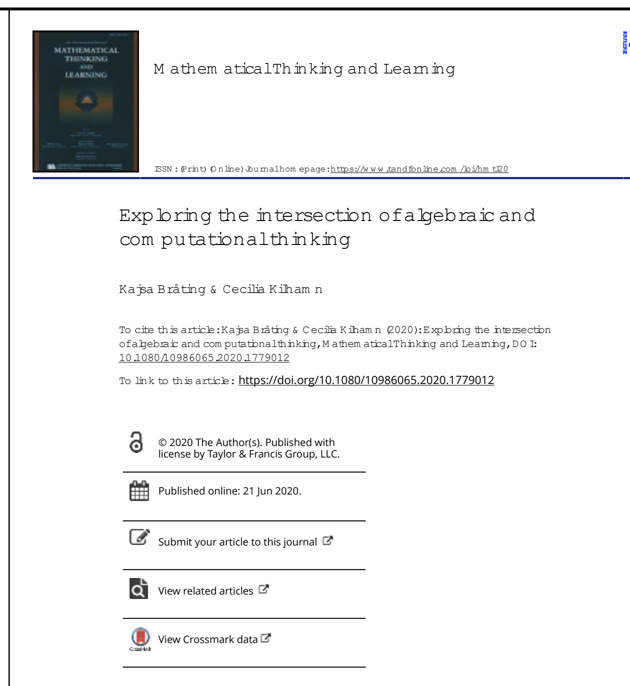
Keywords: Algebra, algebraic thinking, programming, computational thinking.

Introduction and background

In the wake of introducing programming into school mathematics curricula (Mannila et al., 2014) we have identified the intersection of algebraic thinking and computational thinking as an important area to explore (Figure 1). Specifically, this intersection has come to the fore in Sweden, where programming has been inserted into the national mathematics curriculum as part of the core content of algebra. Computational thinking (CT) is a fairly new concept in educational research, first introduced by Papert in 1996. The term involves the kind of thinking skills needed to understand and capitalize on computers. Since Wing launched CT as a didactical term in 2006, researchers in computational science as well as mathematics education have attempted to define it. For example, Hoyles and Noss (2015) describe CT as incorporating four central thinking skills: decomposition, pattern recognition, abstraction and algorithmic thinking. Programming, on the other hand, can be seen as a problem-solving activity that can be used to address the different aspects of CT (Manilla et al, 2014). Although CT is generally considered to encompass more than programming, teaching programming requires the use of CT (Hickmott, 2017). Moreover, programming is a feature of CT that does not necessarily involve writing code in any particular computer language (Bocconi, Chioccarello & Earp, 2018). Programming is thus a more inclusive term than coding, and seen as an activity in which students develop computational thinking. In this paper, we limit the discussion to aspects of CT that can be developed through programming. The aim of this paper is to raise the question of how programming in schools, with the goal of developing CT skills, may potentially interact with, afford or constrain students' development of algebraic thinking.

Algebraic thinking | Computational thinking

Figure 1: The intersection of CT and AT


Mathematical Thinking and Learning

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hmtl20

Exploring the intersection of algebraic and computational thinking

Kajsa Bråting & Cecilia Kilhamn

To cite this article: Kajsa Bråting & Cecilia Kilhamn (2020): Exploring the intersection of algebraic and computational thinking, Mathematical Thinking and Learning, DOI: 10.1080/10986065.2020.1779012

To link to this article: https://doi.org/10.1080/10986065.2020.1779012

© 2020 The Author(s). Published with license by Taylor & Francis Group, LLC.

Published online: 21 Jun 2020.

Submit your article to this journal

View related articles

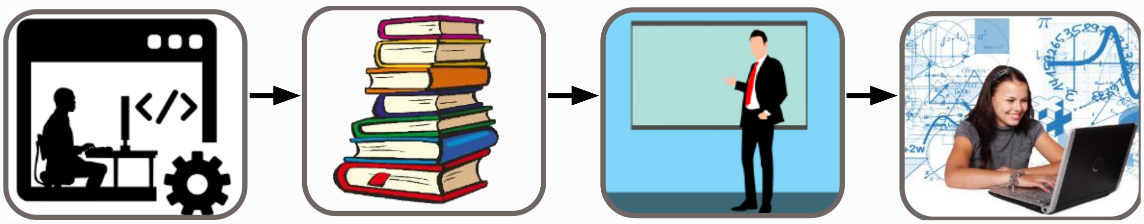
View Crossmark data

6

Theoretical frame of the project

Chevallard's (2006) transposition of knowledge

Scholarly knowledge	Knowledge to be taught	Taught knowledge	Learned knowledge
Intitutions producing and using the knowledge	Curricular documents, educational system, textbooks	Classsroom practice	Students

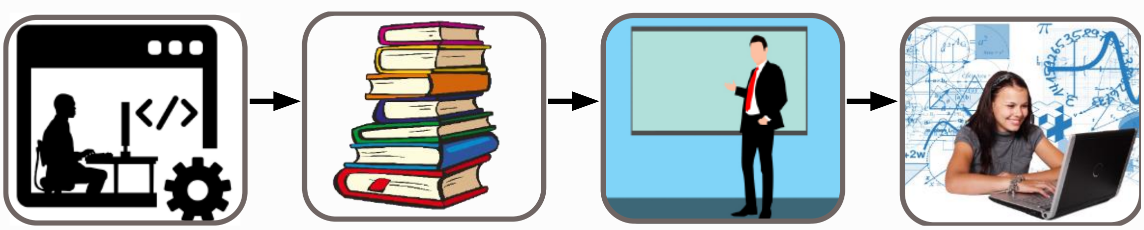


Chevallard, Y. (2006). Steps towards a new epistemology in mathematics education

Bosch, M., & Gascon, J. (2006). Twenty-five years of didactic transposition. ICMI Bulletin, 58, 51–63.

7

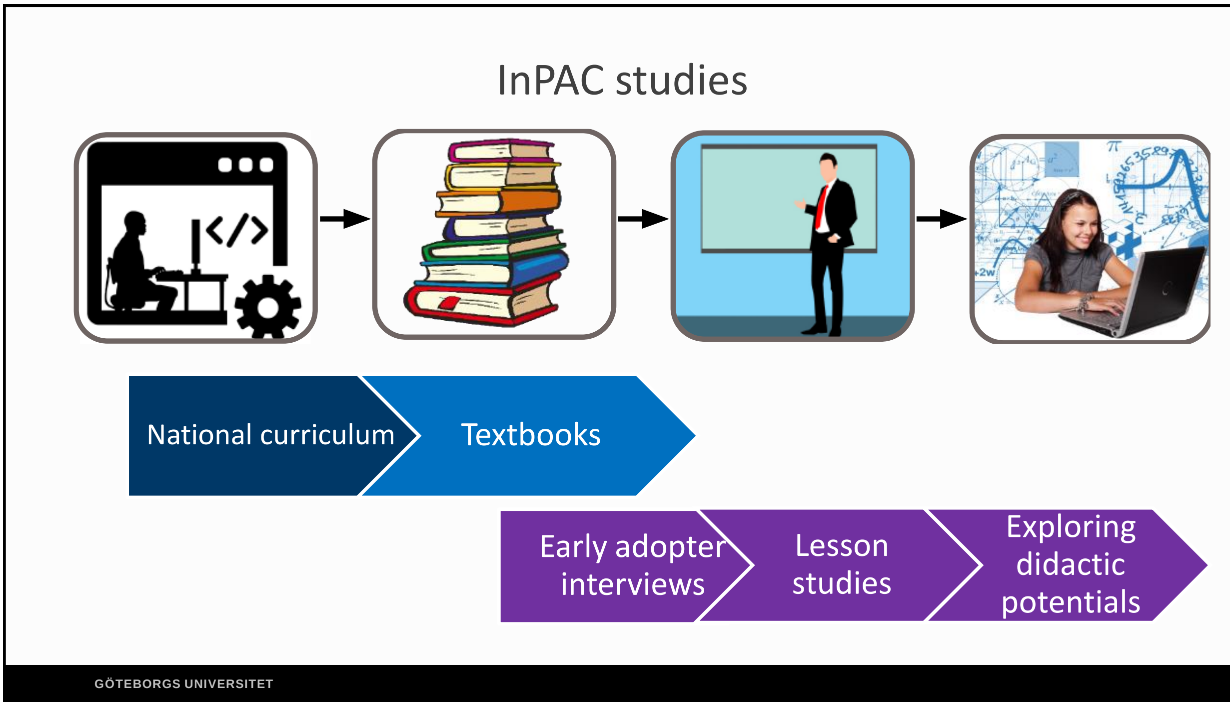
InPAC studies



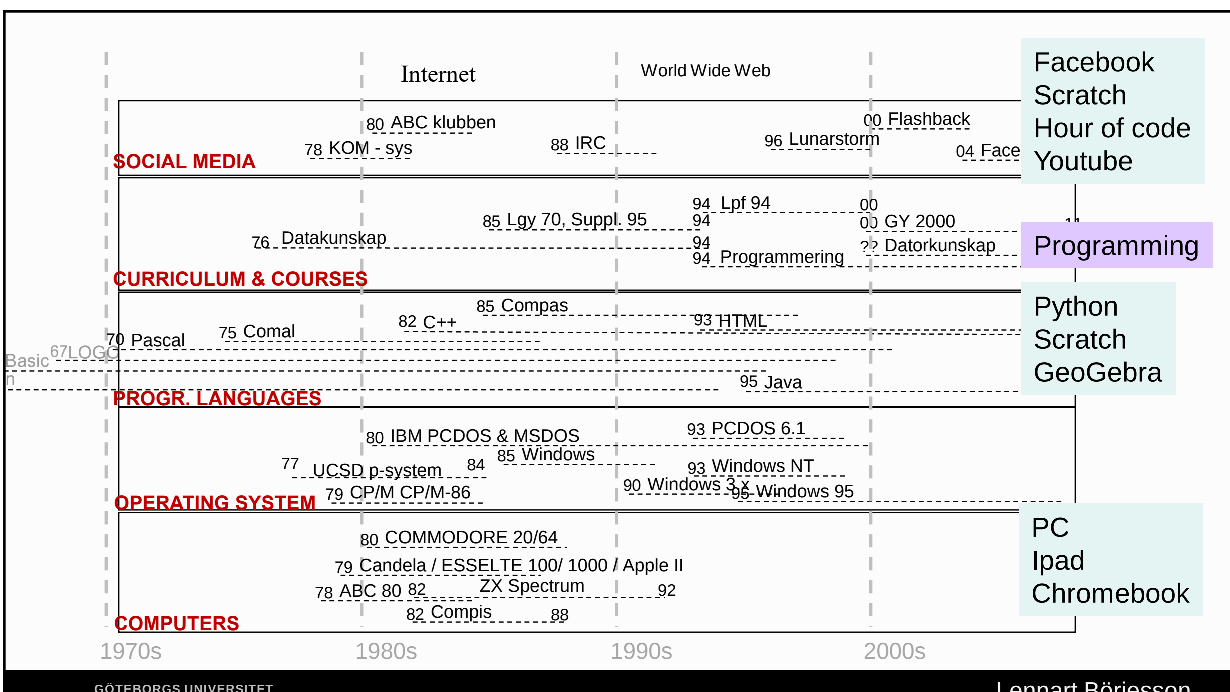
National curriculum

1. Why in mathematics?
2. Why not a core content of its own?
3. Why in algebra?

8


InPAC studies
National curriculum
Textbooks
Early adopter interviews
Lesson studies
Exploring didactic potentials
GÖTEBORGS UNIVERSITET

9


Internet
World Wide Web
SOCIAL MEDIA
80 ABC klubben
78 KOM - sys
88 IRC
96 Lunarstorm
00 Flashback
04 Face
Facebook
Scratch
Hour of code
Youtube
CURRICULUM & COURSES
76 Datakunskap
85 Lgy 70, Suppl. 95
94 Lpf 94
94
94
94 Programmering
00
00 GY 2000
?? Datorkunskap
Programming
PROGR. LANGUAGES
Basic
67LOGO
70 Pascal
75 Comal
82 C++
85 Compas
93 HTML
95 Java
Python
Scratch
GeoGebra
OPERATING SYSTEM
77 UCSD p-system
79 CP/M CP/M-86
80 IBM PCDOS & MSDOS
84
85 Windows
90 Windows 3.x
93 PCDOS 6.1
93 Windows NT
95 Windows 95
COMPUTERS
78 ABC 80 82
79 Candela / ESSELTE 100/ 1000 / Apple II
80 COMMODORE 20/64
82 Compis 88
ZX Spectrum
92
PC
Ipad
Chromebook
1970s
1980s
1990s
2000s
GÖTEBORGS UNIVERSITET
Lennart Börjesson

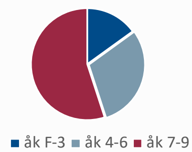
10

Teachers' voices

Early adopter interviews

How do teachers integrate programming in their existing mathematics teaching? What does this integration entail?

- Interviews with 20 ”early adopters”



Early adopters 2018-2019

Rely heavily on social media and internet-based free-of-charge platforms

Content: geometrical shapes, calculations, probability, statistics, (no algebra)

GÖTEBORGS UNIVERSITET

11

Chevallard's praxeology know-how and know-why

Praxis know-how	Types of tasks (what)
	Technique (how)
Logos know-why	Technology (why this technique for this task)
	Thoery (why these tasks and techniques at all, why do we use them and trust them, what do they tell us)

Reference epistemological model (REM) Programming in mathematics

Praxis know-how	Data handling, statistiscs, proofs, modelling to test outcomes, calculating results …
	Algorithmic techniques, sequencing, tinkering, remixing …
Logos know-why	Using the power and speed of the computer
	Logically constructed unambiguous steps of action give us trustworthy and repeatable results

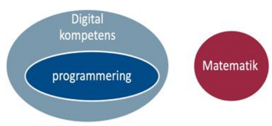
GÖTEBORGS UNIVERSITET

Chevallard, Y. (2006). Steps towards a new epistemology in mathematics education

Bosch, M., & Gascon, J. (2006). Twenty-five years of didactic transposition. ICMI Bulletin, 58, 51–63.

12

Know-how and know-why for early adopters

Praxis **what & how**	*Communicate with the computer.* *Focus: algorithmic thinking, coding, syntax*
what language	*What the curriculum said*
Logos	

GÖTEBORGS UNIVERSITET

13

Teachers' voices

Lesson studies

- 32 Lesson studies – teachers' documentation

Developed a framework describing

the relation between Mathematics and Programming

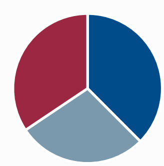
in these lessons (MP framework)

Only programming (no math)

Mathematics as a context for programming

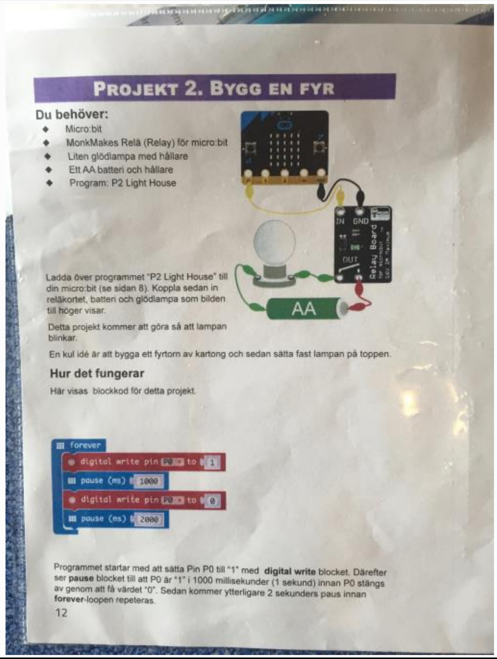
Programming a tool for efficient calculations

Exploring math through programming



GÖTEBORGS UNIVERSITET

14



Projekt 2. Bygg en fyr

Du behöver:

- Micro:bit
- MonkMakes Relä (Relay) för micro:bit
- Liten glödlampa med hållare
- Ett AA batteri och hållare
- Program: P2 Light House

Ladda över programmet "P2 Light House" till din micro:bit (se sidan 8). Koppla sedan in reläkortet, batteri och glödlampa som bilden till höger visar.

Detta projekt kommer att göra så att lampan blinkar.

En kul idé är att bygga ett fyrtorn av kartong och sedan sätta fast lampan på toppen.

Hur det fungerar

Här visas blockkod för detta projekt.

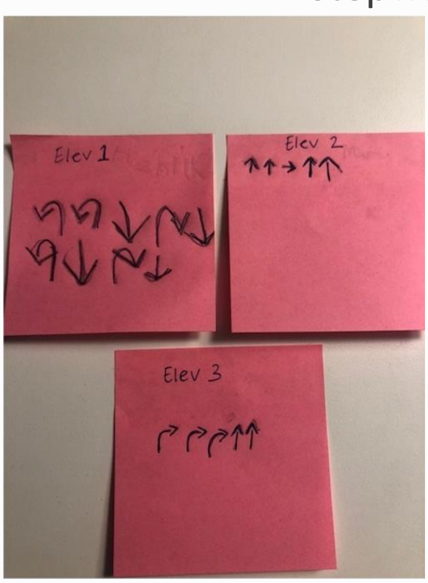
```
forever
    digital write pin P0 to 1
    pause (ms) 1000
    digital write pin P0 to 0
    pause (ms) 2000
```

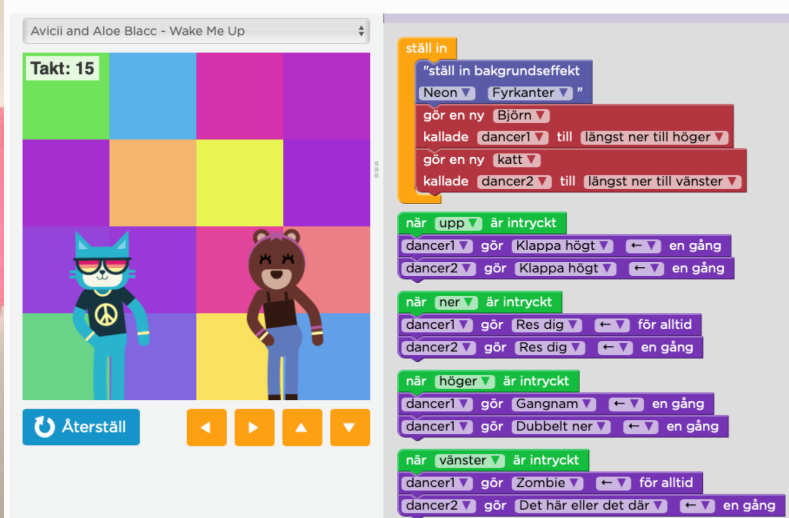
Programmet startar med att sätta Pin P0 till "1" med **digital write** blocket. Därefter ser **pause** blocket till att P0 är "1" i 1000 millisekunder (1 sekund) innan P0 stängs av genom att få värdet "0". Sedan kommer ytterligare 2 sekunders paus innan **forever**-loopen repeteras.

12

15

stepwise instructions

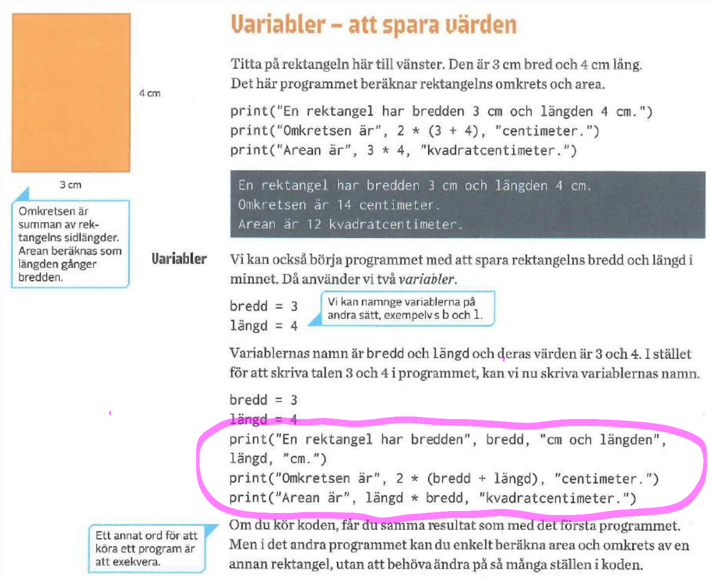




GÖTEBORGS UNIVERSITET

16

Mathematics as a context: perimeter and area in Python, grade 7-8



Variabler – att spara värden

Titta på rektangeln här till vänster. Den är 3 cm bred och 4 cm lång. Det här programmet beräknar rektangelns omkrets och area.

```
print("En rektangel har bredden 3 cm och längden 4 cm.")
print("Omkretsen är", 2 * (3 + 4), "centimeter.")
print("Arean är", 3 * 4, "kvadratcentimeter.")
```

```
En rektangel har bredden 3 cm och längden 4 cm.
Omkretsen är 14 centimeter.
Arean är 12 kvadratcentimeter.
```

4 cm

3 cm

Omkretsen är summan av rektangelns sidlängder. Arean beräknas som längden gånger bredden.

Variabler

Vi kan också börja programmet med att spara rektangelns bredd och längd i minnet. Då använder vi två *variabler*.

```
bredd = 3
längd = 4
```

Vi kan namnge variablerna på andra sätt, exempelvis b och l.

Variablernas namn är bredd och längd och deras värden är 3 och 4. I stället för att skriva talen 3 och 4 i programmet, kan vi nu skriva variablernas namn.

```
bredd = 3
längd = 4
print("En rektangel har bredden", bredd, "cm och längden",
längd, "cm.")
print("Omkretsen är", 2 * (bredd + längd), "centimeter.")
print("Arean är", längd * bredd, "kvadratcentimeter.")
```

Om du kör koden, får du samma resultat som med det första programmet. Men i det andra programmet kan du enkelt beräkna area och omkrets av en annan rektangel, utan att behöva ändra på så många ställen i koden.

Ett annat ord för att köra ett program är att exekvera.

Area = length · width

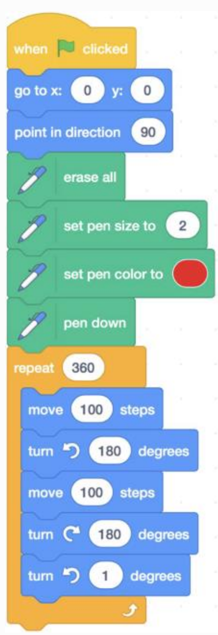
Perimeter = 2 · (length + width)

$A = x \cdot y$

$P = 2 \cdot (x + y)$

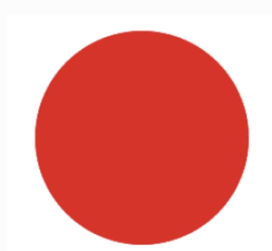
GÖTEBORGS UNIVERSITET

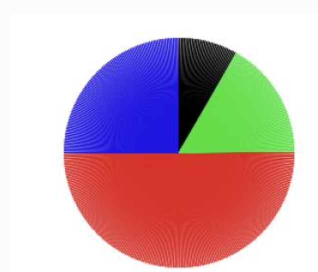
17



Exploring mathematics in grade 6

Explore the area of a whole circle and
change the code so that half the circle is red and half is blue.





GÖTEBORGS UNIVERSITET

18

Content	1.Only programming	2. Math as a context for programming	3. Programming as a tool for efficient calculations	4. Programming as a tool for exploration
Programming only, n=10	Analog: 2 Robot: 4 Block: 4			
Numbers, n=8		Analog: 1 Block: 3 Text: 3	Text: 1	
Geometry, n=8		Analog: 1 Robot: 1 Block: 2 Text: 1	Text: 1	Block: 2
Statistiks and probability, n=3			Block: 1 Text: 1 Excel: 1	
Algebra, n=0				
Problem solving, n=1			Text: 1	
Change and functions, n=2				Block: 2
n=32	n=10 **(31%)** grade 1-4/9	n=12 (**38%**) grade 1-8	n=6 (**18%**) grade 7-9	n=4 (**13%**) grade 4-6

GÖTEBORGS UNIVERSITET

19

Challenges:

- teachers' lack of programming competence,
- accepting failure as a resource for reflection,
- side effects and non-mathematical affordances of the programming environment/language,
- the time issue,
- the unclear relation to mathematics,
- similarities and differences between variables in algebra and in computer science

GÖTEBORGS UNIVERSITET

20

School algebra -------- Programming

= equality, balance

$x \neq x + 1$

same value

= assignement

$x = x + 1$

new value   old value

Variable:
- always numerical
- unknown number
- placeholder (general number)
- varying quantity

Variable:
- storage place for data
- numerical and non-numerical
- varying and non-varying
- changes during the process

GÖTEBORGS UNIVERSITET

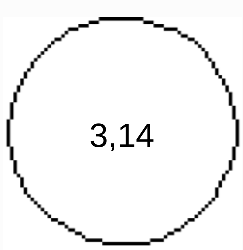
21

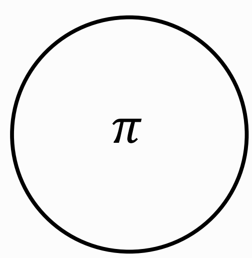
Programming

- Explicit, specific
- Approximate
- Stepwise, dynamic
- Time dependent

Mathematics/school algebra

- Abstract, general
- Precise
- Static
- Time independent





GÖTEBORGS UNIVERSITET

22

Textbook studies

Scandinavian Journal of Educational Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/csje20

The Integration of Programming in Swedish School Mathematics: Investigating Elementary Mathematics Textbooks

Kajsa Bråting & Cecilia Kilhamn

To cite this article: Kajsa Bråting & Cecilia Kilhamn (2022) The Integration of Programming in Swedish School Mathematics: Investigating Elementary Mathematics Textbooks, Scandinavian Journal of Educational Research, 66:4, 594-609, DOI: 10.1080/00313831.2021.1897879

To link to this article: https://doi.org/10.1080/00313831.2021.1897879

International Journal of Mathematical Education in Science and Technology

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tmes20

Can programming support mathematics learning? An analysis of Swedish lower secondary textbooks

Rimma Nyman, Kajsa Bråting & Cecilia Kilhamn

To cite this article: Rimma Nyman, Kajsa Bråting & Cecilia Kilhamn (08 Apr 2024): Can programming support mathematics learning? An analysis of Swedish lower secondary textbooks, International Journal of Mathematical Education in Science and Technology, DOI: 10.1080/0020739X.2024.2329345

To link to this article: https://doi.org/10.1080/0020739X.2024.2329345

Research Article LUMAT Special Issue 2023

Comparing the integration of programming and computational thinking into Danish and Swedish elementary mathematics curriculum resources

Raimundo Elicer[1], Andreas Lindenskov Tamborg[1], Kajsa Bråting[2] and Cecilia Kilhamn[3]

[1] University of Copenhagen, Denmark
[2] Uppsala University, Sweden
[3] University of Gothenburg, Sweden

Computational thinking has become part of the mathematics curriculum in several countries. This has led recently available teaching resources to explicitly integrate computational thinking (CT). In this paper, we investigate and compare how curriculum resources developed in Denmark — digital teaching modules — and Sweden — printed mathematics textbooks — have incorporated CT in mathematics for grades 1–6 (age 7–12). Specifically, we identify and compare the CT and mathematical concepts, actions, and combinations in tasks within these resources. Our analysis reveals that Danish tasks are oriented toward CT concepts related to data, actions related to programming, and mathematical concepts within statistics. This is different from Swedish tasks, which are oriented toward CT concepts related to instructions and commands, actions related to following stepwise procedures, and mathematical concepts related to patterns. Moreover, what is most dominant in one country is almost or completely absent in the other. We conclude the paper by contrasting these two approaches with existing knowledge on computational thinking in school mathematics.

ARTICLE DETAILS

LUMAT Special Issue
Vol 11 No 3 (2023), 77–102

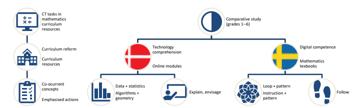
Received 15 February 2023
Accepted 13 November 2023
Published 24 November 2023

Pages: 26
References: 34

Correspondence:
reli@ind.ku.dk

https://doi.org/10.31129/LUMAT.11.3.1940

Keywords: computational thinking, curriculum resources, mathematics, programming



LUMAT: International Journal on Math, Science and Technology Education
Published by the University of Helsinki, Finland / LUMA Centre Finland | CC BY 4.0

Programming i math textbooks

RQ1: *What characterizes the programming content in Swedish textbooks in school mathematics?*

RQ2: *In what ways do the programming tasks create a bridge between programming and mathematics?*

Grades 1–6
4 book series
390 tasks

Grades 7–9
3 book series + 3 special books
62 ”units” , 386 tasks

GÖTEBORGS UNIVERSITET

25

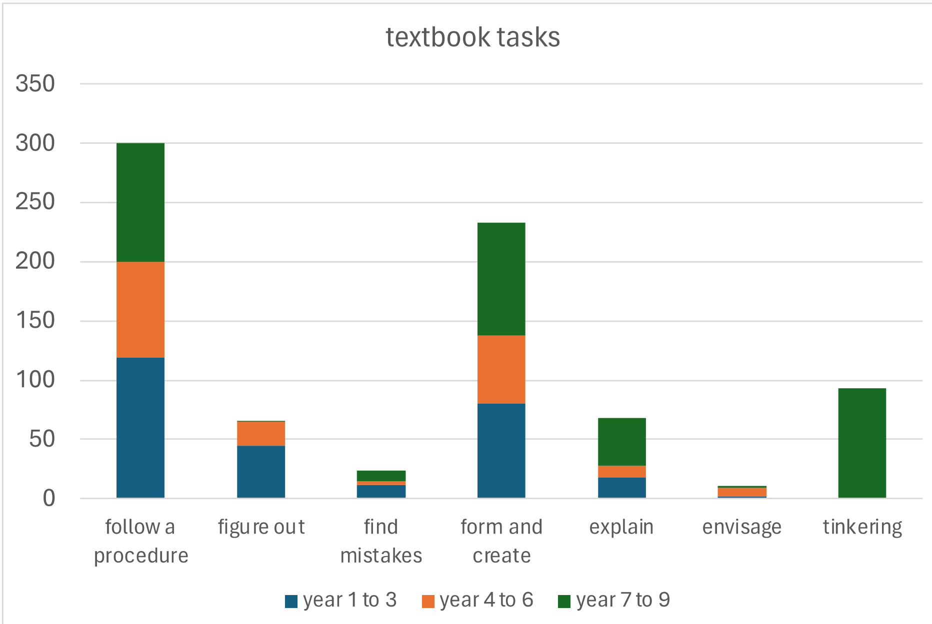
Analysing tasks

Actions (inspired by Benton et al., 2017: Explore, Envisage, Explein, Exchange, bridgE)

- **Follow a procedure -** follow stepwise instructions, copy/run a code and see
- **Figure out –** figure out a rule, see a pattern
- **Find mistakes –** debug a code
- **Form and create -** write code, give instructions, create a pattern
- **Explain -** explain procedure/code using natural language
- **Envisage** - predict and reflect on possible outcomes
- **Tinkering** (only 7–9) – change and manipulate a code

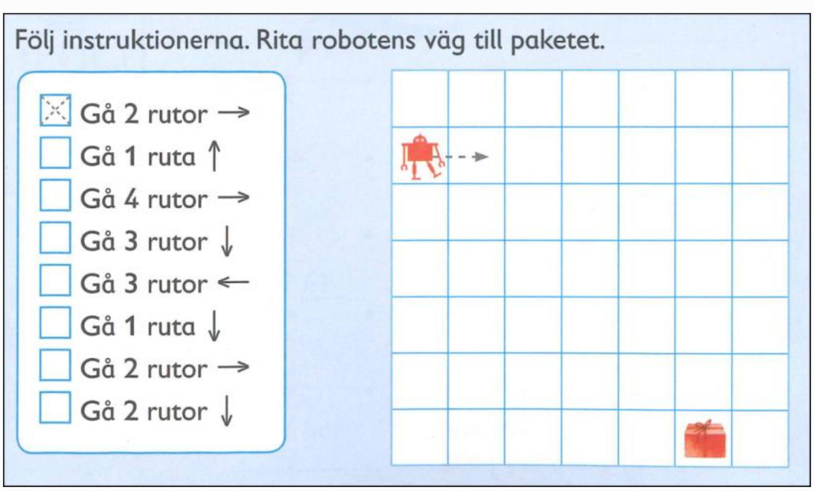
GÖTEBORGS UNIVERSITET

26



27

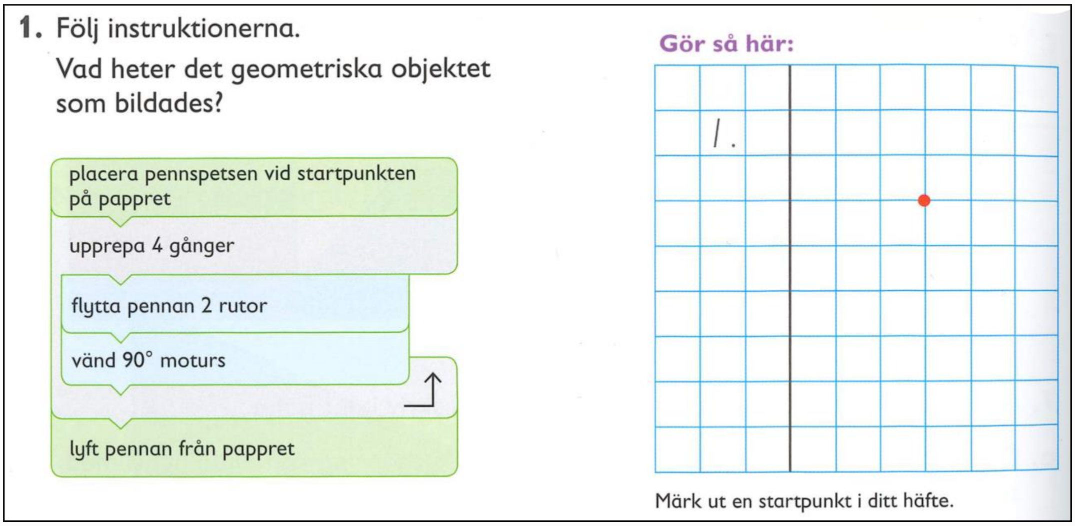
”Follow a procedure”



Favorit matematik 1B, p. 187

28

”Follow a procedure”

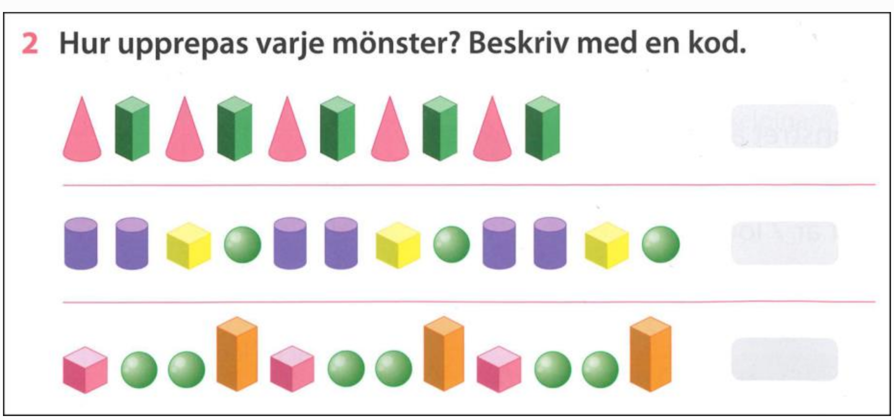


Matematik Favorit 5A, p. 162

GÖTEBORGS UNIVERSITET

29

”Figure out” Bridging math and programming?



Singma 3B, p. 130

GÖTEBORGS UNIVERSITET

31

Bridging math and programming?

Finding prime number

a) Try Emelie's code and find out if it works.

b) Change to other numbers to see if they are primes.

c) There is a bug in the code: when we test with the number 1 the computer answers that it is a prime. How can the bug be corrected?

d) Emelie's teacher tries really big numbers, but then it takes very long time to run the program. How can the code be changed so it gets faster? Can you find solutions on the web which you can use in your code? (A tip: Does the loop need to go through all integers?)

```
var talet = 179424691;
var primtal = true;

var test = 2;
while (test < talet)
{
  if (talet % test == 0) { primtal = false;}
  test += 1;
}

alert(primtal);
```

follow
tinker
debug
envision

exploring the concept of prime numbers

Hitta koden, p. 2

GÖTEBORGS UNIVERSITET

32

The relation between Mathematics and Programming in textbook units for grades 7–9

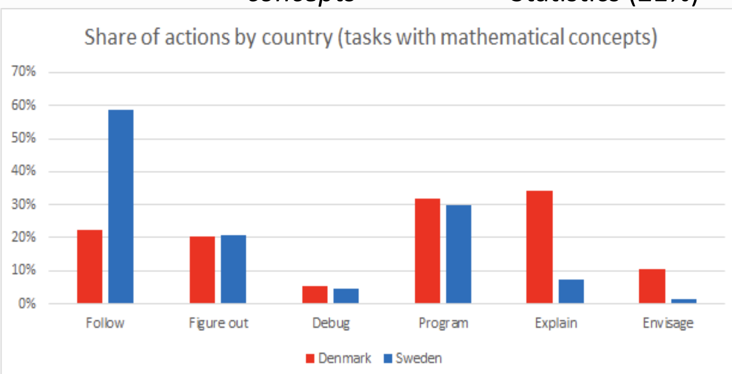
Only programming (no math)	13 %
Mathematics as a context for programming	74 %
Programming a tool for efficient calculations	3 %
Exploring math through programming	10 %

GÖTEBORGS UNIVERSITET

33

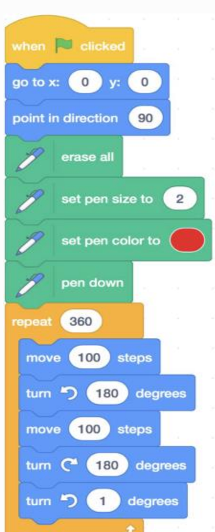
Grades 1-6


	Denmark	Sweden
Computational thinking concepts	Program (20%) Data (12%) algorithm (12%)	Instruction (48%) loop/repetition (26%) algorithm (19%)
Mathematical concepts	Geometry (34%) Statistics (21%)	Pattern, sequence (34%) Geometry (16%) Statistics (1%)



GÖTEBORGS UNIVERSITET

34



ZDM – Mathematics Education
https://doi.org/10.1007/s11858-022-01384-0

ORIGINAL PAPER

Variables in early algebra: exploring didactic potentials in programming activities

C. Kilhamn[1] · K. Bråting[2] · O. Helenius[1] · J. Mason[3,4]

Exploring didactic potentials

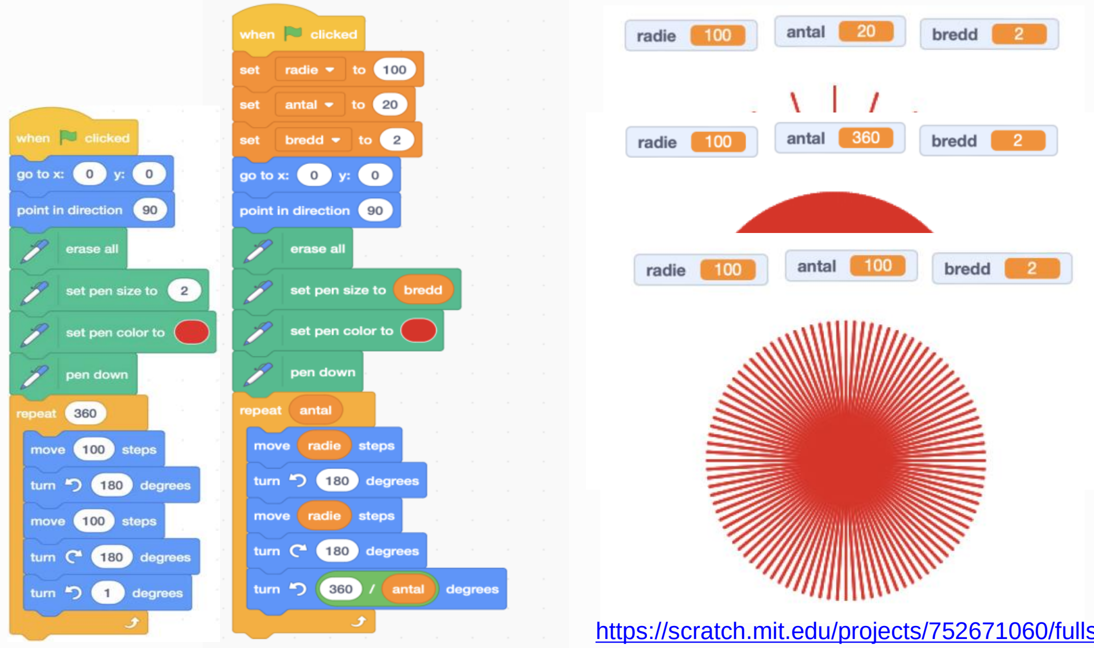
Explore the area of a whole circle and change the code so that half the circle is red and half is blue.

Table 1 Implicit variables in the original code

Implicit variable	Block
sprite_position	go to, move
sprite_rotation	point in direction, turn
pen_size	set pen size to
pen_color	set pen color to
pen_location	pen up, pen down
loop_counter	Repeat

GÖTEBORGS UNIVERSITET

35

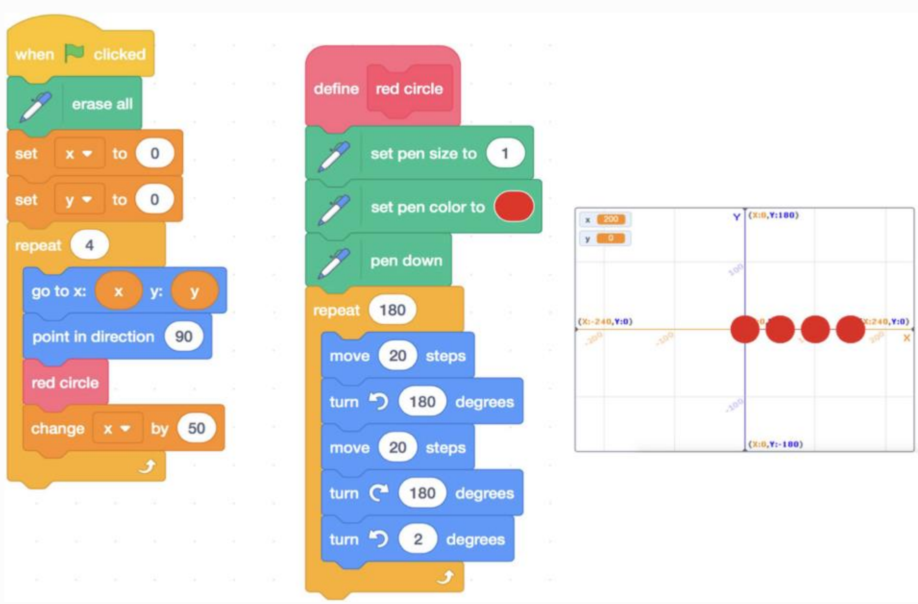


https://scratch.mit.edu/projects/752671060/fullscreen/

GÖTEBORGS UNIVERSITET

Kilhamn, C., Bråting, K., Helenius, O., & Mason, J. (2022). Variables in early algebra – exploring didactic potentials in programming activities. *ZDM - Mathematics Education, 54*(6), 1273–1288

36



GÖTEBORGS UNIVERSITET

Kilhamn, C., Bråting, K., Helenius, O., & Mason, J. (2022). Variables in early algebra – exploring didactic potentials in programming activities. *ZDM - Mathematics Education, 54*(6), 1273–1288

37

How can our research inform practice and curriculum development? What are the potentials and challenges?

1. Describe the knowledge to be taught clearly
2. Make sure teachers have necessary competence
3. Supply teachers with research-based teaching materials and tasks

Give *computational thinking and programming* status as a field of knowledge in its own right with applications outside mathematics (where mathematics is a tool for programming).

For programming to be useful in mathematics – you need to know quite a lot of mathematics and quite a lot of programming.

Use programming where it is really useful: handling large sets of data!

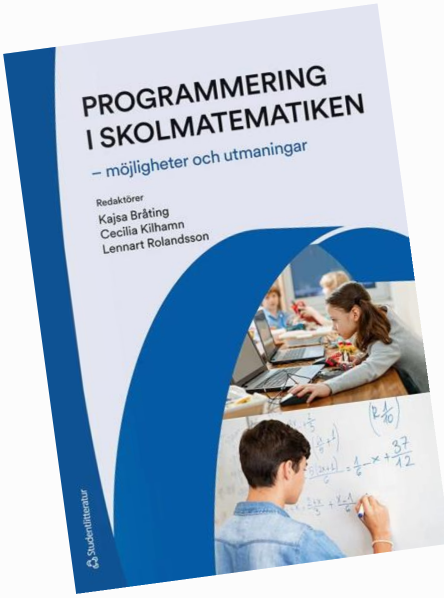
The future — AI

What knowledge will students need?

Chat GPT is good at *producing* code and *explaining* code – students need to tinker, debug, remix …

What are the future practices?

How much will be done in specific programming languages?



40

Journal articles

Bråting, K. (2023) From Symbolic Manipulations to Stepwise Instructions: A Curricular Comparison of Swedish School Algebra Content over the Past 40 Years, *Scandinavian Journal of Educational Research, 67*(2),225–239, DOI: 10.1080/00313831.2021.2006301

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning, 23*(2), 170–185. doi.org/10.1080/10986065.2020.1779012

Bråting, K., & Kilhamn, C. (2022). The integration of programming in Swedish school mathematics: investigating elementary mathematics textbooks. *Scandinavian Journal of Educational Research, 66*(4), 594–609. doi.org/10.1080/00313831.2021.1897879

Elicer, R., Tamborg, A., Bråting, K., & Kilhamn, C. (2023) Comparing the integration of programming and computational thinking into Danish and Swedish elementary mathematics curriculum resources. *LUMAT 11*(3), 77–103. doi.org/10.31129/LUMAT.11.3.1940

Kilhamn, C., Bråting, K., Helenius, O., & Mason, J. (2022). Variables in early algebra – exploring didactic potentials in programming activities. *ZDM - Mathematics Education, 54*(6), 1273–1288. doi.org/10.1007/s11858-022-01384-0

Kilhamn, C., Rolandsson, L. & Bråting, K. (2021). Programmering i svensk skolmatematik. *LUMAT – International Journal on Math, Science and Technology Education, 9*(1), 283–312. https://journals.helsinki.fi/lumat/article/view/1457

Nyman, R., Bråting, K., & Kilhamn, C. (2024). Can programming support mathematics learning? An analysis of Swedish lower secondary textbooks. *International Journal of Mathematical Education in Science and Technology*.(first online) https://doi.org/10.1080/0020739X.2024.2329345

41

Conference papers

Bråting, K., Kilhamn, C., & Rolandsson, L. (2021). *Integrating programming in Swedish school mathematics: Description of a research project.* In: Y. Liljekvist, L. Björklund Boistrup, J. Häggström, L. Mattsson, O. Olande, H. Palmér (eds.), Sustainable mathematics education in a digitalized world. Proceedings of MADIF12, 101–110. Gothenburg: SMDF. https://www.diva-portal.org/smash/get/diva2:1432043/FULLTEXT02.pdf

Kilhamn, C., Bråting, K., & Rolandsson, L. (2021). *Teachers' arguments for including programming in mathematics education*. In: G. A. Nortvedt, N. F. Buchholtz, J. Fauskanger, F. Hreinsdóttir, M. Hähkiöniemi, B. E. Jessen, …, A. Werneberg (red.), Bringing Nordic mathematics education into the future. Preceedings of NORMA 20 The ninth Nordic Conference on Mathematics Education, 169–176. Gothenburg: SMDF. https://uu.diva-portal.org/smash/get/diva2:1541426/FULLTEXT01.pdf

Kilhamn, C. (2022). *Tinkering in algebra - the case of John.* Proceedings of the Twelfth Congress of the European Society for Research in Mathematics Education (CERME12), Feb 2022, Bozen-Bolzano, Italy.

Kilhamn, C., & Bråting, K. (2019). *Algebraic thinking in the shadow of programming*. In: U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (red.), Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education, CERME11 (pp. 566–573). Utrecht, the Netherlands: Freudenthal Group & Freudenthal Institute, Utrecht University and ERME. https://hal.archives-ouvertes.fr/hal-02429028

Misfeldt, M., Jankvist, U.T., Geraniou, E., & Bråting, K. (2020). *Relations between mathematics and programming in school: Juxtaposing three different cases.* In A. Donevska-Todorova, E. Faggiano, J. Trgalova, Z. Lavicza, R. Weinhandl, A. Clark-Wilson & H-G. Weigand (red.), In: Proceedings of the 10th ERME topic conference on mathematics education in the digital era, MEDA 2020, (pp. 255–262). Linz, Australia: Johannes Kepler University. https://www.diva-portal.org/smash/get/diva2:1472841/FULLTEXT01.pdf

Zhang, L., Nouri, J., & Rolandsson, L. (2020). *Progression Of Computational Thinking Skills In Swedish Compulsory Schools With Block-based Programming.* In: Australasian Computing Education Conference (ACE'2020). Melbourne, Australia. https://doi.org/10.1145/3373165.3373173

Book

Bråting, K., Kilhamn, C., & Rolandsson, L. (red.) (2021). *Programmering i skolmatematiken – möjligheter och utmaningar.* Studentlitteratur. https://www.studentlitteratur.se/kurslitteratur/lararutbildning-och-pedagogik/amnesdidaktik-f-6/programmering-i-skolmatematiken

Book chapters

Bråting, K., Kilhamn, C., & Rolandsson, L. (2022). Programming and mathematical competencies: the case of Sweden. In: U.T. Jankvist & E. Geraniou (eds.), *Mathematical Competencies in the Digital Era from a Networking of Theories Point of View* (pp. 293–310). Springer. (First Online: 11 October 2023).

Haspekian, M., Kieran, C., Drijvers, P., Bråting, K., Tabach, M. (2023). Algebra Education and Digital Resources: A Long-Distance Relationship? In: Pepin, B., Gueudet, G., Choppin, J. (eds), *Handbook of Digital Resources in Mathematics Education*. Springer International Handbooks of Education. Springer. https://doi.org/10.1007/978-3-030-95060-6_16-1

Rolandsson, L., Kilhamn, C., & Bråting, K. (2023). Transposition of Computing and Programming Knowledge: The Swedish Upper Secondary School during the 1970s and 1980s. In: J. Hallström & M. de Vries (eds.), *Programming and computational thinking in technology education: Swedish and international perspectives* (pp. 38–59). Brill.

Tamborg, A., Elicer, R., Bråting, K., Geraniou, E., Jankvist, U.T. & Misfeldt, M. (2023). Politics of computational thinking and programming in mathematics education: Comparing curricula and resources in England, Sweden and Denmark. In: B. Pepin, G.Gneudet & J. Choppin (eds.), *Handbook of digital (curriculum) resources in mathematics education*. Springer. (First Online: 16 September 2023)