

# Wie is er bang voor Computational Thinking?

Erik Barendsen – Radboud Universiteit & Open Universiteit

*Minisymposium ter ere van proefschriftverdediging Nataša Grgurina, 28 oktober 2021*



Radboud Universiteit 

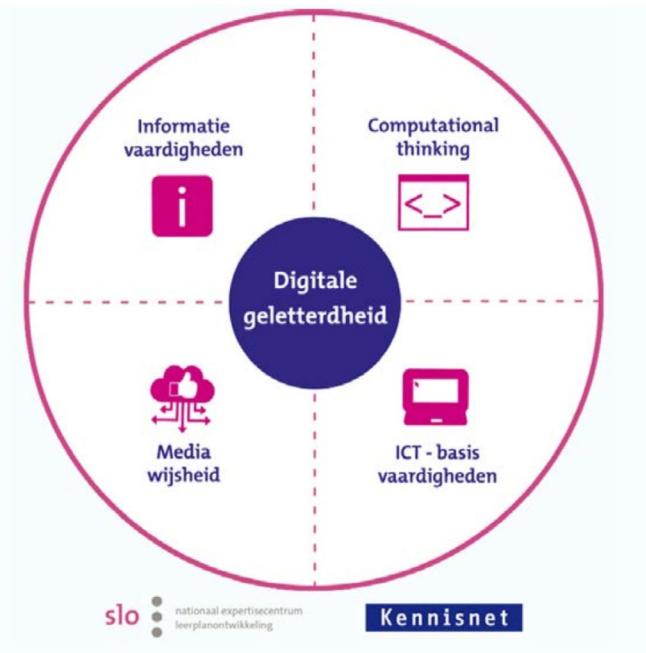
## Wie is die man?

- Erik Barendsen
- Radboud Universiteit en Open Universiteit
- Bètadidactiek, informaticadidactiek



Radboud Universiteit 

## Digitale geletterdheid



Radboud Universiteit

*context*

probleem

reflecteren

toepassing

decontextualiseren

(re)contextualiseren

*CT-inhoud*

computationele elementen

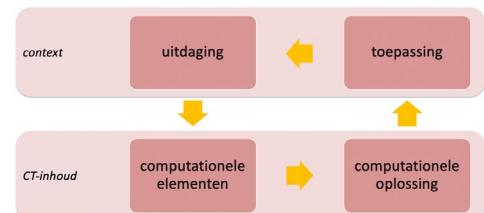
computationaliseren

computationele oplossing

Radboud Universiteit

## CT volgens Selby & Woollard

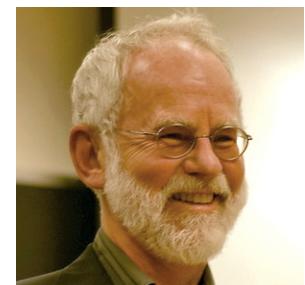
- abstractie
- dekompositie
- algoritmisch denken
- evaluatie
- generalisatie



Veel gebruikt (bv *Computing at School*, Engeland)

## CT concepts en practices volgens Grover & Pea

- **CT-concepten**
  - logica en logisch denken
  - algoritmen en algoritmisch denken
  - patronen en patroonherkenning
  - abstractie en generalisatie
  - evaluatie
  - automatisering
- **CT-praktijken**
  - dekompositie
  - maken van digitale artefacten
  - testen en debuggen
  - stapsgewijs verfijnen (*incrementeel ontwikkelen*)
  - samenwerking en creativiteit (*verbinding met 21<sup>ste</sup>-eeuwse vaardigheden*)



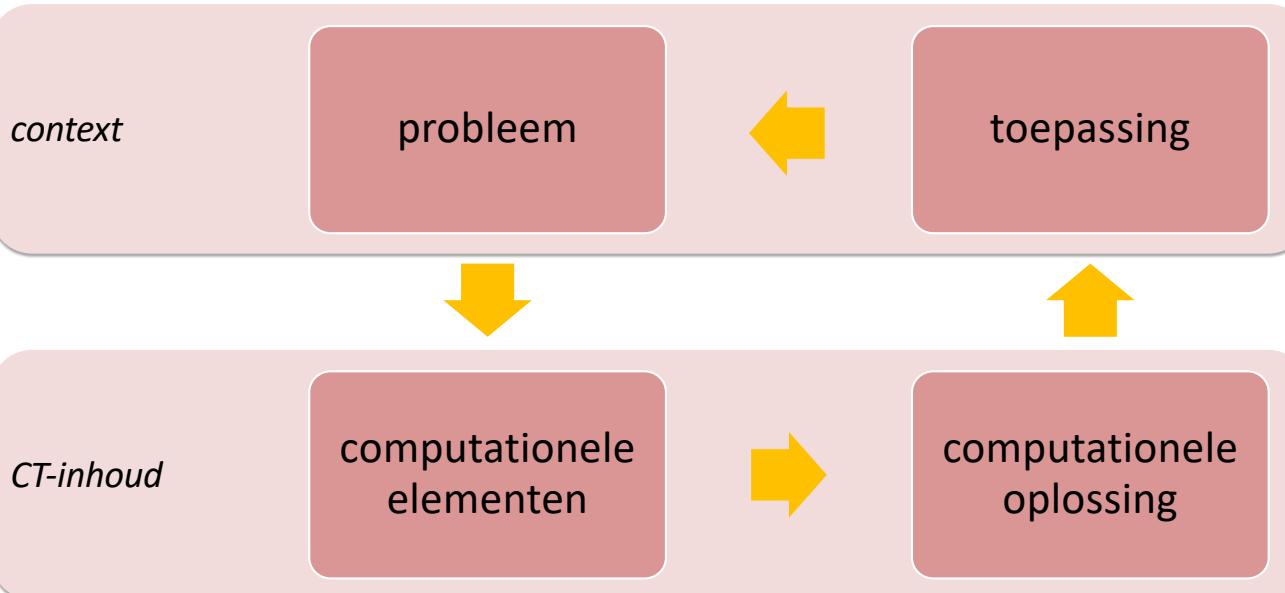
## CSTA-definitie

CT is a problem-solving process that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them
- Logically organizing and analyzing data
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking (a series of ordered steps)
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
- Generalizing and transferring this problem-solving process to a wide variety of problems



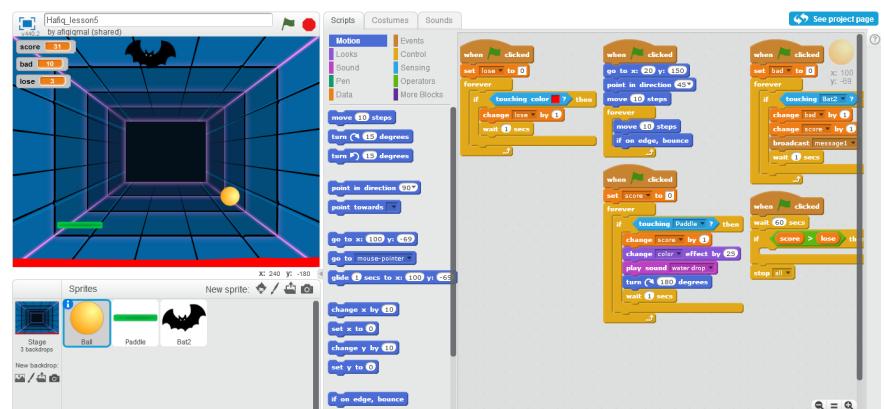
**Topics:** data collection, data analysis, data representation, problem decomposition, abstraction, algorithms & procedures, automation, simulation, parallelization



```

5 public class Main {
6     public static void main(String[] args){
7         Scanner scan = new Scanner(System.in);
8         int five = 5;
9         Integer number = new Integer(five);
10        char[] letters = {'a', 'b'};
11        String[] words = {"apple", "juice"};
12
13        for (String string : words) {
14            System.out.println(string);
15        }
16    }
17 }

```



## Computational thinking met...

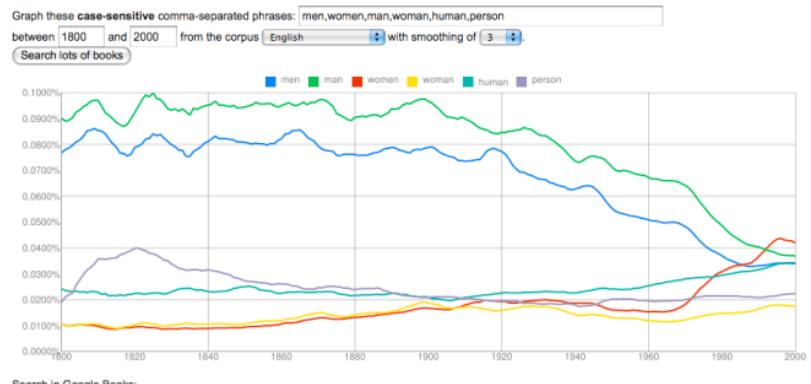
1. Bestaande digitale hulpmiddelen
2. Programmeren
3. Modelleren



## 1. Bestaande digitale hulpmiddelen

- Google N-gram viewer

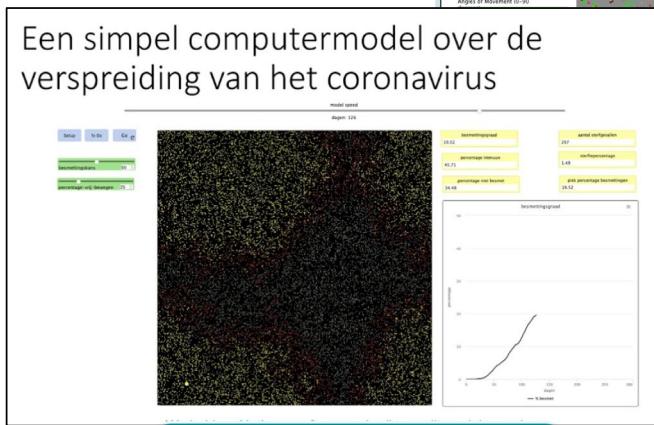
Google labs Books Ngram Viewer



Radboud Universiteit

## 1. Bestaande digitale hulpmiddelen

- Simulaties



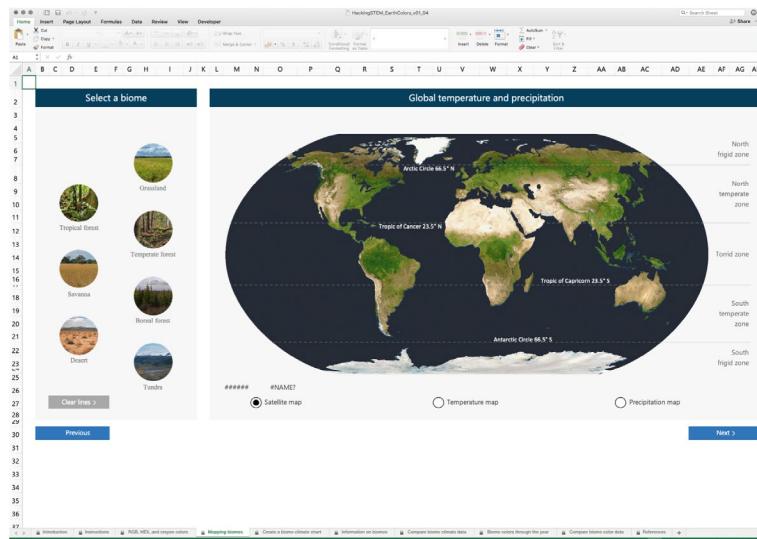
Radboud Universiteit

'Grgurina  
inside'



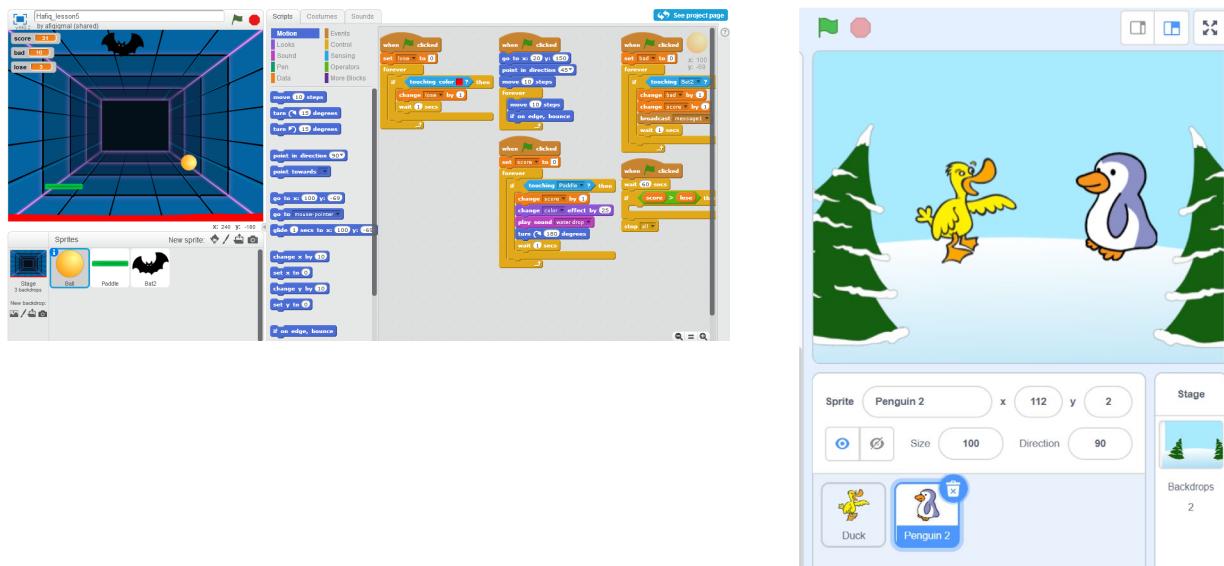
## 1. Bestaande digitale hulpmiddelen

- Excel



Radboud Universiteit 

## 2. Programmeren



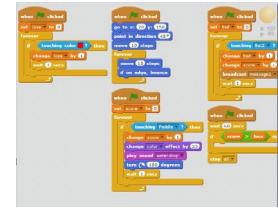
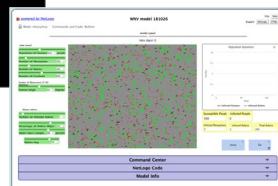
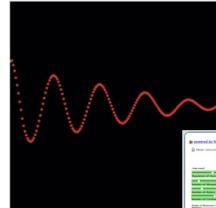
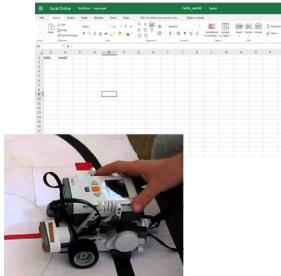
Radboud Universiteit 

## 2. Programmeren

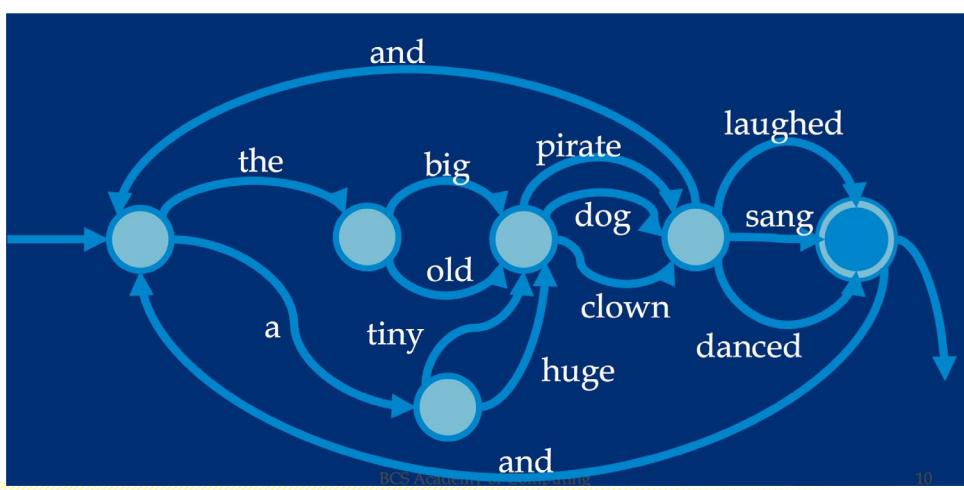
programmeertalen

domeinspecifiek

universeel

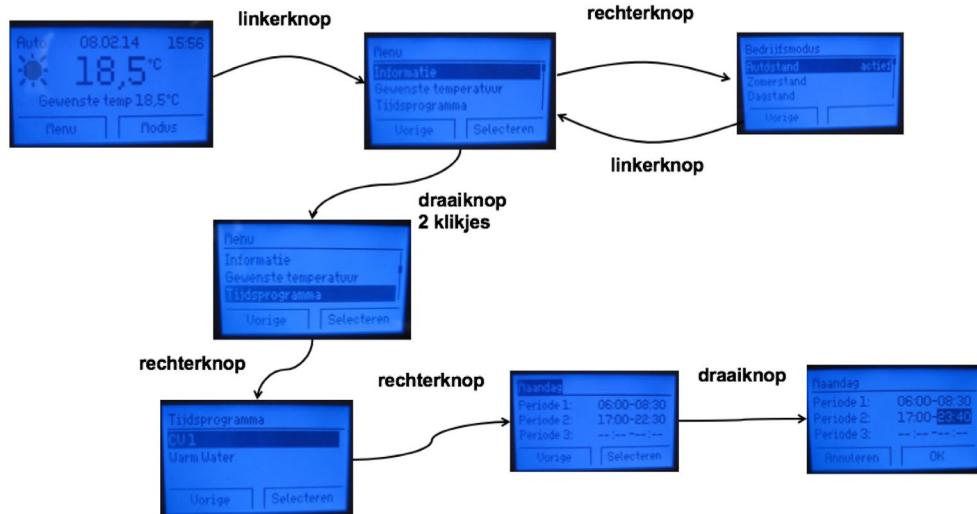


## 3. Modelleeren



met dank aan Simon Peyton Jones

### 3. Modelleren



met dank aan Erik Poll

### 3. Modelleren

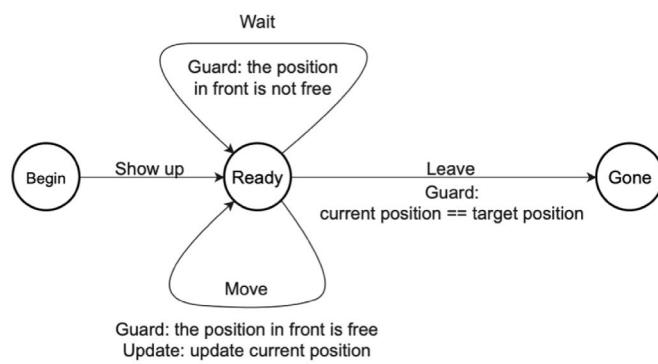
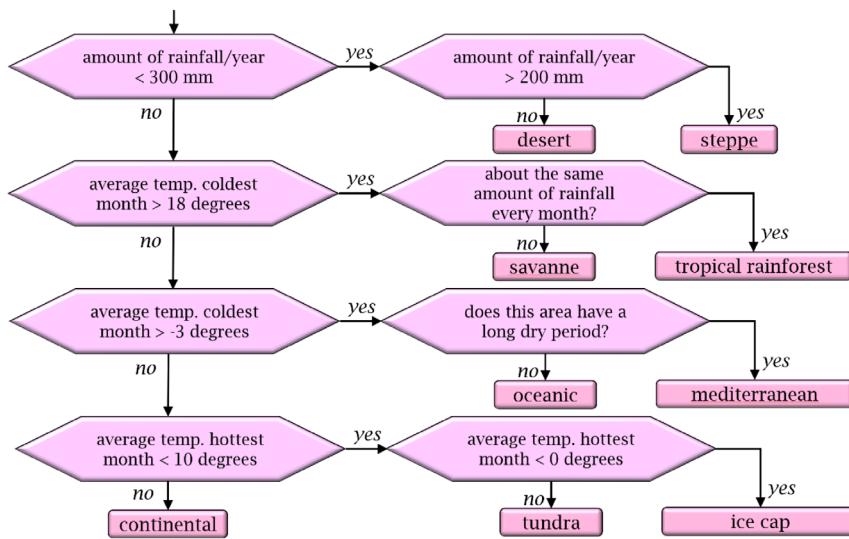


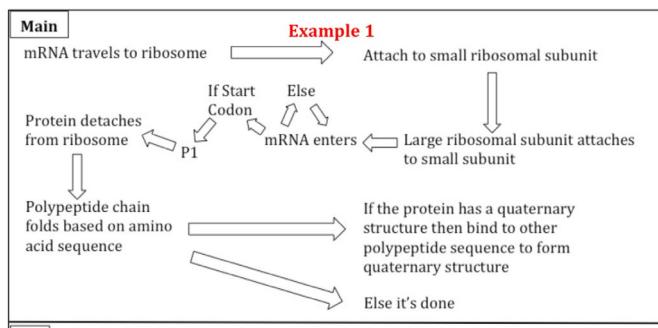
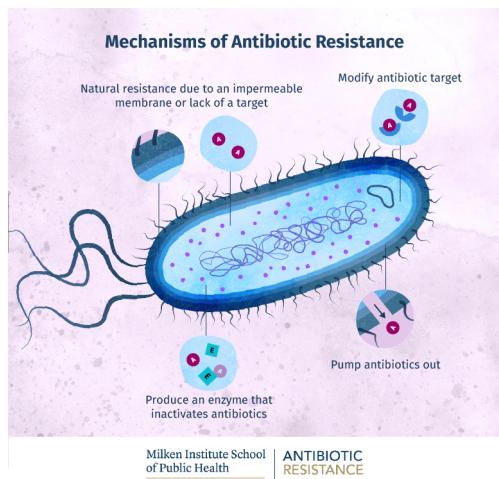
Figure 4: State diagram for vehicle



### 3. Modelleren



### 3. Modelleren



## Wie is er bang voor Computational Thinking?

