# A CONTEXT-BASED INTRODUCTION TO FINITE STATE AUTOMATA IN SECONDARY EDUCATION

Final Version

Supervised by:

Prof. dr. E. Barendsen

S. (Michel) Kok (s4470613)
s1.kok@student.ru.nl

# Table of Contents

# Abstract

The Dutch government has recently accepted a proposed exam program for Computer Science in secondary education. One of the new topics in this exam program is Finite State Automata (FSA). In this thesis, we investigate how we can introduce this theoretical topic, which is often considered hard, in upper secondary education. To answer this question, we have created a pilot lesson of 90 minutes in which the students get acquainted with the basic concepts of FSAs. This lesson consists of a teacher instruction, PowerPoint presentation and several assignments.

Science education is increasingly being designed in a so-called context-based way. In correspondence with this, the pilot lesson was designed in a context-based way as well. Furthermore, neo-Piagetian theories of cognitive development were used to design the assignments.

The pilot lesson was given in a class of sixteen students and was evaluated by analyzing the solutions of individual assignments, the answers on a student survey and observations made during the lesson. The students could use the basic concepts of automata in different contexts and some were even able to reason about these concepts. Still, improvements could be made and these led to an improved, second version of the pilot lesson.

# Acknowledgements

I would like to thank the following people:

- Erik Barendsen, who has been my supervisor throughout the process. Always being available to answer questions, give feedback or inspire me with new ideas he has been a great help to me.
- Jacqueline Nijenhuis-Voogt, who (from a teacher's perspective) helped constructing the lesson and was willing to give the lesson in one of her classes.

# Introduction

The Dutch government has recently adopted a new exam program for upper secondary education regarding Computer Science. Finite State Automata (FSA) is a new topic in this program and it is therefore important to investigate how FSAs can be instructed effectively to this group of students.

Finite State Automata (FSA) are a vital part of Computer Science and related areas. Processes in the 'outside world' can be modeled by automata. A situation of the world at a certain moment is represented by a circle, that we call a state. An event that changes the described state, is depicted by an arrow that points to the next state (situation in the world). A double circle indicates that we have reached a state in which we can choose to stop the process. We can, for example, represent the process of 'heating water' with Figure 1.

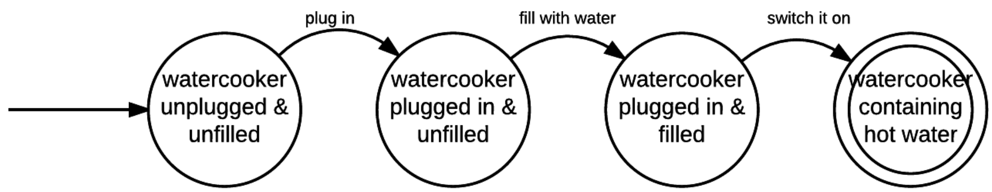



**FIGURE 1: WATER COOKER**

FSAs are part of Theoretical Computer Science and can, for example, be used to reason about the capabilities and limitations of (algorithmic) computation. A typical introductory exercise for undergraduates would be to construct an automaton that models the process of reading a string and determine if the string contains an even number of zeroes. In $s_0$ the number of zeroes is even and in $s_1$ odd. Therefore, we can only choose to stop in $s_0$.

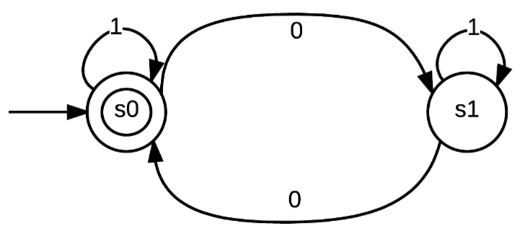



**FIGURE 2**

The official definition of this deterministic finite automaton $M$ is a quintuple, $(Q, \Sigma, \delta, s_0, F)$:

$$Q = \{s_0, s_1\} \qquad \delta(s_0, 0) = s_1$$
$$\Sigma = \{0,1\} \qquad \delta(s_0, 1) = s_0$$
$$F = \{s_o\} \qquad \delta(s_1, 0) = s_0$$
$$\delta(s_1, 1) = s_1$$

Although this is a simple exercise, this already looks more abstract than Figure 1 and the relation with real-world applications is not obvious.

Moreover, when we were studying FSAs, we experienced that undergraduates often find it a though subject. So, studying FSAs gives rise to two problems: the topic is abstract and is considered difficult. These problems may be connected, but this is not necessarily the case.

We will use so-called 'context-based education' to address this first problem. Already more than twenty years ago, some have argued that context-based learning has educational benefits, e.g. it provides practice in applying the fundamental concepts of the disciple, gives students a better understanding of the world and increases the motivation of students (Hellermark & Hollabaugh, 1992). Context-based education keeps the relation of an abstract concept to a corresponding use in the real world clear.

To address the second problem (i.e. studying FSAs is hard), neo-Piagetian theories of cognitive development can be used. These theories describe how knowledge develops in a human brain. Educational instances that design teaching material can, for example, use these theories to make sure that the level of difficulty of an exercise corresponds to the knowledge level of the tested concepts of a student. These theories are used in Computer Science, e.g. by Lister who used neo-Piagetian theories to analyze and improve the teaching of programming in upper secondary education (Lister, 2016).

Several attempts in the case of teaching FSAs to undergraduate students show that an alternative method in a context-rich[1] setting, can be a path to better understanding and higher motivation among students by basing their method on the development of simulators (de Souza, Olivete, Correia, & Garcia, 2015) or via robots with a direct link to coding (Kulyukin et al., 2015). It has even been done via gamification in K-12 education (Isayama, Ishiyama, Relator, & Yamazaki, 2016). This theory used games to mainly test how well the students knew about automata theory.

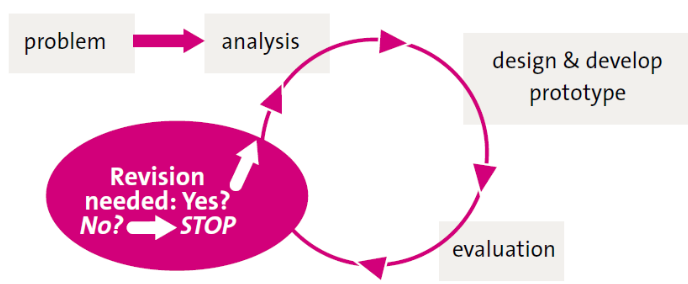



FIGURE 3: EDUCATIONAL DESIGN RESEARCH

However, we will follow a qualitative approach, focusing on how the subject is taught and how the students' understanding develops. We will follow the Educational Design Research strategy as depicted in Figure 3, to develop a pilot lesson that introduces FSAs in upper secondary education. In Part 1 of this thesis, we extend the analysis of the problem by providing theoretical background about context-based education and neo-Piagetian theories of cognitive development. Based on this, we will design and develop a pilot lesson of 90 minutes. In the third part, we will evaluate the first version of the pilot, we will analyze our outcomes and we will develop a second version of the lesson, based on the analysis.

[1] This is slightly different than 'context-based'-education

# 1. Background

Before we can design the lesson, we will first discuss how the work of Jean Piaget regarding his theory of cognitive development has developed over time and how this led to the neo-Piagetian theories of cognitive development. Afterwards, we will discuss the history and use of context-based education.

## 1.1 (Neo-)Piagetian theories of cognitive development

Piaget, a theoretical epistemologist, is known for his theory about the development of human cognition. He came up with four stages in which knowledge is developed. Piaget's theory turned out to be incomplete and not without fallacies. However, Piaget did change our view on children forever, seeing them as cognitive beings instead of imperfect adults. Furthermore, and much more useful for this research, his theory continued to exist, but in a rather different way as he himself probably intended. Although, nowadays, 'Jean Piaget's work is not in fashion' (Lister, 2016, p. 7), it for example directed education in two respects:

> "(a) the development of new teaching methods that would capitalize on the exploratory and inventive activities of the child himself; (b) the strengthening of the teaching of specific school courses, particularly in science and mathematics, by cultivating and consolidating the basic thought structures of scientific and mathematical thinking" (Demetriou, Shayer, & Efklides, 2016, p. 1).

Moreover, much research in other, scientifically related areas has been done that led to the so-called neo-Piagetian theories of cognitive development; theories based on Piaget but without its problems and limitations. These neo-Piagetian theories can also be described as theories that "try to preserve the strengths of Piaget's theory while eliminating its weaknesses" (Demetriou et al., 2016, p. 2). Because of the enormous scope of Piaget's work, a lot of different theories have been developed, either approving or disapproving of Piaget's initial work (Barrouillet, 2015). We will not discuss all differences between Piaget and later researchers, but we discuss those that are important for our research.

Whereas the work of Piaget regarded the development human cognition in general, his stages appeared to apply to the cognitive development of any task at any age. A second change is found in the transition of stages. Piaget stated that humans go from one stage to another via a rapid change, analogous to Thomas Kuhn's paradigm shifts. This change is often analyzed via a staircase metaphor: the stages are compared to the treads and children slowly climb up the stairs. It is important to note that setting a step is just a small move, whereas they are believed to stand still on a tread for a long time. This part of his theory turned out to be problematic and many have argued that many more factors need to be taken in account and that stages, substages and transitions between the stages needed more research (Feldman, 2004). Currently, we see researchers using the overlapping waves metaphor that was developed almost twenty years ago (Siegler, 1998) which shows that stages changes rather gradually and that stages overlap when one comes close to reaching the next stage. Over the years the overlapping waves model has been improved and expanded (Boom, 2015) and this improved model was able to explain empirical data when teaching programming (Lister, 2016).

### 1.1.1 An application in the teaching of programming

A recent example of a successful usage of neo-Piagetian theories of cognitive development in Computer Science education is given by Lister. This table summarizes the initial theory of Piaget and how Lister uses Piaget to formulate the stages in terms of programming.

| | **Piaget** | **Lister** |
|---|---|---|
| **Sensorimotor** | In this stage children develop the ability to organize and co-ordinate their sensations and perceptions with physical movements and actions, and they also operate with a primitive symbol system. | The novice programmer has an incoherent understanding of program execution. |
| **Preoperational** | An increase in their capacity to use language and perceptual images and they can think symbolically. However, several limitations cause a child's thought to fall short of what is seen in later years. These limitations include child's egocentrism, inability to conserve, failure to order objects in a series, and classify them in groups. | The novice can reliably manually execute ("trace") multiple lines of code. These novices often make inductive guesses about what a piece of codes does, by performing one or more traces, and examining the relationship between input and output. |
| **Concrete Operational** | The limitations of preoperational stage disappear. Another change is the child's ability to operate mentally on a series of actions (logic). Children recognize that their perspective and knowledge differs from others. One limitation of concrete thinking is that the child needs to be able to perceive objects and events that he will think about. | The novice programmer reasons about code deductively, by reading the code itself, rather than using the preoperational inductive approach. This stage is the first stage where students show a purposeful approach to writing code. |
| **Formal Operational** | Reasoning gains abstract and hypothetical abilities. The child reaches the most advanced level of thinking. | This is the ultimate stage of Piagetian reasoning. It is the level at which the expert performs. A persons thinking at the formal operational stage can reason logically, consistently and systematically. |

**TABLE 1: PIAGETIAN STAGES**

## 1.2 Context-based education

In this section, we will provide a short overview of recent developments regarding context-based education. Then, we will elaborate on this type of education and we will discuss the benefits of it.

### 1.2.1 Historical developments

In 2003 the Koninklijke Nederlandse Akademie van Wetenschappen (KNAW) published two articles leading to new and adapted curricula in secondary school education that introduce context-based education in science education (KNAW, 2003a, 2003b). At the time, they were motivated by the changing academic structure to bachelor/master structure.

This new type of curricula has three benefits: it focusses on core concepts instead of ever increasing the knowledge in the world and on exam programs. Furthermore, it increases insightful learning. And thirdly, it interconnects different subjects and therefore improves the general cohesion in education (Bruning & Michels, 2013, p. 15).

Moreover, the advisory committee regarding the exam program of Computer Science in the Netherlands stresses the importance of contexts in education, for example, in 'design and development', one of the crucial and characteristic abilities of Computer Science (Barendsen & Tolboom, 2016, pp. 11–12). This type of education, often called context-based education[2], comes from an international change in educational standards that focusses much more on contexts that we were used to, and is increased by the new curricula in science education. When developing the pilot lesson, we follow this advice, since it is very accurate, up-to-date and written for the long term.

### 1.2.2 Concept-context approach

Concept-context approach is the type of context-based education. Contexts and concepts are related to each other and are both core aspects of teaching. Using contexts in education we teach the concepts and, vice versa, concepts become relevant only in a certain context.

#### Concepts

Again, to follow the Dutch educational direction we will not come up with alternative definitions of concepts and contexts but define and use those as they have been used in the past years when the new context-based education was introduced in science education. Concepts are mental images that refer to important ideas in a scientific area (Boersma, Eijkelhof, van Koten, Siersma, & van Weert, 2006). It is important to note that concepts need to be used by students in an agile way, i.e. in multiple contexts (Boersma et al., 2006; Bruning & Michels, 2013, p. 21).

A scientific area can be divided in certain core concepts and underlying concepts based on these core concepts. There is no clear or rational way to do this (yet) and it is probably most of the time based on tradition (Bruning & Michels, 2013, p. 22). Concepts can be chosen internally or externally based on who chooses the concepts that matter. Internally, this is done by teachers and authors of educational material. Externally, they can be established in the exam program as is done in the Netherlands. The advisory committee state that, in order for their advice to maintain up-to-date on the long term, they have constructed the exam program strongly conceptually (Barendsen & Tolboom, 2016).

[2] There is a lot of confusion when people talk about context-rich, context-based or concept-context education. We do not want to confiscate this even more by coming up with a new term and definition, but when we talk about context-rich or context-based education we use the so-called limited notion of context-based (science) education (Bruning & Michels, 2013).

### Contexts

A context is defined as the environment in which the learning takes place: a situation or problem that is relevant for students or becomes relevant for them by performing the learning activities (Boersma et al., 2006). Contexts have two functions, a didactical and a functional one. The former tries to enhance the learning process whereas the latter is used because of their importance for the students (it is important to defend why and how a situation is relevant for a certain group of students) (Boersma et al., 2006).

There are multiple reasons to work with contexts. Contexts show the relation between different scientific areas, improve the image of science among students, improve meaningful learning, improve motivation and attitude of students (Bruning & Michels, 2013). Some of these benefits make it clear why they are used in the new curricula in the Netherlands, since they are almost identical.

Contexts can be categorized in the following four groups: real-world, society, profession, academic (Bruning & Michels, 2013, p. 17). For our pilot lesson, only the first three are worth to look at, since academic contexts require a high level of abstract thinking and a good understanding of the topic.

We also pay attention to Gilbert, who suggested in 2006 that his Model 4 was probably going to be the best and proved empirically five years later that Model 4 was indeed the best model to tackle all problems that chemistry, but as stated computer science in a certain way as well, education face, nowadays (Gilbert, Bulte, & Pilot, 2011). His fourth model *Context as the social circumstances*:

> In this view, the social dimension of a context is essential. A context is situated as a cultural entity in society. It relates to topics and people's activities that are considered of importance to the lives of communities within the society (Gilbert, 2006, p. 969).

## 1.2.3 Benefits

Context-based education has several advantages. First, knowledge is only useful when one can use it in a certain context. Especially in academic regions, it is not always clear how the work that is done relates to real-world applications. To use contexts in teaching already, improves the direction in which knowledge is developed.

In other scientific areas, much research has been done with regards to context-based education, especially in chemistry. Gilbert claims that the use of contexts in education solves one or more of five problems that chemistry education was and is facing (Gilbert, 2006). We think that especially the following three are (becoming) more and more relevant in Computer Science as well:

1. Overload: because of the ever-accelerating accumulation of scientific knowledge;
2. Isolated facts: students are not taught how connections should be formed between different parts of the field or different 'isolated facts';
3. Lack of transfer: students could solve problems presented to them in a way that closely mirrored the problems that they were taught to solve. However, they struggled solving different problems although the problems could have been solved with the same conceptual knowledge.

Lastly, context-based education increases motivation of students. Keller explains the scientifically difficult term of 'motivation' by dividing it in sections indicated by the letters A, R, C and S. 'A' stands for *Attention* which states that students' attention (as part of the motivation) is increased by posing questions or problems. Context-based education provides recognizable problems and will therefore increase the motivation of students. Another aspect of the model is 'R', which stands for *Relevance*, which has a subdivision which stresses the importance of context (Keller, 1987). In this way aims to fulfill all aspects of the ARCS model, causing students' motivation to increase.

# 2. Towards a context-based introduction to FSAs

## 2.1 Conceptual goals

The conceptual goals of this lesson will be constructed with help of the neo-Piagetian theories of cognitive development.

### 2.1.1 Piagetian stages and FSAs

Since Piaget only described lifetimes, we must 'redefine' the content of the several stages when we want to apply his theory to other learning tasks. We have reformulated the four stages of Piaget, specific for teaching FSAs, based on the standard definition by Piaget and inspired by the stages of Lister as follows:

1. **Sensorimotor**: the student has an incoherent understanding of automata. This is the stage that the students are assumed to be in at the beginning of the lesson.
2. **Preoperational**: children do not yet understand logic, but they do make inductive guesses and can come up with solutions by trial and error. This means for our case that the student can create simple automata, but those automata will often be inefficient or incomplete. Furthermore, these novices often make inductive guesses about what automata do, by performing one or more traces.
3. **Concrete operational**: in Piagetian theories children now understand logic. For our case, this means that they can reason about their automata and can come up with more complete or efficient automata. At this stage students also learn to interconnect disciplines, so they better understand the purpose of their doing. From Lister, we learn that especially the way a solution is found distinguishes this stage from the previous.
4. **Formal operational**: at this final level, the student has become an expert in automata theory. In neo-Piagetian theories, this is where abstract thinking is developed and that is exactly the case here too. Students learn to abstract from their automata to mathematical and computational implementations. This state is not even reached in undergraduate classes where FSAs are taught, so it's safe to assume that we can ignore this stage when constructing goals for the pilot lesson.

Since we assume no foreknowledge whatsoever, the student finds himself in the first stage at the start of the lesson. We want at least the whole class to progress from the first to the preoperational stage during the lesson. Some, might even reach or come very close towards the third stage.

In terms of the Piagetian stages for FSAs, we want the whole class to reach the preoperational stage and some of the students to have reached the concrete operational stage. To formulate the conceptual goals for this lesson we also consider the advisory report on the exam program of Computer Sciene (Barendsen & Tolboom, 2016). In the spirit of context-based education (section 1.2), they describe the domain of Computer Science in five knowledge-domains of which *Fundamentals*[3] is one. In *Fundamentals,* we find automata as a subpart of this domain. The goal that was set by this committee was that students can use finite state automata to characterize of certain algorithms.

[3] In Dutch: Grondslagen

To accomplish this, the students will learn the basic concepts of automata: state, start state, end or final state, transition, automata. These are only necessary for the students to be able to construct and understand automata. We also introduce them to concepts that are typical for algorithms: variables and conditions. These are very basic concepts in algorithms, but can also be used in constructing modelling. These concepts relate automata and algorithms and are therefore a vital part of this lesson. For further justification of these concepts see section 1.3.

Based on the concepts we redefine the learning goals as follows:

1. All students can trace simple finite state automata that model real-life situations. 'Simple' means that it only contains the taught concepts and that it clearly relates to a given context;
2. A. All students understand the difference between states and transitions in reading and tracing an automaton;
   B. Some students understand the conceptual difference between states and transitions;
   C. Some students will be able to construct automata efficiently (i.e. least amount of states and transitions);
3. A. All students can update a variable on a transition to use the model;
   B. Some students update variables efficiently;
4. A. All students can read and interpret conditions in an automaton;
   B. Some students can use (e.g. introduce, change) conditions in an automaton;
5. All students can explain why automata are important in Computer Science.

## 2.2 Design approach

We see in the Netherlands a strong focus on skills when we teach foreign languages. In computer science, it is more difficult to focus so strongly on skills only, since we do not wish to merely put it in practice but we also encourage academic reasoning about the theory. We do not wish students to be able to work with computers and machines only, but we want them to start understanding the underlying concepts of this scientific area as well. Therefore, we follow the direction of the Dutch educational system and use the concept-context approach as the type of context-based education.

When developing a lesson series or a single lesson, we need to answer two questions: what determines the content selection of the material and what determines how this is taught (Bruning & Michels, 2013). On both questions, we can either choose concept or context.

When developing the pilot lesson, we choose to let the conceptual structure lead us when we determine the content as we have done in section 1.1. FSAs in general are used to learn how computers work and they describe the underlying structure, the 'concept' one might say. From the current exam program, we cannot find any concept regarding FSAs. However, in the advice for the exam program we find that a candidate must be able to use finite automata to characterize certain algorithms. Their choice to include FSAs is because artefacts can often be described and designed in an elegant and simple way. They should be used instrumentally (Barendsen & Tolboom, 2016, p. 15). Therefore, FSAs can be seen as alternatively drawn UML state diagrams. It also relates closely to model checkers, especially if we introduce variables.

For FSAs the traditional concepts would include states, alphabets, words or strings, languages, transitions and so on. However, since we only want to use FSAs as an instrument, we do not need to introduce the students to these concepts, but only to those that are connected to algorithms. The concepts we chose have been mentioned in section 2.1.

Now, to determine how the material is created, we let the context determine this. The exam program is formulated conceptually and is rather vague in their formulation, leaving a lot of room for the teacher or writer of teaching material. We have chosen for the context because of the previously explained arguments on context-based education. This means that we choose the didactical function of contexts as opposed to the functional function. So, we create a lesson in which we first choose which concepts are to be taught. Based on those concepts we choose a context that covers the concepts and the chosen context will determine the exact material.

### 2.2.1 Choosing a context

Before determining the context, it is important to stress once more that FSAs are used instrumentally and are therefore suitable to be taught in overarching contexts, e.g. when teaching algorithms. We can also teach it as a separate topic in which algorithms should be developed, but only insofar as that automata must be constructed of the algorithm.

We have chosen the professional context of *Working as a consultant for McDonalds* since McDonalds is a company well-known to almost anybody nowadays. Some students may have a part-time job and since McDonalds employs young people, they can relate to working there. Since working at McDonalds as a part-timer is generally not considered a popular job and most students work there for financial reasons only, we chose for the context of a consultant who gets hired by McDonalds. This is more appealing as well since the chance of becoming a consultant or scientist is higher than being an employee in a McDonalds among students of this type of education.

The context is suitable for one lesson, since it needs relatively little introduction. Every city center has multiple McDonalds and everyone has probably been in one at least once in their life. The context is also suitable for future extensions of this pilot lesson, if it appears to be successful. This can be done since McDonalds has a lot of different sides or by simply working for another company as this is relatively easy for a consultant to do.

## 2.3 Design and justification

To develop the lesson, we used a model known as *Direct Instruction* (“Direct Instruction,” n.d.). Using this, we divided the lesson in several parts. We will discuss why a part matters and how it contributes to our goal of developing a context-based introductory lesson for teaching FSAs. The result of this design is the lesson as can be found in the appendices A, B and C. At the end of this section a table can be found in which the activities of the students and teacher are

| Stage | Name | Activity |
|---|---|---|
| **1** | Introduction/ Review | Set the stage for learning |
| **2** | Development | Model the expected learning outcomes by providing clear explanations and examples |
| **3** | Guided Practice | Monitor and engage pupils with assigned learning tasks |
| **4** | Closure | Bring the lesson to a conclusion by highlighting what was covered |
| **5** | Independent Practice | Provide learning tasks that are independent of teacher assistance |
| **6** | Evaluation | Assess pupil progress |

**TABLE 2: DIRECT INSTRUCTION**

summarized along with the Direct Instruction phase (see Table 2) and the concepts that are introduced per part of the lesson.

The lesson is developed in two cycles of this model. In the first cycle the students are concerned with modelling the behavior of customers and their payment methods. The second cycle shifts the focus from creating the model, to practically using the model to collect the data.

## Introduction

Normally, a lesson would start by looking back on previous lessons, recalling learned concepts and knowledge that is required for the current lesson. However, this lesson assumes no knowledge regarding FSAs and this is an introductory lesson, so we skip this step of retrospection.

We start off by introducing the lesson to the students. They are told what the goal of this lesson is: creating models by constructing automata. We leave out any conceptual goals or terminology. We summarize the learned concepts at the end of the lesson.

After the general introduction, the teacher explains why modelling is important as a part of the *Development* phase. The teacher mentions that modelling or constructing automata[4] is important for computer scientists in the field of programming, especially while working with algorithms. This is deliberately stressed, because, as argued before, in the Netherlands the students need to study FSAs in their curriculum only in their relationship to algorithms (Barendsen & Tolboom, 2016).

Before we start with FSAs, we first present them with a different kind of model. This model suits the purpose of explaining why models matter well. The model is simple and with no foreknowledge one can still understand what the model is about.

The students are afterwards introduced to the context of working for McDonalds as a consultant and are confronted with an automaton in this context. The teacher explains how the automaton reflects the behavior of the customers. This introduces the students to concepts like start state, final state, state and implicitly to transition but it also introduces them indirectly, deliberately done incorrectly, to the possibilities of loops and multiple transitions to or from one state.

## Classroom assignment I

From the introduction, we move to the *Guided practice* phase (see Table 2). This simplifies the initial model and corrects the error. The students get a better understanding of a state, namely that it is not defined by a single word, but that it is a real situation of the world. They may recognize that waiting for a meal is different than waiting in line to order.

The teacher can ask the students to take 'walks' through the model and thus involving the students. This 'debugging' technique can come in handy for the coming exercises. When alternative paths are taken, students find out that one of the paths does not properly reflect the situation in the real world.

## Classroom assignment II

We remain in the same instruction phase, but the level of difficulty is increased. We increase the level of detail in our model, setting it up for the individual assignments that are to follow. The

[4] Although 'modelling' and 'constructing automata' are different concepts, we can interchange the terms in this lesson because we model by constructing an automaton.

students learn that by adding a simple transition, a lot more can be told. The teacher asks students for solutions, and can ask others to find FSAs in presented solutions or walk the 'path' through the model a few times, to present a way to find errors in this kind of models to the students: walking through the model.

At the end of this assignment we come to *Closure.* Usually, this is used to end a lesson and give the students some individual homework since the *Independent Practice* requires no teacher. However, since we do not have a follow-up lesson, we cannot give them individual assignments to do at home. Therefore, the teacher summarizes what has been done. This can be done shortly, because the teacher will remain available in the classroom during the next phase and the model is the summary of the previously done assignments.

### Individual assignment I

As we just argued, we enter the *Independent Practice* phase with an individual assignment. This assignment adds two states and is in difficulty comparable to the second classroom assignment. This assignment describes a more difficult part of the model. They are presented with some states and are asked to complete this model by placing the transitions. The distinction between states and transitions becomes clear after classroom assignment I and this individual assignment. This exercise also presents an alternative way to construct models for the students: define states and add transitions later.

Although this is the last part of the first cycle, we will use this to assess the students. This exercise should rather be seen as the most difficult exercise of the first half. This exercise requires reasoning about the model, since many things (17 transitions) should be done and the chance of accidentally coming up with 17 correctly placed transitions is very small.

### Classroom evaluation I

The students that have not yet solved the problems have a chance to keep up with the pace of the lesson by seeing the solution presented by other students. This increases the participation of the students. They are also presented with a new strategy. By applying this way of working, students should identify the important parts for a certain exercise, 'ignore' the rest by 'putting it in a box' and then see how the assignment can be done. This strategy comes in handy at this point, since the automaton is getting bigger and bigger.

This automaton resembles the first correctly constructed automaton, with a former state being replaced by a box now. Students can see that this technique works both ways: we can describe a part of the model in much more detail, but on the other hand, we can also reduce the size of a model by creating a box.

There are two other benefits of this strategy. These benefits do not come in handy for our lesson or for this topic only, but for the whole field of Computer Science. The first of these being that students are introduced to the concept of black boxes. Computer Scientists often do not know why a specific part of program or computer works in a certain way, but they know how it functions, i.e. if they provide the right input, they will get a correct, related and predictable output. The second benefit is the following: they learn how to deduce, to create abstract objects. We chose not to elaborate on this and apply this technique multiple times or by using several instances of this box, since that is not a goal of this lesson. Although the students are probably not aware of this, they get a little experience in two core techniques of Computer Science.

### Classroom assignment III

The third classroom assignment brings us back to the *Guided Practice*. This assignment contains of two parts. The first part focusses on the provided strategy for bigger automata and models: find relevant part of the model and then see what exactly should be done. The students have just been introduced with a way to simplify the model by wrapping up several states and transitions in one single box. Now the students must identify the relevant states and transitions themselves. They will find it much easier to solve the exercise once this has been done. The students are presented with the possibility of maintaining a variable in a state and updating this variable at certain transitions. They also know that multiple correct solutions can be found although one may be better, more efficient, than the other.

### Individual assignment II

This exercise asks the students to come up with a new model. This can be challenging since they should change the model fundamentally. Not many states have to be added, but the old model is just a small part of the new model. It sets up the next individual assignment that is primarily focused on variables and it tests the capability of the students to rethink and adapt their model. This situation occurs often in Computer Science when working with models: after creating a model (or some other artifact), you realize that it misses crucial information and that a new way of thinking is required.

### Individual assignment III

Multiple variables have to be added to the model. However, the operations are not very hard: they only increase and they always increase by one. This means that the focus should be on maintaining the right variables and updating them on the correct transition. This assignment is mostly meant to familiarize the students with using variables in models.

### Classroom evaluation II

This second evaluation serves the same purpose as the first. We move to *Closure,* which is in our pilot lesson the last phase, as explained before. Students will present multiple solutions and argue for or against a presented solution, which indicates that they understand the underlying concepts well.

### Individual assignment IV

This exercise goes away from McDonalds and moves towards a different context. Students must be able to apply the learned concepts in different contexts. Only then can knowledge be used, since a situation in life is hardly ever the same as a learned context. This exercise takes a more technological perspective; instead of an ordinary company we know analyze typical Computer Science products. Also, we focus more on updating variables. The operations are a little harder and we introduce conditions on transitions. These conditions can be seen as if-else statements.

This exercise looks like a model checker, FSA and UML state diagram all in one. Although this may be confusing for computer scientists, we deliberately chose for this combination. This combination closely resembles an algorithm and that was the reason to introduce the concept of FSAs in secondary education in the Netherlands.

The question is split in two, with the students only receiving the second exercise if they can solve the first. The first exercise tests how well they understand the given automaton. The last exercise is challenging and expands the given automaton.

### Classroom evaluation III and conclusion

This part closes the lesson by discussing the last exercise in the same way as the previous assignments have been discussed. After this the teacher wraps up the learned concepts and thereby ends this pilot lesson about FSAs.

| Part | Students | Teacher | Concepts | DI - phase |
|---|---|---|---|---|
| **A** | Listen to introduction and answer questions about the displayed models. | Introduces the lesson and context. | State, start state, end/final state, transition, automata. | 1 & 2 |
| **B** | 'Walk' through an automaton, suggest and discuss several solutions. | Introduces the exercise and leads the discussion. | | 3 |
| **C** | Idem | Idem | | 3 & 4 |
| **D** | Do individual assignment I and discuss solution with a neighbor. | Answers questions and allows students to work together after a certain amount of time. | | 5 |
| **E** | Present and discuss presented solutions in the classroom. | Leads the discussion. | | 5 |
| **F** | Find solutions for the problem together. | Introduces the exercise and leads the discussion. | Variable | 3 |
| **G** | Do individual assignment II and discuss solution with a neighbor. | Answers questions and allows students to work together after a certain amount of time. | | 5 |
| **H** | Do individual assignment II and discuss solution with a neighbor. | | | 5 |
| **I** | Present and discuss presented solutions in the classroom. | Leads the discussion. | | 4 |
| **J** | Do individual assignment IV. | Answers questions. | Condition | 5 |
| **K** | Present and discuss presented solutions in the classroom. | Leads the final discussion and summarizes the lesson. | | 4 |

**TABLE 3: LESSON PLAN**

# 3. Evaluating the given lesson

In the third part of this thesis we will evaluate the constructed lesson. Several (sub-)questions make clear if we have constructed an effective lesson to teach FSAs to students in secondary education. The evaluation leads to a second version of the pilot lesson.

## 3.1 Research questions

We set the conceptual goals as follows:

a) All students can trace simple finite state automata that model real-life situations. 'Simple' means that it only contains the taught concepts and that it clearly relates to a given context;
b) A. All students understand the difference between states and transitions in reading and tracing an automaton;
   B. Some students understand the conceptual difference between states and transitions;
   C. Some students will be able to construct automata efficiently (i.e. least amount of states and transitions);
c) A. All students can update a variable on a transition to use the model;
   B. Some students update variables efficiently;
d) A. All students can read and interpret conditions in an automaton;
   B. Some students can use (e.g. introduce, change) conditions in an automaton;
e) All students can explain why automata are important in Computer Science.

This leads us to the first research question:

1. To what extent have the conceptual goals been achieved?

Our second sub-question handles about the chosen context. When the context was chosen poorly, the conceptual goals will not be achieved. Although the context cannot be seen apart from the conceptual evaluation, the context itself needs some evaluation as well.

2. To what extent did the students find the context appealing, realistic and understandable?

Thirdly, we need to analyze if students can follow the pace of the lesson and if the amount of new information in this lesson is adequate. Therefore, our next sub-question:

3. Was the lesson well-structured in terms of new information, difficulty level and time management?

## 3.2 Method

The lesson is taught by Computer Science teacher Jacqueline Nijenhuis-Voogt of the GSG Guido de Brès, a secondary school in Amersfoort. The lesson is given to a VWO-5 class that consists of sixteen students.

To answer the research questions, we use three data sources: individual assignments, observations and a student survey.

### 3.2.1 Individual assignments

To examine the progress during the lesson or to identify the place where students get stuck, we collect all individual assignments and will try to find general patterns. These assignments are mainly used to answer sub-question 1.

The assignments were handed in directly after a student did the assignment and students were not obliged to write their name on top, so we cannot relate different assignments to one person.

Since we do a qualitative analysis of this pilot lesson, we are not interested in the amount of correct solutions, except for assignment 4. By analyzing the common mistakes, we can identify improvements for the lesson. We are however interested in the number of correct solutions for assignment 4, as this exercise was not created to familiarize the students with the concept, but to test the conceptual knowledge at the end of the lesson.

#### Individual assignment I

We will collect the mean and standard deviation of the found correct transitions. In the case that students appear to have many conceptual difficulties in the other assignments, we will closely analyze the mistakes to see why this assignment does not succeed in familiarizing the students with the basic concepts of automata.

#### Individual assignment II

We can distinguish between the different Piagetian stages in this exercise. When one can expand and adapt this automaton with states and transitions at the same time, this indicates a stronger understanding of the learned concepts than the previously done assignments which mainly focused on backtracking or making simple changes. 'Walking paths' will only help the students to validate that a solution is correct. If students succeed in doing this exercise, we know that they have reached the third stage of Piaget considering the concepts of transitions and states. We will use the answers on question 2a and the number of found solutions on question 2b to provide a part of the answer on the first research question.

#### Individual assignment III

We are interested if students apply the correct techniques when using variables as this partly answers the first research question. We collect the number of students that update variables in states. This informs us about the progress the students are making during the lesson.

#### Individual assignment IV

This assignment is a test assignment and used to answer many of research' sub-questions. This assignment is split in two parts, indicating if students think on the second or third stage of Piaget. The first part corresponds with the preoperational and the second with the concrete operational stage.

Since we expect all the students to be capable of solving assignment 4a, we will analyze all incorrect solutions if they are there. For assignment 4b we will identify common errors and identify how many students were able to complete this exercise.

### 3.2.2 Observations

During the lesson, we observe to what extent the teacher instructs the students in the way as this was described in the teacher instruction (see Appendix A). Furthermore, we observe how students do their assignments by walking around in the classroom. We also pay special attention to remarks and questions from students, as these may indicate confusion or a lack of instruction. Lastly, we pay special attention to remarks and questions from students, as these may indicate confusion or a lack of instruction.

### 3.2.3 Survey

The survey aims to answer the sub-questions 2 and 3. These sub-questions can be found back in the survey as shown in the table. We will collect the mean and standard deviation of the answers on survey questions 1, 2 and 5. Next to this, we will analyze the *Toelichting* if students filled this in and we will collect the answers given on the open questions, i.e. questions 3, 4 and 6.

| # | Survey question | Research question |
|---|---|---|
| **1** | I found this lesson (each part on a scale from 1 to 5): nice, interesting, difficult? | 4 |
| **2** | How do you rate (each part on a scale from 1 (bad) to 5 (good): amount of new information, difficulty of explanation, difficulty of assignments? | 4 |
| **3** | What was the easiest part of the lesson and why? | 4 |
| **4** | What was the hardest part of the lesson and why? | 4 |
| **5** | I found the context (each on a scale from 1 to 5): appealing, realistic, understandable. | 3 |
| **6** | How can you use automata in Computer Science? | 2 |

**TABLE 4**

We are mainly concerned with the collective results of the survey questions to answer our research question. However, we may also analyze individual responses if these can give extra information to answer other questions.

## 3.3 Results

The class consisted of 16 students that all participated in this pilot lesson.

### 3.3.1 Individual assignments

The correct solutions for all assignments can be found in Appendix A.

#### Individual assignment I

One student completed the assignment during the classroom evaluation, so we will not use this assignment for the results and analysis.

| Mean | Standard deviation |
|---|---|
| 13.467 | 2.232 |

**TABLE 5**

There were seventeen transitions to be found by the students and the results can be seen in Table 6.

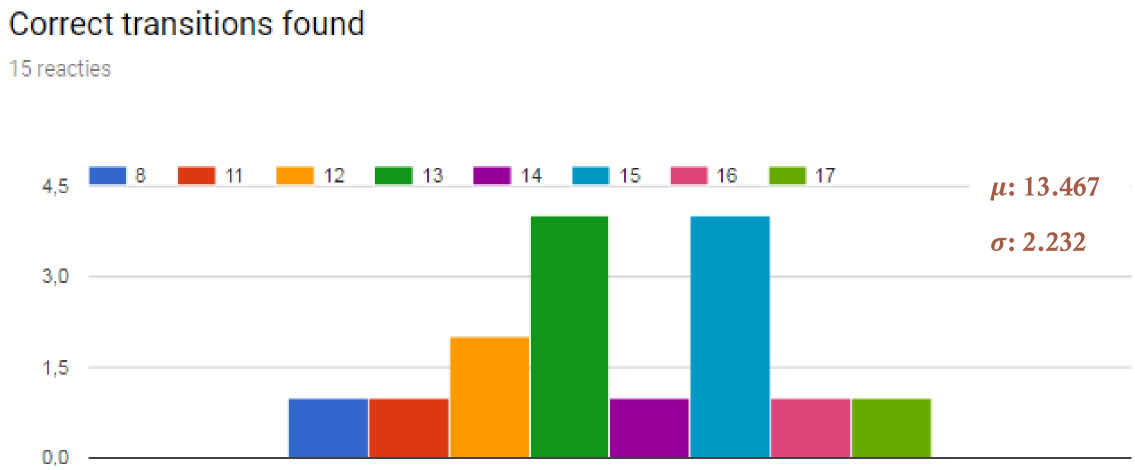



**TABLE 6**

We found one common mistake: seven students updated the loop on ‘Item ingevoerd’ instead and two created the loop via ‘Aan de beurt’.

## Individual assignment II

Four students did not answer question 2a. Two students mentioned that it was impossible to store information of multiple customers. Ten students answered that there was no option to choose where to order. Note that one student mentioned both, so he or she is included in both groups. Finally, a student stated that there was no distinction between paying contactless or cash.

Ten students indeed added a state for self-ordering (the transitions to and from this state were not always correct). Although not correctly done, one student started the automaton with some state indicating that a new day had started. This student was not one of the two answering on assignment 2a that it was impossible to store data of multiple customers in our current model.

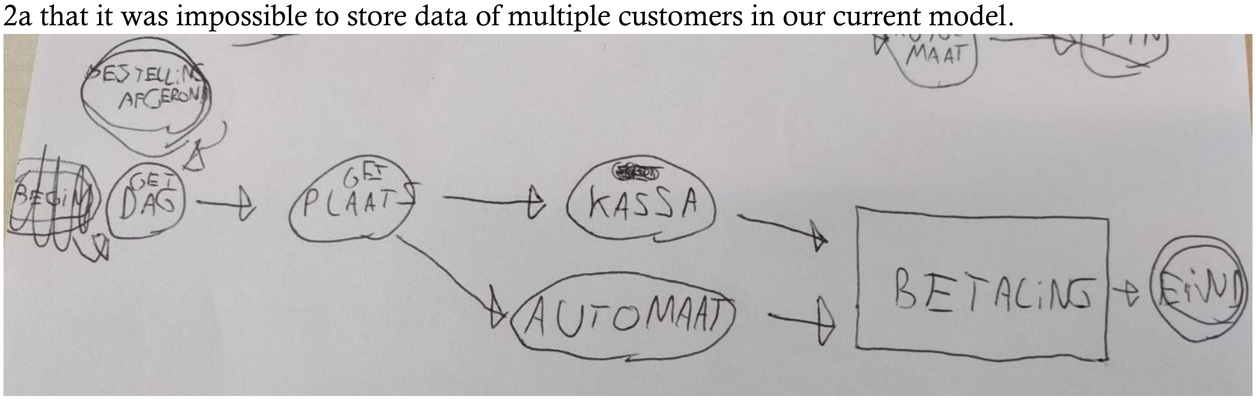



**FIGURE 4: SOLUTION THAT MODELLED A DAY**

## Individual assignment III

Six students updated variables in states and on transitions and three of those only updated variables in states. One student did not update any variable.

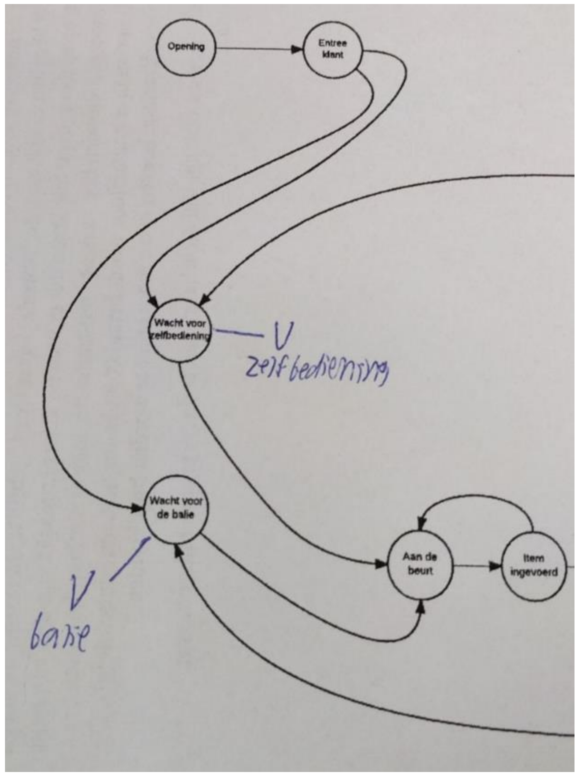



**FIGURE 5: UPDATING VARIABLES ON STATES**

One unexpected solution was that someone introduced one variable 'Bijhouden' to store all possible types of information.

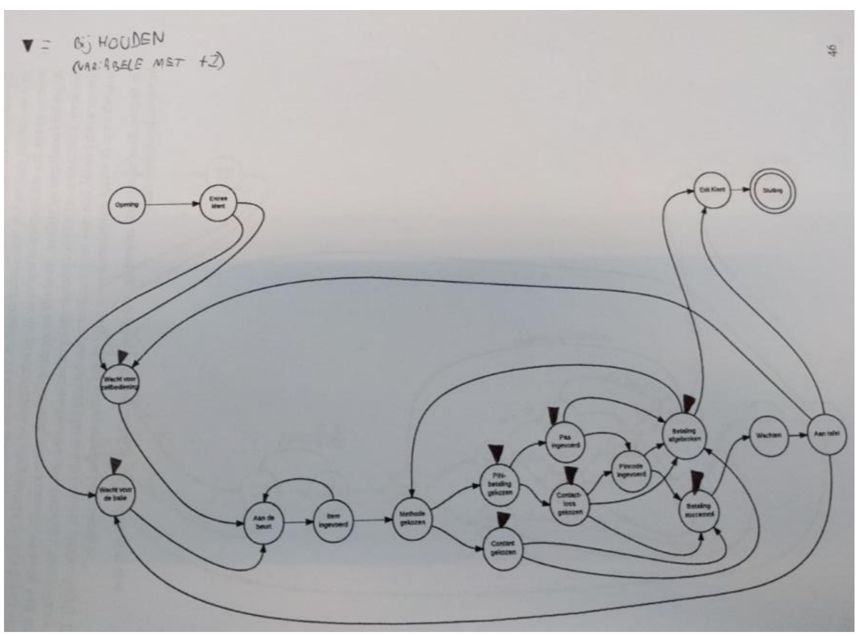



**FIGURE 6: ONLY ONE VARIABLE**

## Individual assignment IV

Fifteen students handed in the correct solution for assignment 4a. The failed solution:

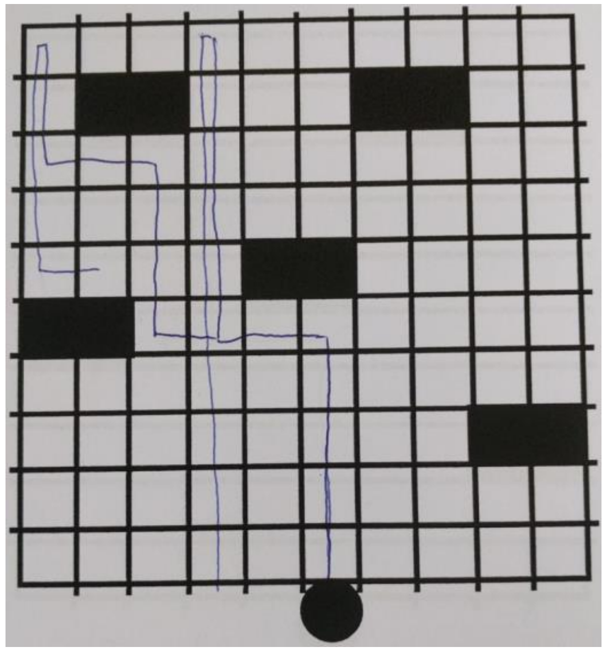


**FIGURE 7: WRONG SOLUTION FOR ASSIGNMENT 4A**

Five students corrected themselves while doing this exercise in this way:

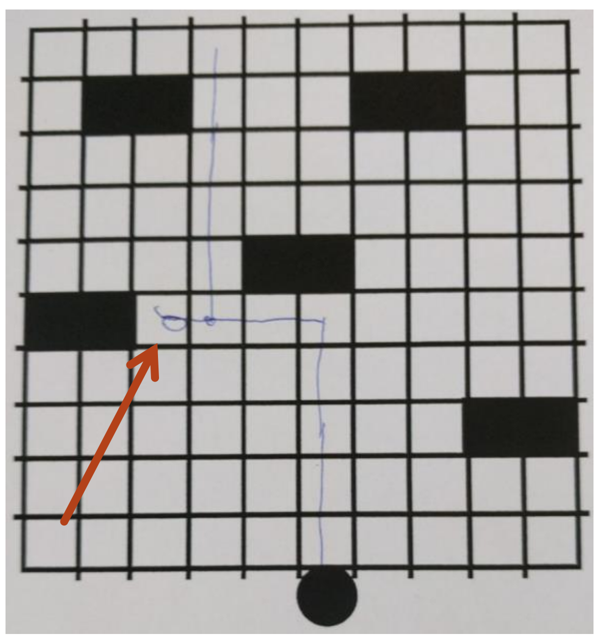


**FIGURE 8: STUDENTS CORRECT THEIR MISTAKE**

In the second part of the assignment we found one, almost correct, efficient solution. The student seemed to have the correct idea in his mind, although he did not draw the final state and failed to include a path from 'Noord' to 'Oost' in both directions.

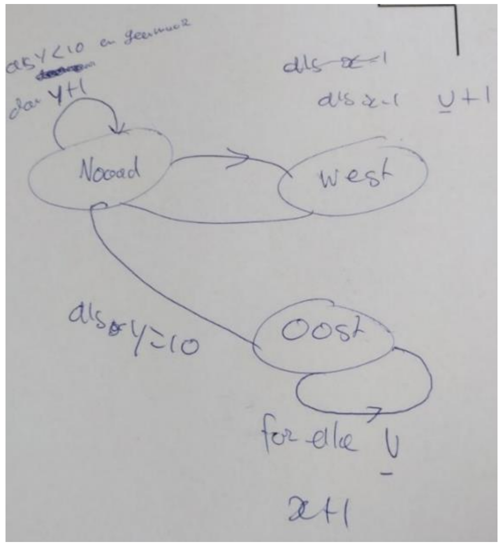



**FIGURE 9: SOLUTION WITHOUT DUPLICATING 'KIJKT NOORD'**

There was another student that used the current states to find a solution, but this solution is incorrect since the student moves from 'Kijkt west' to 'Kijkt oost', which is a move the robot could never make.

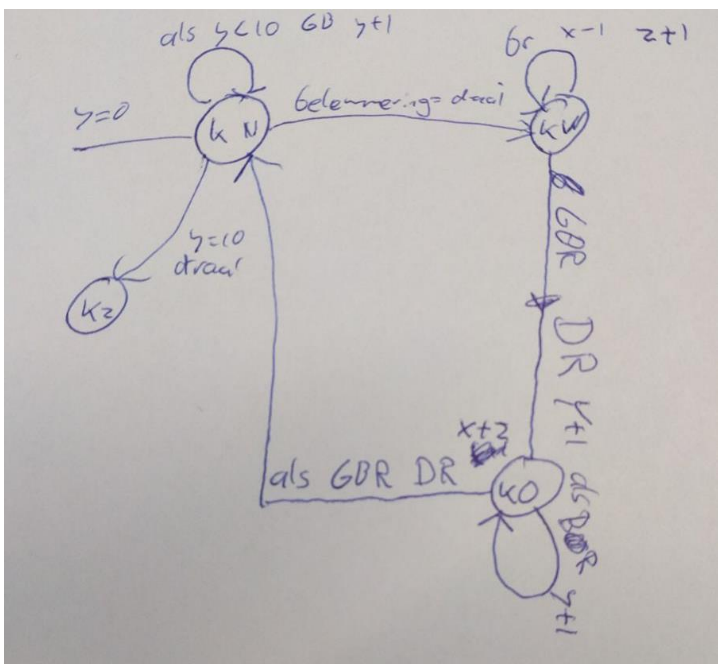



**FIGURE 10: SOLUTION INCLUDING A 180-DEGREES TURN**

Furthermore, seven students added a second state called ‘Kijkt noord’.

12 students handed in a solution (we consider assignment that only one included one state or variable not among those). 8 of these students introduced a variable to save the amount of steps taken to the left.

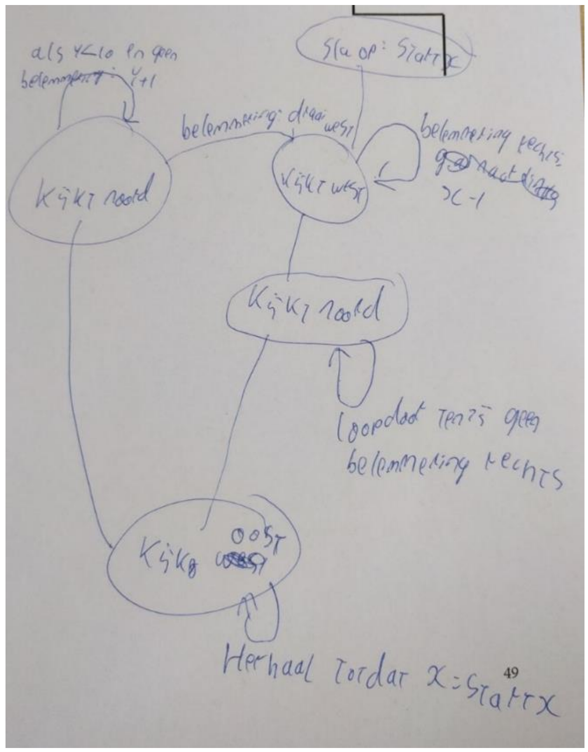



**FIGURE 11: EXAMPLE OF A SOLUTION THAT DUPLICATED THE STATE 'KIJKT NOORD'**

## 3.3.2 Observations

The letters in the part column correspond with the table in section 1.3.3.

| Part | Instruction (changes, remarks) | Student (questions (Q), remarks (R)) | Other |
|---|---|---|---|
| **A** | Teacher does not mention that models or automata are used for algorithms, but mentions only databases.<br>Little stress on the evaluation part of the job (in context explanation). | Q. What does a double ring mean? | A few options are named to extend the model. |
| **B** | Teacher does not ask the students to 'walk' through the model. | Students come up with many extensions of the schema, but not with a solution to the error (answering the next question already).<br>They also start discussing what would happen when groups would enter. | |
| **C** | | | Solution was already proposed during the previous exercise. |
| **D** | | Q: What does 'Bestelling afgerond' mean?<br>Another student answered from across the classroom immediately.<br>Students work from left to right when doing the assignment. | Link to the context was hard to find. |
| **E** | | Students participated actively by helping the student that was drawing his solution on the board. In this way they found the solution quickly. | Students find transitions that are correct but not yet in the given solution: multiple pin insertions and a transition from 'Pin gekozen' to 'Afgebroken'. Also, 'Item ingevoerd' can contain a loop instead of going back to a previous state. |
| **F** | Teacher explains the purpose of resetting or initializing the variable very well and this seems | Students are confused about the success of an ordered item. If it is not successful, should they still update the variable(s)? | No abstraction is done. Exactly 45 minutes passed. |

| | | | |
|---|---|---|---|
| | to lead to more understanding.<br>Teacher does not ask the students to find the alternative solution and to create a block.<br>Teacher lost track of the recurring automata in the PowerPoint. | Student indicates directly that variables should be used here.<br>They neither understand the purpose of the transition between 'Item ingevoerd' to 'Bestelling afgerond'. | |
| **G** | | Students struggle to start with this exercise and start staring around or checking out their neighbors' work.<br>Q: is it about a person or about all people together. Another student answers this question by citing the exercise.<br>Students introduce variables (assignment III), but this helps them discover why this model is not sufficient. | It appears to be hard to let the students work individually.<br>This exercise took too much time. No discussion of this part and the solution was only shortly shown. |
| **H** | | Again, confusion if orders that have an unsuccessful payment should be counted as an order or not. | Again, due to a lack of time, the solution was not discussed nor displayed. |
| **I** | | | Skipped |
| **J** | | Students find it hard to do this on their own. Especially during the second part. They could not find a solution themselves and therefore turned to their neighbor for ideas.<br>Students solved the first part fast.<br>Students have the correct idea in their mind (store variable to maintain the steps taken to the left) about assignment 4b. However, they do not know how to draw this in a model. | |

**TABLE 7: OBSERVATIONS**

### 3.3.3 Survey

12 students filled out the survey. We will first consider the closed questions, i.e. questions 1, 2 and 5. Then we will describe the results of the other questions.

#### Closed questions

For the closed questions we got the following results:

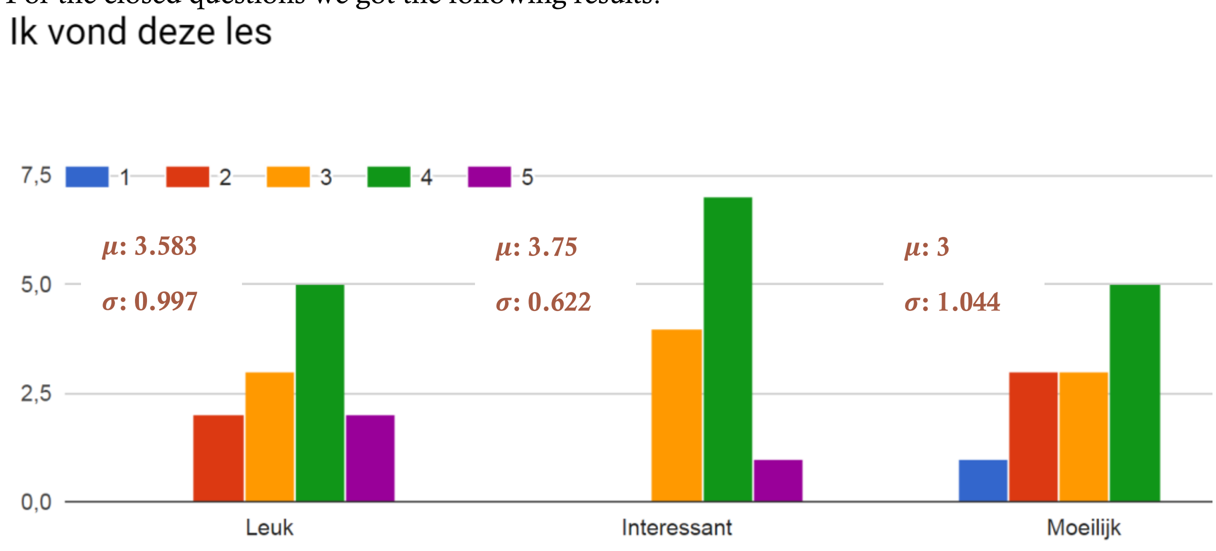



**FIGURE 12**

One student said that he was bothered by the warm temperature, but that he in general liked the lesson. Another student indicated that he was often confused about what to do and what was important.

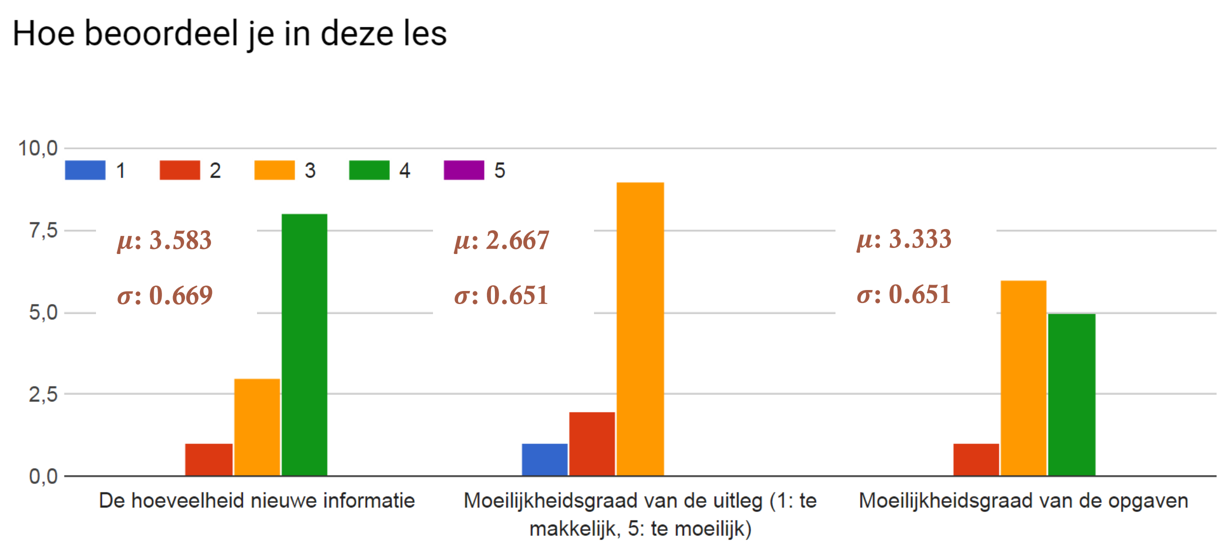



**FIGURE 13**

The comments correspond with the tables as one student states that the explanation was logical, but the assignment were more difficult. Three of the four comments indicate that, in general, the amount of new information and the level of difficulty was good.

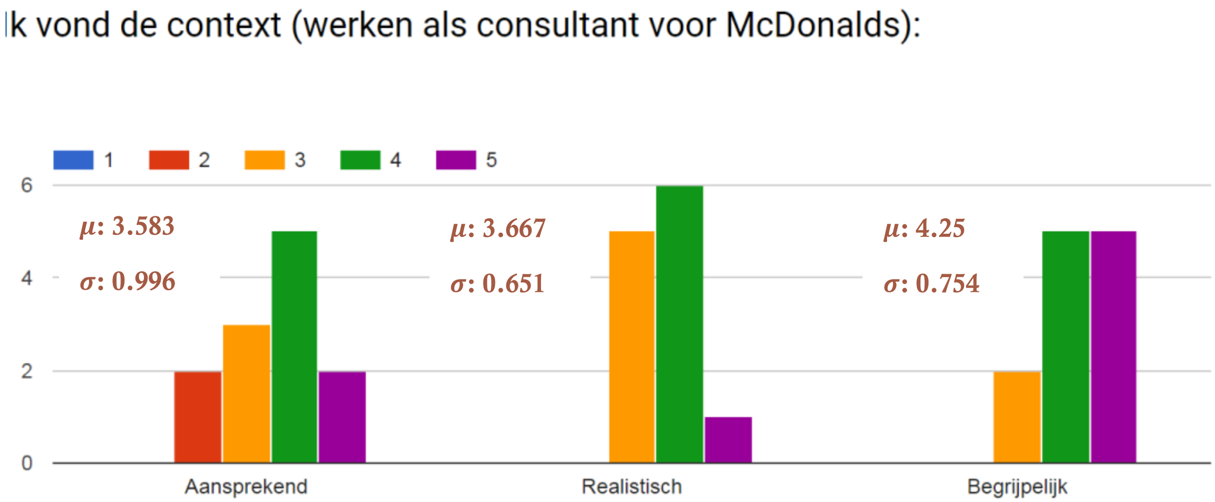



FIGURE 14

Two of the students were very positive and two others were somewhat more negative, but still indicating that it could have been worse. One stated that it was better than abstract alternatives. The students that gave 'Aansprekend' a 2, were also the ones with the somewhat more negative comments. They stated that they just do not like McDonalds, but they are not negative about the context itself.

## Open questions

The following two tables show the results of questions 3 and 4 indicating which part of the lesson was respectively easiest and hardest.

| Easiest part | Number of students |
|---|---|
| **Start of the lesson/first exercises** | 6 |
| **Explanation** | 2 |
| **Understanding a model** | 1 |
| **Idea behind the lesson** | 1 |
| **Creating a model** | 1 |
| **Assignments** | 1 |

TABLE 8

| Hardest part | Number of students |
|---|---|
| **Last assignment** | 5 |
| **Creating a model yourself** | 2 |
| **Assignment 2** | 2 |
| **Listening/concentrating** | 3 |

TABLE 9

This table shows the answer on the last open question about the relevance of automata for Computer Science:

| Answer | Number of students |
|---|---|
| **No idea** | 4 |
| **Everywhere** | 2 |
| **With everyting** | 1 |
| **Programming/program writing** | 2 |
| **Do research** | 1 |
| **Describe systems** | 1 |
| **Products of factory processes** | 1 |

**TABLE 10**

## 3.4 Conclusion

### 3.4.1 Conceptual goals

We will conclude if we achieved our set goals, by answering the sub-questions one by one.

#### Modelling

On the one hand, fifteen out of sixteen students handed in a correct solution to assignment 4a. Most of the students found the solution fast. On the other hand, one student made mistakes when doing assignment 4a.

#### States and transitions

We did not include an assignment at the end of the lesson to specifically evaluate this goal. Many students understood how to use and read a transition. Without a rough understanding of states and transitions, a solution on assignment 4a would not have been found. Assignment 1 was used to train this distinction by letting the students only fill out transitions and this exercise was generally done well, especially when solutions were combined.

However, the survey shows that students found assignments 2 and 4b hard, both assignments in which students need to add states and transitions to a given model. This is confirmed by the solutions of the individual assignments, where no one came up with an efficient and perfect solution for these assignments. But still, we see in their solutions that they do distinguish between states and transitions as they should.

The distinction between states and transitions is closely related to our fourth conceptual goal. Some students solved the third assignment by updating variables in a state. This may also indicate that do not students fully grasp the concepts of transitions and states.

To evaluate the third aspect of this goal, we can analyze the same assignments but now focus more on the found solutions. One student came up with a solution for assignment 4b. He or she introduces a new variable *v* at state 'Kijkt west' and decreases this in 'Kijkt oost', without duplicating states.

So, this goal was achieved. All students can work with transitions and states and understand how to use them. On the other hand, we see students struggle to introduce states and transitions and some update variables on a state.

### Variables

Since assignments 3 and 4 contain variables, these assignments are suitable to verify if we achieved this goal. Some students used variables falsely, by updating them on states. Assignment 4 shows that the students know how to read and interpret variables and some know how to introduce variables in different ways and have therefore left the preoperational and moved towards the operational stage.

Despite these results, we also see a student introducing just one variable in assignment 3, while others found a small number of variable updates. In conclusion, this goal was not achieved. Some find themselves at the operation stage, but others not even on a preoperational stage yet.

### Conditions

The first part of this goal has been accomplished since almost everybody succeeded in doing assignment 4a. With regards to the second aspect of this goal, we have found multiple solutions where students used conditions on transitions and although the solutions were not always correct, this was not caused by a lack of understanding of the use of conditions.

### Relevance of automata in computer science

The results of the last survey question can be grouped in different categories. The first group of students indicated that they did not know how automata could be used in Computer Science. The second group of students gave answers that were very vaguely formulated ('overal or 'met alles') and therefore give us little information. A third group answered very detailed how automata can be used in Computer Science. We conclude that this goal was not achieved.

## 3.4.2 Contextual goals

In the second part of the survey we considered the chosen context of working as a consultant for McDonalds. During the lesson, we did not observe bored faces when this was introduced and we heard or saw no one sighing or getting bored with this context.

Moreover, the fifth question of the survey (about the context) gets the highest rating. Especially the understandability (begrijpelijkheid) scores very high. This is also stated in the comments that students had after answering this question. One stated that it was better than abstract alternatives.

From the first question in the survey we also find that students generally liked this lesson and found it 'interesting' to 'very interesting'. And lastly, the fact that students generally did well at assignment 4 indicates that students were not bothered by the context to understand the concepts.

## 3.4.3 Lesson structure

Based on the observations, the second assignment cost much time, which resulted in a lack of time for the rest of the lesson. Students were unable to compare their solutions plenary, could not see the solution and did not know any more if they were on track or not. Moreover, students found it hard to come up with a solution for this problem.

Students found the amount of new information quite high. In the same question, they indicated that especially the assignments were quite hard and the explanation was easy to normal.

## 3.5 Discussion

Although many students can use the basic concepts of FSAs on a preoperational stage, and some already on a concrete operational stage, there is still room for improvements. We will therefore first analyze our conclusions and see how this impacts the pilot lesson. Afterwards, limitations and future work will be discussed.

### 3.5.1 Findings

Assignment 2 had some unexpected results and we will therefore discuss this first. Then, we will pay attention to the results from assignment 4, since this was the assignment that shows us how students progressed. The conclusion about the conceptual understanding of variables indicated that this differed a lot among the students. That is the third observation that we will discuss here. Finally, we will discuss seven other, smaller, changes to the pilot lesson.

#### Assignment 2

At this point, we do not expect all students to fully understand the concepts of states and transitions (this would mean an operational stage speaking in Piagetian terms). We therefore decided to remove the second assignment, and replace this by letting the teacher introduce the new context, give the students the new model and let them do the third (which now becomes the second) assignment.

We deliberately chose not to remove assignment 4b, although most of the students indicated in the survey that they considered assignment 4b to be the hardest. This exercise is meant to test if students have reached the operational stage and it is not important that everyone does reach this stage. It should be challenging for those that have reached that level, especially now without the second assignment. Not everyone found the lesson difficult, and this is in correspondence with the results of assignment 4b, so therefore we keep this assignment the way it is and in the same place in the lesson.

This change may as well increase the motivation of the students even more, since they already find it interesting and they also found the lesson a little too difficult. More attention can be given to evaluation of the assignments and students can understand the concepts at an earlier stage, which may result in better results at the end of the lesson and thus increase the chance of achieving the conceptual goals.

This solution solves another point, namely that students found the amount of new information quite high. In the same question, they indicated that especially the assignments were hard and the explanation was ‘easy’ to ‘normal’. By the removal of this assignment, the difficulty of the exercises in general is decreased, making it possible for the students to process the high amount of new information. We do not want to skip or remove certain concepts in the revised version of the lesson, since we got promising results per concept. We want the students to understand the concepts better. As shown by Lister, this should not be done by giving the students exercises that require thinking on another Piagetian stage, but this is done by practicing more in the current stage of thinking.

#### Assignment 4a

We see that the difficulty in this assignment lies in turning right when facing west. However, all except one found out that they should change the direction and state. The one student that failed to find a solution continued to walk straight ahead until he could not continue and only then turned right. This may be caused by the fact that we poorly formulated the transition from ‘Kijkt west’ to

‘Kijkt noord’, where the word ‘belemmering’ was suddenly gone and replaced by ‘niets’. The student may have thought that ‘niets’ meant no grid at all. However, the student tried to go to the right as well at the turning point, but makes another mistake there as well. He does turn south but just continues to walk back. So, either the students failed to understand the finite state, or he just got mixed up with the states and transitions.

An explanation may be found in the survey. The survey shows that one student had difficulty concentrating on the lesson due to the due to the high temperature in the classroom and the most difficult part of the lesson was, according to this student, doing the exercises because it was much too warm. However, we cannot know for certain that this assignment is connected to the same person as the one that had difficulty concentrating.

The observations also showed that students knew what they were doing. At the first individual assignment, it was observed that students worked from left to right, completing the model by following the process that was in their mind.

We implement a minor change in the assignment: change the vague description on the transition from ‘Kijkt west’ to ‘Kijkt noord’. We also noticed that we mixed up state names in the solution, so we change those as well. Thirdly, we update the loop on state ‘Kijkt noord’ in the solution since this should only happen if the robot should not first walk back to the original *x*-position. We cannot know that these changes result in all students to be able to finish this assignment, but it only improves the assignment and it takes away the problems that may have arisen.

### Variables

Since assignments 3 and 4 contain variables, these assignments are suitable to verify if we achieved this goal. Some students used variables falsely, by updating them on states.

We think this is due to a conceptual misunderstanding of states and transitions, instead of a problem with the understanding of variables, because the students indeed update the variables (although in the wrong place). They did not always manage to update all the variables, but the variables that were updated, where updated at the right transition (or the state just after the transition).

It is remarkable that some students left the preoperational stage since assignments two and three were not plenary discussed, due to a lack of time. We did not verify but assume the students to have experience in using variables from other areas in Computer Science.

Based on the observations during the third classroom assignment, where students luckily directly introduced a variable, we think that we can solve this problem by making ‘basic knowledge of variables’ a pre-requisite for this lesson. Since variables as a means of storing information is not characteristic for automata, we will not teach how to use variables in a lesson (or course) on automata. We still want to use variables for reasons explained when we constructed the model.

### Other changes

During the lesson, we made some other, unforeseen observations that are important for the revised lesson. We will discuss those in a chronological way.

Firstly, students kept making comments about the incompleteness of the model. Especially the absence of a toilet state appeared to be hard to deal with. Apart from the mistakes he or she makes when drawing states (it looks like a database model), one student drew a toilet state in the model as soon as he or she got the possibility to change the model. Although this is can be considered humorous, it indicates a lack of explanation in the introduction. Students should know that models are created to visually display certain aspects of the world. Those 'certain aspects' being some interesting properties or situations that you want to reason about. This means that a model necessarily omits details. Also, students did not clearly know that we were not modelling a group of customers, but only one single customer. We therefore add to the introduction the incapability of a model to represent all possible situations and we stress that the automaton is about a single customer.

Secondly, during first individual assignment confusion arose about what exactly needed to be done and what exactly was the context. Therefore, we move the introduction to the exercise to the assignment paper that is handed out.

Thirdly, some mistakes were found in the assignments. A loop on 'Pincode invoeren' should be added since customers get multiple (we do not care about the exact number here) chances of giving their PIN without having to go back to a previous state. Also, a transition can be added from the state in which a paying method has been chosen to the 'Bestelling afgebroken' state. It can also be argued that the transition from 'Item ingevoerd' to 'Aan de beurt' can simply be a loop on 'Item ingevoerd'. This depends on the interpretation of the states, but both solutions are correct, therefore we add this option as well. Also, there was confusion about the transition between 'Item ingevoerd' and 'Bestelling afgerond'. We chose to describe this state in the revised lesson by stating that a customer has indicated that the ordering is finished.

Fourthly, before the third assignment the teacher has to provide much information, probably causing confusion about how or what the students need to know. The teacher skipped finding alternative solutions and creating blocks. However, this appeared to work out fine, as we saw a few students using blocks in individual assignment II and given the fact that abstraction is not a concept we want to teach the students in this lesson. We choose to mark these two parts of this assignment as optional, leaving it up to the teacher to discuss it or just leave it out of the lesson. Next to this, we observed that students did not clearly know if only successful orders had to be updated. So, we choose to mark the creating of blocks and finding of alternative solutions as optional and we clearly state in the assignment that we want to know the exact amount of orders, without considering if they fail or succeed.

Fifthly, the teacher was confused about the slides. This was due to some automata appearing on multiple slides. We will therefore name the models and show on the PowerPoint on what slide we are.

Furthermore, we observed that it is hard to force students to think individually. We therefore only want the last assignment to be done individually. The other assignments will still be individual, but the teacher can allow students to discuss the solution with each other when they get stuck.

Students found it hard to explain why automata matter in Computer Science. We think that this is explained well enough in the lesson, but that students cannot answer this because they do not know that an automaton is a special kind of model and we mainly used the word 'model' when referring to automata. This is indicated by the observations where we noted that the teacher did not mention that finite state automata are a way of modelling. We have argued before why we think that the focus should be on modelling an algorithm instead of stressing the fact that we do this by finite state automata, since we use aspects of model checkers as well. Still, we will put more stress on the words 'automaton' and 'automata' instead of 'model' in the teacher instruction.

Seventhly, the observations show that the teacher did not stress the consultancy part of the context. Students may find it more realistic and appealing for a computer scientist to end up as a consultant than as a McDonalds employee. We therefore give more attention to the consultancy part instead of focusing the most on McDonalds and only naming consultancy shortly at the beginning of the lesson.

### 3.5.2 Limitations and future work

We must always keep in mind that we cannot simply apply (neo-) Piagetian methods, simply because it works in some cases. Multiple, sometimes contradicting, methods have proven successful or unsuccessful in different cases (Teague, 2015). Therefore, the proposed 'redefinition' can be researched and analyzed, since it was beyond the scope of this research to thoroughly develop a neo-Piagetian method based on empirical evidence.

Moreover, we have only evaluated this pilot on 16 students and we cannot know that multiple lessons in this area will result in the same conclusion. This thesis can therefore function as the basis for a quantitative analysis of this lesson, as we have not evaluated the improved lesson and we only tested our pilot on 16 students. Furthermore, the second version of the pilot should be evaluated, which may lead to more cycles of the design and creation cycle as displayed in the introduction. This thesis can also be used as the first lesson of a new series that teaches students FSAs with special attention to their use in algorithms.

# Bibliography

Barendsen, E., & Tolboom, J. (2016). *Advies Examenprogramma Informatica vwo-havo: inhoud en invoering*. Enschede.

Barrouillet, P. (2015). Theories of cognitive development: From Piaget to today. *Developmental Review*, *38*, 1–12. https://doi.org/10.1016/j.dr.2015.07.004

Boersma, K., Eijkelhof, H., van Koten, G., Siersma, D., & van Weert, C. (2006). *De relatie tussen context en concept. Intern werkdocument Vernieuwingscommissies bètavakken.* Utrecht.

Boom, J. (2015). A new visualization and conceptualization of categorical longitudinal development: measurement invariance and change. *Frontiers in Psychology*, *6*, 289. https://doi.org/10.3389/fpsyg.2015.00289

Bruning, L., & Michels, B. (2013). *Concept-contextvenster - Zicht op de wisselwerking tussen concepten en contexten in het bèta-onderwijs*. Enschede.

de Souza, G. S., Olivete, C., Correia, R. C. M., & Garcia, R. E. (2015). Teaching-learning methodology for formal languages and automata theory. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1–7). IEEE. https://doi.org/10.1109/FIE.2015.7344185

Demetriou, A., Shayer, M., & Efklides, A. (Eds.). (2016). *Neo-Piagetian Theories of Cognitive Development: Implications and Applications for Education.* Routledge.

Direct Instruction. (n.d.). Retrieved May 1, 2017, from http://www.worksheetlibrary.com/teachingtips/directinstruction.html

Feldman, D. H. (2004). Piaget's stages: the unfinished symphony of cognitive development. *New Ideas in Psychology*, *22*(3), 175–231. https://doi.org/10.1016/j.newideapsych.2004.11.005

Gilbert, J. K. (2006). On the Nature of "Context" in Chemical Education. *International Journal of Science Education*, *28*(9), 957–976. https://doi.org/10.1080/09500690600702470

Gilbert, J. K., Bulte, A. M. W., & Pilot, A. (2011). Concept Development and Transfer in Context-Based Science Education. *International Journal of Science Education*, *33*(6), 817–837. https://doi.org/10.1080/09500693.2010.493185

Hellermark, P., & Hollabaugh, M. (1992). Teaching problem solving through cooperative grouping. Part 2: Designing problems and structuring groups Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics American Journal of Physics American Journal of Physics*, *601*(66). https://doi.org/10.1119/1.18809

Isayama, D., Ishiyama, M., Relator, R., & Yamazaki, K. (2016). Computer Science Education for Primary and Lower Secondary School StudentsIsayama, D., Ishiyama, M., Relator, R., & Yamazaki, K. (2016). Computer Science Education for Primary and Lower Secondary School Students. ACM Transactions on Computing Education, 1. *ACM Transactions on Computing Education*, *17*(1), 1–28. https://doi.org/10.1145/2940331

Keller, J. M. (1987). Development and use of the ARCS model of instructional design. *Journal of Instructional Development*, *10*(3), 2–10. https://doi.org/10.1007/BF02905780

KNAW. (2003a). *Ontwikkeling van talent in de tweede fase. Advies van de KNAW klankbordgroep voortgezet Onderwijs.* Amsterdam.

KNAW. (2003b). *Robuuste profielen in het voortgezet onderwijs. Advies van de KNAW aan de minister van Onderwijs, Cultuur en Wetenschappen over het curriculum in de tweede fase van het voortgezet onderwijs.* Amsterdam.

Kulyukin, V., Kiran, S., Melodi, A., Heaps, O. C., Rasmussen, S., & Rawlins, W. (2015). Exploring Finite State Automata with Junun Robots: A Case Study in Computability Theory.

Lister, R. (2016). Toward a Developmental Epistemology of Computer Programming. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education on ZZZ - WiPSCE '16* (pp. 5–16). New York, New York, USA: ACM Press. https://doi.org/10.1145/2978249.2978251

Siegler, R. S. (1998). *Emerging Minds: The Process of Change in Children's Thinking - Robert S. Siegler - Google Boeken*. Oxford: Oxford University Press.

Teague, D. (2015). *Neo-Piagetian Theory and the novice programmer. School of Electrical Engineering & Computer Science; Science & Engineering Faculty*. Queensland University of Technology.

# Appendices

## Appendix A: Teacher Instruction (version 1)

In this appendix, the teacher instruction can be found. Important parts of the instruction are bold. We have chosen to provide an introduction, but we do not expect the teacher to read this or memorize it by hard. It is important though that the teacher presentation is almost equal to this, since a lot of sentences are important for the process or analysis afterwards. A PowerPoint presentation for the lesson is also provided (Appendix B). Instructions on how to use this, are underlined. Since the pilot is given in a Dutch class, the teacher instruction is in Dutch.

### a) Introductie (0 – 10)

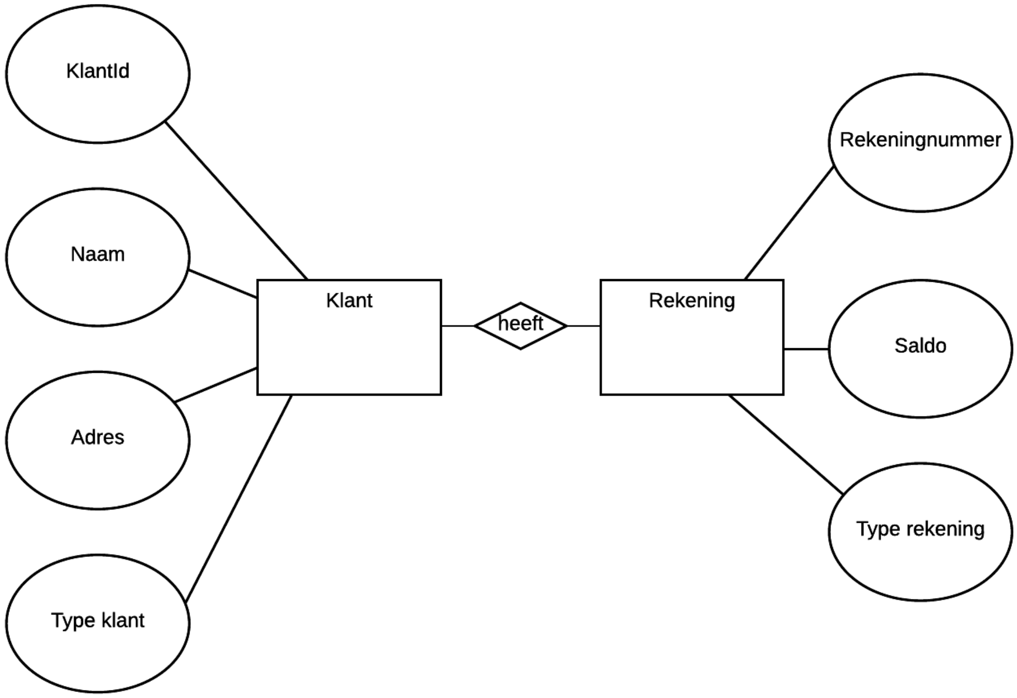
De opgaven moeten voorafgaand aan de les nog niet worden uitgedeeld. De docent kan vragen tijdens of na bepaalde onderdelen toestaan.

<u>PowerPoint slide 1</u>

Docent: vandaag gaan we een model maken *(optioneel: docent geeft informatie hoe dit in het curriculum past)*. Modellen worden in Informatica bijvoorbeeld gebruikt om schematisch weer te geven hoe systemen werken. Door ergens een model van te maken, kunnen veel mensen, zelfs zij die geen verstand hebben van Informatica, toch vaak meepraten over informatica toepassingen. Ook kunnen modellen erg goed worden gebruikt voor het programmeren. **Met name bij het opstellen van algoritmes worden vaak modellen gebruikt.**

<u>PowerPoint slide 2</u>

Docent: neem bijvoorbeeld dit model:




**Docent: waar gaat dit model over?**

Dit model gaat over een bank. Een bank heeft klanten. De klanten hebben bepaalde eigenschappen, maar iedere klant heeft ook een rekening. Die rekening heeft zelf ook weer allerlei eigenschappen. De docent kan dit naar eigen inzicht uitleggen. Type klant is bijvoorbeeld particulier of bedrijf terwijl een type rekening het verschil tussen een spaar- of betaalrekening weer kan geven. Het grote voordeel is dat bijna iedereen een model ongeveer snapt.

**Docent introduceert de volgende context**:

<u>PowerPoint slide 3</u>

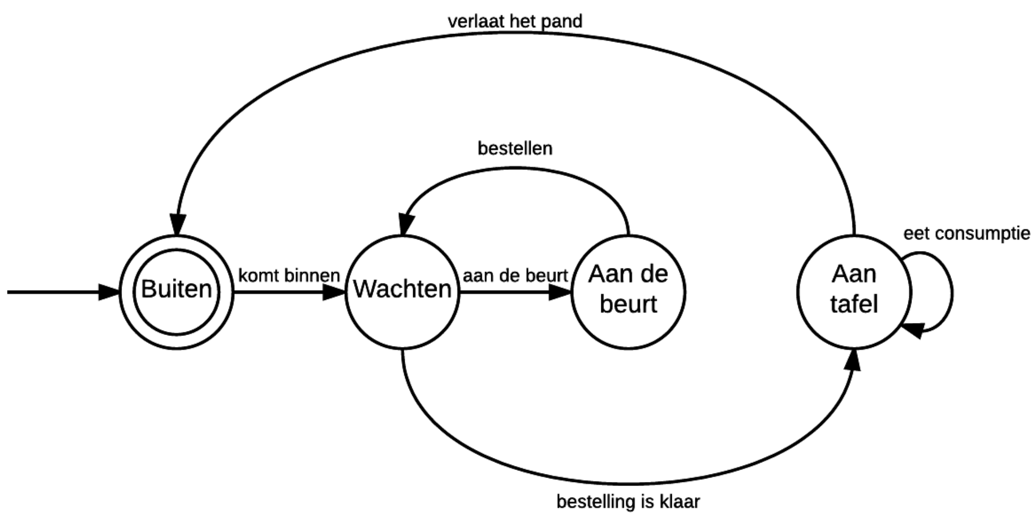
Drie jaar geleden ben je een eigen onderzoeksbureau begonnen. Inmiddels heb je je eerste echt grote opdracht binnen. McDonalds heeft jou namelijk gevraagd om voor hun een onderzoek uit te voeren. Ze willen graag een onafhankelijke evaluatie van enkele veranderingen die ze de laatste tijd hebben doorgevoerd in hun filialen. Zo hebben ze in de afgelopen tijd overal contactloos betalen mogelijk gemaakt en zijn er de zogenaamde ‘bestelpalen’ toegevoegd waar mensen zelf kunnen bestellen. Ze verwachten dat door het contactloos betalen mensen sneller kunnen afrekenen en er dus minder lange wachtrijen zijn. Met de ‘bestelpalen’ kunnen er meer mensen tegelijk bestellen, wat minder wachttijd oplevert. Daarnaast zijn er geen McDonalds medewerkers nodig om de bestelling op te nemen, waardoor McDonalds ook verwacht te besparen op medewerkers.

Om deze veranderingen te kunnen evalueren, besluit je een model op te stellen. **Het soort model dat je gaat maken heet een automaat**. Dit model kan je dan presenteren aan de managers bij McDonalds en je verwacht duidelijk bij te kunnen bijhouden efficiënt de vernieuwingen hopelijk zijn.

Vandaag ben je afgereisd naar een filiaal van McDonalds. Je arriveert, drinkt een bak koffie en kijkt goed rond. Na een uur heb je een globaal van de meeste klanten. De klanten komen binnen, plaatsen hun bestelling, eten deze op en vertrekken als ze uitgegeten zijn. Je besluit om heel ongedetailleerd het eerste gedeelte van het model alvast op te stellen, door globaal het gedrag van de klanten te beschrijven.

**Leerlingen luisteren naar de introductie.**

PowerPoint slide 4




**Docent: dit is een automaat. Iedere cirkel of bol noemen we een toestand. We beginnen bij de pijl die uit het niets lijkt te komen en we tekenen die meestal helemaal links.**

Vervolgens:

4. De docent beschrijft het proces aan de hand van de powerpoint en zoals in de introductie aangegeven, door als volgt te beginnen: **"Vanuit deze begintoestand gaan we…"**
5. Het is belangrijk dat de docent het jargoneske woord transitie *niet* noemt. Simpelere varianten als 'overgang' mogen wel.

**Docent: een dubbel omcirkelde toestand noemen we een eindtoestand** en als we hier zijn, kunnen we stoppen. Dat hoeft echter niet per se, we kunnen ook weer de automaat nogmaals ingaan. Dus als iemand naar buiten is gegaan, kan hij of zij terug komen en het hele proces nogmaals doorlopen.

### b) Klassikale opdracht I (10 – 15)

Docent: dit schema bevat helaas een fout. We willen geen uitbreiding van het schema, maar alleen de fout in deze automaat. Kunnen jullie die vinden?

**Leerlingen geven suggesties hoe de fout kan worden opgelost.**

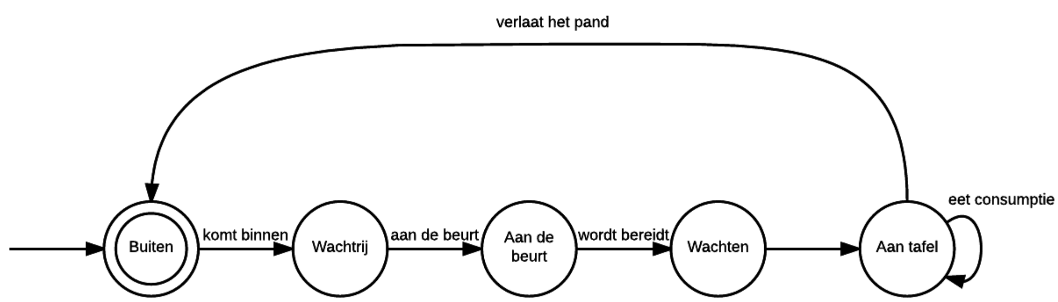
Docent:

6. Als leerlingen er niet achter komen, kan de docent vragen om nogmaals stap voor stap door de automaat te gaan. De docent kan ook vragen of er alternatieve manieren zijn om door de automaat te gaan.

**Uitwerking:**

PowerPoint slide 5

De fout is dat in dit model met binnen kan komen, wacht, eet en vervolgens naar buiten gaat. Dit is echter geen correcte weergave van de werkelijkheid. De correcte versie van dit model is deze:



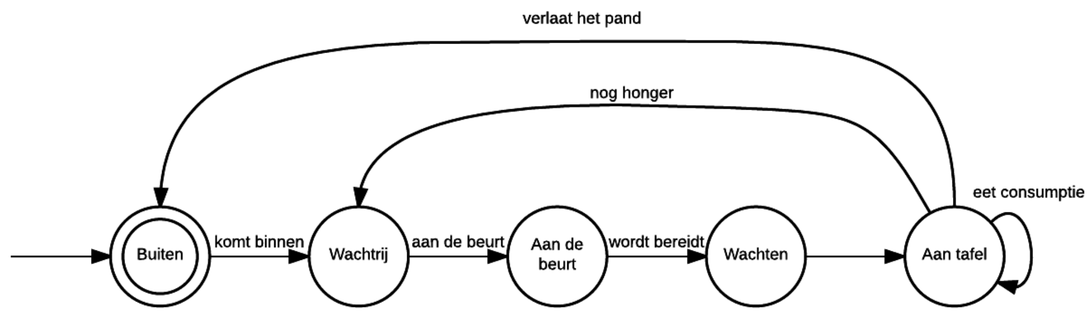

### c) Klassikale opdracht II (15-20)

Docent: mensen bestellen soms meerdere keren voordat ze weer naar buiten gaan. Ze hebben nog niet genoeg gegeten of ze willen een toetje. Voeg dit toe aan het model.

**Leerlingen geven suggesties voor de toevoeging.**

**Uitwerking:**

PowerPoint slide 6




Docent: nu is het tijd voor een aantal individuele opdrachten. De docent vertelt de opdracht en laat die daarna op het scherm zien.

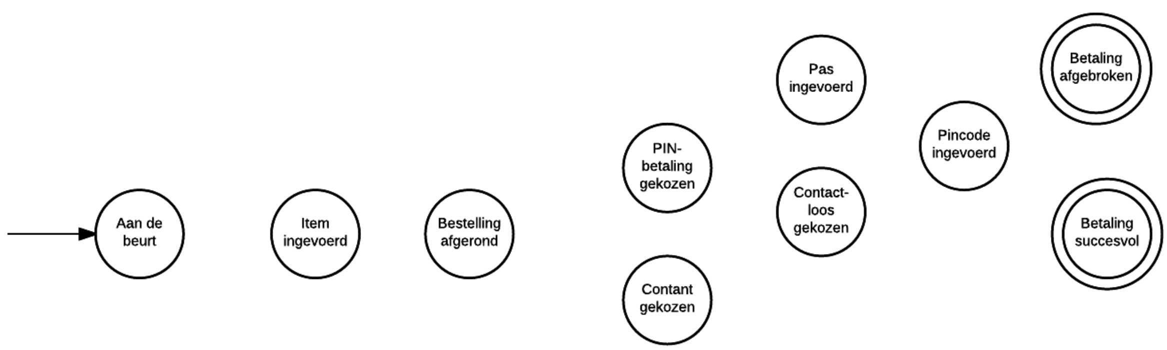
**Het is voor mijn scriptie van belang dat duidelijk geïnstrueerd wordt dat leerlingen de uitwerking niet kopiëren op hun opgavenformulier. De uitwerkingen worden beschikbaar gesteld via ItsLearning, zodat ze altijd achteraf nog de juiste uitwerkingen kunnen bekijken.**

### d) Individuele opdracht I (20-30)

**De docent**: omdat je vooral geïnteresseerd bent in hoe klanten precies bestellen en deze bestelling vervolgens afrekenen, besluit je dit iets nauwkeuriger weer te geven. Je hebt inmiddels de toestanden beschreven en gaat nu de automaat afmaken.

Docent deelt opdracht 1 uit.

De docent vraagt de leerlingen dit enkele minuten alleen te doen. Hierna kunnen leerlingen dit in paren vergelijken en het model nog aanpassen. Hierna kan de docent vragen of iemand aanvullingen kan vinden of fouten ziet in het getekende schema. Het model bevat 17 pijlen, de docent kan dit gebruiken om de leerlingen te stimuleren. Bijvoorbeeld als iemand verwacht klaar te zijn en slechts 13 pijlen heeft, kan de docent hem of haar erop wijzen dat er nog wel het een en ander moet gebeuren.



## e) Klassikale bespreking (30-35)

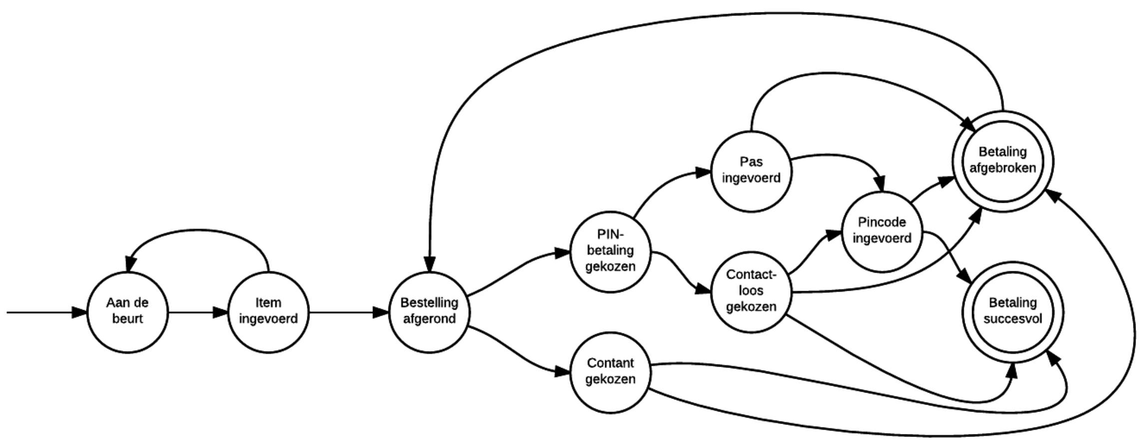
<u>PowerPoint slide 7</u>

In deze slide tekenen de leerlingen hun oplossing(en).

De docent laat iemand naar voren komen die denkt de juiste uitwerking te hebben en deze wordt klassikaal besproken, tot het getekende model overeenkomt met de uitwerking.

***Uitwerking:***

<u>Powerpoint slide 8</u>




<u>PowerPoint slide 9 (dezelfde als 5)</u>

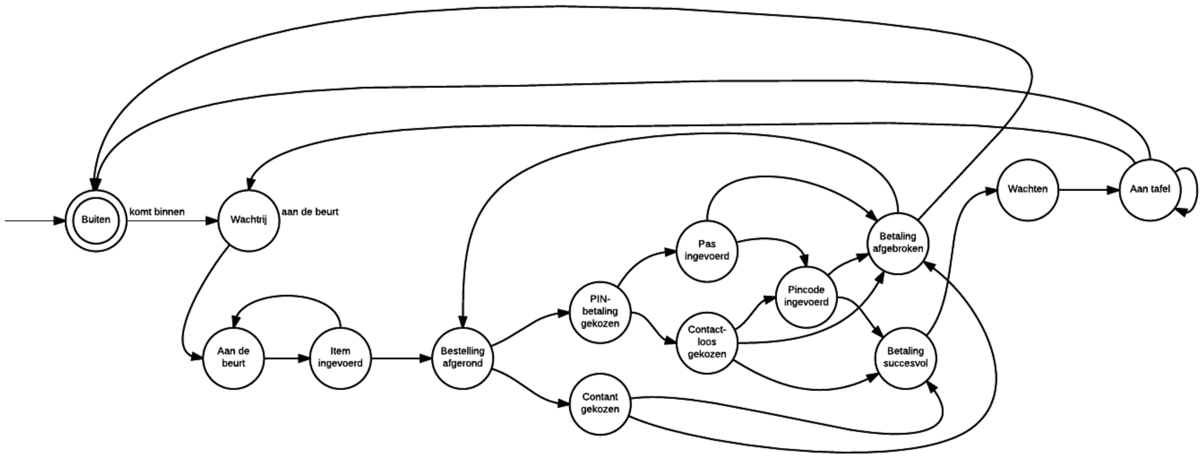
De docent vraagt aan de studenten of iemand weet hoe het nieuwe model in dit model past. We hebben namelijk de toestand ‘Aan de beurt’ vervangen door dit model van ‘Bestellen en betalen’.

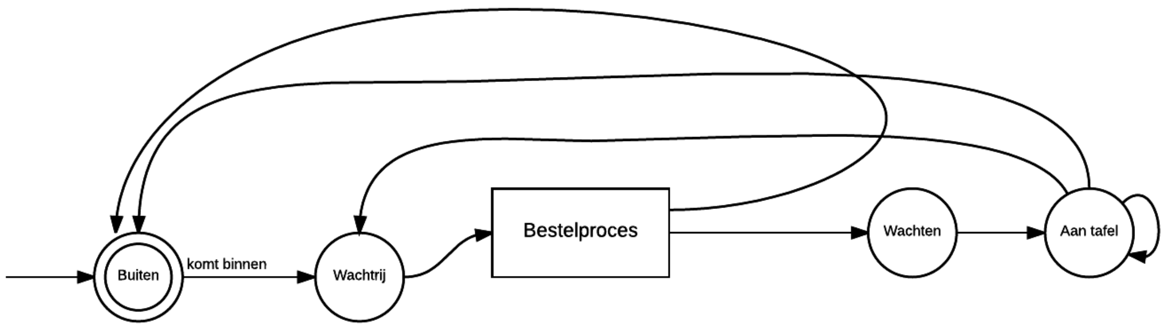
***<u>Uitwerking: PowerPoint slide 10</u>***

De docent toont daarna hoe we gedeeltes uit een model kunnen vereenvoudigen in een grote tekening, door het te vervangen door een vierkant blok met de in- en uitgangen die ook in het schema zichtbaar waren.

<u>PowerPoint slide 11</u>

Om duidelijk te maken hoe dit blok het gedeelte vervangt kan de docent de leerlingen vragen om de transitie te beschrijven, met andere woorden, wat kan er op de pijlen staan die aan het blok verbonden zijn.




De docent vraagt aan de leerlingen wat het verschil is tussen 'Aan de beurt' en 'Bestelproces' (slide 11 en 12). Het antwoord luidt als volgt: bestelproces is een blok waarin eigenlijk een model in zit. 'Aan de beurt' is slechts een toestand. Het gedetailleerder beschrijven van deze toestand, heeft ertoe geleidt dat we ook een pijl hebben van 'Bestelproces' naar 'Buiten'.

## f) Klassikale opdracht III (35-45)

**Docent geeft een korte introductie**: inmiddels hebben we al een best uitgebreid model gemaakt. Toch zegt dit nog vrij weinig. We gaan dit gedrag daarom uitbreiden met gegevens die nodig zijn om een goede beslissing te kunnen maken voor McDonalds. We gaan daarom vastleggen *hoeveel* menu-items iedere klant toevoegt aan zijn of haar bestelling tijdens één bezoek aan het filiaal.

PowerPoint slide 13

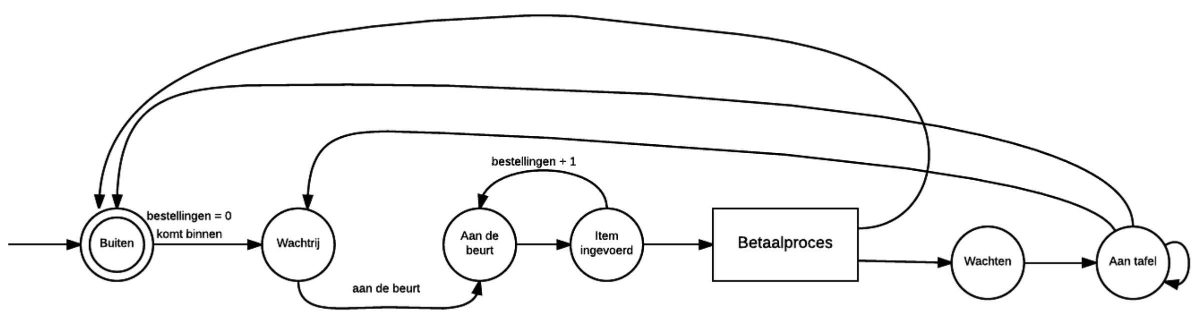
De studenten bedenken de oplossing en tekenen dit in de slide. Ze worden geleid in het proces door de docent. Deze vraagt eerst om welk gedeelte van het model het gaat. Er kan namelijk een 'Betaalproces'-blok worden getekend, maar het bestelproces moet wel aanwezig blijven. Het is zaak dat de leerlingen **minimaal** dit blok vinden, maar ze mogen ook het blok 'Betalen en eten' vinden waarbij de twee rechter toestand ook meegenomen worden in het blok. De docent kan leerlingen motiveren verder te denken als ze een kleiner gedeelte samenpakken, bijv. “Kunnen we wellicht nog meer samenvatten?”

Hierna wordt aan de leerlingen gevraagd of ze een manier kunnen bedenken hoe de hoeveelheid menu-items kan worden opgeslagen. Afhankelijk van de voorkennis van een bepaalde klas kan de docent ze naar eigen inschatting leiden naar dit proces (bijv. weten ze wat variabelen zijn).

Er zijn twee mogelijkheden waarbij de onderstaande te prefereren is, aangezien een bestelling pas definitief is als die is ingevoerd. De andere optie is namelijk om de bestelling variabele te updaten van ‘Aan de beurt’ naar ‘Item ingevoerd’. De klas moet beide oplossingen vinden, zodat men kan zien waarom het een beter is dan het ander. De kans is aanwezig dat men probeert de variabele in een toestand te stoppen en die daar te verhogen. De docent kan vragen of iemand snapt waarom dit niet kan. Er staat nu geen tekst op de pijl maar er kan gevraagd worden wat er op de pijl zou kunnen staan. Eventueel kan anders de docent zelf uitleggen dat het ‘Aan de beurt zijn’ slechts een beschrijving is van de wereld, niet een daadwerkelijke handeling. Het is van belang dat het woord state of transitie niet letterlijk wordt gebruikt, maar dit verduidelijkt wordt door de voorbeeldzinnen.

Ook is belangrijk dat ze de bestelling op 0 zetten als de klant binnenkomt, aangezien een klant anders twee keer binnen kan komen en dit doorgeteld wordt terwijl het een ander bezoek is. Overigens is de notatie hier niet heel belangrijk, dit mag ook gewoon worden beschreven als ‘bestellingen wordt 0’. Dit kan ook worden geplaatst op de pijlen die naar *Buiten* wijzen, maar dan weten we niet 100% zeker dat *bestellingen* tijdens het eerste bezoek wel echt 0 zijn, dus is de eerste de gewenste variant (maar de tweede niet per se fout).

***Uitwerking: PowerPoint slide 14:***




## g) Individuele opdracht II (45 – 55)
De docent deelt opdracht 2 uit en neemt opdracht 1 in.

De docent instrueert de leerlingen om de opdracht individueel te gaan maken. Na 5 minuten geeft de docent toestemming om het werk te vergelijken met de naast zittende persoon.

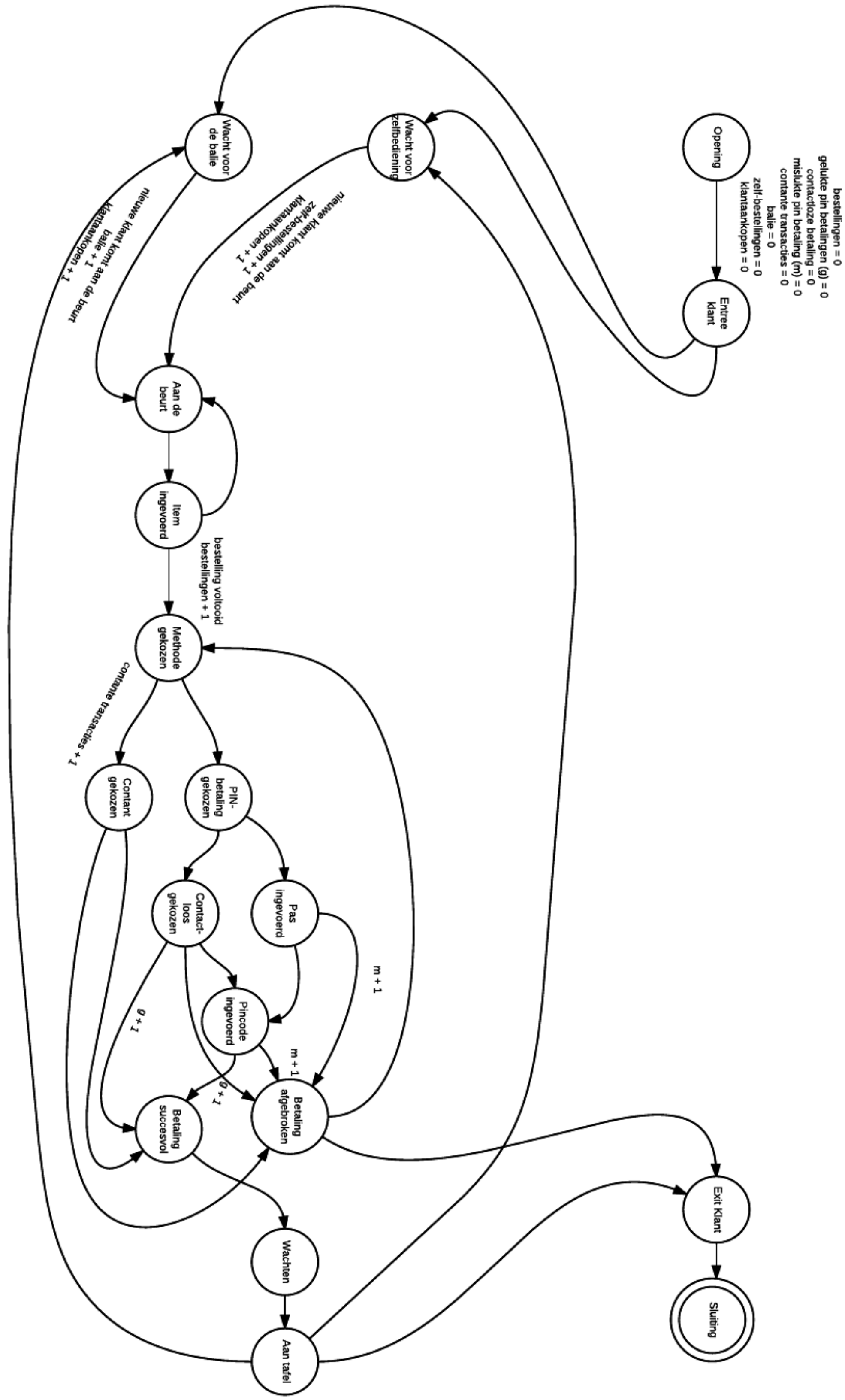
## h) Individuele opdracht III (55 – 65)
De docent deelt opdracht 3 uit en neemt opdracht 2 in.

## i) Klassikale bespreking (65 – 75)
PowerPoint slide 15

De docent laat iemand naar voren komen die denkt de juiste uitwerking te hebben en deze wordt besproken, tot het getekende model overeenkomt met de uitwerking. Alle alternatieve opties worden besproken als ze aan bod komen, inclusief afwegingen waarom op welke transitie bepaalde variabelen worden opgehoogd.

***Uitwerking: PowerPoint slide 16***

Inderdaad kunnen al deze gegevens in één model komen:


bestellingen = 0
gelukte pin betalingen (g) = 0
contactloze betaling = 0
mislukte pin betaling (m) = 0
contante transacties = 0
balie = 0
zelf-bestellingen = 0
klantaankopen = 0
Opening
Entree klant
Wacht voor zelfbediening
Wacht voor de balie
nieuwe klant komt aan de beurt
zelf-bestellingen + 1
klantaankopen + 1
nieuwe klant komt aan de beurt
balie + 1
klantaankopen + 1
Aan de beurt
Item ingevoerd
bestelling voltooid
bestellingen + 1
Methode gekozen
contante transacties + 1
PIN-betaling gekozen
Contant gekozen
Pas ingevoerd
Contact-loos gekozen
Pincode ingevoerd
m + 1
g + 1
m + 1
g + 1
Betaling afgebroken
Betaling succesvol
Wachten
Aan tafel
Exit Klant
Sluiting

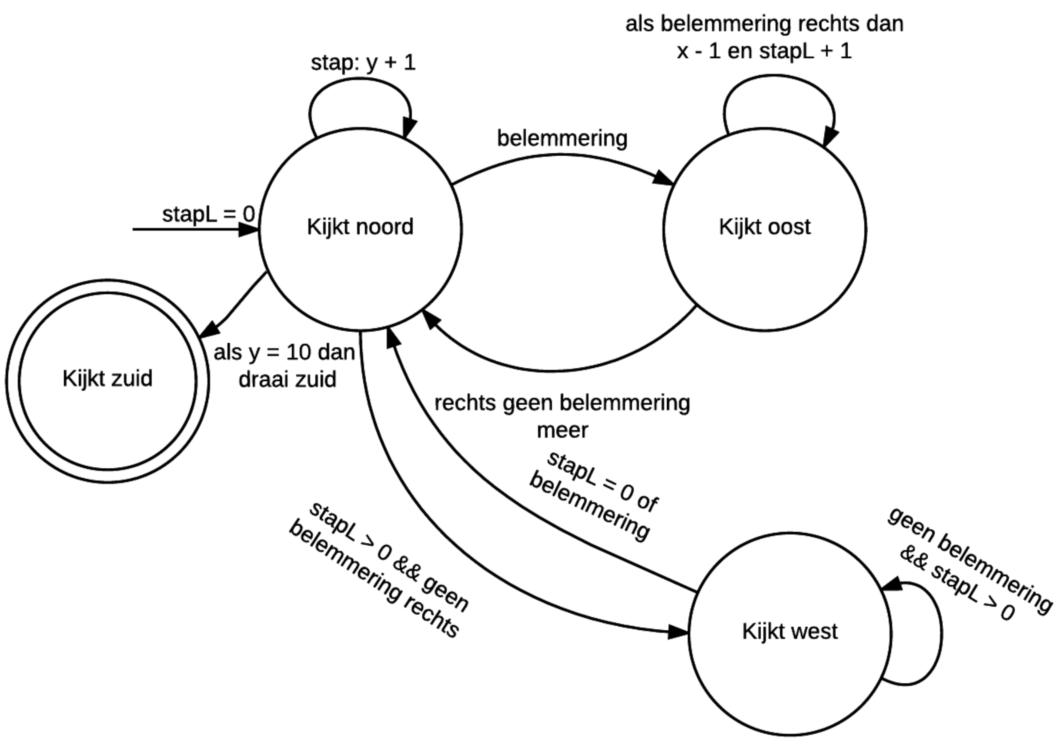
## j) Individuele opdracht IV (75 – 85)

**Deze opdracht moet echt individueel opgelost worden.**

Docent: je hebt genoeg gezien in de McDonalds voor een voorlopig advies. Je bent gevraagd door Jeroen, een vriend die bij de TU Delft, die onderzoek doet naar robots om hem te helpen. Hij werkt mee aan een groot project dat uiteindelijk een zorg-robot moet opleveren. Een robot die bij hulpbehoevende mensen in huis wordt geplaatst en daar allerlei taken gaat doen die de persoon zelf niet meer kan. Jeroen moet ervoor zorgen dat de robot zich in ieder mogelijke ruimte moet kunnen bewegen.

De docent instrueert de leerlingen opgave 4 te maken. Deze opgave bestaat uit twee delen. Opgave 4 en opgave 4a worden uitgedeeld en studenten krijgen pas het tweede deel als ze de eerste hebben ingeleverd. Aangezien dit gaat om het beschrijven van een automaat is dit voor de bespreking geen probleem.

**Uitwerking:**

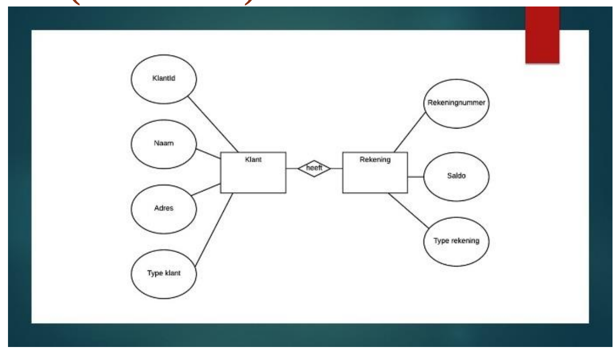



## k) Klassikale bespreking en afsluiting

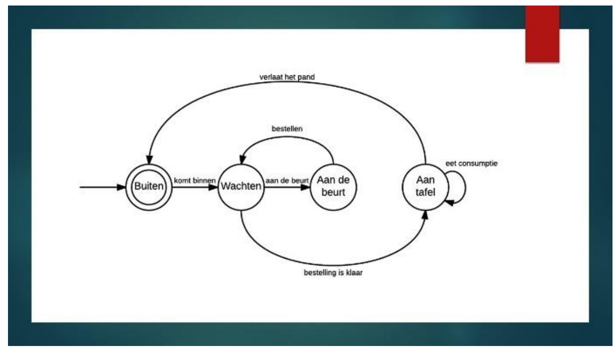
PowerPoint slide 17

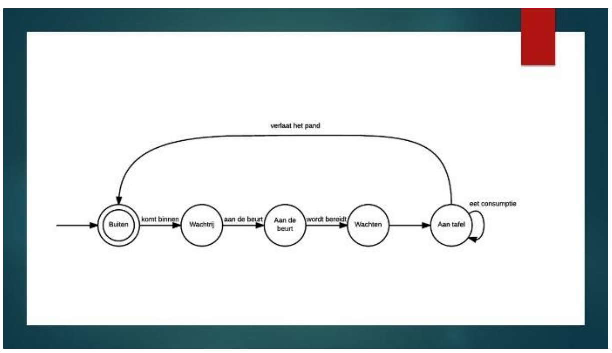
De docent bespreekt de laatste opgave klassikaal op dezelfde manier als de voorgaande keren en laat de leerlingen weer een uitwerking tekenen. Hierna toont de docent de uitwerking: PowerPoint slide 18 en bespreekt deze kort. De docent sluit af met een samenvatting: we zijn gekomen aan het einde van de les. In deze les hebben jullie geleerd hoe je modellen kan maken. Jullie hebben een automaat getekend van een McDonalds filiaal en van het looppatroon van een robot. Jullie hebben gezien hoe modellen kunnen bijdragen aan het inzichtelijk maken van allerlei situaties, wat ook met name in Informatica gebeurt. Inmiddels kunnen jullie een simpel automaat opstellen met een begintoestand, eindtoestand en normale toestanden. Daarnaast kunnen jullie een model ook gebruiken om bepaalde gegevens te meten door variabelen te gebruiken en deze te veranderen bij een toestandsverandering.

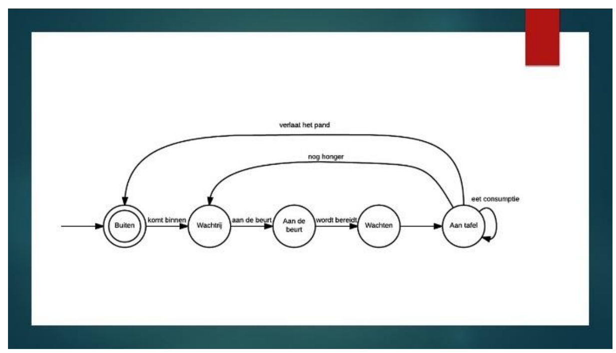
# Appendix B: PowerPoint Presentation (version 1)

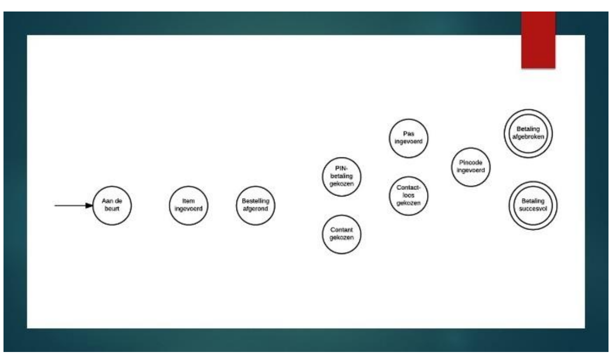



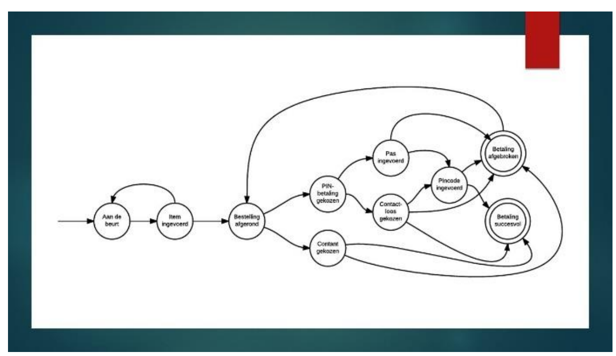


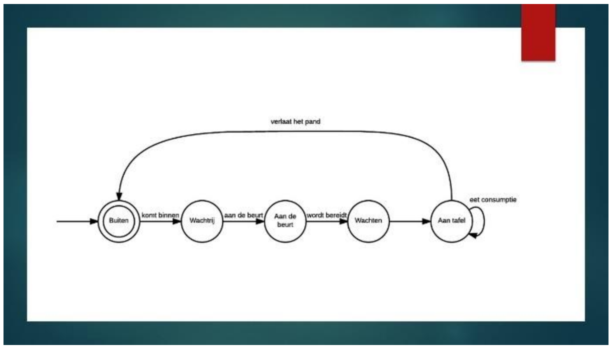


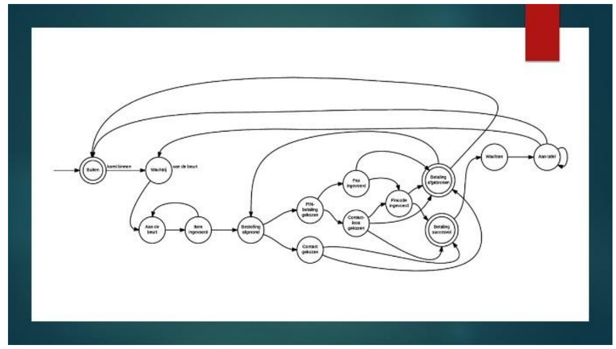



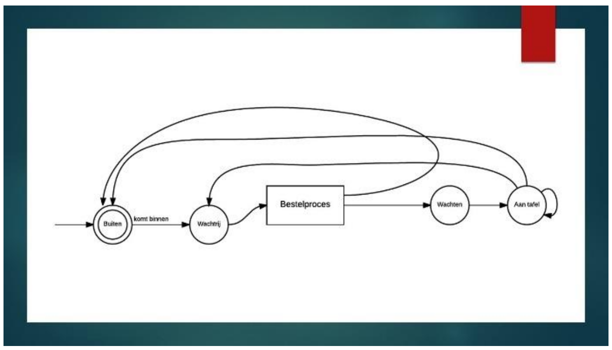



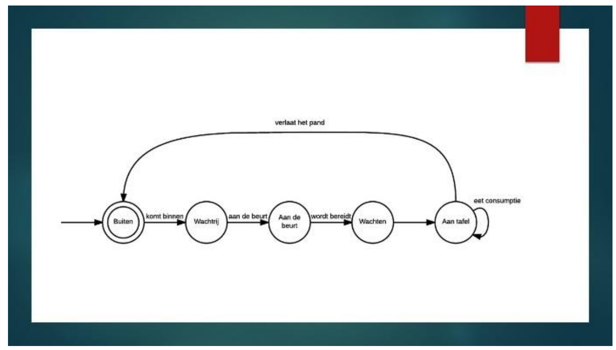



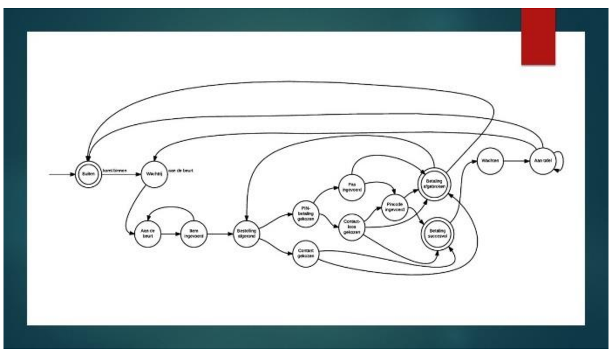



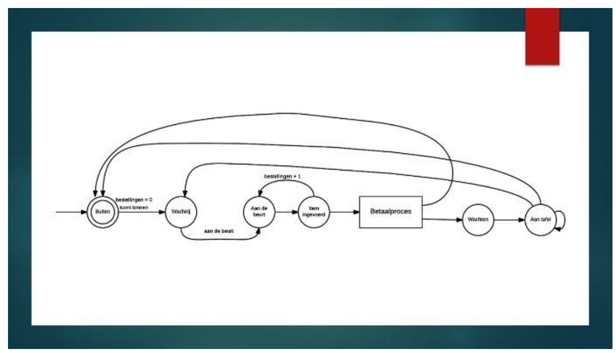

verlaat het pand
Buiten
komt binnen
Wachtrij
aan de beurt
Aan de beurt
wordt bereidt
Wachten
Aan tafel
eet consumptie

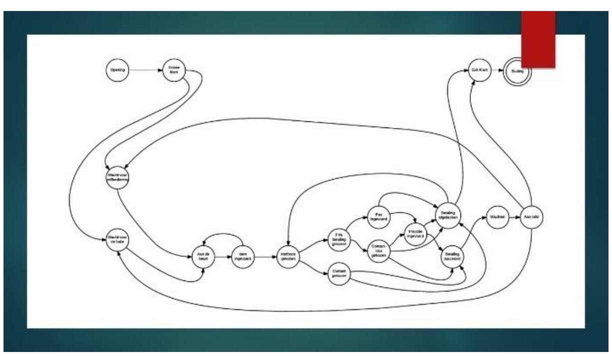


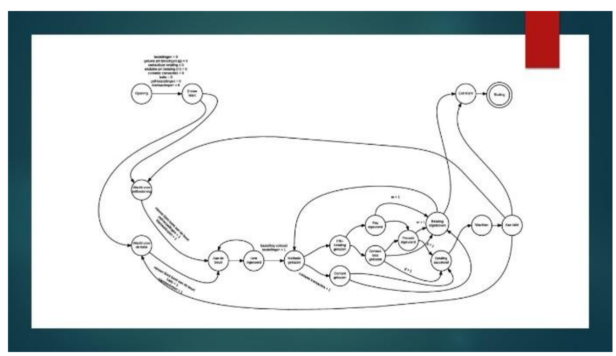

Buiten
komt binnen
Wachtrij
Bestelproces
Wachten
Aan tafel

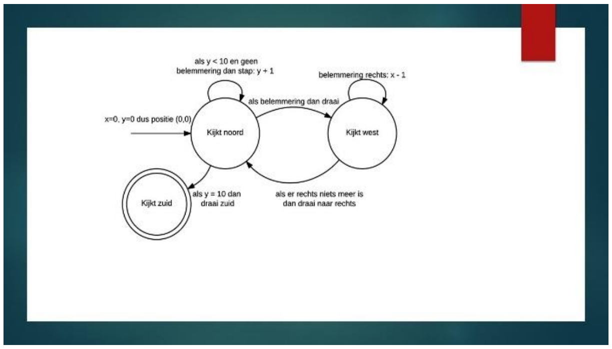

verlaat het pand
Buiten
komt binnen
Wachtrij
aan de beurt
Aan de beurt
wordt bereidt
Wachten
Aan tafel
eet consumptie

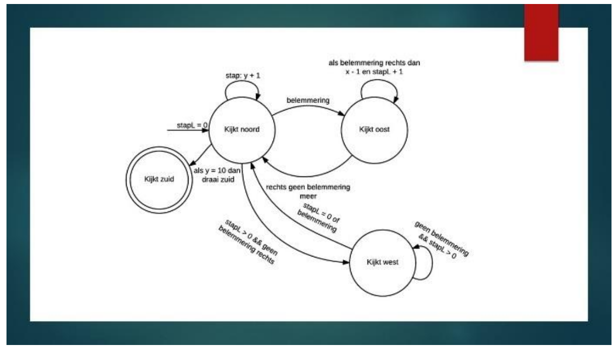



bestellingen = 0
komt binnen
Buiten
Wachtrij
aan de beurt
bestellingen + 1
Aan de beurt
Item ingevoerd
Betaalproces
Wachten
Aan tafel






als y < 10 en geen
belemmering dan stap: y + 1
belemmering rechts: x - 1
als belemmering dan draai
x=0, y=0 dus positie (0,0)
Kijkt noord
Kijkt west
Kijkt zuid
als y = 10 dan
draai zuid
als er rechts niets meer is
dan draai naar rechts

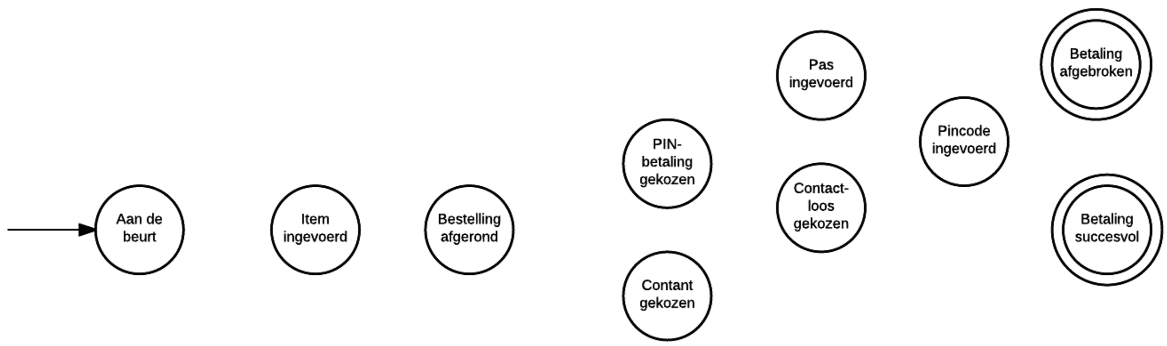

als belemmering rechts dan
x - 1 en stapL + 1
stap: y + 1
belemmering
stapL = 0
Kijkt noord
Kijkt oost
Kijkt zuid
als y = 10 dan
draai zuid
rechts geen belemmering
meer
stapL = 0 of
belemmering
stapL > 0 && geen
belemmering rechts
geen belemmering
&& stapL > 0
Kijkt west

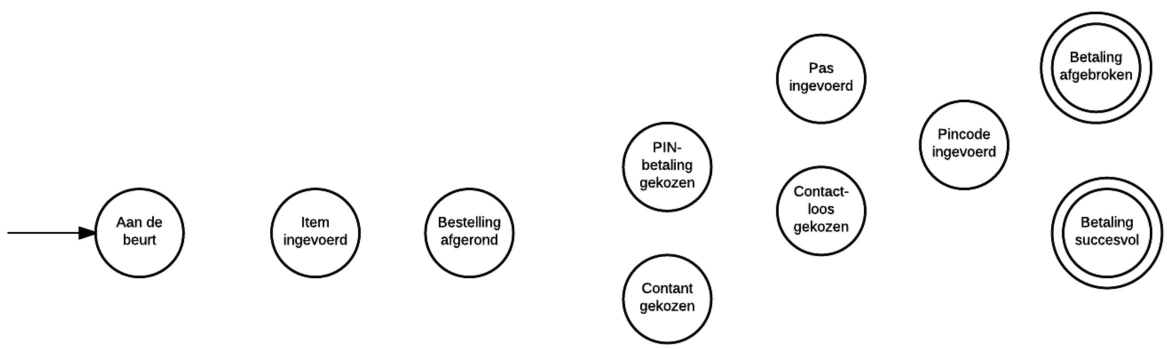
# Appendix C: Student Handout (version 1)

This appendix contains the handout that is provided to the students at the beginning of the lesson. It contains the individual assignments of the lesson and it will be collected and used to evaluate afterwards.

**Opdracht 1**

Maak het schema af. Gebruik de eerste versie als kladversie en maak de tweede als je denkt klaar te zijn.



**Opdracht 2**

Om een goed beeld te kunnen vormen van het bestellen en betalen ga je nu opslaan waar en hoe vaak klanten bestellen. Je wilt ook graag dat je precies weet hoeveel mensen op een dag zelf bestellen of hoe vaak de 'gewone' kassa wordt gebruikt. Je hoeft niet alleen de gelukte bestellingen op te slaan, maar alle keren dat er iemand gebruikt maakt van een van beide manieren. Daarnaast moet je bijhouden hoe vaak een pin-transactie lukt of mislukt, hoe vaak er contant wordt betaald, hoe vaak er contactloos wordt betaald (dat is dus ook een pinbetaling!). De variabelen van de vorige opdrachten hoeven in dit model niet te worden bijgehouden.

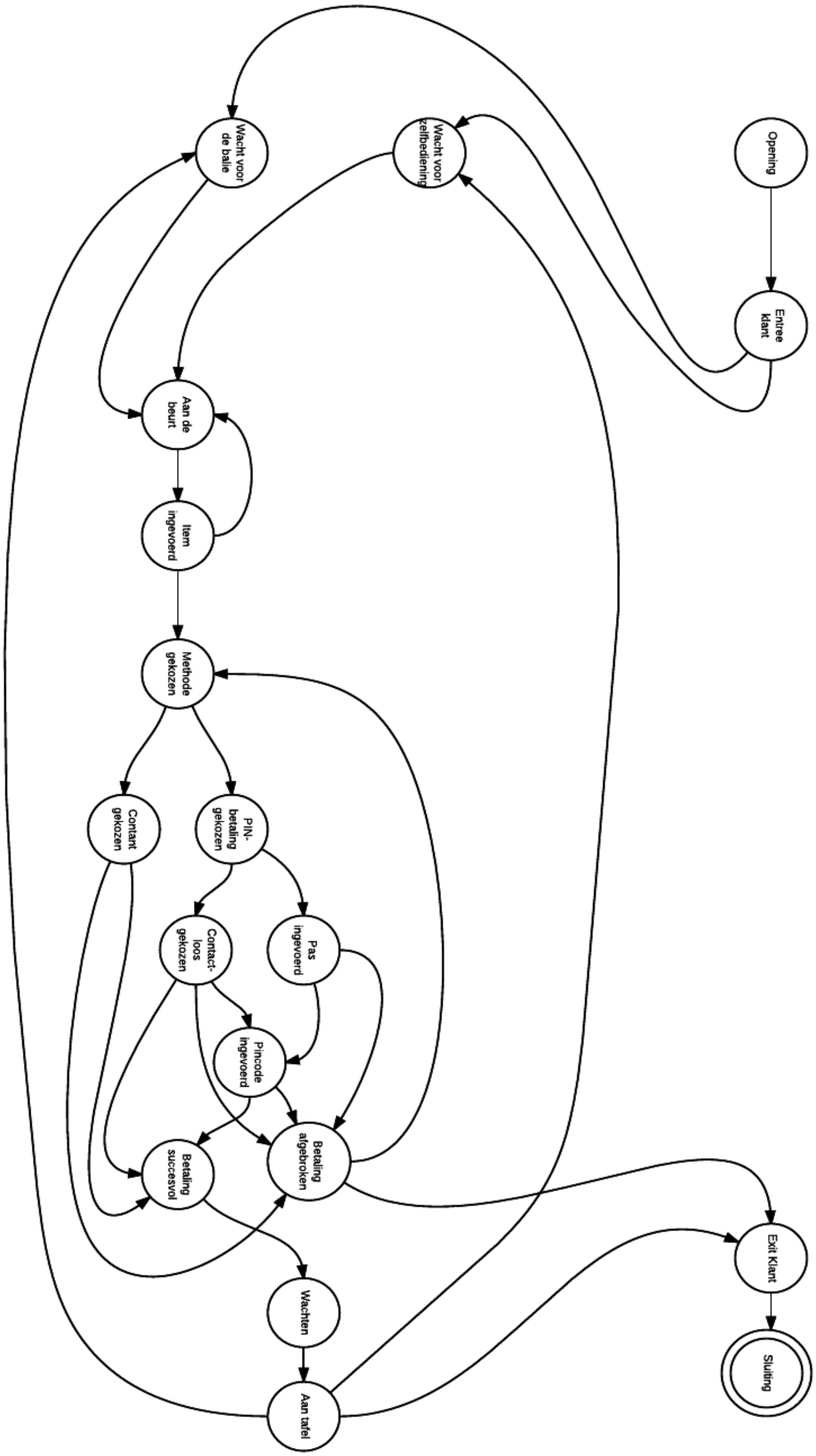
a. Deze gegevens kan je niet opslaan in je huidige model. Waarom niet?

b. Maak hieronder een model van het filiaal van McDonalds, waarin deze gegevens wel opgeslagen kunnen worden (je mag gebruik maken van blokken):

**Opdracht 3**

Dit is nog een keer opdracht 2: "Om een goed beeld te kunnen vormen van het bestellen en betalen ga je nu opslaan waar en hoe vaak klanten bestellen. Je wilt ook graag dat je precies weet hoeveel mensen op een dag zelf bestellen of hoe vaak de 'gewone' kassa wordt gebruikt. Je hoeft niet alleen de gelukte bestellingen op te slaan, maar alle keren dat er iemand gebruikt maakt van een van beide manieren. Daarnaast moet je bijhouden hoe vaak een pin-transactie lukt of mislukt, hoe vaak er contant wordt betaald, hoe vaak er contactloos wordt betaald (dat is dus ook een pinbetaling!). De variabelen van de vorige opdrachten hoeven in dit model niet te worden bijgehouden."

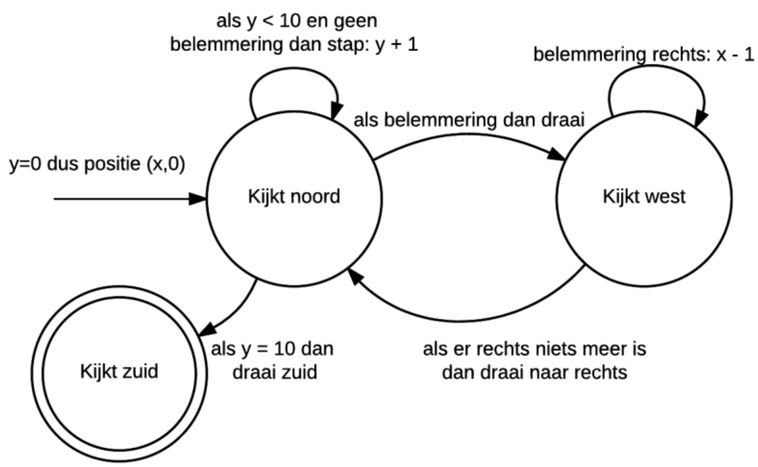
Op de achterkant vindt je een uitwerking. Vul dit schema aan zodat je precies bijhoudt wat hierboven beschreven staat.


Opening
Entree klant
Wacht voor zelfbediening
Wacht voor de balie
Aan de beurt
Item ingevoerd
Methode gekozen
Contant gekozen
PIN-betaling gekozen
Contact-loos gekozen
Pas ingevoerd
Pincode ingevoerd
Betaling succesvol
Betaling afgebroken
Wachten
Aan tafel
Exit Klant
Sluiting

**Opgave 4**

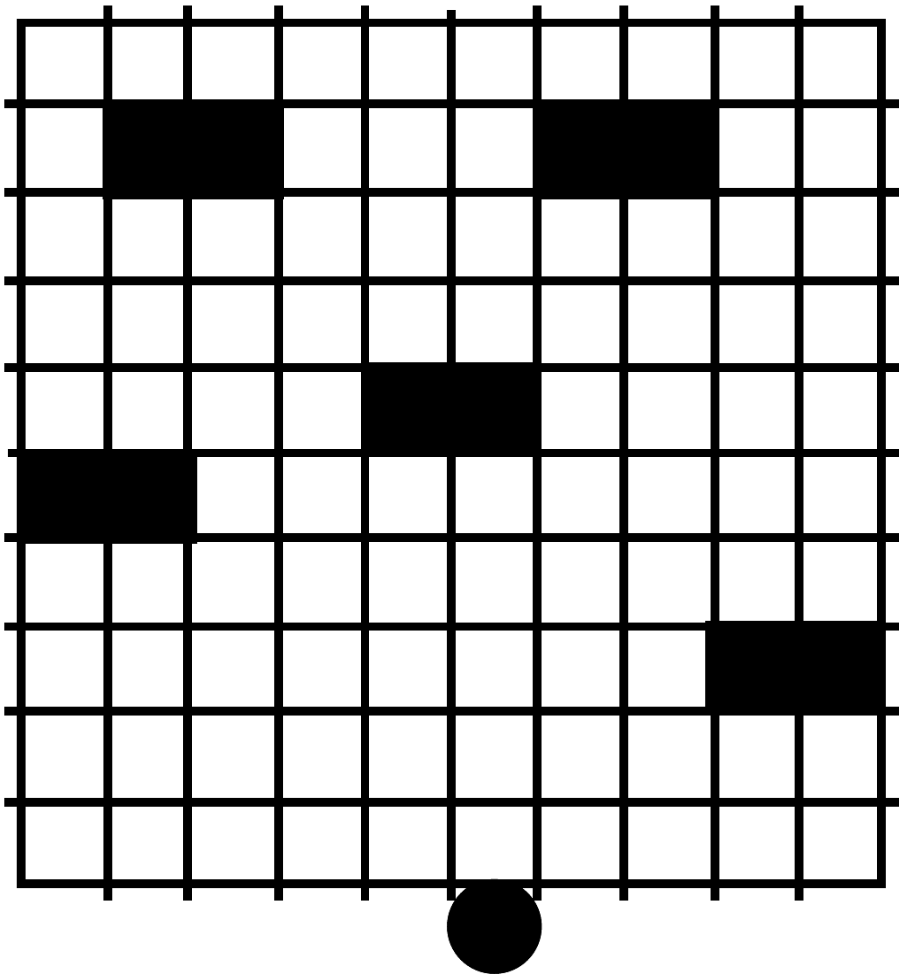
De robot kan een scan uitvoeren. Die scan kijkt naar de sensoren van de robot die afstand meten tussen de robot en een muur of andere belemmering. Naar aanleiding van die scan, zet de robot vervolgens de volgende stap.

De robot kan alleen 90 graden naar links of naar rechts draaien en stappen vooruit zetten. De robot heeft een positie *(x,y)*, zodat de robot weet waar hij is. Bij iedere stap die de robot doet, wordt er een scan uitgevoerd waarbij de robot merkt of er om hem heen belemmeringen zijn. De robot moet 10 plaatsen naar voren bewegen. Als de robot is aangekomen, moet deze omdraaien. Jeroen heeft het volgende model gemaakt:



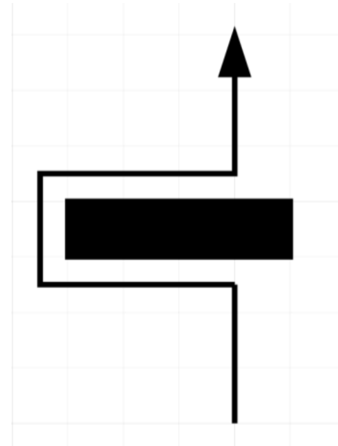
**Opdracht 4a**

Teken hieronder hoe, volgens Jeroens model, de robot zich zal verplaatsen.



**Opdracht 4b**

Jeroen wil echter dat de robot precies 10 stappen boven de startpositie uitkom zal komen (zoals in het plaatje).



Breidt het model uit zodat de robot om zulke blokken heen kan lopen. Je mag aannemen dat je oneindig ver naar links kan blijven lopen.

Als je tijd over hebt, probeer het schema uit te breiden met de situatie dat je niet oneindig ver naar links kunt lopen. Je kunt dan niet altijd meer op een blok heen, door linksaf te slaan.

# Appendix D: Evaluation Form

## Evaluatie

*Vereist

### Ik vond deze les *

1: helemaal niet, 5: helemaal wel

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Leuk | ○ | ○ | ○ | ○ | ○ |
| Interessant | ○ | ○ | ○ | ○ | ○ |
| Moeilijk | ○ | ○ | ○ | ○ | ○ |

### Toelichting

Jouw antwoord

### Hoe beoordeel je in deze les *

1: slecht, 5: goed

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| De hoeveelheid nieuwe informatie | ○ | ○ | ○ | ○ | ○ |
| Moeilijkheidsgraad van de uitleg (1: te makkelijk, 5: te moeilijk) | ○ | ○ | ○ | ○ | ○ |
| Moeilijkheidsgraad van de opgaven | ○ | ○ | ○ | ○ | ○ |

### Toelichting

Jouw antwoord

Wat vond je het makkelijkste gedeelte van de les en waarom? *

Jouw antwoord

Wat vond je het moeilijkste gedeelte van de les en waarom? *

Jouw antwoord

Ik vond de context (werken als consultant voor McDonalds): *
1: helemaal niet, 5: helemaal wel

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Aansprekend | ○ | ○ | ○ | ○ | ○ |
| Realistisch | ○ | ○ | ○ | ○ | ○ |
| Begrijpelijk | ○ | ○ | ○ | ○ | ○ |

Toelichting

Jouw antwoord

Hoe kan je gebruik maken van automaten in de Informatica? *

Jouw antwoord

## Appendix E: Teacher Instruction (version 2)

In this appendix, the teacher instruction can be found. Important parts of the instruction are bold. We have chosen to provide an introduction, but we do not expect the teacher to read this or memorize it by hard. It is important though that the teacher presentation is almost equal to this, since a lot of sentences are important for the process or analysis afterwards. A PowerPoint presentation for the lesson is also provided (Appendix G). Instructions on how to use this, are underlined. Since the pilot is given in a Dutch class, the teacher instruction is in Dutch.

The students must be familiar with the concept of variables or the teacher should explain this carefully when they are used.

### a) Introductie (0 – 10)

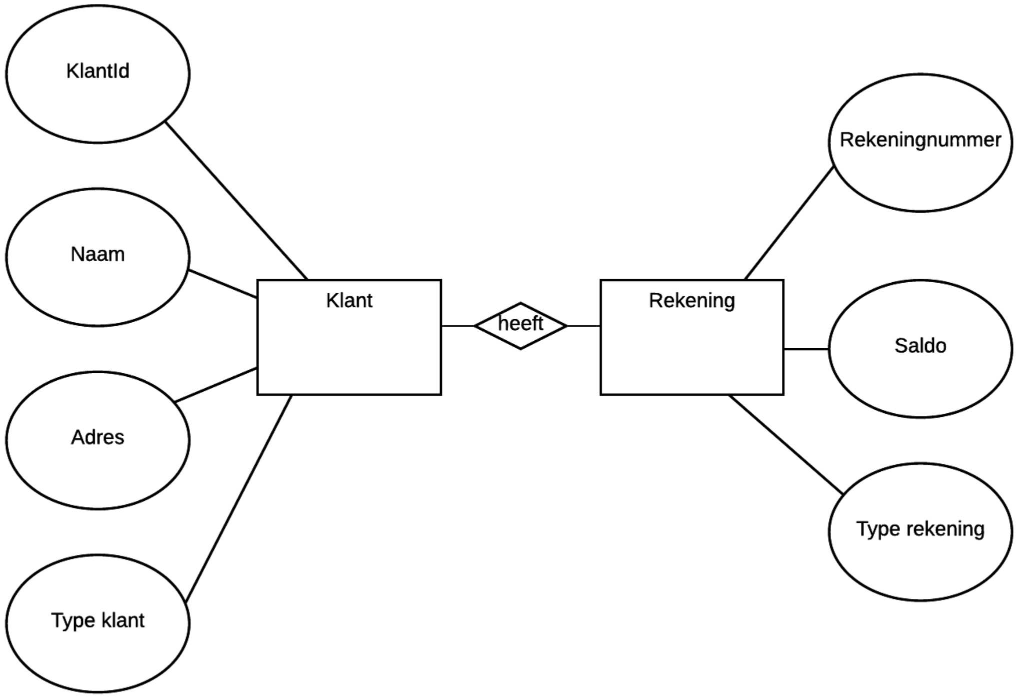
De opgaven moeten voorafgaand aan de les nog niet worden uitgedeeld. De docent kan vragen tijdens of na bepaalde onderdelen toestaan.

<u>PowerPoint slide 1</u>

Docent: vandaag gaan we een model maken *(optioneel: docent geeft informatie hoe dit in het curriculum past)*. Modellen worden in Informatica bijvoorbeeld gebruikt om schematisch weer te geven hoe systemen werken. Door ergens een model van te maken, kunnen veel mensen, zelfs zij die geen verstand hebben van Informatica, toch vaak meepraten over informatica toepassingen. Vandaag maken we gebruik van een speciaal soort modellen: eindige automaten. Automaten kunnen onder andere goed worden gebruikt voor het programmeren. **Met name bij het opstellen van algoritmes worden vaak automaten gebruikt.** Automaten beschrijven altijd een gedeelte van de werkelijkheid, we kunnen nooit alles in een automaat weergeven. Een automaat zou er ook heel onoverzichtelijk van worden. We spreken daarom af dat we alleen in een automaat opslaan wat we echt belangrijk is.

<u>PowerPoint slide 2</u>

Docent: neem bijvoorbeeld dit model:




**Docent: waar gaat dit model over?**

Dit model gaat over een bank. Een bank heeft klanten. De klanten hebben bepaalde eigenschappen, maar iedere klant heeft ook een rekening. Die rekening heeft zelf ook weer allerlei eigenschappen. De docent kan dit naar eigen inzicht uitleggen. Type klant is bijvoorbeeld particulier of bedrijf terwijl een type rekening het verschil tussen een spaar- of betaalrekening weer kan geven. Het grote voordeel is dat bijna iedereen een model ongeveer snapt.

**Docent introduceert de volgende context**:

PowerPoint slide 3

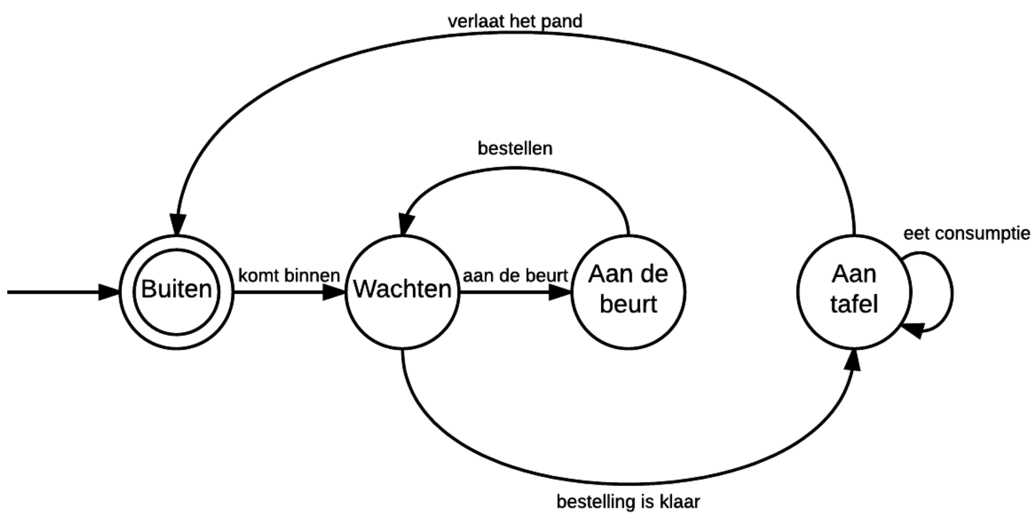
Drie jaar geleden ben je een eigen onderzoeksbureau begonnen. Inmiddels heb je je eerste echt grote opdracht binnen. McDonalds heeft jou namelijk gevraagd om voor hun een onderzoek uit te voeren. Ze willen graag een onafhankelijke evaluatie van enkele veranderingen die ze de laatste tijd hebben doorgevoerd in hun filialen. Zo hebben ze in de afgelopen tijd overal contactloos betalen mogelijk gemaakt en zijn er de zogenaamde 'bestelpalen' toegevoegd waar mensen zelf kunnen bestellen. Ze verwachten dat door het contactloos betalen mensen sneller kunnen afrekenen en er dus minder lange wachtrijen zijn. Met de 'bestelpalen' kunnen er meer mensen tegelijk bestellen, wat minder wachttijd oplevert. Daarnaast zijn er geen McDonalds medewerkers nodig om de bestelling op te nemen, waardoor McDonalds ook verwacht te besparen op medewerkers.

Om deze veranderingen te kunnen evalueren, besluit je een automaat op te stellen. Deze automaat kan je dan presenteren aan de managers bij McDonalds en je verwacht duidelijk bij te kunnen bijhouden efficiënt de vernieuwingen hopelijk zijn.

Vandaag ben je afgereisd naar een filiaal van McDonalds. Je arriveert, drinkt een bak koffie en kijkt goed rond. Na een uur heb je een globaal van de meeste klanten. De klanten komen binnen, plaatsen hun bestelling, eten deze op en vertrekken als ze uitgegeten zijn. Je besluit om heel ongedetailleerd het eerste gedeelte van het model alvast op te stellen, door globaal het gedrag van de klanten te beschrijven.

**Leerlingen luisteren naar de introductie.**

<u>PowerPoint slide 4</u>




**Docent: dit is een automaat. Iedere cirkel of bol noemen we een <u>toestand</u>. We beginnen bij de pijl die uit het niets lijkt te komen en we tekenen die meestal helemaal links.**

Vervolgens:

> De docent beschrijft het proces aan de hand van de powerpoint en zoals in de introductie aangegeven, door als volgt te beginnen: **"Vanuit deze <u>begintoestand</u> gaan we…"**
> Het is belangrijk dat de docent het jargoneske woord transitie *niet* noemt. Simpelere varianten als 'overgang' mogen wel.

**Docent: een dubbel omcirkelde toestand noemen we een <u>eindtoestand</u>** en als we hier zijn, kunnen we stoppen. Dat hoeft echter niet per se, we kunnen ook weer de automaat nogmaals ingaan. Dus als iemand naar buiten is gegaan, kan hij of zij terug komen en het hele proces nogmaals doorlopen.

## b) Klassikale opdracht I (10 – 15)

Docent: dit autoaat bevat helaas een fout. We willen geen uitbreiding van het schema, maar alleen de fout in deze automaat. Kunnen jullie die vinden?

**Leerlingen geven suggesties hoe de fout kan worden opgelost.**

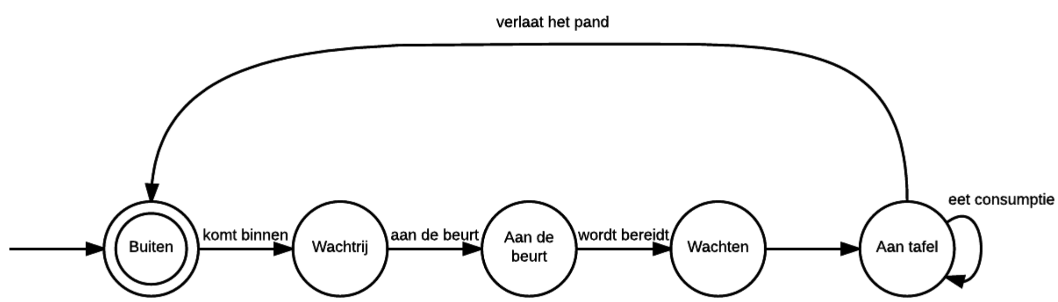
Docent:

Als leerlingen er niet achter komen, kan de docent vragen om nogmaals stap voor stap door de automaat te gaan. De docent kan ook vragen of er alternatieve manieren zijn om door de automaat te gaan.

**Uitwerking:**

PowerPoint slide 5

De fout is dat in dit model met binnen kan komen, wacht, eet en vervolgens naar buiten gaat. Dit is echter geen correcte weergave van de werkelijkheid. De correcte versie van dit model is deze:



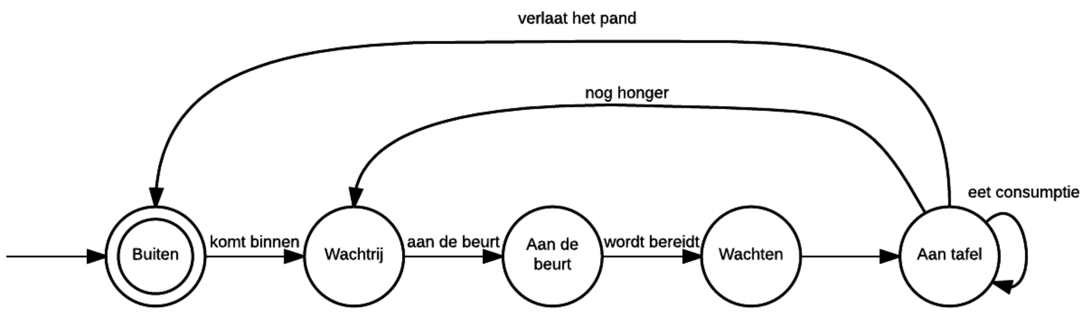

## c) Klassikale opdracht II (15-20)

Docent: mensen bestellen soms meerdere keren voordat ze weer naar buiten gaan. Ze hebben nog niet genoeg gegeten of ze willen een toetje. Voeg dit toe aan de automaat.

**Leerlingen geven suggesties voor de toevoeging.**

**Uitwerking:**

PowerPoint slide 6



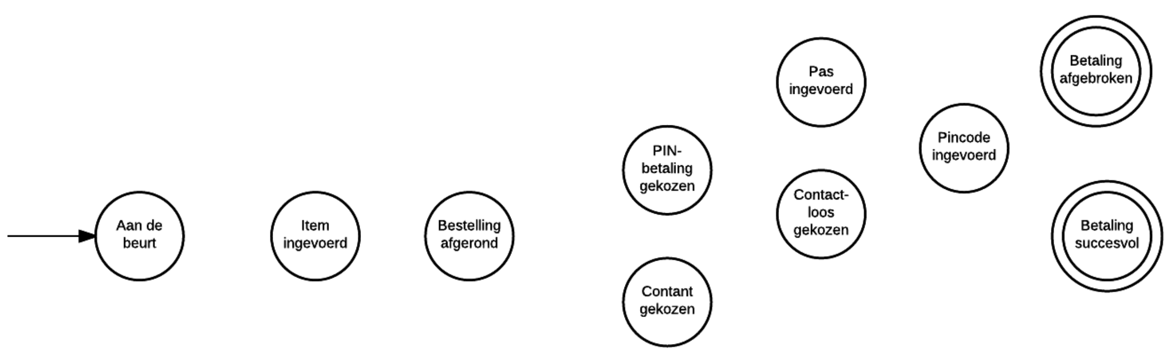

Docent: nu is het tijd voor een aantal individuele opdrachten. De docent vertelt de opdracht en laat die daarna op het scherm zien.

**Het is voor mijn scriptie van belang dat duidelijk geïnstrueerd wordt dat leerlingen de uitwerking niet kopiëren op hun opgavenformulier. De uitwerkingen worden beschikbaar gesteld via ItsLearning, zodat ze altijd achteraf nog de juiste uitwerkingen kunnen bekijken.**

### d) Individuele opdracht I (20-30)

Docent deelt opdracht 1 uit.

De docent vraagt de leerlingen dit enkele minuten alleen te doen. Hierna kunnen leerlingen dit in paren vergelijken en het model nog aanpassen. Hierna kan de docent vragen of iemand aanvullingen kan vinden of fouten ziet in het getekende schema. Het model bevat 17 pijlen, de docent kan dit gebruiken om de leerlingen te stimuleren. Bijvoorbeeld als iemand verwacht klaar te zijn en slechts 13 pijlen heeft, kan de docent hem of haar erop wijzen dat er nog wel het een en ander moet gebeuren.




### e) Klassikale bespreking (30-35)

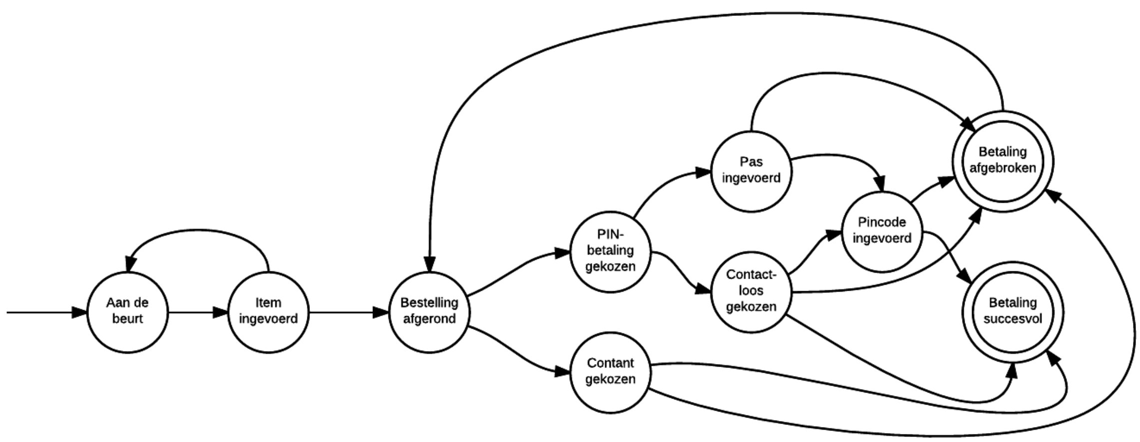
<u>PowerPoint slide 7</u>

In deze slide tekenen de leerlingen hun oplossing(en).

De docent laat iemand naar voren komen die denkt de juiste uitwerking te hebben en deze wordt klassikaal besproken, tot het getekende model overeenkomt met de uitwerking.

***Uitwerking:***

<u>Powerpoint slide 8</u>




<u>PowerPoint slide 9 (dezelfde als 5)</u>

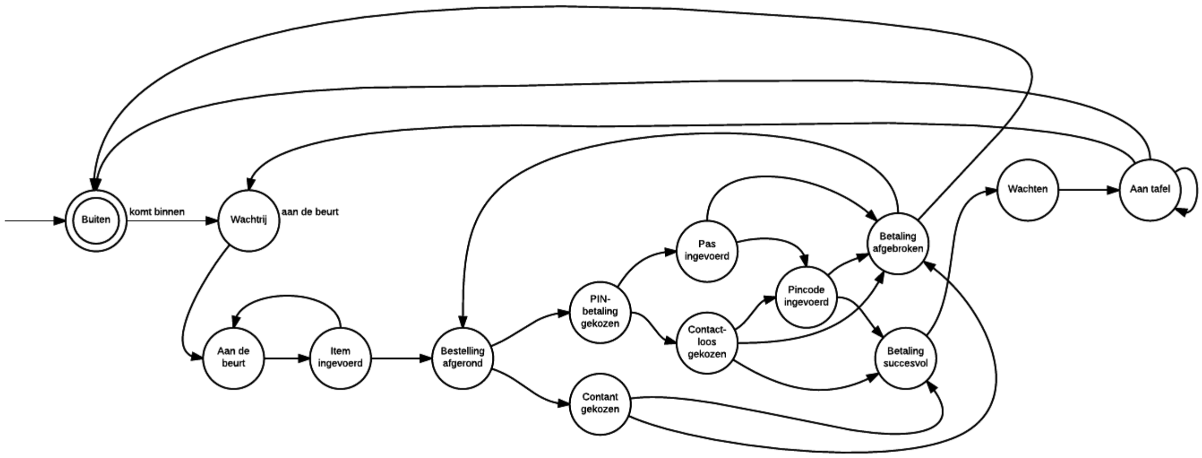
De docent vraagt aan de studenten of iemand weet hoe de nieuwe automaat in deze automaat past. We hebben namelijk de toestand ‘Aan de beurt’ vervangen door deze automaat van ‘Bestellen en betalen’.

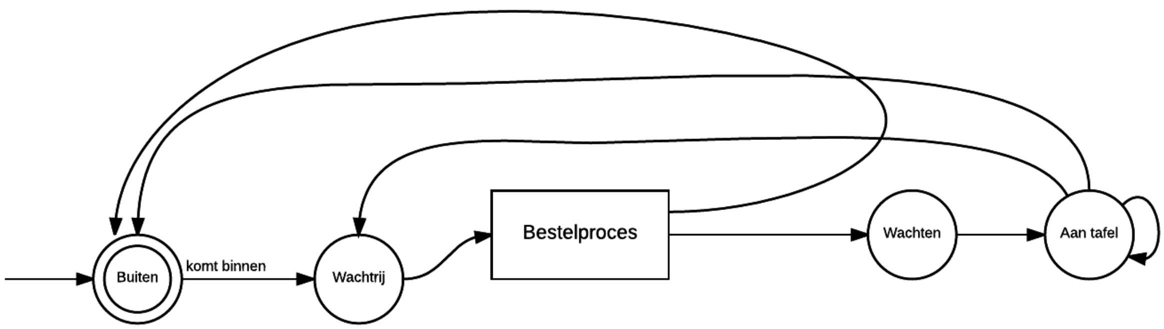
***<u>Uitwerking: PowerPoint slide 10</u>***

De docent toont daarna hoe we gedeeltes uit een automaat kunnen vereenvoudigen in een grote tekening, door het te vervangen door een vierkant blok met de in- en uitgangen die ook in het schema zichtbaar waren.

<u>PowerPoint slide 11</u>

Om duidelijk te maken hoe dit blok het gedeelte vervangt kan de docent de leerlingen vragen om de transitie te beschrijven, met andere woorden, wat kan er op de pijlen staan die aan het blok verbonden zijn.




De docent vraagt aan de leerlingen wat het verschil is tussen ‘Aan de beurt’ en ‘Bestelproces’ (slide 11 en 12). Het antwoord luidt als volgt: bestelproces is een blok waarin eigenlijk een automaat in zit. ‘Aan de beurt’ is slechts een toestand. Het gedetailleerder beschrijven van deze toestand, heeft ertoe geleidt dat we ook een pijl hebben van ‘Bestelproces’ naar ‘Buiten’.

### f) Klassikale opdracht III (35-45)

**Docent geeft een korte introductie**: inmiddels hebben we al een best uitgebreide automaat gemaakt. Toch zegt dit nog vrij weinig. We gaan dit gedrag daarom uitbreiden met gegevens die nodig zijn om een goede beslissing te kunnen maken voor McDonalds. We gaan daarom vastleggen *hoeveel* menu-items iedere klant toevoegt aan zijn of haar bestelling tijdens één bezoek aan het filiaal.

PowerPoint slide 13

De studenten bedenken de oplossing en tekenen dit in de slide.

Afhankelijk van de tijd kan het volgende gedeelt op twee manier worden gedaan:

**Lang:**

Ze worden geleid in het proces door de docent. Deze vraagt eerst om welk gedeelte van het model het gaat. Er kan namelijk een ‘Betaalproces’-blok worden getekend, maar het bestelproces moet wel aanwezig blijven. Het is zaak dat de leerlingen **minimaal** dit blok vinden, maar ze mogen ook het blok 'Betalen en eten' vinden waarbij de twee rechter toestand ook meegenomen worden in het blok. De docent kan leerlingen motiveren verder te denken als ze een kleiner gedeelte samenpakken, bijv. "Kunnen we wellicht nog meer samenvatten?"

**Kort:**

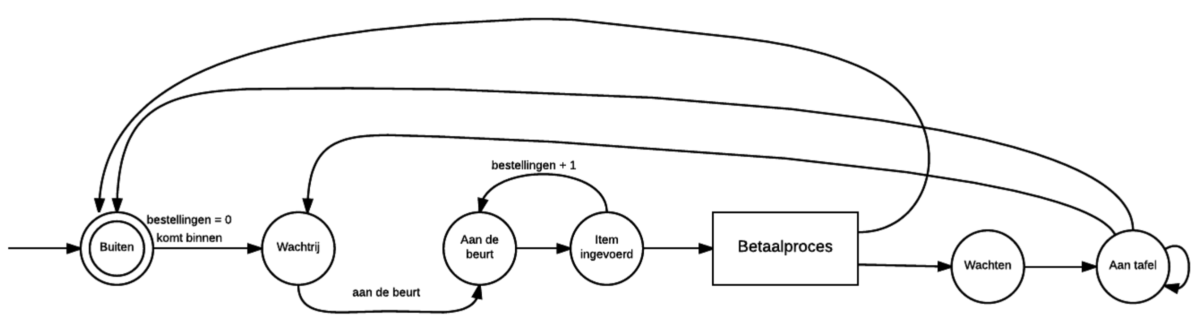
Een blok is een automaat. We tekenen dit nu zo, omdat we dit gedeelte van de automaat nu niet interessant vinden.

Hierna wordt aan de leerlingen gevraagd of ze een manier kunnen bedenken hoe de hoeveelheid menu-items kan worden opgeslagen. Afhankelijk van de voorkennis van een bepaalde klas kan de docent ze naar eigen inschatting leiden naar dit proces (bijv. weten ze wat variabelen zijn).

Er zijn twee mogelijkheden waarbij de onderstaande te prefereren is, aangezien een bestelling pas definitief is als die is ingevoerd. De andere optie is namelijk om de bestelling variabele te updaten van 'Aan de beurt' naar 'Item ingevoerd'. De klas kan beide oplossingen proberen te vinden, zodat men kan zien waarom het een beter is dan het ander. De kans is aanwezig dat men probeert de variabele in een toestand te stoppen en die daar te verhogen. De docent kan vragen (indien er tijd is) of iemand snapt waarom dit niet kan. Er staat nu geen tekst op de pijl maar er kan gevraagd worden wat er op de pijl zou kunnen staan. Eventueel kan anders de docent zelf uitleggen dat het 'Aan de beurt zijn' slechts een beschrijving is van de wereld, niet een daadwerkelijke handeling. Het is van belang dat het woord state of transitie niet letterlijk wordt gebruikt, maar dit verduidelijkt wordt door de voorbeeldzinnen.

Ook is belangrijk dat ze de bestelling op 0 zetten als de klant binnenkomt, aangezien een klant anders twee keer binnen kan komen en dit doorgeteld wordt terwijl het een ander bezoek is. Overigens is de notatie hier niet heel belangrijk, dit mag ook gewoon worden beschreven als 'bestellingen wordt 0'. Dit kan ook worden geplaatst op de pijlen die naar *Buiten* wijzen, maar dan weten we niet 100% zeker dat *bestellingen* tijdens het eerste bezoek wel echt 0 zijn, dus is de eerste de gewenste variant (maar de tweede niet per se fout).

***Uitwerking: PowerPoint slide 14:***



### g) Individuele opdracht II (55 – 70)

Docent: we hebben niet veel aan de informatie van 1 klant. We willen graag ook gegevens bijhouden als er meerdere mensen komen. We hadden eerst een automaat van een klant, maar nu hebben we dit aangepast naar een automaat voor een heel McDonalds filiaal.

De docent deelt opdracht 2 uit en neemt opdracht 1 in. Leerlingen worden aangespoord zo veel mogelijk zelf dit op te lossen, maar mogen wel overleggen.

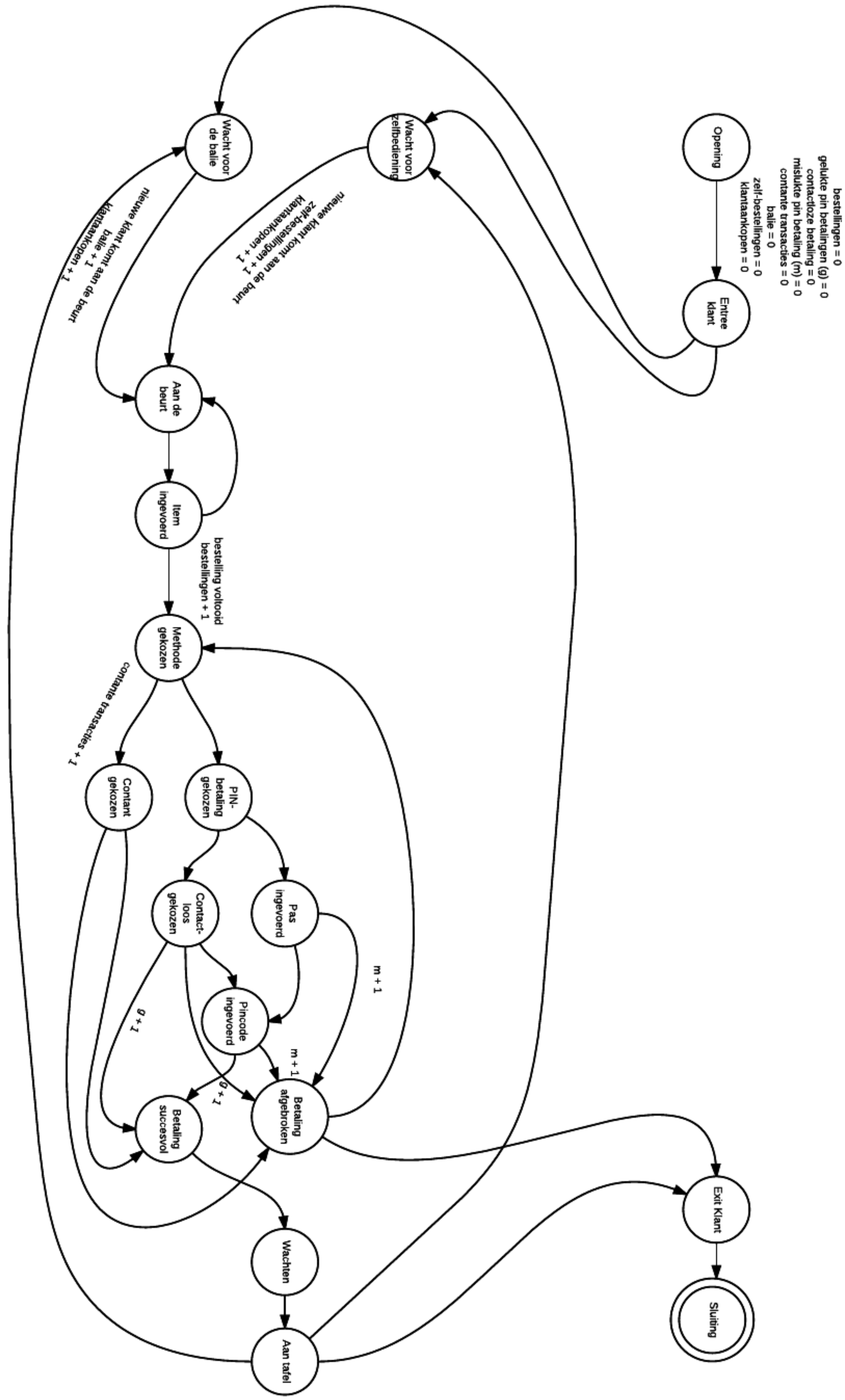
### h) Klassikale bespreking (70 – 75)

PowerPoint slide 15

De docent laat iemand naar voren komen die denkt de juiste uitwerking te hebben en deze wordt besproken, tot de getekende automaat overeenkomt met de uitwerking. Alle alternatieve opties worden besproken als ze aan bod komen, inclusief afwegingen waarom op welke transitie bepaalde variabelen worden opgehoogd.

***Uitwerking: PowerPoint slide 16***

Inderdaad kunnen al deze gegevens in één automaat komen:


bestellingen = 0
gelukte pin betalingen (g) = 0
contactloze betaling = 0
mislukte pin betaling (m) = 0
contante transacties = 0
balie = 0
zelf-bestellingen = 0
klantaankopen = 0
Opening
Entree klant
Wacht voor zelfbediening
Wacht voor de balie
nieuwe klant komt aan de beurt
zelf-bestellingen + 1
klantaankopen + 1
nieuwe klant komt aan de beurt
balie + 1
klantaankopen + 1
Aan de beurt
Item ingevoerd
bestelling voltooid
bestellingen + 1
Methode gekozen
contante transacties + 1
Contant gekozen
PIN-betaling gekozen
Pas ingevoerd
Contact-loos gekozen
Pincode ingevoerd
m + 1
m + 1
g + 1
g + 1
Betaling afgebroken
Betaling succesvol
Wachten
Aan tafel
Exit Klant
Sluiting

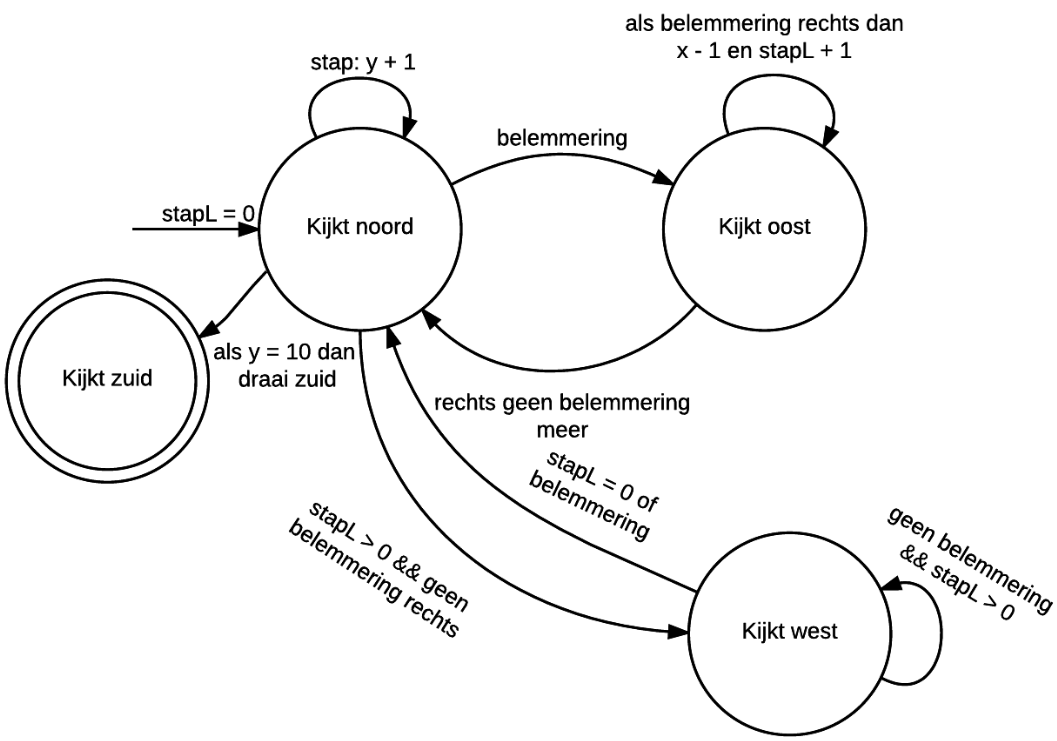
### i) Individuele opdracht III (75 – 85)

**Deze opdracht moet echt individueel opgelost worden.**

Docent: je hebt genoeg gezien in de McDonalds voor een voorlopig advies. Je bent gevraagd door Jeroen, een vriend die bij de TU Delft, die onderzoek doet naar robots om hem te helpen. Hij werkt mee aan een groot project dat uiteindelijk een zorg-robot moet opleveren. Een robot die bij hulpbehoevende mensen in huis wordt geplaatst en daar allerlei taken gaat doen die de persoon zelf niet meer kan. Jeroen moet ervoor zorgen dat de robot zich in ieder mogelijke ruimte moet kunnen bewegen.

De docent instrueert de leerlingen opgave 4 te maken. Deze opgave bestaat uit twee delen. Opgave 4 en opgave 4a worden uitgedeeld en studenten krijgen pas het tweede deel als ze de eerste hebben ingeleverd. Aangezien dit gaat om het beschrijven van een automaat is dit voor de bespreking geen probleem.

**Uitwerking:**

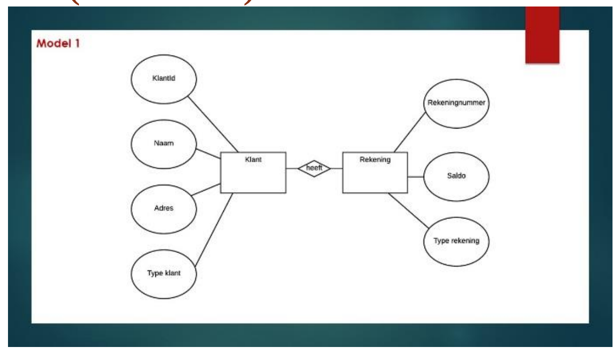



### j) Klassikale bespreking en afsluiting

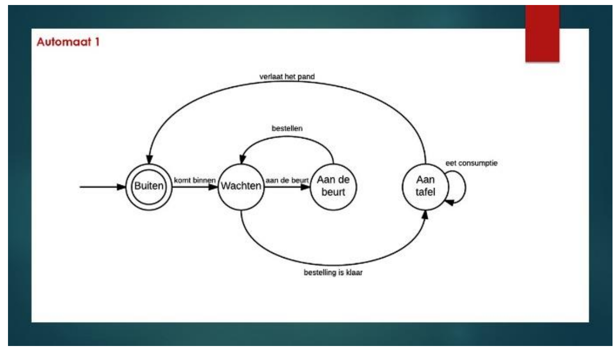
<u>PowerPoint slide 17</u>

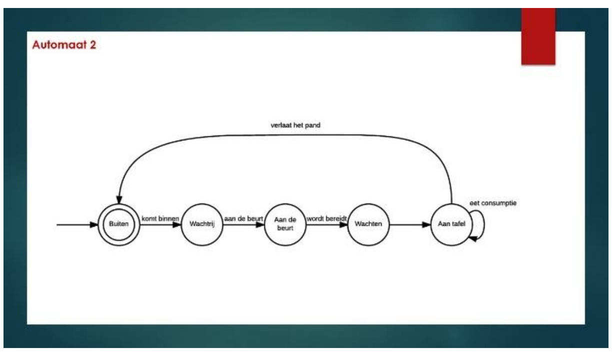
De docent bespreekt de laatste opgave klassikaal op dezelfde manier als de voorgaande keren en laat de leerlingen weer een uitwerking tekenen. Hierna toont de docent de uitwerking: <u>PowerPoint slide 18</u> en bespreekt deze kort. De docent sluit af met een samenvatting: we zijn gekomen aan het einde van de les. In deze les hebben jullie geleerd hoe je automaten kan maken. Jullie hebben een automaat getekend van een McDonalds filiaal en van het looppatroon van een robot. Jullie hebben gezien hoe automaten kunnen bijdragen aan het inzichtelijk maken van allerlei situaties, wat ook met name in Informatica gebeurt. Inmiddels kunnen jullie een simpel automaat opstellen met een begintoestand, eindtoestand en normale toestanden. Daarnaast kunnen jullie een model ook gebruiken om bepaalde gegevens te meten door variabelen te gebruiken en deze te veranderen bij een toestandsverandering.

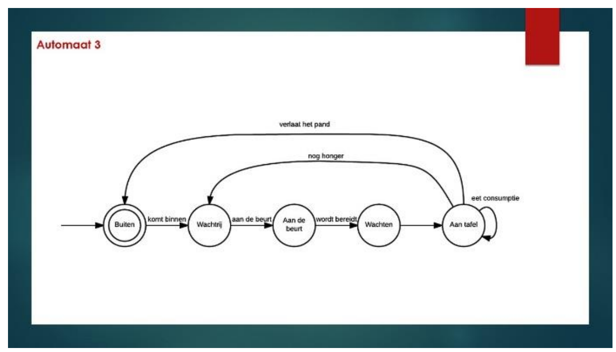
# Appendix F: PowerPoint Presentation (version 2)

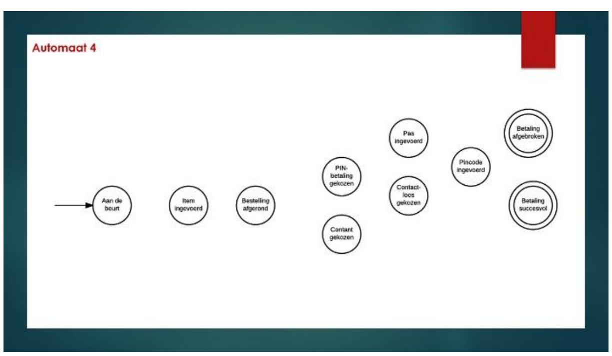



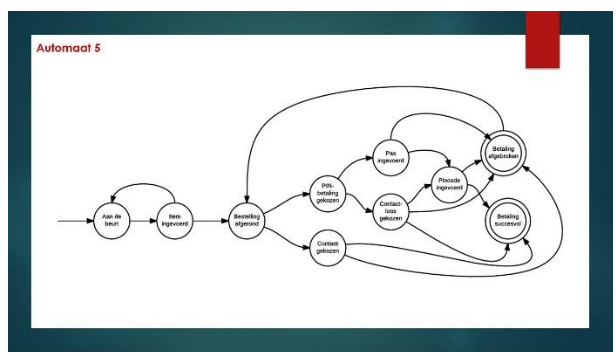


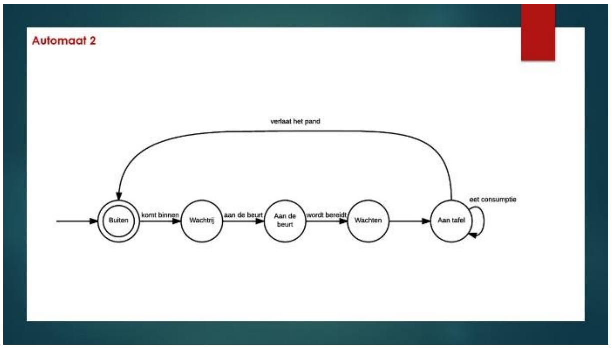


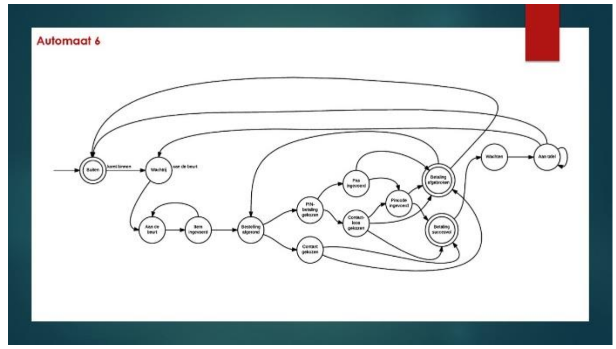



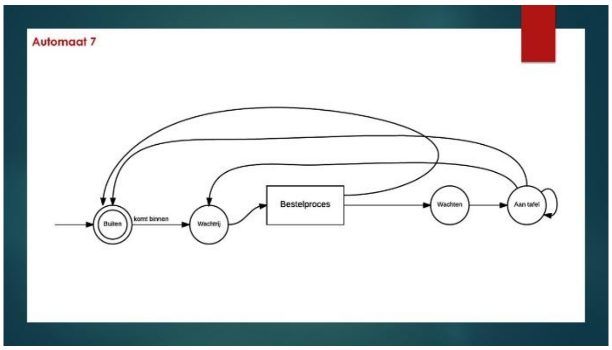



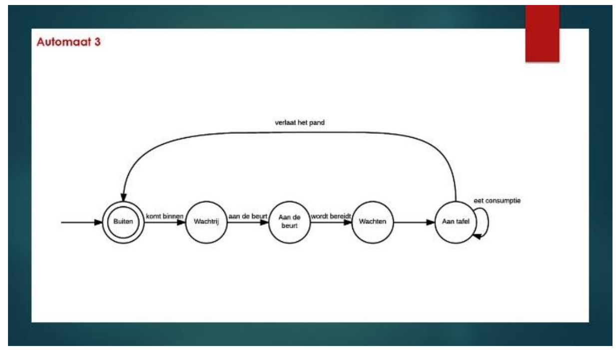



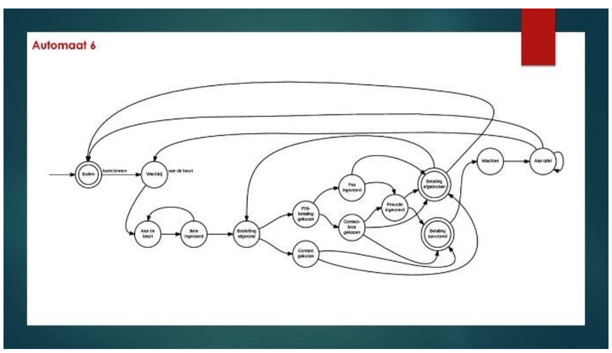



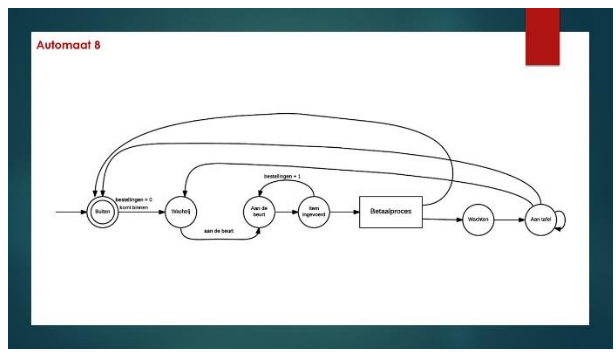

Automaat 2
verlaat het pand
Buiten
komt binnen
Wachtrij
aan de beurt
Aan de beurt
wordt bereidt
Wachten
Aan tafel
eet consumptie

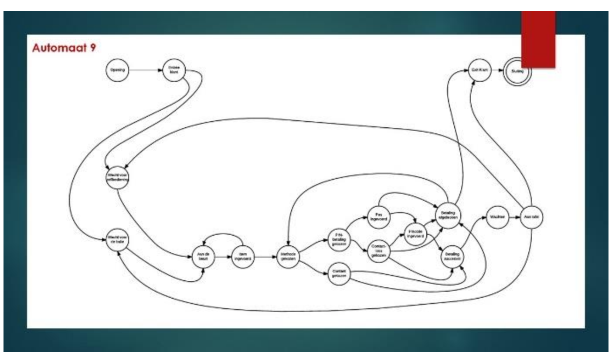

Automaat 6

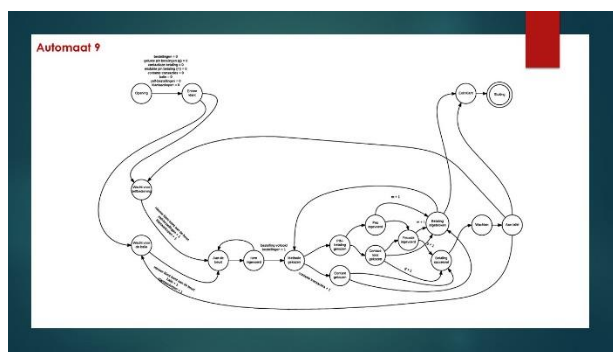

Automaat 7
Buiten
komt binnen
Wachtrij
Bestelproces
Wachten
Aan tafel

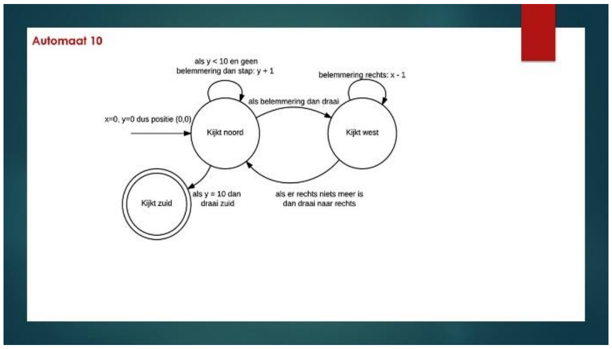

Automaat 3
verlaat het pand
Buiten
komt binnen
Wachtrij
aan de beurt
Aan de beurt
wordt bereidt
Wachten
Aan tafel
eet consumptie

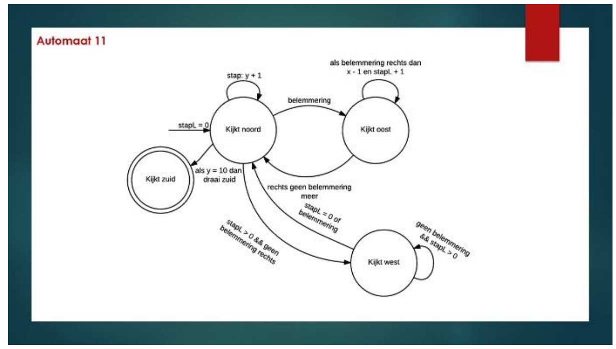

Automaat 6


Automaat 8
bestellingen = 0
komt binnen
Buiten
Wachtrij
aan de beurt
bestellingen + 1
Aan de beurt
Item ingevoerd
Betaalproces
Wachten
Aan tafel


Automaat 9


Automaat 9


Automaat 10
als y < 10 en geen
belemmering dan stap: y + 1
belemmering rechts: x - 1
als belemmering dan draai
x=0, y=0 dus positie (0,0)
Kijkt noord
Kijkt west
Kijkt zuid
als y = 10 dan
draai zuid
als er rechts niets meer is
dan draai naar rechts

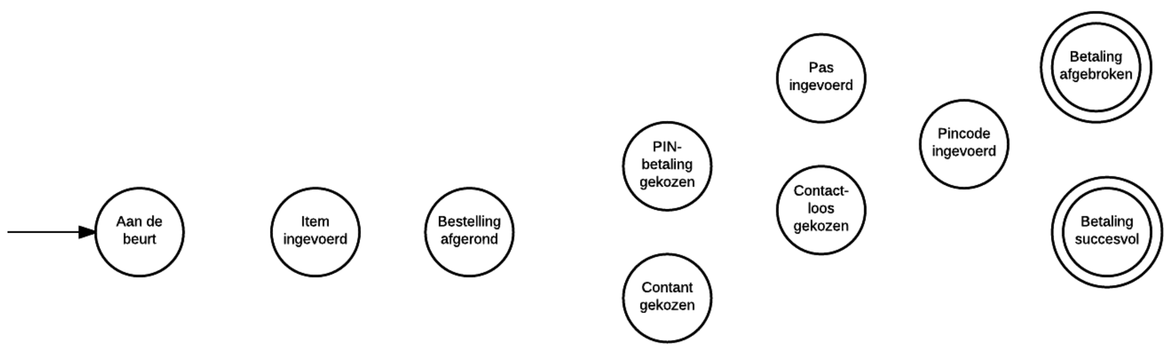

Automaat 11
als belemmering rechts dan
x - 1 en stapL + 1
stap: y + 1
belemmering
stapL = 0
Kijkt noord
Kijkt oost
Kijkt zuid
als y = 10 dan
draai zuid
rechts geen belemmering
meer
stapL = 0 of
belemmering
stapL > 0 && geen
belemmering rechts
geen belemmering
&& stapL > 0
Kijkt west

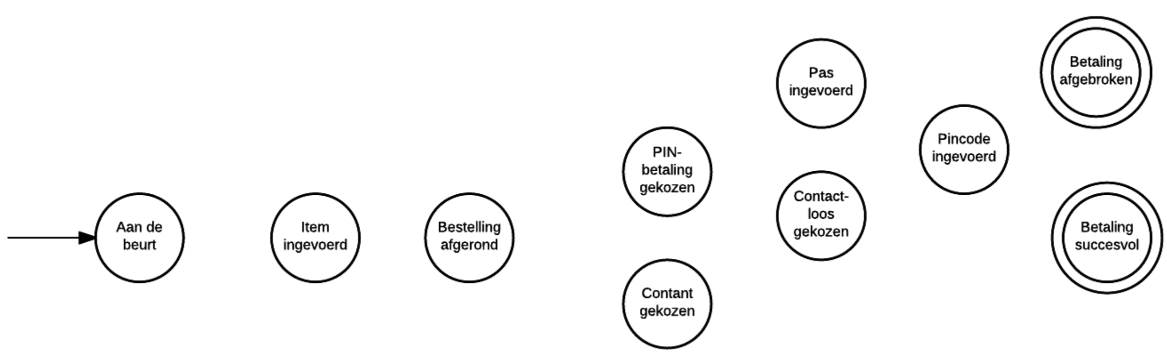
# Appendix G: Student Handout (version 2)

This appendix contains the handout that is provided to the students at the beginning of the lesson. It contains the individual assignments of the lesson and it will be collected and used to evaluate afterwards.

**Opdracht 1**

Omdat je vooral geïnteresseerd bent in hoe klanten precies bestellen en deze bestelling vervolgens afrekenen, besluit je dit iets nauwkeuriger weer te geven. Je hebt inmiddels de toestanden beschreven en gaat nu de automaat afmaken. Gebruik de eerste versie als kladversie en maak de tweede als je denkt klaar te zijn.

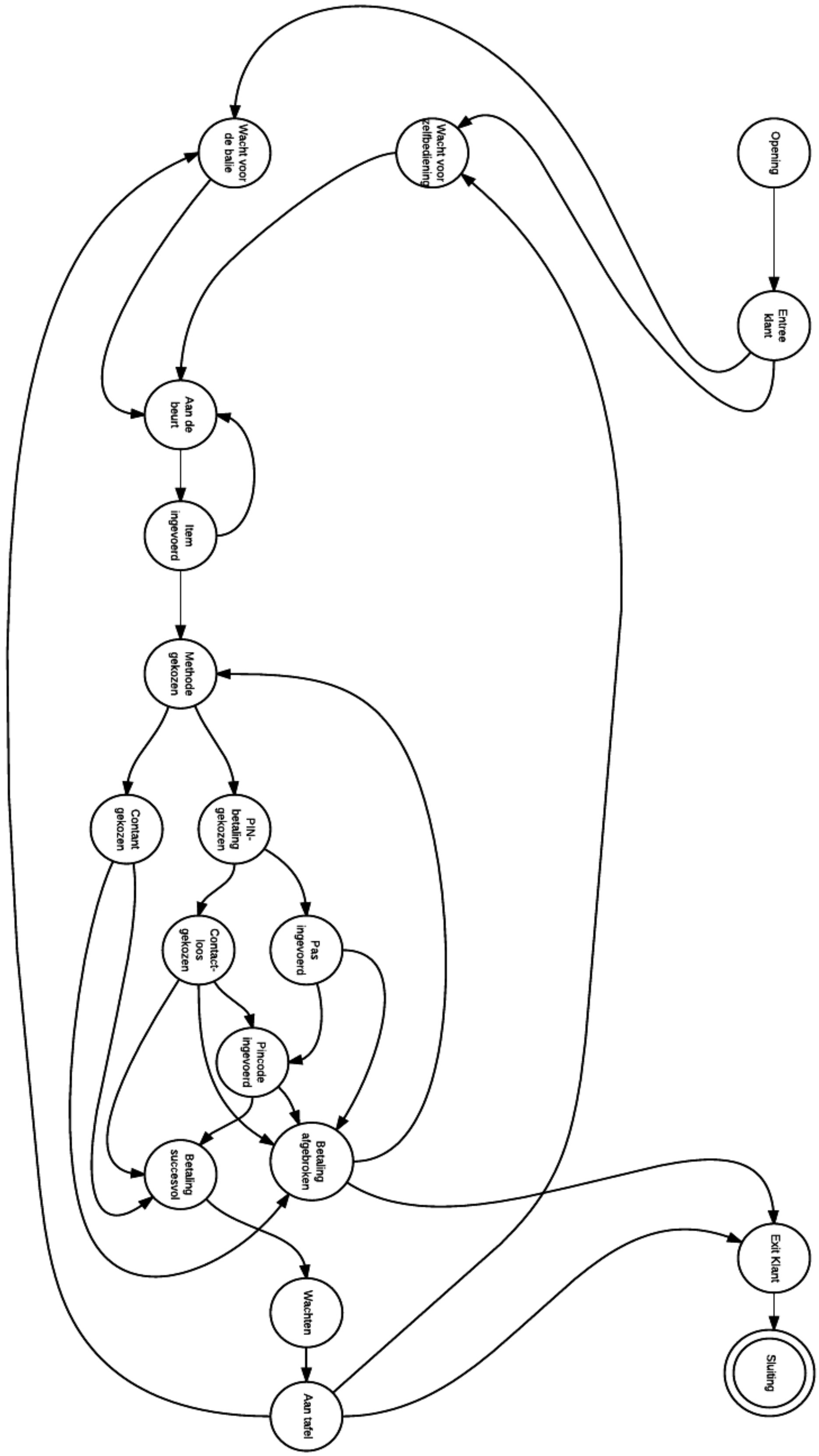


**Opdracht 2**

Om een goed beeld te kunnen vormen van het bestellen en betalen ga je nu opslaan waar en hoe vaak klanten bestellen. Je wilt ook graag dat je precies weet hoeveel mensen op een dag zelf bestellen of hoe vaak de 'gewone' kassa wordt gebruikt. Je hoeft niet alleen de gelukte bestellingen op te slaan, maar alle keren dat er iemand gebruikt maakt van een van beide manieren. Daarnaast moet je bijhouden hoe vaak een pin-transactie lukt of mislukt, hoe vaak er contant wordt betaald, hoe vaak er contactloos wordt betaald (dat is dus ook een pinbetaling!). De variabelen van de vorige opdrachten hoeven in dit model niet te worden bijgehouden.

Op de achterkant vindt je een uitwerking. Vul deze automaat aan zodat je precies bijhoudt wat hierboven beschreven staat.

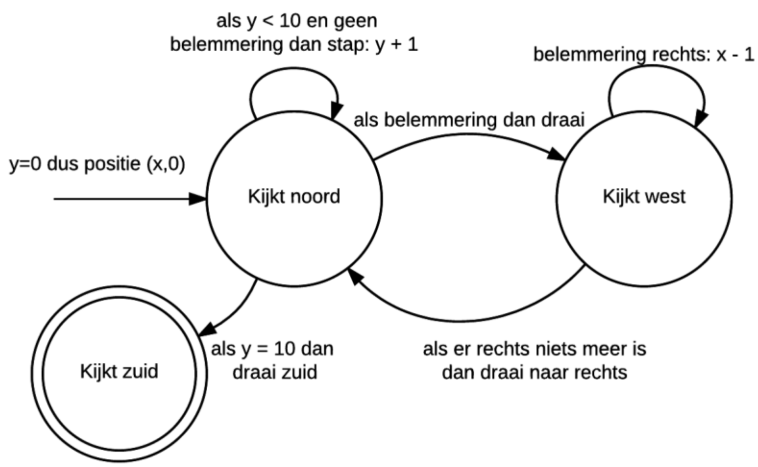


Opening
Entree klant
Wacht voor zelfbediening
Wacht voor de balie
Aan de beurt
Item ingevoerd
Methode gekozen
Contant gekozen
PIN-betaling gekozen
Contact-loos gekozen
Pas ingevoerd
Pincode ingevoerd
Betaling succesvol
Betaling afgebroken
Wachten
Aan tafel
Exit Klant
Sluiting

**Opgave 3**

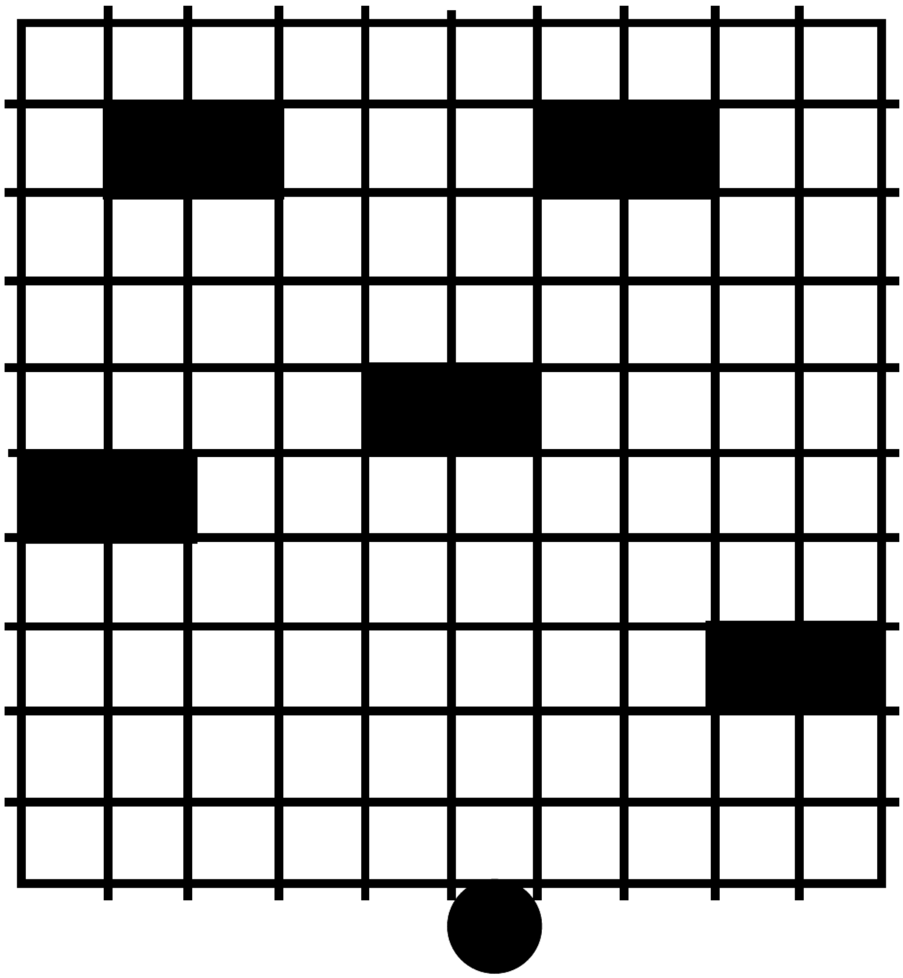
De robot kan een scan uitvoeren. Die scan kijkt naar de sensoren van de robot die afstand meten tussen de robot en een muur of andere belemmering. Naar aanleiding van die scan, zet de robot vervolgens de volgende stap.

De robot kan alleen 90 graden naar links of naar rechts draaien en stappen vooruit zetten. De robot heeft een positie *(x,y)*, zodat de robot weet waar hij is. Bij iedere stap die de robot doet, wordt er een scan uitgevoerd waarbij de robot merkt of er om hem heen belemmeringen zijn. De robot moet 10 plaatsen naar voren bewegen. Als de robot is aangekomen, moet deze omdraaien. Jeroen heeft de volgende automaat gemaakt:



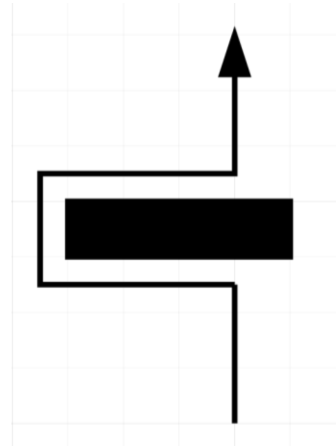
**Opdracht 4a**

Teken hieronder hoe, volgens Jeroens model, de robot zich zal verplaatsen.



**Opdracht 4b**

Jeroen wil echter dat de robot precies 10 stappen boven de startpositie uitkom zal komen (zoals in het plaatje).



Breidt het model uit zodat de robot om zulke blokken heen kan lopen. Je mag aannemen dat je oneindig ver naar links kan blijven lopen.

Als je tijd over hebt, probeer het schema uit te breiden met de situatie dat je niet oneindig ver naar links kunt lopen. Je kunt dan niet altijd meer op een blok heen, door linksaf te slaan.