Masters Thesis Information Sciences
Radboud University Nijmegen

Braille in Mathematics Education

Marc Bitter

April 4, 2013

Supervisors
Prof.Dr.Ir. Theo van der Weide
Dr. Henny van der Meijden

IK183

Abstract

This research project aimed to make improvements to the way blind learners in the Netherlands use mathematics in an educational context. As part of this research, contextual research in the field of cognition, braille, education, and mathematics was conducted. In order to compare representations of mathematics in braille, various braille codes were compared according to set criteria. Finally, four Dutch mathematics curricula were compared in terms of cognitive complexity of the mathematical formulas required for the respective curriculum.

For this research, two main research methods were used. A literature study was conducted for contextual information regarding cognitive aspects, historic information on braille, and the education system in the Netherlands. Interviews with experts in the field of mathematics education and braille were held to relate the contextual findings to practical issues, and to understand why certain decisions were made in the past.

The main finding in terms of cognitive aspects, involves the limitation of tactile and auditory senses and the impact these limitations have on textual aspects of mathematics. Besides graphical content, the representation of mathematical formulas was found to be extremely difficult for blind learners. There are two main ways to express mathematics in braille: using a dedicated braille code containing braille-specific symbols, or using a linear translation of a pseudo-code into braille. Pseudo-codes allow for reading and producing by sighted users as well as blind users, and are the main approach for providing braille material to blind learners in the Netherlands. A comparison based on set criteria allowed us to conclude that dedicated braille codes are significantly better at assisting the reader than pseudo-codes are. The comparison of mathematics curricula in the Netherlands confirmed the representation problems; the less popular mathematics curricula involve more mathematical formulas of higher cognitive complexity.

Contents

1. Introduction **1**
1.1. Objective . . . 1
1.2. Research questions . . . 1
1.3. Research methods . . . 2

2. Braille code **3**
2.1. Standard Braille . . . 3
2.2. Linearity . . . 5
2.3. Braille Grades . . . 5

3. Cognitive Aspects **7**
3.1. Acquiring cognitive skills . . . 7
3.2. Understanding language, numbers and shapes . . . 8
3.3. Spatial concepts . . . 10

4. State of the art **11**
4.1. Braille Notations . . . 11
4.1.1. Encoding numbers . . . 12
4.1.2. Encoding mathematical equations . . . 13
4.2. Mathematics Education . . . 15
4.2.1. Mathematics Curricula . . . 15
4.2.2. Teaching Goals . . . 17
4.2.3. Teaching Materials . . . 19
4.3. Technology . . . 20
4.3.1. Hardware . . . 21
4.3.2. Software . . . 23

5. Comparing braille codes **27**
5.1. Requirements Analysis . . . 28
5.2. Demonstration of braille codes . . . 29
5.2.1. Linear Formulas . . . 30
5.2.2. Fractional Formulas . . . 32
5.2.3. Spatial and fractional formulas . . . 33
5.3. Assessment of braille codes . . . 35
5.3.1. Analysis . . . 35
5.3.2. Comparison . . . 37

6. Comparing mathematical formulas **39**
6.1. Model for cognitive complexity of mathematical formulas 39
6.2. Quantification of complexity 42
6.3. Assessment of mathematical formulas 44

7. Conclusions **47**
7.1. Contextual findings 48
7.2. Comparing braille codes 50
7.3. Comparing mathematics 51
7.4. Discussion 51
7.5. Future research 52

Appendices **55**

A. Braille translations of mathematical formulas **55**
A.1. Spatial 55
A.2. Summation and Fraction 56
A.3. Integral 57
A.4. Group 58

B. Cognitive Complexity Survey **59**
B.1. Mathematics A, C 60
B.2. Mathematics B, D 60
B.3. Mathematics A, B, C, D 61

1. Introduction

For many years it has been common to enroll blind children in regular, non-special education, making use of the same teaching methods and materials that are used for sighted children. These teaching methods and mateirals are made accessible by introducing assistive aids such as laptops, computer software and digitized or tactile textbooks. In some subjects, however, the way blind children learn in combination with the available assistive aids form a problem. One of these subjects is mathematics, where not only multi-dimensional graphics are used, but mathematical formulas and the braille system are limiting factors as well. The teaching methods offered, often rely on visual aspects, increasing the difficulty blind children will have.

1.1. Objective

In this thesis the focus lies in the limitations of the braille system in the field of mathematics. The aim is to research the possibilities of braille codes that support mathematics and thereby make mathematics education for the blind more accessible. Because a variety of braille codes is used in the world, we want to know how these braille codes can be compared. Additionally we show that one of the reasons certain mathematics curricula in the Netherlands are relatively unpopular among blind learners, is related to the complexity of its mathematical formulas. We achieve this by using a model to compare mathematical formulas in terms of complexity, and applying this model on Dutch mathematics curricula.

1.2. Research questions

The main question that is answered in this thesis is:

- How complex are mathematical formulas and how can they be represented in Braille?

In order to answer this main question, the following sub-questions are answered in the chapters of this thesis. Question 1 will be answered in chapter 3, whereas chapter 4 will cover questions 2, 3 and 4. The core of this research are questions 5 and 6, that are answered in chapters 5 and 6 respectively. A summary of the answers to these sub-questions, as well as the answer to the main question, is presented in the concluding chapter on page 47.

1. How do blind children learn in comparison to sighted children?
2. What advancements to the braille system have been made to support blind learners in the field of mathematics?

3. How is mathematics taught in the Netherlands
4. What technological development support blind learners in the field of mathematics?
5. In what way can braille codes be compared?
6. In what way can mathematics levels be compared in terms of complexity of mathematical formulas?

1.3. Research methods

For this research, two main research methods have been applied, being literature research and interviews. Scientific papers as well as published articles and books have been referenced in order to answer the contextual questions related to cognition of blind children, the braille system, and technology. Interviews have been conducted to answer questions regarding education, the use of technology, and the way blind learners in the Netherlands experience mathematics. These interviews were held with mathematics teachers and educational coaches of blind learners, and blind individuals that already passed high school. Interviews often led to new sources of information such as articles and books.

2. Braille code

Braille is a tactile writing system used by blind people around the world, that was originally described in 1829 by Louis Braille. Louis Braille had been blind since the age of three, and started development of the braille system when he was 16 years old, to publish it four years later. At the time, the system did not cover much more than the French alphabet and few punctuation marks. Since then, many countries have customized the original Braille system to meet the specific requirements with respect to diacritical letters of their own language. Since many of these changes were made on a national level because of differences in language and character set, many braille tables are currently in use around the world.

In addition to literary braille, braille systems for special application areas such as music, mathematics and science have been developed. There is also a braille system for languages that mainly use symbols, such as Chinese. These systems often use a similar representation to writing without the use of symbols, by describing the sounds[1]. Although the music notation has been widely accepted[2], braille codes and tables for mathematics and science vary a lot[3]. The differences between these braille codes for mathematics and science, as well as the differences between braille tables are described in more detail in chapter 4.1. The following paragraphs cover the differences between classes of literary braille that are used around the world.

2.1. Standard Braille

The standard braille system is directly based on the work by Louis Braille, and uses a character set composed of 6 dots per cell. In this system, combinations of raised dots form characters that span one or multiple cells. The main structure of a braille cell, as shown in figure 2.1, is the 6-dot pattern that is also used on dice. Dots in the left column are numbered 1-2-3, and dots on the right column are numbered 4-5-6.

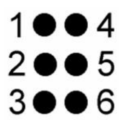


Figure 2.1.: Standard 6-dot braille cell pattern

When represented on paper, braille dots are 1.44mm in diameter and distance between the center of two horizontal or vertical dots is 2.340mm. Raised dots are typically 0.48mm in height, whereas inactive dots are not raised at all[4]. Braille is by nature entirely linear, which means that readers use their (often index) fingers to read from left to right, to continue on the next line. This reading tactic is necessary mainly because of the successive nature of tactile senses opposed to the simultaneous nature of sight. Seeing large structures and recognizing words at the same time is possible, whereas touch is limited to the position of fingers or other body parts.

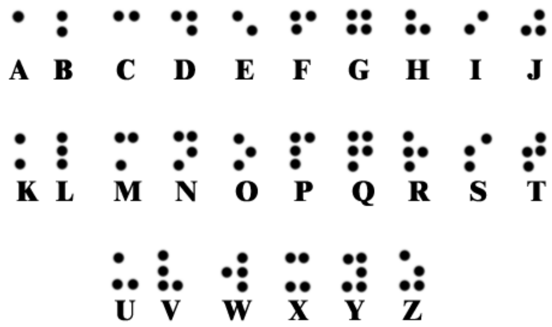


Figure 2.2.: Original braille table as defined by Louis Braille

Using only combinations of six dots essentially results in 64 unique characters including one space character in which no dots are raised. The standard alphabet of braille, as shown in Figure 2.2, follows a repeating pattern in the upper four dots that resets after each 10th letter. The first ten letters, a-j, use 00 (no raised dots) in the bottom row. These ten letters are followed by 10 characters for letters k-t with 10 in the bottom row (dot 3 is raised). The final letters u-z have 11 in the bottom row (dot 3 and 6 are both raised). The letter w is an exception to the ordering in the upper four dots, because at the time Louis Braille designed the system in 1825, this letter was not considered part of the standard French alphabet. This standard alphabet leaves 37 dot patterns for further encoding, of which some are used for punctuation marks and context indicators (e.g. upper case characters, numbers). The sequence of patterns in the original alphabet is, unlike the unicode implementation of braille symbols [5], not based on a mathematical model. This can be explained by the age of Louis Braille when he first developed the system; he was only 16 years old when development started. Although no standard is described for international braille, ISO standards starting with ISO/TR 11548-1[1] provide multiple (but possibly not all) standardized correspondence tables as well as invocation sequences to indicate context switches.

In order to increase the number of braille symbols, special multi-character representations of symbols are used. This is often achieved using modifier signs, that change the context of the upcoming structure. Figure 2.3 shows examples of modifier signs to indicate numbers and capitals letters. In many countries a similar approach is used for foreign letters and accented letters. As one only needs upper-, and lower case letters, numbers, punctuation marks and a small set of special characters for literary material[6], the use of modifier characters for such material is considered satisfactory. However, it has been argued that for mathematics the system of modifier symbols is "not an ideal

[1] ISO/TR 11548-1 refers to "Communication aids for blind persons"

solution"[3]. The main reasoning behind such statements is related to the high amount of context switches that is required to minimize problems of ambiguity.

Modifier	**Meaning**	**Braille**	**Example**	
$\backslash cf$	Capital Follows	⠠	⠠⠁	$\backslash cf$ a $\cong$ A
$\backslash nf$	Number Follows	⠼	⠼⠁	$\backslash nf$ a $\cong$ 1

Figure 2.3.: Modifier signs for capitals and numbers in braille

2.2. Linearity

For mathematical equations, all normal characters can be used as well as a large number of special characters such as arithmetic operations and Greek letters. Because the use of upper-, and lower case letters in combination with numbers, the need for modifier symbols is higher in the field of mathematics, in comparison to literary material. This increases the size and complicates mathematical material in braille. Another issue for learning mathematics in braille is linearity. Text is typically linear, which means that letters are on the same line and typically span no more than one line in height. In mathematics, anything but the simplest math is not normally represented in a linear way[3]. Very basic equations such as y=ax+b can be considered linear, but when powers, fractions or spatial functions such as summations are involved, translation to braille becomes a problem. For that reason, spatial structures in written text have to be linearized in order to be represented in braille[3]. This linearization process often increases the size of the formula, and also makes reading more difficult. The previously mentioned context-sensitive meaning by modifier characters, may cause even more problems in the field of mathematics. An example of such ambiguity problems involves the letter "a", that could, depending on the braille table that is used, represent an "a", "A", and "1", be also the Greek letter alpha (α), or a comma symbol. When grouping by means of parentheses, braces or brackers is used, mathematical content increases in size even more, and thereby becomes harder to read. An increased size of braille content will require more time to read, as one can only read one character at a time.

2.3. Braille Grades

Braille reading is done in a linear way, from left to right. This linear method of reading makes reading braille material a lot slower than reading print would be for sighted readers; depending on individual reading skills, it can take twice as long to read braille. To reduce the problem of generally slower reading, many adjustments to the braille system have been made in the form of so-called "Braille grades". Most of these changes were made on a national level, because of the different languages in the world, and the lack of standardization in the past. Standard braille is, in many countries, also referred to as Grade-1 braille or uncontracted braille. In contrast, Grade-2 braille is referred to as contracted braille and aims to decrease size by contracting common words in the applicable language. In Unified English Braille[7] some contractions use alphabetic word signs, and thus add an additional meaning to certain letters. A list of single-letter abbreviations for words in Unified English Braille is shown in Figure 2.4. Examples of alphabetic word signs are the letters "b" and "c". Their meaning can be the word "but" and "can" as well. Other contractions make use of modifier characters or multi-character abbreviations. Special modifier characters are used to switch between various braille grades in a similar way context switches are used for capital letters and numbers. Because of the various languages in the world, these contractions are commonly used on a national level only.

⠁	⠃	⠉	⠙	⠑	⠋	⠛	⠓	⠚	⠅
a	but	can	do	every	from	go	have	just	knowledge
⠇	⠍	⠝	⠏	⠟	⠗	⠎	⠞	⠥	⠧
like	more	not	people	quite	rather	so	that	us	very
⠺	⠭	⠽	⠵	⠯	⠿	⠷	⠮	⠾	⠡
will	it	you	as	and	for	of	the	with	child/ch
⠣	⠩	⠹	⠱	⠫	⠻	⠳	⠪	⠆	⠒
gh	shell/sh	this/th	which/wh	ed	er	out/ou	ow	bb	cc

⠲	⠢	⠶	⠔	⠌	⠬	⠜
dd	en	gg/were	in	st	ing	ar

Figure 2.4.: Single-letter abbreviations in Grade-2 Unified English Braille. Based on Simpson[7]

In addition to Grade-2 braille, there have been initiatives for an even higher level of contractions in braille[8]. The most well-known ones being Adam Speed Braille, Braille Shorthand and Grade Three braille. These braille codes are, like Grade-2 braille, based on the English language and offer abbreviations for words as well as phrases. Learning of these braille codes either involves special training courses, or reading a lot of textual material on the rules of these codes. In some methods, such as the Braille Shorthand, these rules may conflict with the standard Grade-2 braille rules.

A well-known method for shortening written braille material is the use of 8-dot cells rather than 6-dot cells. In this approach dots 7 and 8 are placed directly below dots 3 and 4 respectively. These dots can then be used to modify characters without the need for additional modifier symbols; an extra dot-8 could indicate a capital or number. This approach is mostly used in computer-braille, because most refreshable braille displays contain cells that count 8 dots. 8-dot braille is not common in printed material. An alternative to using dots 7 and 8 for modifier characters, is the use of a different character set; having 8 dots in total means there are 256 possible braille patterns (including a space) instead of the 64 combinations in standard braille. As many braille readers are already familliar with the existing system, adoption of such expanded system seems unlikely.

3. Cognitive Aspects

Aside from the way the braille system is generally used, it is also important to look at the way blind individuals learn in comparison to sighted people. This mainly involves the use of memory and acquisition of cognitive skills. In order to relate this to mathematics further on in this research, three main subjects are discussed in this chapter. These subjects involve acquisition of cognitive skills, understanding of language, numbers and shapes, and concept formation of spatial objects. Most research on this subject, in relation to learning, is done on children.

One important factor in each of these subjects involves the concept of Cognitive Load. Cognitive Load is described by Cooper[9] as "The total amount of mental activity imposed on working memory at an instance in time". Cognitive Load greatly influences the way people learn; it is believed that people cease to learn when the Cognitive Load is either extremely low or extremely high[10]. For blind learners this is particularly important, as they are limited to senses with a successive nature that donot allow for an overview of the material they are reading. This practically means that material that is read, has to be remembered entirely in order to apply it, increasing the cognitive load.

3.1. Acquiring cognitive skills

Many differences between sighted and blind children have been proven by research focussed on young children. These differences are mostly related to the senses used, and how these senses make for an mental 'image' of reality that is significantly different from that of sighted children. Acquiring cognitive skills by blind individuals is limited by the use of all senses except vision. The most important channels of skill learning are auditory, tactile and kinesthetic senses. Both auditory and tactile senses are successive by nature, in comparison to the simultaneous character of visual perception. This limitation of tactile senses makes the perception field of blind individuals significantly more narrow than that of sighted individuals.

Gouzman[11] states that *"the process of concept formation in blind learners is dominated by two extremes: extremely abstract verbal notions that have little support in the learners' experience, and the extremely concrete tactile images of everyday life objects that possess little potential for generalization."*. In addition, Vanlehn[12] applied a framework for motor skill acquisition on cognitive skill acquisition, and describes sub-phases of the early, intermediate and late phase of skill acquisition. During the intermediate phase, the concept of generalization is described as "the process of modifying one's understanding of an example or principle in such a way that surface information does not play a role in retrieval, mapping, and application". Because most concrete objects blind individuals perceive in daily-life don't allow for much generalization, the skill of generalization (something Vanlehn suggests is not an automatic process) could be limited as

	Nelson 1973 (sighted)	**Mulford 1988 (blind)**
Specific nominals	14	22
General nominals	52	38
Action words	13	24
Modifiers	11	6
Personal-social	9	13
Function words	5	3

Table 3.1.: Percentages of word categories part of the first 50 words of blind and sighted children, source: Warren[13].

well. In education, this could form a limitation, especially with the educational methods that rely on relations between abstract information and real-life objects.

Spontaneous exploration of two dimensional graphics by blind individuals is often done using a similar approach they apply for reading braille. Gouzman[11] describes that individuals scan for horizontal lines using one finger, a method he calls "absolutely inadequate for the exploration of tactile images". During his experiment some individuals also applied this technique on three-dimensional objects. The lack of adequate technique could lead to underdevelopment of skills such as comparative behavior directed at tactile images. Spontaneous comparison, something important for generalization mentioned before, is therefore less common.

3.2. Understanding language, numbers and shapes

Understanding language starts with speech, but on the subject of 'first speech' no consistent differences between blind and sighted children has been proven[13]. When looking at the first 50 words of sighted children compared to blind children, differences can however be observed. Warren[13] summarizes the results of various research conducted in his book. A copy of this summary is shown in Table 3.1.

The categories of words are *specific nominals* (names of people, toys, pets), *general nominals* (names of classes of objects, e.g. "dog"), *action words* (showing a manner or direction of action, e.g. "up"), *modifiers* (qualities of objects, e.g. "hot"), *personal-social words* (used in social interaction, e.g. "thank you") and *function words* (e.g. "what's this?"). The research, although among a small sample group (n=9 for blind, n=18 for sighted), significant differences can be seen. The biggest difference is identified with nominals; blind children tend to speak more of instances of objects/persons, whereas sighted children speak significantly more of classes of objects.

Besides speaking, reading is also an important factor when it comes to understanding language. Research has proven that consuming written material depends highly on the reading skills of the individual[14]. For assessing reading skills in the United Stats, the Oral Reading Fluency measure is used, which is based mostly on the amount of words per minute that can be correctly read. The study conducted by Helwig[14] also proves that reading skills highly influences mathematical skills, and thereby influence the understanding of numbers. Helwig states that *"Low Oral Reading Fluency - High*

Mathematics" students tend to be more successful when solving complex problems via a video presentation, indicating that reading can be a significant obstacle in mathematics. Gordon[15] found that the median reading speed for braille readers was 124 words per minute, compared to a median of 251 words per minute for normally sighted print readers.

Researchers have studied the role of hearing in an educational context, and related this to the memory of blind individuals. When hearing numerical sequences, studies show that sighted children are able to perceive up to seven numbers from hearing, by using their fingers as a tool to remember them. Blind children are able to perceive seven to eight numbers without using any tools, whereas some blind children were able to perceive many more[16]. During the research Ahlberg conducted, not one of the blind children used their fingers as a tool, which indicates that they were not aware of their fingers when doing arithmetic operations. This also suggests that the cognitive load while doing the same arithmetic operations is significantly higher for blind children, in comparison to sighted children.

Although not specifically for blind children, research on how children of a higher age learn best has been conducted. In the context of mathematics, it has been proven that learning from examples greatly increases the speed at which material is learnt[12]. There are two main ways for learners to use examples: studying examples before starting the assignment, or referring back to the examples while doing an assignment. Research in 1989 showed that the first approach, that involves explaining an example for oneself, results in the pupil learning faster. Referring back to examples while doing the assignment can have a spontaneous or deliberate nature, where the spontaneous nature is commonly based on superficial similarities between the example and the assignment. These similarities can be lingual or structural, and in some cases result in wrong use of formulas in particular. Deliberate retrieval of examples is less common, but generally has better results. When this approach is aimed at by teachers, the relation between the assignment and the example is typically hidden. Generally speaking, structural similarities often resulting in spontaneous retrieval of examples are less of a distraction for blind pupils, but the lingual ones may therefore play a bigger role. Strategies for learning by either studying examples before starting the assignment or referring back to them later can be used by blind pupils as well, although reading back (and thereby searching) may take more time.

When looking at the way mathematical equations are read, sighted pupils use different strategies. It seems that in half of the equations tested by Gillian[17], the sighted pupil made an initial scan to get to know the structure of the mathematical equation. Additionally, most of the participants in his research read from left to right in the same way they would read text, but frequently scanned back to previously read elements. Finally, the concept of chunking was very common. This concept involves solving parts of the equation first, often those within parentheses, to apply the outcome on the equation later. Most of these approaches are a lot harder for blind pupils to apply because of the forced linear way of reading. Scanning for the global structure and scanning for chunks is impossible, so reading the full equation is required. This increases the previously mentioned cognitive load.

An important part of high school mathematics involves reading of shapes, maps, diagrams and graphs. When reading such material, various cognitive problems play a role. The previously mentioned successive nature of tactile perception forms the first problem. Gouzman[11] describes that *"The whole tactile picture thus remains beyond the spontaneous grasp of the learner"*. Another related problem has to do with size, direction and proportion: two circles of different sizes may be perceived to be identical by blind learners, and blind learners often don't know their own body size in relation to the surrounding objects and environments. In terms of mathematics, exploring graphs and reasoning about steepness of lines in comparison to other lines become problems.

3.3. Spatial concepts

Describing spatial concepts logically involves physical properties of those concepts. However, when blind children are asked to describe properties of an object, they often mention visual properties they cannot perceive themselves. Other researchers have confirmed that speech of blind children is often *"less firmly connected to their sensory experience"*, but whether this in fact forms a problem is unclear[6].

When assigning a task related to the position of objects or body parts in a multi-dimensional environment, it has been proven that blind children complete this task differently. One of the research examples included the task of drawing a straight line between two objects on a drawing board. Sighted children (aged 6-11) were able to use external reference points (e.g. borders, edges). In contrast, blind children of the same age were less aware of the relation between the objects and their position on the board; they tended to rely on internal accuracy instead[6]. David Warren stated: *"Although there was a general improvement in performance of increasing age from 6 to 11 years, age was not a strong predictor for performance"*.

A research project by Hermelin and O'Connor in 1971[18] is slightly different in approach. They studied the concept of external referencing for blind, seeing and autistic children. During a training session they assigned words to the first two fingers of each hand, that were carefully placed on adjoining surfaces. No instruction was given on how they should remember the words, but it turned out that most blind children (75%) used their fingers as reference points rather than their location. With sighted children, only 40% used their fingers as reference. Researchers call this "Egocentric referencing of spatial information", which in relation to mathematics can be an obstacle when it comes to graphs, tabular information, lines and geometric figures.

4. State of the art

In the previous chapters we discussed how the the standard braille system works and what the influences of blindness are on cognitive skills and cognitive skill acquisition. In order to relate this to mathematics education, we will look at the state of the art from three viewpoints. These viewpoints include the braille system and its dialects and advancements to support mathematics, the way mathmatics is taught to the blind, and the way technology assists blind learners.

4.1. Braille Notations

In order to make mathematics more accessible for the blind, many projects arose[3]. Karshmer describes the braille code to be *"sufficient but far from perfect for normal writing"*, but with mathematics and science more problems occur. Although the 26 letters of Western alphabet remained mostly the same, the braille tables used vary by country for most other symbols. The need for specific mathematics codes in braille comes from the inability to write spatially arranged formulas in braille, which is especially a problem in the field of algebra. To target this limitation of braille, multiple codes specifically for mathematics and science were developed including the Nemeth Braille Code in the United States, Unified English Braille in Great Britain and the Woluwe Code in Dutch-speaking parts of Belgium[6]. Other mathematics codes include the Marburg Code (on which Woluwe was based), the French Maths Code and ItalBra (used in Italy). Additionally, the LaTeX notation, and variations of it, are commonly used to allow for easier communication with sighted users. Such notations are however not considered braille codes, because they still requires translation to braille[19].

Karshmer describes the various notations as a static approach to making mathematics more accessible. The mathematical content is simply translated to braille, where the user can navigate through the content but the content does not change. An alternative for this so-called static approach, would be to provide mathematical content in a more dynamic way[3]. In such approach, the user can, after a conversion process, navigate through the mathematical content in accordance with its mathematical structure. This could theoretically reduce the problems discussed in the previous chapter.

Approaches to making mathematics more accessible vary on two levels:

- The way numbers are encoded
- The way mathematical equations are encoded

4.1.1. Encoding numbers

There are four main ways to encode numbers in braille: Standard Braille, French grade 1 and 2, Antoine and the US Maths and Science notation. These approaches all come with advantages and disadvantages when it comes to ambiguity or size of the resulting structure. Number notations have been point of discussion for a long time; the International Council on English Braille (ICEB) documented a discussion on numbers that was held between November 1993 and January 1994[20]. The decisions made during that discussion were eventually implemented in the Unified English Braille standard, and involve the use of the standard number indicator from standard braille. In the Netherlands no such discussion has been documented, but the same approach is used in education. When refreshable braille displays are used, the French notations are also used. The following is an overview of the used number notations in the world.

Standard braille makes use of the number number sign as introduced by Louis Braille in the original braille table. It is known around the world, and has only one disadvantage: It takes one extra symbol for each group of numbers.

French grade 1 and 2 make use of the sixth dot as a replacement for the number sign, and adds the sixth dot to the normal characters for letters a-j. There is no real advantage when it comes to size of the structure, but ambiguity is reduced by numbers no longer having the exact same symbol as letters. This ambiguity is still there with less common symbol.

Antoine uses the additional sixth dot in a similar way as French grade 1 and 2, but only adds a number sign if ambiguity is an issue. Size is therefore decreased compared to standard braille. The other advantage over standard braille is that using 8-dot braille, where dots 7-8 are below 3-6, the 8th dot can be used as a number sign. This saves space and resolves ambiguity. 8-dot braille is common when refreshable braille displays are used, more information on this subject can be found in chapter 4.3. The disadvantage is only visible with 6-dot braille; some numbers may appear as less common symbols, in which case the number symbol is still required.

US Maths and Science makes use of lowered letters a-j, and only uses the standard number sign in case of ambiguity. The disadvantage is that many numbers will appear as other characters such as punctuation marks. For that reason the traditional number sign is still required in many cases.

To illustrate the differences between the previously discussed number notations, the number 12 is encoded using the above methods in Table 4.1.

Notation	Representation
Print	12
Standard braille	⠼⠁⠃
French grade one two	⠠⠡⠣
Antoine	⠡⠣
U.S. Maths and Science	⠂⠆ ⠼⠂⠆

Table 4.1.: Ways to write numbers in Braille

4.1.2. Encoding mathematical equations

Besides number notations and varying symbols, the way mathematical equations are represented in braille varies by country as well. A linear translation of symbols into braille is uncommon; most countries offer a set of dedicated symbols to encode equations instead. This way mathematical structures such as fractions and summations are encoded using newly introduced symbols. A comparison of known braille codes that are discussed in this paragraph, can be found in chapter 5.2, which also provides examples of formulas from the mathematics curricula in Dutch secondary education.

The first English braille code for mathematics is the Taylor code, which was used in the United States in 1946 and originated from Great Britain. The Taylor code contained a lot of grouping symbols, was very simplistic[1] and did not provide all mathematical symbols and structures that are used in high school mathematics. It was therefore considered not suitable for secondary (or higher) education by the blind mathematics teacher Dr. Abraham Nemeth.

Nemeth code

In order to solve the problems people experienced with the Taylor code, Dr. Abraham Nemeth described a more efficient way of encoding mathematics, that was called Nemeth code. This code has been in use in North America, Australia and New Zealand. Development started in 1946, and the specification was published for the first time in 1952. Two major revisions were published in 1965 and 1972[21]. The Nemeth code was originally based on the way Dr. Nemeth wanted pupils to read out formulas, where he replaced the words (e.g. "Fraction start", "Square root", "Superscript") by symbols unique to braille. The weakness of the Nemeth code is its complexity and high number of indicators and additional symbols. A survery conducted by Amato[22] shows that in the United States 20% of the teacher preparation courses don't offer instructions for the Nemeth code. Additionally, in 25% of the programs graduates were believed not to be competent in the Nemeth code because of limited instructional time. Amato[23] states that 39.6% of the participants in his survey completed a course in both literary braille

[1] Information about Taylor code, only available through the Google cache of http://www.unifiedbrailleforall.com/

and the Nemeth code, and 33.6% completed a course that covered both codes.

Unified English Braille

A cooperation between multiple countries called the International Council on English Braille developed a braille code called Unified English Braille (UEB). This standardized notation was developed in the 1990s and uses a different set of symbols and grammar than the Nemeth code. It is mainly based on the notation previously in use in the United Kingdom, which is called English Braille. Development of UEB code is done by the International Council on English Braille (ICEB) of which Australia, Canada, New Zealand, Nigeria, South Africa, the United Kingdom and the United States are member. The goal of the cooperation is to create a unified standard for both literary and mathematical content for countries of which English is the main language[24]. Although the UEB code and the Nemeth code vary a lot in terms of braille tables and specific symbols, the theory behind it is the same; symbols are introduced to replace mathematical functions and structures.

During user research in the United Kingdom, conducted prior to the official introduction in 2011, technical material encoded with UEB was provided to blind students without providing symbol lists or other explanations of UEB. These users were not used to UEB prior to this research, but had the knowledge to complete the given assignments. This research showed both advantages and disadvantages of the UEB code[25]. The biggest advantage over other braille codes is that is is relatively easy to learn; only few mistakes were made in technical material, and most confusion was related to symbols being changed from the code they were used to. Other advantages are related to the ease of transcription and conversion; only one braille code is used for both technical and literary material, and this code more closely reflects print. The biggest disadvantage and criticism against UEB, is the space it requires. Many common mathematical symbols (such as arithmetic operators) require two or three braille cells[7], which results in significantly longer formulas that in turn are harder to understand. Additionally, the amount of space characters used in UEB are significantly higher than is common in other braille codes.

Woluwe code

In 1975 a braille code was developed for Dutch-speaking parts of Belgium and the Netherlands. This code is known as Notaert as well as Woluwe code, and is based on the German Marburg code. The code had been in use in education for 30 years without much discussion[26], but has seen revisions, some of them recently in 2011[27]. Similar to the Nemeth code and UEB, symbols are introduced to replace common mathematical functions and structures, and a different braille table and set of special symbols are used. Most criticism on the Woluwe code is targeted at practical problems also occurring in other codes; creation using qwerty keyboards isn't possible, and teachers have problems teaching the code to blind learners. Additionally, the non-lingual nature of the notation makes it hard to understand for sighted users. The advantages are that encoded formulas using Woluwe are relatively short, even when spaces are included to indicate arithmetic operators.

Dedicon code

In the Netherlands a new encoding for mathematical equations was introduced in 2009, that aimed to make information equally readable for blind and sighted users. This code is not a dedicated braille code and does not describe braille symbols; instead, symbols used on a computer keyboard are used. When physical books are produced, the Dutch braille table is used, but personal communication indicated that most learners don't use the Dutch braille table when working on a computer. The Dedicon code can be best described as a linearization using a similar approach as used in mathematical computer software such as Microsoft Excel. Figure 4.1 shows an example formula using Dedicon encoding in comparison to a standard (two-dimensional) notation.

Standard formula	$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$
Dedicon encoded formula	$x = -b + sqrt(b\hat{}2 - 4ac)/2a$

Figure 4.1.: Linearization of a mathematical formula using the Dedicon code

Although this linearization is readable for both blind and sighted users when a computer is used, it is still believed to be complex to read. The use of grouping symbols is required, abbreviations are relatively long (especially compared to dedicated braille codes), and exact displaying of mathematical content depends highly on the software and braille table that is used. In personal communication it has been said that the code is based too much on what is displayed on the computer screen, rather than the refreshable braille display of the blind learner.

4.2. Mathematics Education

In order to look at the way braille can be applied on mathematics in the context of education in the Netherlands, we will assess the way mathematics is taught in the Netherlands. In this chapter we will discuss the structure of Dutch mathematics education, the teaching goals and methods that are used, and the teaching materials that are used. We will focus on the VWO (pre-university) levels. Most information in this chapter was gathered using interviews with mathematics teachers, educational coaches and blind learners.

4.2.1. Mathematics Curricula

The focus of this research is on the various VWO (pre-university) curricula for mathematics. Learners choose one of four available *levels*: Mathematics A, B, C and D. These levels involve different domains of mathematics, and aim to prepare learners for specific study directions. The levels are not ordered in increased or decreased complexity by definition; mathematics B does not necessarily have a higher or lower workload than mathematics A. The system in use now was introduced in 1999, but has seen its latest revision in 2007[28].

Mathematics A has a focus on applied mathematics, and prepares students for economic, social and medical studies. Mathematics A contains the domains of functions and graphs, discrete analysis, collections and probabilities, differential equation calculations and application, and statistics and probability calculations[29]. Mathematics B is considered to cover more of the fundamentals of mathematics, and prepares students for the more technical (Beta) studies. Mathematics B contains domains of functions and graphs, discrete analysis, differential and integral equation calculations, goniometric functions and advanced geometry[29]. Table 4.2 shows the domains associated with the Dutch mathematics curricula A and B. Mathematics Mathematics C is considered a preparation for social, cultural, legal, language and societal studies on a scientific level, and is similar to Mathematics A[30]. Mathematics D is an expansion of mathematics B, and has a more scientific nature[28].

	Mathematics A	**Mathematics B**
Functions and graphs	X	X
Discrete Analysis	X	
Collections and probabilities	X	
Differential equation calculation and application	X	X
Statistics and probability calculation	X	
Integral calculation		X
Goniometric functions		X
Advanced geometry		X

Table 4.2.: Domains covered in Dutch pre-university mathematics levels

From a list of blind learners enrolled in secondary education in the Netherlands that was made available by Koninklijke Visio, as well as interviews with education coaches from the same organisation, it has become clear that mathematics levels B (and D) are rarely chosen by blind learners. This is also confirmed by teachers of blind learners, and blind former-students. Reasons for this phenomenon that were mentioned in interviews, involve the more complex formulas and equations, as well as the spatial nature of mathematical content mainly in the domains of geometry and goniometry. Additionally, graphical calculators, spatial figures, and the braille code to express mathematical content form problems for blind students in the Netherlands[31]. No research has been done to identify or prove the causes for this phenomenon as far as we could find.

4.2.2. Teaching Goals

In the Netherlands it has become common to enroll blind learners in regular education as much as possible, which implies that the same teaching goals and methods are being used for blind learners as for sighted learners. Possibly for that reason, special needs education in the Netherlands makes use of the same textbooks and domains of mathematics. Differences between special needs education and regular education that were mentioned in interviews, involve the time spent on certain domains or subjects, and the more personal approach in special needs education.

An important domain of all mathematics curricula in the Netherlands, is the field of algebra, that contains a lot of formulas and equations. As mentioned in chapter 4.1, this is also where the use of braille complicates mathematics most. To understand how algebraic skills are taught, and what problems occur for blind students, it is important to look at the didactical methods and goals, and their limitations used in regular education. In the Netherlands a didactics guide for mathematics is available that covers this subject[32], and speaks of six main fields of didactics that will be covered. These fields are the process-object duality, visual properties of expressions, basic skills and symbol sense, meaning of algebraic expressions, practicing of skills and the development of schemas. It has been said in interviews that these fields are hard to teach sighted learners, many fields however base on visual properties that are inaccessible for the blind.

Duality of processes and objects

Most students will consider an algebraic expression as a process that can be used to get an outcome or result. This approach is however no longer usable when one needs to solve a mathematical equation by using the balance method, considering the left and right side as weights. An example of looking at the algebraic object rather than the process, involves solving the mathematical equation "4xˆ2 - 8a*x = 0". The solution to this equation is not a usual outcome, but yet another algebraic object, that also contains variables - "x = 2a". For blind learners this teaching goal is not necessarily harder than for sighted learners, but the methods used to illustrate this problem (e.g. the balance method) is targeted at visual comparison and learning to recognize possible tactics.

Visual properties of expressions

The first thing sighted students will see when looking at a mathematical expression is often its structure, as is also backed up by research[17]. Some teaching methods focus on visual similarities between the mathematical structures in rules of algebra, to give the learner pointers on what step to take next. Examples of this include numbers being placed closely together, and symmetrical looking rules. Based on previously taught rules that look similar (e.g. reductions of parentheses), one may try and use a similar approach. This sometimes leads to invalid generalization of these rules, when (for instance) the actual meaning of parentheses or powers is ignored. The desire for similar looking or symmetrical structures is used to guide students into a first step, or to mislead students so they can learn from their mistake. Both didactics methods are not directly applicable on blind learners for two main reasons; firstly, blind learners can only perceive one expression at a time, as discussed in chapter 3.1. Secondly, there have been indicators that blind learners have problems comparing sizes and other properties of structures, as explained in chapter 3.2. This implies that even when adequate tools are used, it remains difficult for blind learners to pass domains building highly on visual content.

Basic skills and symbol sense

Basic skills are required to work using set procedures, to look at small parts of a problem, and to use algebra to solve these problems. Symbol sense requires basic skills, but speaks more of using a strategy to solve a problem, and keeping a broad view on the expressions. It also involves looking the global expression and sub-expressions to know the meaning and context of mathematical symbols, so conversion of the same expression to another structure is possible. Finally, reasoning using algebra is an important part of symbol sense. Acquisition of basic skills is therefore more based on learning procedures, where symbol sense requires a mental image of the problem, to develop a strategy for solving the problem in multiple steps. The tactic for analyzing mathematical expressions by looking at the global structure as well as sub-structures as researched by Gillian[17] is not usable for blind learners directly. This could indicate that acquiring symbol sense could be more difficult as well.

Meaning of algebraic expressions

In order to give algebraic expressions a meaning for learners, algebra in the Netherlands often makes use of concrete examples and situations recognize from real life experiences. An example of a meaningful situation involves the repair for a car, where the cheapest of two options is to be chosen. In order to give meaning to algebra, this situation is translated into a graph with two lines in it that intersect at some point. From this translation on, it is considered important that the meaning is less related to the original problem; instead, one should use algebra to solve the problem instead. For blind learners it is imaginable that the step containing a visualisation of the problem is skipped; reading of graphs has proven to be difficult for blind learners. This implies that problem solving in a mathematical context more or less starts at algebra. Whether skipping of this step has a positive or negative impact on progress of the learner is unclear.

Practicing of skills

Developing, expanding, maintaining and practicing of algebraic abilities requires time, there is however also a risk involved in this. When practicing basic skills, retention of insight may decrease due to students memorising solutions or common procedures and operations. This routine can therefore reduce the meaning of algebra for the learner. To target this problem, alternative forms of exercises are often provided. The didactics guide advises to use exercises in which underlying insights and the previously mentioned symbol sense skills are also maintained. No clear limitations for blind learners apply on this field, as it mainly bases on other fields.

Development of schemas

In order to recognize and solve algebraic problems faster, one will need to know of the applicability of algebraic operations and techniques. An expert in algebra will be able to predict the effects of various strategies, and will also be able to judge if a certain operation helps in solving the problem. The collection of procedures, techniques and terminology is called a (mental) schema. One of the goals of teaching algebra, is to let students develop their own schemas that help solve mathematical problems. Development of such schema depends on skills of the previous five aspects as well. Blind learners that have problems with the previous fields are more likely to have problems with developing mental schemas for problem solving as well. This is however also the case for sighted learners.

4.2.3. Teaching Materials

Whereas the teaching goals and methods are generally the same for blind students and sighted students, teaching materials are different in most cases. The main difference in terms of reading is the braille system that is used, as explained in chapter 4.1. Besides textual content and mathematical expressions, however, mathematical content is often provided in other formats that can form obstacles for blind learners. Smith[33] states that *"this obstacle is especially obvious in the area of data analysis, with its strong emphasis on graphical representation of data in tables, charts, graphs and plots"*. Karshmer[6] describes five general approaches to making (mathematical) content more accessible: tactile as in braille or other raised representations, audio aids that read equations out loud, tonal representations of equations and graphs, haptic or force feedback devices that represent shapes of objects and curves, and integrated solutions. The use of technical aids and software to achieve these methods of accessibility is discussed in more detail in chapter 4.3.

Tactile approaches for mathematics often build on embossed paper, that can be produced in various manual and automatic ways. Two main ways to emboss paper are the use of swelling paper, and the use of braille printers. Swelling paper uses chemically treated paper that, when heated, swells dark areas on the paper, whereas braille printers use dots of various height and spacing instead of, or in combination with, ink[34]. Either system is usable for production of mathematical content that is best read spatially, and also supports production of charts, graphs, tables and drawings. Exploration of tactile material by blind users takes significantly more time than it would take sighted users to read, which in turn increases the cognitive load for the blind learner. Inadequate tactics for spontaneous exploration of tactile graphical material (as described in chapter 3) increases this problem.

In addition to using the standard braille system and the 8-dot braille systems mentioned in chapter 2.3 there have been 8-dot braille systems that target mathematics and science specifically[35]. The most well-known system being the dots-plus system[34][6].

This braille system, besides using 8 dots (4 rows, 2 columns) for regular characters, can also be printed using braille printers. When dots-plus content is printed on compatible printers, dots can have variable spacing and height. These properties are used to draw symbols similar to the print symbols, rather than convert them into (complex) braille symbols. An example of this approach is displayed in figure 4.2. The image shows the use of a square root symbol that is spatially written, symbols for plus, minus and plus or minus being written as they would be in print, and exponents being drawn on a higher position with no special indicator. Letters (variables) and numbers were not altered, and are readable in braille.

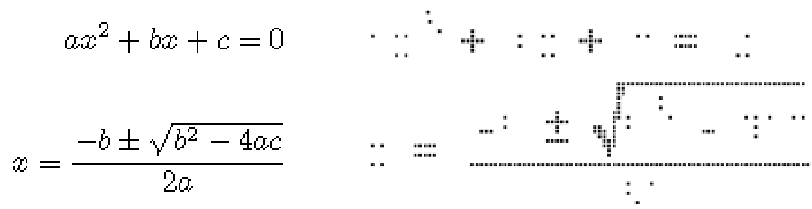


Figure 4.2.: Quadratic equations in print and dots-plus format. Source: Gardner[34].

When looking at examination of mathematics, tests are typically converted in a similar way as textbooks are. In the Netherlands this is done by Dedicon, a Dutch publisher of accessible textbooks. The main changes in exams are related to the displaying of tables, charts and linearization of formulas. When tables are used, rows and columns may be swapped to allow faster searching, and additional spacing is used to indicate the start of the next row or column. Charts may be provided in a tactile format, but the use of descriptions or tables instead is also common. An example of a description that goes with the added table is *"The figure contains three parallel straight lines"*, which is something a sighted user would instantly notice, but that is harder to recognize in a table. Mathematical formulas are translated in a similar way as is used in textbooks. In some cases, additional notes that mention possible confusing use of parentheses are added. An example of this is the comment *"Note that the denominator of the fraction is part of the radical"*, as was added to a mathematics level A exam from 2011.

4.3. Technology

Knowing about the braille systems for mathematics, and the teaching methods and materials for mathematics, it is important to look at technology to assist blind students. This technology can be separated in two categories: Hardware and software. The reason for this division, is the fact that many projects to assist blind learners involve the use of computers and specific hardware. This paragraph will explain some of these projects, as well as the hardware involved in mathematics education.

4.3.1. Hardware

As discussed in chapter 4.2, physical tactile material is commonly used in education and can be produced using braille printers or braille typewriters. The Perkins brailler is a typewriter that punches braille dots in paper. The device was developed in 1951[2] and has been sold over 300,000 times in 170 countries. Although it has been mostly replaced by technical aids in the Netherlands, the Perkins brailler is still popular in many countries. Although the newer generations of the device are significantly lighter in weight (3.5kg compared to 4.85kg of the standard brailler) and come with a reduction of noise, the devices are still generally considered bulky. Both versions of the Perkins Brailler are shown in Figure 4.3.

Figure 4.3.: Standard Perkins brailler and next generation Perkins Brailler

Braille printers are able to punch dots in paper, in combination or instead of traditional ink. The downsides of using traditional paper for braille, however, involve the fact that braille printed documents are bulky and deteriorate with use[36]. The production process itself is relatively time consuming and noisy in comparison to ink printing.

A popular alternative to physical braille material using paper, is the refreshable braille display. Refrehsable braille displays typically present between 40 and 80 braille cells with support for 8-dot braille. Refreshable braille displays are either connected to a computer using a USB or wireless bluetooth connection, or function as a standalone device. When the device is used in combination with a computer, special computer software can control the braille output, as will be explained in further detail section 4.3.2. When a braille display is used as a standalone device, however, it can also be used as a note-taker. This way digital braille material can be produced and read, as well as imported and exported to the computer when a connection is established[3]. Refreshable braille displays are not suitable for graphical content, as their *resolution* is limited to 4 dots in height, by typically 80 dots (40 braille cells) in width. There is however an experimental German project called the Hyper Braille, that offers a resolution of 120 dots in width and 60 in height, and is thereby able to display spatial content in a similar way to the output of braille printers[37]. Although a solution like this solves many problems of traditional braille, it is still in a development phase and is likely to be very

[2]Information from `http://www.perkins.org/store/braillers/` visited at 20-12-2012

[3]HandyTech Active Braille from `https://handytech.de/produkte.php?produkt=31` visited at 20-12-2012

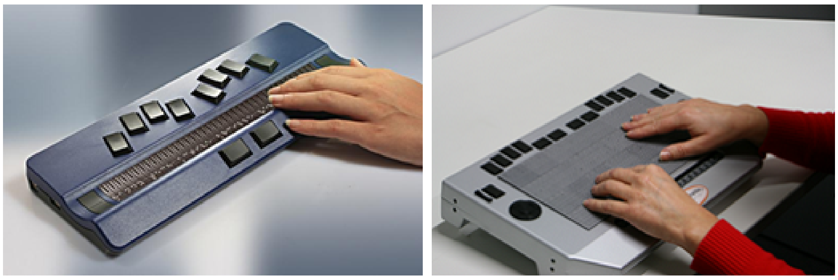


Figure 4.4.: Active Braille refreshable braille display and Hyperbraille tablet.

expensive when commercially available. Figure 4.4 shows the Active Braille refreshable braille display[4] and the Hyper Braille tablet[5].

In addition to tactile approaches to making content accessible, Karshmer[6] describes audio aids (and software) as an approach as well. Audio aids such as the DAISY Standard for Digital Talking Books[38] can be used to read (digital) content, including mathematics. According to blind students braille cannot be substituted entirely by speech in the field of mathematics and science[19]. The main reason for this is that when reading formulas, a different method of reading is required. Additionally, quick navigation is important in order to track back, and fully understand the (often context-sensitive) meaning of (sub-)expressions. Abraham Nemeth[21] came up with rules for reading formulas out loud himself, and Dick[39] indicates that taped versions of mathematics books come with drawbacks related to the way mathematics is read as well. In addition to the previously mentioned problems, another problem lies in the use of parentheses and grouping symbols, that when read character-by-character make the global formula hard to understand. An example Dick[39] uses involves the formula ”2(x+2)(x+2)” that is possibly read out as ”*two left parenthesis x plus two right parenthesis left parenthesis x plus two right parenthesis*” rather than ”*two times the quantity x plus two, times the quantity x plus two*”. The same problem occurs when regular screen reading software is used to read digital books. Karshmer[6] mentions software projects that allow reading of equations out loud in different formats, using non-speech sounds to support the user as well. These projects will be covered in section 4.3.2.

Although secondary education in the Netherlands makes use of graphical calculators, these calculators are inaccessible for the blind[31]. Talking calculators are available[39], and offer features such as speaking of pressed keys, display content, and a *learning mode* for key identification at any moment that does not affect the calculation[40]. The main limitations of such talking calculators, involve the fact that they lack graphing capabilities. Because of this limitation of physical talking calculators, the use of computer software is more common.

Besides methods to make information tactile by means of refreshable braille displays and braille printers, two other techniques have been researched for making spatial in-

[4] Source: `http://handytech.de/produkte.php?produkt=31&lang=en`
[5] ource: `http://www.hyperbraille.de/press/`

formation more accessible. These techniques rely on haptics and force feedback devices, that present vibrations or force when the user attempts to navigate in a virtual space, using these devices. Haptics are already in use in phones and touch screen devices. Toennies et. al[41] showed that most users are able to "find requested locations on a grid, determine the locations of displayed points, and differentiate between lines and shapes, with haptic feedback, auditory feedback and various combinations of the two". In this study a single touch screen device was used in combination with a host computer. An alternative to using touch screen devices, are devices that offer force feedback. Examples of such devices are the commercially available Sensable Phantom, Logitech Wingman Force Feedback Mouse or the Microsoft Sidewinder Force feedback 2 joystick. These devices can, as demonstrated by Carmeiro et. al[42] be used to illustrate shapes, locations of objects, relative sizes of objects and identifying the various surfaces (e.g. bump, friction, vibration). Experiments with haptic or force feedback devices show that blind users are able to identify shapes, sizes of objects and different surface types, but use of these devices seems to be mostly experimental at the moment.

4.3.2. Software

In the early 80s, software applications called screen readers were introduced, aiming to produce a vocal rendering of the text contents of the computer screen under control of the keyboard using a text-to-speech (TTS) converter. Screen readers typically create a separate off-screen model, and interpret messages from the operating system. These applications are commonly used, often in combination with speech synthesizers and refreshable braille displays. Commercially available screen readers are JAWS by Freedom Scientific, HAL by Dolphin and Window Eyes by GW Micro[42]. Free alternatives are available too, either by using built-in features of the operating system such as Microsoft's Narrator and Apple's VoiceOver, or dedicated applications such as NVDA[6]. The limitations of screen readers across all operating systems lie in the fact that images, graphs and visual layouts cannot be expressed using speech at all, and objects such as tables become confusing or lengthy[42]. Karshmer[6] states that an example of a screen reader (JAWS) is primarily designed for the general user interface, and is not well suited for more technical user interfaces. Because not all applications are accessible using screen readers, the concept of scripting is available in many screen readers. Scripting allows a user to give instructions to the screen reader on what to read out. This is typically not configured by users themselves, as it involves highly technical knowledge.

Specific software has been created for th purpose of reading and producing mathematical content. These two classes of use will be covered in the next paragraphs.

Reading Mathematics

There are various ways to make mathematical formulas readable for the blind. Often this requires converting the formulas to a format compatible with speech output, rather than images that are often used to express mathematics. In order to read out mathematical

[6]NVDA Screen reading software from `http://www.nvda-project.org/`

formulas on the internet using a screenreader, MathPlayer can be used[43]. MathPlayer requires the mathematical content to be described using MathML. MathML is a standard that provides a low-level specification for describing maths on the web. There are several ways to output MathML code, including conversion from LaTeX or the Microsoft Office Word Equation Editor[44]. Although MathML works well on the web, it still relies on the time spent by the creator of mathematical content. If MathML was not used when the webpage was created, a screen reader will not be able to display it[7]. There are Javascript files available that are able to translate formulas in a LaTeX-format to MathML when the visitor loads the page, an example of this approach is AsciiMathML[8]. Unfortunately many electronic formats, including PDF and Postscript (commonly used for textbooks) are inaccessible in braille and speech[19].

Several projects exist that allow reading of mathematical expressions that are not described using the MathML format, including ASTER, InftyReader and InftyEditor, and AudioMath. ASTER, Audio System for Technical Readings, was designed in 1994 and allows the user to read out and navigate through technical content in a document using a type of text editor referred to as emacs[45]. ASTER also allows for different *audio views* to skip certain objects, use sonification in combination with speech, use higher or lower pitched voices to indicate a change of level (to superscript, subscript), and bars/accents. The Infty project was first presented in 2000, and consists of multiple applications. The InftyReader application uses Optical Character Recognition (OCR) to scan a file from a variety of digital formats including PDF and several image formats. After scanning, the file can be converted into more accessible formats including LaTeX, MathML, HTML, braille codes such as Unified Braille Code and the Japanese code, and Human Readable Tex (HrTeX)[46]. Whereas LaTeX, MathML and HTML are quite technical and contain many grouping symbols, the HrTeX standard, although not compatible with LaTeX, aims to increase readability. HrTeX was first described in 1996, and increases readability by using abbreviations, reducing the amount of backslashes needed, and using a different notation for fractions[47]. The method of outputting HrTex or dedicated braille codes is suggested for use by blind users[46]. AudioMath was developed in 2004, and uses a similar approach to the Infty project, by scanning digital files into the MathML format before presenting it to the user. When expressing mathematics using speech only, information is generally lost. Especially grouping symbols such as parentheses and powers form problems. To overcome this problem, AudioMath makes use of formal rules for text generation based on existing research that keep the right structural information of the formula. This structure information results in pauses of various length and the use of tones with a higher or lower pitch[48], and is similar to the approach used in the ASTER project. Unfortunately AudioMath is mainly aimed at the Portuguese language.

[7] University of Washington about screen readers and mathematical equations at `https://www.washington.edu/doit/Faculty/articles?404` visited at 10-01-2013

[8] AsciiMathML at `http://www1.chapman.edu/~jipsen/mathml/asciimath.html` visited at 10-01-2013

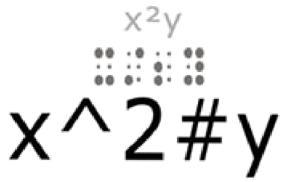


Figure 4.5.: SensoMath output in a spatial, braille and linear way

Writing Mathematics

Writing or editing mathematical content can be done on several levels, varying from typing braille on a braille notetaker to using LaTeX on a computer. When a notetaker is used, one is able to use the braille code desired, and transfer it to a computer. Karshmer[3] describes the need for braille translators that allow conversion of LaTeX to braille, and back-translators, to convert technical braille to a more uniform format such as LaTeX. Although such software is available for the Nemeth braille code (the MAVIS and Insight projects), Marburg code (the Labradoor project) and two French notations (the Bramnet project), no applications exist for the braille codes used in the United Kingdom (Unified English Braille) or Dutch-speaking Belgium (Woluwe code) as far as I could find.

As an alternative to writing mathematics in braille, one can use computer software to support the user with speech or braille output on a refreshable braille display. Projects that specifically support writing of mathematics for blind users are the Infty project described earlier, MathType (for Microsoft Office Word) and SensoMath. The Infty project includes InftyEditor, a text editor that is able to edit content previously scanned with InftyReader. When the extended version, called ChattyInfty is used, speech is available as well, allowing a blind user to edit or create mathematical formulas using the computer's keyboard. ChattyInfty offers speech feedback and key combinations for inserting special mathematical structures, and (like InftyReader) is able to save files in the MathML, LaTeX and HrTeX format. One is also able to use braille output of LaTeX on a refreshable braille display[49]. MathType is a plugin for Microsoft Office Word, that is able to convert formulas created using the Microsoft Equation Editor to the MathML format. The MathML output that is generated is accessible using a screen reader in combination with the previously discussed MathPlayer. SensoMath is a plugin for Microsoft Word that allows instant conversion between the linear mathematics code used in Dutch-speaking Belgium, a spatial arrangement, and direct braille output to a refreshable braille display[26]. Similar to the Dedicon code discussed in chapter 4.1.2, it displays linearly on the computer screen to make it readable for both blind and sighted users, but in addition it also allows reading using a refreshable braille display using the familiar Woluwe code as well as the spatial notation as used by sighted users. An example of a formula in the three views this plugin offers is shown in Figure 4.5.

Calculating

The most common software applications that allow calculating in the Netherlands include Microsoft Office Excel and AllerCalc[9]. Both are accessible using screen reading software (including speech) and allow for an intuitive input method that is also usable for sighted users. The limitation these applications, and most other computer calculators, have in comparison to graphical calculators as used in secondary education, is the fact that graphics cannot be presented using speech or a refreshable braille display. As a solution to this problem, one can use tonal representations generated by so-called audio-graphing calculators. In such application, one can write a formula that is then made audible using tones that gain pitch for increasing steepness, and lower pitch for decreasing steepness. Additional features include locating of intersections and finding the highest and lowest point. Examples of audio graphing calculators are the ViewPlus Audio Graphing Calculator[50] and the NASA MathTrax application[51].

[9] As stated in personal communication

5. Comparing braille codes

As described in chapter 4.1, the braille code varies mostly by country and is usually not decided by users. Users often use the braille code and (if applicable) the associated braille tables that are common in their country of residence. In the United States one is actively taught the Nemeth code, in the United Kingdom the English Braille Code or Unified English Braille are used, and in the Netherlands one will use the Dedicon code. In most countries a braille authority decides which braille code is used; examples include the Braille Authority of North America (BANA) for the United States[1] and the UK Association for Accessible Formats (UKAAF)[2] in the United Kingdom. On top of that there is the International Council on English Braille (ICEB)[3] that focuses on braille for countries in which English is the main language. BANA and UKAAF, as well as braille authorities of Australia, Canada, New Zealand, Nigeria and South Africa are a member of ICEB. In the Netherlands no real braille authority exists. Most institutions and the main publisher of accessible books in the Netherlands cooperate to make print material accessible, but no clear decisions or future plans are made and documented as far as we know. The "braille quality" division of the NLBB alliance[4] was formed in 2011 and aimed to improve and assess the implmentation of the existing code. From what is documented on their website, no plans for improving the braille code as used in the Netherlands have been made.

Braille codes and tables are typically used for all purposes except for music; the braille music code from 1997 is agreed upon internationally[2]. The braille code that is decided on by a braille authority is commonly a generic code that is not targeted at a specific use case (e.g. writing or reading) or material type (e.g. science, literature, mathematics). The reason for this as named in personal communication, is that the generic code has always been expanded to suit new requirements. This has traditionally been done by institutions or braille authorities on a national level, or on an individual scale that drew the attention of others. As mentioned, the braille music notation is the only real exception to this.

As described in chapter 4.1, existing braille codes vary on many levels ranging from numeric notation and the use of number symbols, the way formulas are encoded and the code-specific symbols that are introduced. The combination of these factors results in braille codes that are difficult to compare. In 2010, a Comenius school partnership called Touching Maths arose, that focussed on mathematics education for braille users

[1] Braille Authority of North America at `http://www.brailleauthority.org`
[2] UK Association for Accessible Formats at `http://www.ukaaf.org`
[3] International Council on English Braille at `http://www.iceb.org`
[4] NLBB at `http://www.nlbb.nl/`

integrated in mainstream secondary education[5]. Participants of the project include Norway, the Netherlands, France, Estonia, Germany and Belgium. One of the results of this project is a set of requirements for the - in their opinion - ideal braille code. Using these requirements, as well as some additional requirements, the existing braille codes have been assessed. In the following paragraphs these requirements are explained in more detail and a demonstration of five main methods of encoding mathematical work in braille is provided. This demonstration is then used to assess and compare the braille codes in terms of the given requirements.

5.1. Requirements Analysis

Results from the Touching Maths project explicitly state that the use of a laptop in combination with braille a display offers advantages over the more traditional braille typewriters or note-takers. Whereas laptops can use software to make translations and allow sharing of information with sighted individuals, braille typewriters and note-takers lack this functionality and require braille reading skills. At the same time, the laptop's keyboard can be used as well, reducing the need for a braille code when producing mathematics. Using the laptop's keyboard also has the advantage of faster entering of textual material. During personal communication it was made clear that the use of speech as a sole mean to access mathematics results in a high cognitive load, and is therefore not preferred. In personal communication with multiple braille users, it was mentioned that many blind students prefer to use speech in combination with braille, allowing them to track-back while reading the formulas.

It is my hypothesis that different requirements for braille codes can be useful for different types of usage. In order to decide what types of usage are important, teaching methods and factors by projects such as the Touching Maths project were considered. A general aspect of braille involves writing of braille. Although shortness is an important factor when writing braille, it is also important that a braille code offers guidance for the future reader (which could be someone else). In contrast, note-taking benefits far more from shorter formulas because of the speed at which production takes place. The Touching Maths projects' documentation also describes *communication between blind and sighted peers and teachers* as an important aspect. We will refer to this as *"Sharing"*, as essentially material is shared by the blind pupil to other (sighted) individuals.

When looking at the braille codes that were mentioned in chapter 4.1, a few clear differences can be seen. Most notably for mathematics, there is the factor of feed-forward. In braille codes focussed on mathematics and science, it is common to tell the user what kind of structure is coming before beginning the actual formula. Especially with fractions this can be very useful to reduce the cognitive load and the need to remember the entire mathematical structure at once. With a fractional formula, a braille reader will only realize a fraction is involved when reading the division symbol. When nested divisions are used this will make it even more complex, and tracking back

[5] Touching Maths project at `http://www.touchingmaths.net` visited at 10-01-2013

will consume significantly more time. Another factor that especially plays a role in longer formulas or equations, is the way a structure is terminated. The main reason for this is the fact that mathematical structures may span multiple lines. When this is the case, a user will need to know that a mathematical structure has ended. Whereas parentheses can be used for this as well, especially with larger structures with a high level of nesting, dedicated symbols may improve readability. We will refer to the use of symbols to terminate structures as feed-back. Besides fractional formulas, feed-back is also used when switching from sub-, or superscripts to the standard base-line level.

The first two requirements described in reports from the Touching Maths project involve transparency and intuitivity for both teachers and peers. The explanation for these requirements is that lingual or intuitive codes are more suitable for secondary education in comparison to the paper-braille codes. Examples of lingual codes include LaTeX or variations of it, that use keywords or abbreviations of keywords to describe structures. For instance, a square root in LaTeX will start with sqrt. In addition, these codes are typically easier to learn for those that gradually lose vision. The third requirement from the Touching Maths project states that a dedicated braille code must be suitable for secondary and higher education. The fourth requirement in short is called compactness, but involves many methods that influence this. The first part describes the total length of the structure, which is typically greater than the print version. It does however also explain the use of spacing, key signs and brackets to prevent ambiguity. Finally, the support of 8-dot braille is mentioned, to reduce the length of structures by using dot-7 and dot-8 for key symbols. An example of this would be the exclusion of the regular number symbol, and using dot-8 together with the number instead. The same concept applies to greek symbols, capitals, or even braille grade changes. Although the use of spaces increases the total size of a mathematical structure, it has been said in personal communication, that a popular method of reading equations involves adding spaces to it, to separate sub-structures. This indicates that an increased size of a mathematical structure in a braille code is not necessarily a bad thing. When comparing braille codes, we will separate the requirements for assistive spacing and required size.

In Table 5.1, the relative importance of the previously mentioned requirements is shown. Requirements marked with an asterisk (*) are the ones that were also described by the Touching Maths project. The remaining requirements are mostly based on personal communication with braille users. Using the demonstration in section 5.2 of this document, several braille codes will be assessed according these requirements in section 5.3.

5.2. Demonstration of braille codes

In order to assess existing braille codes using the requirements mentioned in chapter 5.1, formulas from Dutch mathematics education were used. Before making a selection of formulas, all formulas in the VWO A and B curriculum were listed and compared in search for common structures (e.g. fractional, linear, spatial, summation, integral) and symbols (Greek letters, arithmetic operators) used in these formulas. I argue that these

	Writing	**Note-taking**	**Reading**	**Sharing**
Feed-forward	0	0	1	N/A
Feed-back	0	0	1	N/A
*Required size	1	2	1	0
*8-dot braille support	0	0	1	0
*Lingual	0	0	1	2
*Seconday and higher education	1	1	1	1
*Production on qwerty-keyboards	1	1	0	0
*Assistive spacing	0	0	1	1

Table 5.1.: Relative importance of requirements for specific use-cases

different structures as well as symbols increase or decrease the need for certain requirements, and also increase the differences between braille notations that may look similar at first sight. The full assessment of formulas used in Dutch mathematics education can be found in chapter 6.3. The reason mathematics curricula A and B are used, involves the fact that the remaining levels are very similar to these levels; mathematics C is very similar to mathematics A, and mathematics level D is an expansion of mathematics B.

As an indication, the first occurrence of common mathematical structures is translated to the various braille notations. The structures found, and the first formula of this structure is shown in Table 5.2. The numbers between parentheses refer to the identification numbers used in the full analysis of mathematical formulas, which can be found in Appendix B. To increase readability of this thesis, only the translations of formulas of the type linear, linear and fraction and spatial and fraction are included. The remaining translations can be found in Appendix A.

Because LaTeX is not a braille-specific system and is based on characters on a computer keyboard, a braille table had to be selected to match these symbols. Because the default braille table in various screen reading applications in the Netherlands is the braille table from the United States, the symbols from this braille table are used for the LaTeX translations.

For the beforementioned common structures, the first occurance of this structure in a list of formulas used in Dutch mathematics education of both levels A and B is encoded using various braille codes. In most braille notations, special characters are used for capitals, number or other special symbols to increase the amount of characters that can be represented. In the print explanation below the translations, the ; character will be used to denote a special character such as a Greek letter or an enlarged grouping symbol. The # symbol is used to illustrate the regular number symbol (dots-3456 ⠼).

5.2.1. Linear Formulas

A linear formula is a formula that spans one line in height, and requires no spatial symbols that cover multiple characters in width or height. An example of a linear formula is

Structure Type	Example formula
Linear	$n! = n * (n-1) * .. * 2 * 1$
Fraction	$\sigma(\bar{X}) = \frac{\sigma(X)}{\sqrt{n}}$
Spatial	$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$
Spatial and Fraction	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$
Summation and Fraction	$S = \sum_{k=0}^{n-1} ar^k = a\frac{1-r^n}{1-r}$
Integral	$L = \int_a^b \sqrt{1 + (f'(x))^2} dx$
Group	$\begin{cases} x(t) = m + r \cos \omega (t - t_0) \\ y(t) = n + r \sin \omega (t - t_0) \end{cases}$

Table 5.2.: Example formulas for common mathematical structures

the formula shown below, which was translated using LaTeX, Woluwe, Unified English Braille, Dedicon and Nemeth code.

$$n! = n * (n-1) * .. * 2 * 1$$

LaTeX

The LaTeX formula is written as "n!=n*(n-1)*..*2*1" and counts 17 characters in print. The representation below is a character-by-character translation of the formula that counts 18 characters when using the American braille table. It is worth noting that the equality symbol in the American braille table counts two cells, and numbers are lowered so no number indicator is necessary.

⠝⠯⠨⠅⠝⠡⠷⠝⠤⠂⠾⠡⠲⠲⠡ ⠆⠡⠂

Woluwe encoed

The Woluwe encoded formula counts 21 characters and contains no braille-specific characters. It appears the same as the LaTeX version of the same formula, but contains number indicators and a sign of equality preceded by a space.

⠝⠖⠀⠶⠝⠦⠣⠝⠤⠼⠁⠜⠦⠲⠲⠦⠼⠃⠦⠼⠁

Unified English Braille encoded

The UEB encoded formula counts 30 characters and contains no braille-specific characters. The representation of the asterisk (*) and period (.) are very similar, as are the equals sign (=) and the number indicator (#). Most arithmetic operators span two braille cells.

⠝⠖⠀⠐⠶⠀⠝⠐⠲⠐⠣⠝⠐⠤⠼⠁⠐⠜⠐⠲⠲⠲⠐⠲⠼⠃⠐⠲⠼⠁

Dedicon encoded

The Dedicon encoded formula appears as ”n! = n * (n - 1) * .. * 2 * 1” and counts 29 characters in print. When translated using the American braille table it counts 32 characters. This size is mostly explained by the number of spaces used and the use of number indicators.

⠝⠯⠀⠨⠅⠀⠝⠀⠡⠷⠝⠀⠤⠀⠼⠁⠶⠀⠡⠀⠲⠲⠀⠡⠀⠼⠃⠀⠡⠀⠼⠁

Nemeth encoded

The Nemeth encoded formula is equal to the LaTeX representation of the formula, but adds assistive spacing around the equals sign. It counts 20 characters.

⠝⠯⠀⠨⠅⠀⠝⠡⠷⠝⠤⠂⠾⠡⠲⠲⠡⠆⠡⠂

5.2.2. Fractional Formulas

Although fractions can also occur linearly using the linear division symbol, which appears as a slash (/) in print, we will only consider the spatially written version of a fractional formula. In such case, the numerator and denominator are written on different lines, separated by a horizontal line. This structure is very common in high school level formulas.

The formula we will use as an example, reads as follows: ”Sigma of X-bar = fraction of sigma of X divided by the square root of n”. The spatial expression is shown below:

$$\sigma(\bar{X}) = \frac{\sigma(X)}{\sqrt{n}}$$

LaTeX

The LaTeX encoded formula reads as ”\sigma{(\bar(X))}=\frac{\sigma{(X)}} {\sqrt{(n)}}” and counts 49 characters. When translated to braille using the American braille table, it counts 59 characters.

⠳⠎⠊⠛⠍⠁⠨⠷⠷⠳⠃⠁⠗⠷⠠⠭⠾⠨⠾⠨⠅⠳⠋⠗⠁⠉⠨⠷⠳⠎⠊⠛⠍
⠁⠨⠷⠷⠳⠠⠭⠾⠳⠳⠨⠾⠳⠎⠿⠗⠞⠨⠷⠷⠝⠾⠨⠾⠨⠾

Woluwe encoded

The Woluwe encoded formula counts 25 characters and uses Woluwe-specific symbols to indicate the start of a fraction. It reads as ";S(X;;) =start fraction ;s(X) / radicand(n) end fraction". The bar on top of the X is denoted by two different indicators; braille dots 45, followed by dots 36 (⠘ ⠤).

⠰⠎⠣⠰⠨⠭⠘⠤⠜⠀⠶⠐⠆⠰⠎⠣⠨⠭⠜⠳⠩⠣⠝⠜⠱

Unified English Braille encoded

The UEB encoded formula counts 33 characters, and (like the Woluwe code) includes specific characters for divisions. It reads like ";s(X;;) = start fraction ;s(X) / radicand(n) end fraction". Like the Woluwe code, the indicator for a bar on top of the X, two different indicators are used. For UEB this symbol in braille is described using the braille dots 56, followed by dots 156.

⠨⠎⠐⠣⠠⠭⠰⠱⠐⠜⠀⠐⠶⠀⠷⠨⠎⠐⠣⠠⠭⠐⠜⠨⠌⠰⠩⠐⠣⠝⠐⠜⠨⠾

Dedicon encoded

The Dedicon encoded formula can be directly translated to print, and looks like "~s({~X) = ~s(~H) / sqrt(n)" and counts 26 characters. When translated using the American braille table, it counts 31 characters.

⠸⠘⠎⠷⠨⠷⠤⠠⠭⠾⠀⠸⠅⠀⠸⠘⠎⠷⠨⠭⠿⠀⠌⠀⠎⠟⠗⠞⠷⠝⠾

Nemeth encoded

The Nemeth encoded formula counts 25 characters, and (similar to Woluwe and UEB) uses a special character to indicate the start of a fraction. It reads like ";s(X;) = start fraction ;s(X) / radicand(n) end fraction".

⠨⠎⠷⠠⠭⠱⠾⠀⠨⠅⠀⠹⠨⠎⠷⠠⠭⠾⠨⠌⠜⠷⠝⠾⠼

5.2.3. Spatial and fractional formulas

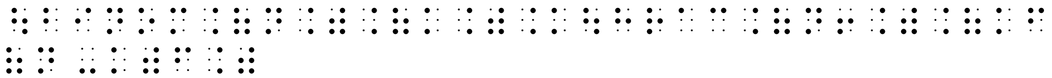
A spatial formula in the context of this research, either involves symbols that span multiple lines, or has a different meaning because of the spacing between (sub)expressions. Examples include the binomial coefficient, the Greek capital Sigma (Σ) and the "long s" ($\int$) used for integrals.

The example formula that makes use of a spatial structure as well as a fractional structure, is the formula for binomial coefficients. The formula is shown below, and reads like "n choose k = n factorial divided by k factorial times the factorial of n minus k".

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

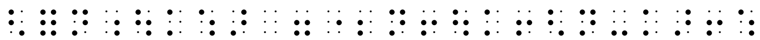
LaTeX

The LaTeX encoded formula reads as "\binom{n}{k}=\frac{n!}{k!(n-k)!}", and counts 32 characters. When translated using the American braille table, it counts 41 characters. Its representation in braille is shown below.



Woluwe encoded

The Woluwe encoded formula counts 24 characters, and reads as "(; n newline k) end = start fraction n! / k!(n-k)! end fraction". The newline character indicates the start of a new line within enlarged parentheses. Starting on an actual new line is not recommended with relatively small expressions. Within a fraction, the numerator and denominator are written without spaces.

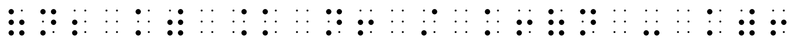


Unified English Braille encoded

The UEB encoded formula counts 29 characters, and reads as ";(n binom k) = fraction of n! / k!(n-k)! end fraction". Dedicated symbols for binomial coefficients are used.

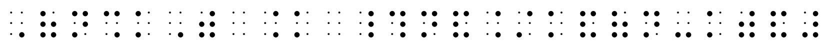
Dedicon encoded

The Dedicon encoded formula counts 24 characters, and reads as "(n; k) = n! / k!(n - k)!". In this formula, the semicolon is not a braille character to indicate a change of context, but the actual semicolon. Using the American braille table, it counts 25 characters.



Nemeth encoded

The Nemeth encoded formula counts 26 characters, and reads as ";(n directly under k;) = start fraction n! / k!(n-k)! end fraction". The directly under symbol indicates that the next character appears directly below the previous character. The user will have to recognize this as a binomial coefficient.



5.3. Assessment of braille codes

In the previous paragraph translations of three mathematical formulas into various braille notations were provided. In addition, four other example formulas are provided in appendix A. In this paragraph we use these examples to compare the braille notations in terms of the requirements described in section 5.1.

Looking at the example formulas, we see two main classes of braille notations: pseudo-codes and dedicated braille codes. Pseudo-codes can typically be written using a qwerty-keyboard and have a visual nature. Examples include LaTeX and the Dedicon code, as well as variations of LaTeX such as Human Readable Tex (HrTex) [46]. Dedicated braille codes, on the other hand, involve braille-specific symbols and are developed purely for the blind. Production on qwerty-keyboards is not directly possible, and displaying is only possible on paper or refreshable braille displays. Examples of dedicated braille codes are Woluwe code, UEB and Nemeth code.

In the following paragraphs, we will analyse how each notation compares to the others according to the set requirements. Finally, we will define a scoring scheme to compare braille codes from a global perspecive. Depending on user preference, this scoring scheme could also be used to choose a braille code for specific use cases.

5.3.1. Analysis

Feed-back and feed-forward

Dedicated braille codes typically introduce special characters that are used to guide the braille reader. This is most commonly achieved by providing feed-forward and feed-back, and also aims to minimize the need for abbreviations of function names. Whereas pseudo-codes abbreviate function names such as summations, square roots or integrals, dedicated braille codes implement (sometimes multi-cell) characters. Feed-back is provided by both LaTeX and Dedicon code by the use of grouping symbols, which could cause problems when nested groups are involved. Whereas LaTeX commonly uses curly braces (two-cell symbols in most braille tables), Dedicon uses regular parentheses (one-cell symbols in most braille tables). Counting of either is still required when nested structures are used in either notation.

Required size

When looking at the size of our example formulas using the selection of braille notations, it can be stated that LaTeX (a pseudo-code) and Unified English Braille (a dedicated braille code) both produce relatively lengthy results. This is shown in more detail in Table 5.3. The remaining notations that include both a pseudo-code (Dedicon) and dedicated braille codes (Woluwe code and Nemeth code) are comparable in terms of size; depending on the structure type (e.g. fractional, summation, integral) one is shorter than the other. Dedicon being a pseudo-code offers significant improvements over LaTeX in terms of size. The downside of pseudo-codes in general is however also visible; grouping

	LaTeX (Br)	**Woluwe**	**UEB**	**Dedicon (Br)**	**Nemeth**
Linear	17 (18)	21	30	29 (32)	20
Linear, fraction	49 (59)	25	33	26 (31)	25
Spatial	33 (46)	36	51	35 (41)	39
Spatial, fraction	32 (41)	24	29	24 (25)	26
Summation, fraction	41 (58)	41	55	54 (68)	43
Integral	31 (40)	29	41	36 (44)	30
Group	87 (96)	61	78	63 (68)	58

Table 5.3.: Formula types and the amount of braille cells required when translated using various braille codes

symbols such as parentheses are common and don't necessarily assist the reader. Looking at the difference in size of LaTeX in comparison to Unified English Braille, it can be observed that LaTeX contains a high amount of grouping symbols such as curly braces and parentheses, whereas the UEB-encoded formulas contain symbols that span up to four braille cells (including up to two spaces). It can be concluded that the length of UEB-encoded formulas is a direct result of the braille table and spacing rules applied, as has also been said during user research by RNIB in the United Kingdom[25].

Taking the average size of our example formulas, Woluwe code requires the least amount of characters, followed by Nemeth code, UEB and LaTeX (equal size), and Dedicon.

8-dot braille

Looking at the specifications of the braille codes, no explicit support for 8-dot braille has been mentioned. It is however possible to combine 8-dot braille with all braille codes. In such situation, the size of all mathematical work can be decreased thanks to modifier symbols as described in paragraph 4.1.

Linguality

All pseudo-codes are relatively lingual; function names are abbreviated and easy to understand. Pronunciation of formulas is therefore relatively easy as well. Examples of abbreviated function names involve *"sqrt"* for square root and *"int"* for integral. All dedicated braille codes make use of braille-specific symbols to decrease the size of mathematical formulas. When computers are involved, speech output of dedicated braille codes could be similar to that of pseudo-codes, as translation to a universal system is possible. More information on this subject was covered in chapter 4.3.

Support for secondary and higher education

The example formulas were chosen to include the most common mathematical structures in secondary education in the Netherlands. All braille codes were able to express these formulas, suggesting that all these notations are suitable for secondary education. When it comes to higher education, LaTeX is the most flexible notation; it can be expanded by the user, and expansions are provided in the form of packages by a supporting community. Whether or not higher education is covered in other notations is hard to say, although judging from the specifications of both the Nemeth code and Unified English Braille these standards seem mature enough for use in higher education as well. The Woluwe code specifications used to express the example formulas only covers the basics, but the enhancements that were made in 2012[52] should make it possible to cover higher education mathematics as well. The Dedicon code appears to not cover subjects from higher education at the moment, as has been confirmed in personal communication with users. The Dedicon code is however the most recently developed notation. This code can, in theory, be expanded by users in a similar way this is done by LaTeX users. The userbase of this code is currently relatively low.

Production on qwerty-keyboards

As was the case with linguality, the division between pseudo-codes and dedicated braille codes are key to production on qwerty-keyboards. Pseudo-codes such as LaTeX and variations, and Dedicon code, can be expressed using computer keyboards and can be read directly from a computer screen without the need for conversion. Dedicated braille codes lack this functionality directly, but with the right software conversions are possible (as has been described in chapter 4.3).

Assistive spacing

Assistive spacing in the context of this research involves the separation of mathematical content by the use of spaces. This can be done with function names, arithmetic operators, comparison signs, or to start or end sub-expressions. To quantify the level of assistive spacing, we look at the positions at which assistive spaces are used in formulas according to the official specification. Possible positions of spaces include before or after signs of comparison and mathematical operations. When the same rules apply for non-linear context (e.g. square roots, superscripts) these are counted as well.

The highest level of assistive spacing is offered by the Dedicon notation; every sign of comparison and every sign of operation is preceded and followed by a space. In both Unified English Braille and Nemeth code, signs of comparison are preceded and followed by a space, but arithmetic operators are usually not preceded or followed by a space[53][54]. The Woluwe code specification describes that spaces only precede signs of comparison and arithmetic operators. The Woluwe code also describes an exception; when the context is non-linear (square roots, fractions, integrals, summations, sub-scripts, super-scripts) no spaces are used at all[55].

5.3.2. Comparison

Having looked at the set requirements, a scoring schema was developed to provide a quantitative comparison. This schema is based on a scoring of 0-5, where 0 means exclusion of the requirement, and 5 means that the related notations scores best on the requirement. Scores can be the same when the way two notations meet the requirement are equal. This can be the case when the exact same spacing rules or size is required for our example formulas. When a requirement is not met by each notation, the maximum score is reduced by the number of notations that don't meet the requirement. The requirement for required size is based on the size in braille. For pseudo-codes the American braille table was used, whereas other notations use the braille table that is common in the country/countries the related braille code is used in.

Our comparison, as shown in table 5.4, shows that the quantitative differences between dedicated braille code does not vary much. Pseudo-codes are less supportive towards the reader, as can be concluded by the lower scores for feed-forward, feed-back, and required size.

	LaTeX	**Woluwe**	**UEB**	**Dedicon**	**Nemeth**
Feed-forward	2	4	4	1	4
Feed-back	1	4	4	2	4
Required Size	3	5	3	1	4
8-dot braille support	1	1	1	1	1
Lingual	2	0	0	1	0
Secondary and higher education	5	4	4	3	4
Production on qwerty-keyboards	2	0	0	2	0
Assistive spacing	1	2	4	5	4
Total	17	20	20	16	21

Table 5.4.: Assessment of braille codes, higher scores are better for each requirement

6. Comparing mathematical formulas

As described in chapter 4.2.2, it has become common to enroll blind students in regular, non-specialized, education where regular teaching methods and materials are used. In personal communication it has been said that mathematics levels B and D are, in comparison to mathematics A and C, less popular among blind learners. We believe that cognitive aspects could be a reason for this phenomenon, as these mathematics levels include fields of mathematics, such as advanced geometry, that are harder to perceive by blind learners. The use of graphical calculators, spatial figures and complex mathematical formulas could also make mathematics levels B and D harder for blind learners. Although specialized software and hardware exists [1] to support blind learners with graphical representations of information, braille users are still limited to the braille system for reading mathematical formulas. Because reading braille is, in most cases, slower than reading print[15], another reason for the lower demand for mathematics B and D curricula by blind learners could be related to the reading handicap that braille forms with the claimed[2] more complex mathematical formulas.

It is my hypothesis that the complexity of mathematical formulas indeed forms a greater obstacle to blind learners for mathematics B and D than mathematics A and C. In order to prove this hypothesis, we need to compare the complexity of mathematical formulas in various mathematics levels of Dutch education. In this chapter, we introduce a model to define how hard it is to read and understand a formula. This model will then be used to compare mathematics level B with the more popular mathematics levels A and C. As discussed in section 4.2.1, mathematics level D is an expansion of mathematics B. For that reason, it is not treated separately in our comparison.

6.1. Model for cognitive complexity of mathematical formulas

As briefly described in chapter 5, the structure of a mathematical formula influences the length of representation in the various braille codes. The table on page 36 shows that mathematical structures such as integrals, summations and fractions take a lot more space than, for instance, a linear formula. In order to compare complexity of mathematical formulas, it is important to consider the meaning of sub-structures as well as the size of entire formulas. We consider a fomula as a set of sub-expressions. Looking at the deep sentence structure of such sub-expression, it can be said that a sub-expression:

- Either is an elementary expression, in which case it may be a number or a variable

[1] Hardware and software to assist blind users is described in chapter 4.3

[2] Mathematics teachers in interviews mentioned that formulas in mathematics A are generally shorter and easier to read than those in mathematics B

- Or consists of an operator (predicate) with a number of associated components (arguments)

A predicate can either be an arithmetic operator, or a function. Examples of binary arithmetic operators are the addition and subtraction symbols. Functions can be unary, such as the sine and cosine function, but can also take multiple arguments. Examples of compound expressions with binary operators are:

- 2+3*5
- (2+3)*5

An example of an operator that takes four arguments, including a sub-expression, is the Greek capital sigma that is used to represent a summation function. The example below shows the four arguments a, b, c and d. The argument d is typically a function that is executed c-b times, in which argument a can be used.

$$\sum_{a=b}^{c} d$$

The deep sentence structure of an expression can be shown using a parse tree. A parse tree shows the operators, and connects those operators to its associated arguments by lines, forming a tree. Parse trees corresponding with the examples containing binary operators are shown in figure 6.1. In these parse trees, each level (operator) as well as each argument adds complexity. We will quantify this complexity by introducing a model for Cognitive Complexity.

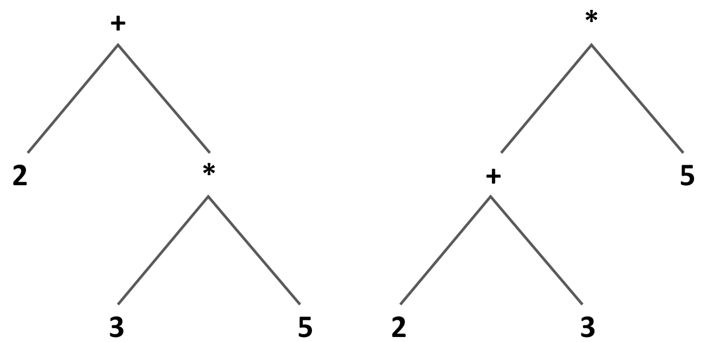


Figure 6.1.: Deep sentence structure of mathematical expressions

We define the Cognitive Complexity of a mathematical formula as a numeric value that gives an indication of how hard it is to not only read, but also understand the given formula. It is understandable but not proven that a formula with a high Cognitive Complexity will also have a higher Cognitive Load for the reader. This model was designed mainly to compare complexity of mathematical formulas. It is believed that complexity depends on three main factors:

- The numbers and variables used in the expression
- The operators and functions used in the expression
- The grouping characters used in the expression

The Cognitive Complexity of an expression (formula) in this model, is the cognitive complexity of all sub-structures and an additional factor for each sub-structure. In the parse tree shown before, each level indicates a sub-structure, and starts with one or more operators. In the formula below, Op_i stands for Operation i of the main expression E. A_{ij} refers to Argument j of operator i. The additional factor $i - 1$ adds the index of an operation to the cognitive complexity value, where the first operator adds 0. The variable n refers to the number of operators, whereas m refers to the number of arguments of the associated operator.

$$CC(E) = \sum_{i=1}^{n}(CC(Op_i) + (i - 1) + \sum_{j=1}^{m} CC(A_{ij}))$$

The Cognitive Complexity of numbers and variables, operators and functions, and grouping characters involved in the formula are defined below.

Numbers and variables

Although the length of a mathematical expression is an important factor in Cognitive Complexity, we argue that the length of a number is not a good method of measurement. For example, the number 83,000,000 may be easier to remember (and apply) than the number 8371 - which is significantly shorter. For this reason, the scientific notation is used to write numbers in the shortest possible way. This notation has the format $a * 10^b$ or aEb, in which a is called the mantissa, and b is called the exponent. The number 83,000,000 can be written as $83 * 10^6$ or $83E6$ using the scientific notation. The number 8371 can be written as $8.371 * 10^3$ or $8.371E3$, which only increases its length. The scientific notation allows for a better comparison of size of mathematical expressions; intuitively 8371, counting 4 characters in normal notation and 7 in scientific notation, is harder to remember than 83,000,000 - counting 8 characters in normal notation, but only 4 in scientific notation.

We define the Cognitive Complexity of a number CC(N) equals the length of the mantissa L(mantissa) plus the length of the exponent L(exponent) plus 1 (for the letter E).

$$CC(N) = L(mantissa) + L(exponent) + 1$$

The Cognitive Complexity of a variable CC(V) consisting of letters, is defined as the length of the letters required to write the variable name L(V). When this L(V) is larger than 5 and forms one or more words, CC(V) will be maximized at 5. In formula form:

$$CC(V \mid L(V) \leq 5) = L(V)$$

$$CC(V \mid L(V) \geq 6) = 5$$

Operators and functions

The Cognitive Complexity of an operator or function depends on the type of operator and the Cognitive Complexity of its arguments. The Cognitive Complexity of an operator or function itself is a constant, because remembering or applying a division operator is believed to be harder than, for instance, remembering or applying an addition or subtraction operator. The Cognitive Complexity of the arguments are based on rules for numbers, variables, operators and grouping characters, as arguments themselves are considered sub-expressions as well.

Functions surrounding expressions, such as $f()$ and $g()$ will have a constant Cognitive Complexity similar to operators. The general rule is that the Cognitive Complexity equals the number of characters used to name the function. The exceptions to this rule are common textual functions, such as sine (sin) and cosine (cos), that are believed to have a lower cognitive complexity.

Grouping symbols

Similar to operators, a set of grouping characters, such as parentheses, is believed to have a constant Cognitive Complexity. When nested, grouping characters do add complexity, as more reading is involved, and confusion can occur. In addition to symbols such as parentheses solely used as a method of grouping, such symbols can also span multiple rows. In such ituation they either carry a special meaning, such as the binomial coefficient, or group multiple rows of content, such as a matrix. Enlarged curly braces, spanning multiple rows to group a set of functions are used as well. Such enlarged grouping symbols are believed to add complexity for each row they span.

6.2. Quantification of complexity

In section 6.1 we described a model for cognitive complexity of mathematical formulas, that can, when applied, be used to compare complexity of mathematical formulas. In order to do this, we need to define the complexity of mathematical operators, functions and grouping symbols. This quantification will be used in section 6.3 to assess the complexity of mathematical formulas used in mathematics levels A, B, C and D in Dutch pre-university (VWO) education. The decisions made to quantify complexity of mathematical operators, functions and grouping symbols is not backed by research, and is used solely as an instrument to compare complexity.

The relevant mathematical, functions andgrouping symbols operators are divided in four main classes of complexity:

- Class 1: Basic operators
- Class 2: Operators with a spatial nature
- Class 3: Functions
- Class 4: Operators and functions with more than two arguments

Basic operators

Basic operators are operators that are mainly taught in primary education, require two arguments, and are not commonly written in a spatial way. Basic operators have a Cognitive Complexity of either 1 or 2. Table 6.1 shows the basic operators, the Cognitive Complexity assigned, and an example illustrating the calculation. One example will be explained in more detail below the table.

Operator	**CC(Op)**	**Example**
= (Equals symbol)	1	$CC(a = b)$ =1+1+1=3
+ (Plus symbol)	1	$CC(a + b)$ =1+1+1=3
- (Minus symbol)	1	$CC(a - b)$ =1+1+1=3
* (Multiplication symbol)	2	$CC(a * b)$ =1+2+1=4
(){}[] (grouping symbols)	1	$CC(a * (b * c))$ =1+2+0+1+1+2+1+1=9
' (prime symbol)	1	$CC(f'(x))$ =1+1+1+1=4

Table 6.1.: Table of basic operators, their Cognitive Complexity, and an example showing calculation of this Cognitive Complexity

The formula $a * (b * c)$ has a Cognitive Complexity of 9, which is based on the sum of variables, operators, and parentheses. From left to right it contains the variable a, the multiplication operator (with index 0), an opening parenthesis, the variable b, the multiplication operator (with index 1) and the variable c. This sums up to $CC(V) = 1 + 1 + 1 = 3$, $CC(Op) = 2 + 0 + 2 + 1 = 5$, $CC(Grp) = 1$. The closing parenthesis is ignored in this calculation; the Cognitive Complexity for grouping symbols is calculated per group.

Operators with a spatial nature

Mathematical operators that are commonly written in a spatial way are, in terms of this research, more complex than basic operators. Even though some of these operators are taught in primary education, it is believed that these operators are more complex for people with a reading handicap.

Operator	**CC(Op)**	**Example**
ˆ (Exponent symbol)	2	$CC(a\hat{}b)$ =1+2+1=4
$\surd$ (Square root symbol)	3	$CC(\sqrt{(25)}$=3+1+2=6
/ (Division symbol)	4	$CC(a * b/c)$ =1+2+0+1+4+1+1=10

Table 6.2.: Table of operators with a spatial nature, their Cognitive Complexity, and an example showing calculation of this Cognitive Complexity

The formula $a * b/c$ has a Cognitive Complexity of 10. From left to right it contains the variable a, the multiplication symbol (with index 0), the variable b, the division symbol (with index 1) and the variable c. This sums up to $CC(V) = 1 + 1 + 1 = 3$, $CC(Par) = 0$, $CC(Op) = 2 + 0 + 4 + 1 = 7$.

Functions

Common mathematical functions include placeholder functions such as $f()$ and $g()$, but also include functions such as sine ($\sin$) and cosine ($\cos$). The amount of letters to write these function names are typically higher than what we believe is the Cognitive Complexity of such functions. Regular words, such as sin and cos are relatively easy to read and understand.

Function	**CC(F)**	**Example**
Single-letter function	1	$CC(f(x))$ =1+1+1=3
Common three-letter functions	2	$CC(\sin a)$=2+1=3
Expansions of common three-letter functions	3	$CC(\arcsin a)$ =3+1=4

Table 6.3.: Table of common mathematical functions, their Cognitive Complexity, and an example showing calculation of this Cognitive Complexity

The formula $f(x) = 1$ has a Cognitive Complexity of 5. From left to right, the function f, the opening parenthesis, the variable x, the equals symbol and the variable 1 are used. This sums up to $CC(F) = 1$, $CC(V) = 2$, $CC(Op) = 1$, $CC(Grp) = 1$.

Operators and functions with multiple arguments

The final class of operators and functions are those with multiple arguments. Besides the increased number of arguments, that count as sub-expressions or variables as well, it is believed that this class of operators and functions form an obstacle to blind users. We believe this is the case because when reading from left to right, all arguments must be remembered as well as the relation to the operator.

Operator	**CC(F)**	**Example**
$\int$ (Integral)	6	$CC(\int_a^b v(t)dt)$ =6+1+1+1+1+1+1+1=13
$\sum$ (Summation)	7	$CC(\sum_{i=1}^n a * r^i)$ =7+1+1+0+1+1+1+2+1+1+2+2+1=21

Table 6.4.: Table of common mathematical functions and operators that take multiple arguments, their Cognitive Complexity, and an example showing calculation of this Cognitive Complexity

The formula $\sum_{i=1}^n a * r \hat{} i$ has a Cognitive Complexity of 21. From left to right, it contains the summation symbol, the variable i, the *equals* operator (with index 0), the variable 1, the variable n, the variable a, the multiplication symbol (with index 1), the variable r, the exponent symbol (with index 2) and the variable i. This sums up to $CC(F) = 15$, $CC(V) = 6$, $CC(Grp) = 0$.

6.3. Assessment of mathematical formulas

Having introduced a model for Cognitive Complexity of mathematical formulas and a quantification of common sub-structures of mathematical formulas, we are able to compare complexity of mathematical formulas. It is our aim to use this model to compare Dutch mathematics curricula in terms of complexity of mathematical formulas. In order to achieve this, we take all mathematical formulas used in mathematics levels A, B, C and D as used in Dutch secondary (pre-university) education. These formulas were copied from formula summaries from publisher Wolters-Noordhoff[56][57] and checked by a mathematics expert. These formulas are the formulas learners must be able to understand and apply in exams.

In order to assess all formulas, a simplified version of our model for Cognitive Complexity was used. In this simplified version, we eliminate the increased complexity for each additional operator or function. This means that, for example, $CC(1+2+3+4+5)$ will not equal 15 (Var=5, Op=1+2+3+4) but 9 (Var=5, Op=1+1+1+1=4). Although the influence of this factor, especially with more complex formulas, is noticeable, it is believed that the outcome of this analysis will be comparable to using the original model. The advantage of this model is the fact that it allows for quick analysis. It is believed that the outcome will comparable because the initial hypothesis, "mathematics B is more complex than mathematics A and C in terms of mathematical formulas", was partially based on the mathematical structures identified in these formulas. The structures we identified, as was briefly discussed in chapter 5.3, include linear formulas, fractions of varying complexity, spatial formulas, summations, integrals, groups, and combinations of these structures. Table 6.3 shows the amount of mathematical formulas of each identified structure and combination of structures in mathematics levels A+C, B and the formulas used in all curricula. Mathematics level D is not assessed separately, as it is considered an expansion of mathematics B. In this analysis, fractions were counted and categorized in three levels because it is believed that, for instance, $\frac{a}{b}$ is significantly less complex than $\frac{a}{f(b)}$, which in turn is less complex than $\frac{f(a)}{g(b)}$. The levels are defined as follows:

- Category 1: Numerator and denominator contain no functions
- Category 2: Either the numerator or denominator contains a function
- Category 3: Both the numerator and denominator contain a function

In table 6.3, category 1 fractions are shown first, followed by categories 2 and 3. From this quick analysis, one can conclude that mathematics B requires understanding of more formulas in general, but in relative terms the percentage of more complex structures is significantly higher as well. In the previous paragraphs we showed that more complex mathematical structures often result in a higher Cognitive Complexity, supporting our hypothesis.

	A+C	**B**	**A+B+C(+D)**
Linear	9	44	39
Fraction	2,0,1	5,2,2	13,4,2
Spatial and Fraction	1	0	0
Summation and Fraction	0	1	1
Integral	0	0	3
Group	0	0	1

Table 6.5.: Amount of mathematical formulas of a certain structure in mathemtics curricula

Applying our simplified version of the Model for Cognitive Complexity of mathematical formulas involved counting of variables (and their lengths), grouping symbols, and mathematical structures. The sum of the Cognitive Complexity of these parts was then calculated. All formulas and their Cognitive Complexity according to this simplified model are shown in Appendix B. The main criterium for mathematics B being more complex in terms of complexity of its mathematical formulas is the average and median cognitive complexity being higher than those of mathematics A and C. Table 6.6 shows a summary of the analysis of Appendix B and shows properties of mathematics levels A and B exclusively, as well as the combination with the formulas shared among all curricula. It can be seen that formulas in mathematics level B have a higher average and median Cognitive Complexity than those in mathematics levels A and C. From this we can conclude that our hypothesis is confirmed; mathematics level B is more complex than mathematics levels A and C in terms of mathematical formulas.

	A-only	**Shared**	**B-only**	**A**	**B**
Average	14.5	13.6	16.4	13.8	15.2
Minimum	4	5	2	4	2
Maximum	33	35	41	35	41
Median	13	12	15	13	14
n	14	51	68	65	119

Table 6.6.: Average, Minimum, Maximum, Median values of Cognitive Complexity of formulas in the associated curriculum

7. Conclusions

In this research we assessed and compared braille codes and mathematical formulas in the context of blind learners in secondary education. The results allowed us to find out why Dutch mathematics education, and specific parts of it, cause problems for blind learners. In order to place this in a broader context, the cognitive aspects of blind learners and the developments in the field of education, braille and technology were researched. Combining the results allows for drawing conclusions about the problems in Dutch mathematics education for the blind. With this information and future research, it may be possible to make mathematics education more accessible for the blind. The main question that will be answered in this concluding chapter is:

- How complex are mathematical formulas and how can they be represented in Braille?

In order to answer this question, sub-questions were described and answered in separate chapters of this thesis. The sub-questions are categorized as contextual (chapters 3-4), comparing braille (chapter 5) and comparing mathematics (chapter 6). The same structure and order is used in this concluding chapter.

Based on findings from contextual questions, the main question is answered by combining the answers to sub-questions 5 and 6. Briefly put, we conclude that mathematical formulas can be compared in terms of complexity by introducing a model for Cognitive Complexity of mathematical formulas. In Dutch high school pre-university mathematics, the mathematics B curriculum requires use of more complex mathematical formulas than mathematics A and C. Combined with the knowledge that this curriculum is rarely chosen by blind learners, we conclude that the braille representation for mathematics forms an obstacle. During our research we found that there are many braille codes describing grammar-like rules for mathematics, of which the resulting formulas can also be compared. This comparison was partially based on requirements from previous research, and showed a separation of dedicated braille codes and codes we refer to as *pseudo-codes* that can also be represented in print. We concluded that dedicated braille codes are better at assisting the braille reader than pseudo-codes are. The differences between dedicated braille codes were not significant; using our requirements and a scoring scheme where a higher score indicates more assistive properties, the resulting scores were 20 (Woluwe code), 20 (Unified English Braille) and 21 (Nemeth code). The pseudo-codes Dedicon and LaTeX scored 16 and 17 respectively. Looking at individual requirements, user preference as well as the braille table that is used have more influence on the strenghts of braille codes.

7.1. Contextual findings

This section contains conclusions based on the findings in chapters 3 and 4, and answers the following questions:

1. How do blind children learn in comparison to sighted children?
2. What advancements to the braille system have been made to support mathematics?
3. How is mathematics taught in the Netherlands?
4. What technological developments support blind learners in the field of mathematics?

How do blind children learn in comparison to sighted children?

In order to research the way blind children learn, literature was referenced and backed up by interviews with experts in the field of pedagogical and educational sciences. We expected to find that graphical content would be the main problem for blind learners, as perception of this content using tactile or auditory senses is extremely difficult. The main reason for this difficulty is the successive nature of these senses, allowing to perceive just one piece of information at a time, lacking any form of overview. In addition to graphical content, it has become clear that this limitation of tactile senses is also an important factor when reading non-graphical mathematical content, such as formulas, in braille. Tactics for reading mathematical formulas that are often applied by sighted users are not suitable for braille readers, resulting in an increased cognitive load when reading mathematical content. The cognitive load is defined as *"the amount of mental activity imposed on the working memory at an instance in time"*, and is an important factor when looking at reading tactics for mathematical content. These tactics include the concept of *chunking* and getting an overview of the structure of a formula prior to reading it in detail. Chunking involves reading and/or solving parts of the formula and remembering the outcome, whereas getting an overview involves generation of a mental image to understand the meaning of a sub-formula at a certain position. Either tactic is unusable for blind children without memorizing the entire formula.

What advancements to the braille system have been made to support mathematics?

Through literature and interviews with teachers and educational coaches of blind learners, the strengths and weaknesses of braille in the context of mathematics were investigated. Previous research has indicated that reading braille is typically 50% slower than reading print. Reading braille can in that sense be considered a reading handicap, that is much stronger for slow braille readers than it is for faster readers. It has been proven that learners with low reading abilities are also less skilled in mathematics, indicating that it is possible that braille readers require more effort in order to learn mathematics, regardless of two-dimensional content. The braille system itself, from an international perspective, varies a lot; only the 26 letters of the alphabet as developed by the Frenchman Louis Braille are internationally agreed upon. All special symbols, indicators and punctuation marks are different around the world, mainly because of the differences in language. In order to display non-French characters, and to decrease the size required for braille content, changes to the braille system have been made in many countries. Besides the braille table itself, abbreviations or contractions of words and the addition of braille rules (referred to as braille code) were made.

It can be said that all but the simplest of mathematics is non-linear, whereas the braille system is entirely linear; every character is placed at the same vertical position on the baseline, and has the same (maximum) size of 2x3 dots. In order to express mathematics, braille codes for mathematics have been introduced, that allow linearization of non-linear mathematical content. This development has happened in various ways, independently, in many countries in the world. Although the field of mathematics, like music, can be considered language-independent, no universal braille code was ever introduced for mathematics. In this research we focussed on the notations for mathematics used in North America, the United Kingdom, Belgium and the Netherlands. North America, the United Kingdom and Belgium have used braille codes specifically designed for mathematics and science for several years, and changes are still being made to them. In the Netherlands, no real braille code exists, but a pseudo-code that is also readable by sighted users is being used.

How is mathematics taught in the Netherlands?

Besides the differences in perceiving mathematical content and the increased cognitive load for blind learners, mathematics education is highly focussed on visual aspects. Teaching goals for secondary mathematics education rely on the learner understanding certain concepts, such as the differences and similarities between objects and processes, symbol sense, the meaning of algebraic expressions and the development of mental schemas. Teaching methods used to achieve these goals make use of common reading tactics to guide learners; properties such as symmetry and visual similarities between exercises and examples, and color-coding are common. These methods are not suitable for blind learners, but as the same textbooks are used for blind learners it is understandable that blind learners have more problems with mathematics. Looking at the curricula offered in the Netherlands, four mathematics levels can be chosen in the final years of secondary (pre-university) education; levels A, B, C and D. Mathematics A and C are similar in contents, and focus on applied mathematics and prepare mostly for Alpha-studies. Mathematics B and D use a more fundamental approach, and prepare for Beta-studies. By definition, mathematics level B is not more complex than mathematics A or C. Mathematics D is an expansion of mathematics B, containing parts of mathematics A and C.

What technological developments support blind learners in the field of mathematics?

Technology in the context of this research involves the possibilities of hardware and software to assist blind learners. Most of this research was based on published articles, online references, personal communication with teachers and blind students, and information provided by vendors of assistive aids and software for the blind. Refreshable braille displays, braille note-takers and braille typewriters have been available for many years, as well as screen reading and speech software to support these devices. Such software however offers limited support for mathematis; both braille and speech output for mathematics is, in most cases, a character-by-character output of the original content. Alternatives to a character-by-character output exist and rely on automated translation of braille codes and visual representations (often using a universal format such as MathML), but these are not widely used. The variation in braille codes and tables limits usefulness of such translation and back-translation software as well; no universal application is available, and braille codes and tables constantly change. In order to share mathematical work with sighted users, many braille users use pseudo-codes such as LaTeX or variations of it. These notations generally require more space than braille codes when represented on a refreshable braille display, thereby increasing the cognitive load for the reader.

7.2. Comparing braille codes

This section contains conclusions based on the results from chapter 5 and answers the following question:

5. In what way can braille codes be compared?

It was our aim to assess and compare the possibilities of using braille to represent mathematical formulas. In order to compare the notations used to represent mathematics in braille, requirements were described that were partially based on existing research. The requirements used are: support of feed-forward, feed-back and 8-dot braille, the level of linguality, the required size, coverage of high school and higher mathematics, production on qwerty-keyboards and the use of assistive spacing. Braille notations compared include both dedicated braille codes that are not suitable for visual representation (e.g. containing characters unique to braille), and pseudo-codes that are also usable by sighted users without the need for conversion. Dedicated braille codes used in our comparison include the Nemeth code, Unified English Braille and Woluwe code. The pseudo-codes that were also compared are LaTeX and the Dedicon notation. The Dedicon notation is a pseudo-code that was developed in, and is only used in the Netherlands. The braille tables used to represent formulas in these braille codes depend on the country in which the associated braille code is most commonly used, except for LaTeX and Dedicon notation; for these notations the American braille table was used because of its popularity in the Netherlands.

From the comparison found in chapter 5.3, it can be concluded that the braille codes don't vary much in terms of our requirements. Assuming all requirements are equally important, braille codes do score higher than pseudo-codes, but it cannot be stated that one braille code is significantly more supportive to the reader than another. It can however be said that dedicated braille codes, according to these requirements, are more suitable for use by blind users. This advantage is however, with the current technology, mostly limited to reading. Production of mathematics in dedicated braille codes requires conversion from a pseudo-code such as LaTeX, or production using just six keys of the computer keyboard (or braille notetaker), which is significantly slower than typing pseudo-codes using ten fingers on a regular computer keyboard.

7.3. Comparing mathematics

This section contains conclusions based on the results from chapter 6 and answers the following question:

6. In what way can mathematics levels be compared in terms of complexity of mathematical formulas?

One of the findings in an early stage of this research project, is the observation that mathematics level B is rarely chosen by blind learners. Reasons for this phenomenon that were mentioned in personal communication varied, but often included the problems braille users have with notations for mathematics. In order to prove the hypothesis that indeed the braille notation causes problems for mathematical formulas, the mathematical formulas in Dutch secondary (pre-university) education were compared in terms of complexity. In order to do this, a model for Cognitive Complexity of Mathematical Formulas was introduced, that was then used to quantify how hard it is to read and understand a given mathematical formula. Using a simplified version of this model, all formulas in Dutch secondary (pre-university) mathematics education were compared, allowing us to compare mathematics curricula in terms of complexity of its mathematical formulas.

Our simplification of the original model for cognitive complexity reduces the complexity value of more complex formulas, slightly reducing the gap between formulas with a medium complexity and a high complexity. Even with this change, however, it can be concluded that mathematics level B is indeed more complex than mathematics levels A and C, in terms of complexity of the mathematical formulas used. The average complexity of a formula in mathematics A is 13.4, compared to 15.2 in mathematics B. Looking at the median, mathematics A scores 13 in comparison to 14 in mathematics B. Finally, the importance of mathematical formulas in mathematics level B is higher than in mathematics A, as indicated by the number of formulas used; mathematics A requires a total of 65 mathematical formulas, compared to 119 in mathematics B.

7.4. Discussion

We concluded that dedicated braille codes for mathematics have advantages for reading mathematics compared to pseudo-codes. We also concluded that certain mathematics levels are less accessible to the blind because of complexity of mathematical formulas. There is however a drawback in using dedicated braille codes; presentation to sighted users. With the use of computers, it is theoretically possible to translate any braille code for mathematics into a universal format such as MathML. However, projects that aim to achieve this are commonly behind on development of braille codes they rely on. One way to minimize this problem would be to reduce the amount of braille codes altogether, and standardize the braille codes that remain. This also implies that national and international braille authorities need to cooperate more.

During a symposium on Braille in Mathematics Education[1] a discussion resulted in four key recommendations that reduce problems with mathematics education for the blind. These recommendations are:

- Facilitation: Facilitate extra time for teachers of blind learners
- Standardization: Standardize braille codes and tables
- Change management: Form an authority to decide on the future of braille in the Netherlands
- Collaboration: Collaborate with national and international braille authorities and relevant organisations

7.5. Future research

This research resulted in a handful of improvements that can be applied to Dutch mathematics education for the blind. One of the main topics involved the use of a standardized braille code to express mathematical formulas. Although a similar approach has been taken in other countries, this is new for the Netherlands. An introduction of a braille code would therefore require more research. It is possible that, despite the positive characteristics of the found braille codes, the approach cannot be easily applied to the Netherlands. Topics that require more research include:

- The influence on existing braille material
- The influence on production of digital and physical braille material
- The required changes in education for the blind in both regular and specialized education
- The way braille readers can learn about the new braille codes

From a more practical standpoint, research involving braille users is needed in order to decide on the implementatio of a braille code. Questions that need answering include the following:

- Which braille code is most suitable for the Netherlands?

[1]Symposium held at Radboud University Nijmegen, January 25th 2013

- Is it wise to change the Dutch braille table to support mathematics in a better way?
- If a change to the Dutch braille table is required, is there an existing braille table that is suitable for the Netherlands?

Finally, there needs to be more research on a technical level as well. Despite the advancements in software projects over the past years, it is believed that a more integrated approach is required to assist the braille readers most. Screen reader software is still behind on displaying mathematics, and production of mathematics for the blind needs to be researched as well. A recommendation I would personally make, involves a research project combining experts in the field of mathematics, braille, and the vendors of popular screen readers. A collaboration with existing mathematics applications, as mentioned in chapter 4.3.2, would be beneficial as well.

Appendix A Braille translations of mathematical formulas

In chapter 5.2, three mathematical formulas were translated into LaTeX, Woluwe code, Unified English Braille code, Dedicon code and Nemeth code. To improve readability of that chapter, the remaining three formula structures and their translations into these braille codes are shown in this appendix. The formulas translated are shown in table A.1.

Structure Type	Example formula	Page
Linear	$n! = n * (n-1) * .. * 2 * 1$	30
Fraction	$\sigma(\bar{X}) = \frac{\sigma(\bar{X})}{\sqrt{n}}$	32
Spatial	$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$	55
Spatial and fraction	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$	33
Summation and Fraction	$S = \sum_{k=0}^{n-1} ar^k = a\frac{1-r^n}{1-r}$	56
Integral	$L = \int_a^b \sqrt{1+(f'(x))^2}dx$	57
Group	$\begin{cases} x(t) = m + r\cos\omega(t-t_0) \\ y(t) = n + r\sin\omega(t-t_0) \end{cases}$	58

Table A.1.: Example formulas and location of translations in this document

A.1. Spatial

The formula involving a spatial structure is an application of the spatial and fractional formula for the binomial coefficient. The formula is:

$$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$$

LaTeX

In LaTeX the formula is written as "P(X=k)=\binom{n}{k} p^k(1-p)^{n-k}". It counts 33 characters in print, and 46 characters when using the American braille table.

⠠⠏⠷⠠⠭⠨⠅⠅⠾⠨⠅⠳⠃⠊⠝⠕⠍⠨⠷⠝⠨⠾⠨⠷⠅⠨⠾⠡⠏⠸⠇⠅⠷⠂⠤⠏
⠾⠸⠇⠨⠷⠝⠤⠅⠨⠾

Woluwe encoded

The Woluwe encoded formula counts 36 characters.

⠤⠏⠣⠨⠭⠀⠶⠅⠜⠀⠶⠣⠿⠝⠰⠣⠅⠱⠜⠦⠏⠌⠅⠱⠣⠼⠁⠤⠏⠜⠌⠣⠝⠤⠅⠜

Unified English Braille encoded

The UEB-encoded formula counts 51 characters.

⠠⠏⠐⠣⠸⠭⠀⠐⠶⠀⠅⠐⠜⠀⠐⠶⠀⠠⠐⠣⠝⠰⠻⠅⠐⠜⠐⠲⠏⠰⠔⠣⠅⠜⠐⠣⠼
⠁⠐⠤⠏⠐⠜⠰⠔⠣⠝⠐⠤⠅⠜

Dedicon encoded

The Dedicon encoded formula appears as "P(X = k) = (n; k)p^k(1 - p)^(n - k)" and counts 35 characters in print. When translated using the American braille table, it counts 41 characters.

⠨⠏⠷⠨⠭⠀⠨⠅⠀⠅⠾⠀⠨⠅⠀⠷⠝⠆⠀⠅⠾⠏⠘⠅⠀⠷⠼⠁⠀⠤⠀⠏⠾⠘⠷⠝⠀
⠤⠀⠅⠾

Nemeth encoded

The Nemeth-encoded formula counts 39 characters.

⠠⠏⠷⠠⠭⠀⠨⠅⠀⠅⠾⠀⠨⠅⠀⠠⠷⠝⠩⠅⠠⠾⠡⠏⠘⠅⠐⠷⠂⠤⠏⠿⠘⠷⠏⠤⠅⠿⠼

A.2. Summation and Fraction

A summation uses the Greek capital Sigma with arguments directly above, below and next to it. The example formula we use to compare braille codes is shown below.

$$S = \sum_{k=0}^{n-1} ar^k = a\frac{1 - r^n}{1 - r}$$

LaTeX

In LaTeX the formula is written as "S=\sum_{k=0}^{n-1}ar^k=a\frac{1-r^n}{1-r}" and counts 41 characters. When translated using the American braille table, it counts 58 characters.

⠎⠨⠅⠳⠎⠥⠍⠤⠤⠨⠷⠅⠨⠅⠼⠴⠨⠾⠸⠇⠸⠷⠝⠤⠂⠸⠾⠁⠗⠸⠇⠅⠸⠌
⠁⠳⠋⠗⠁⠉⠨⠷⠂⠤⠗⠸⠇⠝⠨⠷⠨⠷⠼⠃⠤⠗⠸⠿

Woluwe encoded

The Woluwe encoded formula counts 41 characters.

⠨⠎⠀⠶⠸⠎⠡⠅⠤⠼⠚⠱⠌⠝⠤⠼⠁⠱⠁⠗⠌⠅⠱⠀⠶⠁⠐⠆⠼⠁⠤⠗⠌⠝⠳⠳⠼⠁⠤⠟⠸

Unified English Braille encoded

The UEB-encoded formula counts 55 characters.

Dedicon encoded

The Dedicon encoded formula appears as "s = ~ S{k = 0..n - 1}ar^k = a * ((1 - r^n) / (1 - r))" and counts 54 characters in print. When translated using the American braille table, it counts 68 characters.

Nemeth encoded

The Nemeth-encoded formula counts 43 characters.

A.3. Integral

An Integral involves a so-called Long-S symbol with arguments directly below, above and next to it. The example formula we used is:

$$L = \int_a^b \sqrt{1 + (f'(x))^2} dx$$

LaTeX

In LaTeX the formula is written as "L=\int_a^b\sqrt(1+(f'(x))^2))dx" and counts 31 characters. When translated using the American braille table, it counts 40 characters.

Woluwe encoded

The Woluwe encoded formula counts 29 characters.

Unified English Braille encoded

The UEB-encoded formula counts 41 characters.

Dedicon encoded

The Dedicon encoded formula appears as "L = intg{a..b}sqrt(1 + (f'(x))^2)dx" and counts 36 characters in print. When translated using the American braille table, it counts 44 characters.

Nemeth encoded

The Nemeth encoded formula counts 30 characters.

A.4. Group

A grouped formula involves an enlarged curly brace on the left, to group together two formulas. The example formula we used contains two formulas.

$$\begin{cases} x(t) = m + r\cos\left(\omega(t - t_0)\right) \\ y(t) = n + r\sin\left(\omega(t - t_0)\right) \end{cases}$$

LaTeX

In LaTeX the formula requires two lines, to separate the two formulas. It counts 87 characters in print, and 96 characters when translated using the American braille table.

```
\begin{cases}x(t)=m+r\cos{(\omega(t-t_0))} y(t)=n+r\sin{(\omega(t-t_0))} \end{cases}
```

Woluwe encoded

The Woluwe encoded formula counts 61 characters.

Unified English Braille encoded

The UEB-encoded formula counts 78 characters.

Dedicon encoded

The Dedicon encoded formula appears as "x(t) = m + r cos(~j(t - t_0)); y(t) = n + r sin(~j(t - t_0))" and counts 63 charactes. When translated using the American braille table, it counts 68 characters.

Nemeth encoded

The Nemeth encoded formula counts 58 characters.

Appendix B Cognitive Complexity Survey

B.1. Mathematics A, C

Num	Formula	CC
1.1.1	$n! = N*(n-1)*..*2*1$	18
1.1.2	$0! = 1$	4
1.1.3	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$	13
1.2.1	$E(X+Y) = E(X) + E(Y)$	13
1.2.2	$\sigma(X+Y) = \sqrt{\sigma^2(X) + \sigma^2(Y)}$	23
1.2.3	$E(S) = n * E(X)$	10
1.2.4	$\sigma(S) = \sqrt(n) * \sigma(X)$	14
1.2.5	$E(\bar{X}) = E(X)$	7
1.2.6	$\sigma(\bar{X}) = \frac{\sigma(X)}{\sqrt(n)}$	13
1.3.1	$P(X=k) = \binom{n}{k} * pk(1-p)^{n-k}$	25
1.3.2	$E(X) = np$	6
1.3.3	$\sigma(X) = \sqrt{np(1-p)}$	14
1.4.4	$Z = \frac{x-\mu}{\sigma}$	10
1.4.5	$P(X \leq g) = P(Z \leq \frac{g-\mu}{\sigma}) = \Phi(\frac{g-\mu}{\sigma}$	33

B.2. Mathematics B, D

Num	Formula	CC
2.2.7	$g \log a = \frac{p \log a}{p \log g}$	16
2.2.8	$g \log a + g \log b = g \log ab$	15
2.2.9	$g \log a - g \log b = g \log \frac{a}{b}$	21
2.2.10	$g \log a^p = p^g \log a$	17
2.3.3	$H = d + a \sin(b(t-c))$	15
2.3.4	$H = d - a \sin(b(t-c))$	15
2.4.3	$S = \Sigma_{k=0}^{\infty} ar^k = a\frac{1}{1-r}$	28
2.6.12a	$f(x) = \sin x$	7
2.6.12b	$f'(x) = \cos x$	8
2.6.13a	$f(x) = \cos x$	7
2.6.13b	$f'(x) = -\sin x$	9
2.6.14a	$f(x) = \tan x$	7
2.6.14b	$f'(x) = \frac{1}{\cos^2 x} = 1 + \tan^2 x$	22
2.6.15	$L(x) = f(a) + f'(a) * (x-a)$	17
2.7.1a	$f(x) = x^n$	8
2.7.1b	$\frac{1}{n+1} x^{(n+1)} + c$	17
2.7.2a	$f(x) = \frac{1}{x}$	10
2.7.2b	$f(x) = \ln(\|x\|) + c$	13
2.7.3a	$f(x) = e^x$	8
2.7.3b	$f(x) = e^x + c$	10
2.7.4a	$f(x) = g^x$	8
2.7.4b	$f(x) = \frac{1}{\ln g} g^x + c$	18
2.7.5a	$f(x) = \ln x$	7
2.7.5b	$f(x) = x \ln x - x + c$	12
2.7.6a	$f(x) = g \log x$	8
2.7.6b	$f(x) = \frac{1}{lng}(x \ln x - x) + c$	21

Num	Formula	CC
2.7.7a	$f(x) = \sin x$	7
2.7.7b	$f(x) = -\cos x + c$	10
2.7.8a	$f(x) = \cos x$	7
2.7.8b	$f(x) = \sin x + c$	9
2.7.9	$L = \int_a^b \sqrt{1 + (f'(x))^2} dx$	24
2.7.10	$I = \pi \int_a^b (f(x))^2 dx$	20
2.8.1	$\cos^2 t + \sin^2 t = 1$	15
2.8.2	$\sin(-t) = -\sin t$	10
2.8.3	$\sin(\frac{\pi}{2} - t) = \cos t$	16
2.8.4	$\sin(\pi - t) = \sin t$	10
2.8.5	$\tan t = \frac{\sin t}{\cos t}$	14
2.8.6	$\cos(-t) = \cos t$	8
2.8.7	$\cos(\frac{\pi}{2} = -\sin t$	15
2.8.8	$\cos(\pi - t) = -\cos t$	10
2.8.9	$\sin(2t) = 2 \sin t \cos 2$	12
2.8.10	$cos(2t) = cos^2 t - sin^2 t = 2cos^2(t-1)$ $= 1 - 2sin^2 t$	39
2.8.11	$\sin(t+u) = \sin t \cos u + \cos t \sin u$	20
2.8.12	$\sin t + \sin u = 2 \sin(\frac{t+u}{2}) \cos \frac{t-u}{2})$	31
2.8.13	$\sin(t-u) = \sin t \cos u - \cos t \sin u$	20
2.8.14	$\sin t - \sin u = 2 \sin(\frac{t-u}{2}) \cos(\frac{t+u}{2})$	31
2.8.15	$\cos(t+u) = \cos t \cos u - \sin t \sin u$	20
2.8.16	$\cos t + \cos u = 2 \cos(\frac{t+u}{2}) \cos(\frac{t-u}{2})$	31
2.8.17	$\cos(t-u) = \cos t \cos u + \sin t \sin u$	20
2.8.18	$\cos t - \cos u = 2 \sin(\frac{t+u}{2}) \sin(\frac{t-u}{2})$	31
2.8.19	$\sin \alpha = \sin \beta$ gives $\alpha = \beta + k * 2\pi$ or $\alpha = \pi - \beta + k * 2\pi$	27
2.8.20	$\cos \alpha = \cos \beta$ gives $\alpha = \beta + k * 2\pi$ or $\alpha = -\beta + k * 2\pi$	25
2.9.1	$v(t) = \sqrt{(x'(t))^2 + (y'(t))^2} dt$	22
2.9.2	$\int_a^b v(t) dt = \int_a^b \sqrt{(x'(t))^2 + (y'(t))^2} dt$	41
2.9.3	$\begin{cases} x(t) = m + r \cos(\omega(t - t_0)) \\ y(t) = n + r \sin(\omega(t - t_0)) \end{cases}$ with $\omega = \frac{2\pi}{T}$	38
2.10.1a	$\frac{dy}{dt} = cy$	11
2.10.1b	$y(t) = y(0) * e^{ct}$	14
2.10.2a	$\frac{dy}{dt} = c(K-y)$	14
2.10.2b	$y(t) = K + (y(0) - K) * e^{ct}$	19
2.10.3a	$\frac{dy}{dt} = c(K-y)$	13
2.10.3b	$y(t) = \frac{G}{1 + ae^{cGT}}, a = \frac{G - y(0)}{y(0)}$	33
3.1.1	$2\pi r$	3
3.1.2	ar	2
3.1.3	πr^2	5
3.1.4	$\frac{1}{2} ar^2$	11
3.2.1	$a^2 + b^2 = c^2$	14
3.2.2	$c^2 = a^2 + b^2 - 2ab * \cos \gamma$	23
3.2.3	$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma}$	26

B.3. Mathematics A, B, C, D

Num	Formula	CC
2.1.1a	$ax^2 + bx + c = 0$	12
2.1.1b	$X = \frac{-b+\sqrt{D}}{2a}$	15
2.1.2a	$X^n = c$	6
2.1.2b	$X = c^{\frac{1}{n}} = \sqrt[n]{c}$	17
2.1.3a	$gx = a$	6
2.1.3b	$X = g\log a = \frac{\log a}{\log g}$	13
2.1.4a	$g\log x = b$	6
2.1.4b	$X = g^b$	6
2.1.5a	$e^x = a$	6
2.1.5b	$x = ln(a)$	5
2.1.6a	$ln(x) = b$	5
2.1.6b	$x = e^b$	6
2.2.1	$a^{-n} = \frac{1}{a^n}$	16
2.2.2	$a^{\frac{1}{n}} = \sqrt[n]{a}$	15
2.2.3	$a^p * a^q = a^{p+q}$	17
2.2.4	$a^p : a^q = a^{p-q}$	16
2.2.5	$(a^p)^q = a^{pq}$	14
2.2.6	$(ab)^p = a^p b^p$	15
2.3.1	$H = b + a * t$	8
2.3.2	$H = b * gt$	9
2.4.1	$S = \frac{Eersteterm+Laatsteterm}{2}$	18
2.4.2	$S = \Sigma_{k=0}^{n-1} ar^k = a\frac{1-r^n}{1-r}$	35
2.5.1	$u(n) = \frac{b}{1-a}(u(0) - \frac{b}{1-a})a^n$	30
2.5.2	$u(n+1) = a * u(n) + b$, $U = \frac{b}{1-a}$	24
2.5.3	$u(n) = U + a^n(u(0) - U)$, $U = \frac{b}{1-a}$	26
2.5.4	$u(n+1) = a * u(n)$	12
2.5.5	$u(n) = u(0) * a^n$	13
2.5.6	$u(n+1) = u(n) + c * u(n) * \frac{G-u(n)}{G}$	30
2.6.1a	$g(x) = c * f(x)$	10
2.6.1b	$g'(x) = c * f'(x)$	12
2.6.2a	$s(x) = f(x) + g(x)$	11
2.6.2b	$s'(x) = f'(x) + g'(x)$	14
2.6.3a	$p(x) = f(x) * g(x)$	12
2.6.3b	$p'(x) = f\prime(x) * g(x) + f(x) * g\prime(x)$	24
2.6.4a	$q(x) = \frac{f(x)}{g(x)}$	14
2.6.4b	$q'(x) = \frac{f'(x)g(x) - f(x)g\prime(x)}{(g(x))^2}$	31
2.6.5a	$k(x) = f(g(x))$	9
2.6.5b 2.6.5c	$\frac{dk}{dx} = \frac{df}{dg} * \frac{dg}{dx}$ $k'(x) = f'(g(x)) * g'(x)$	21 17
2.6.6a	$f(x) = c$	5
2.6.6b	$f'(x) = 0$	6
2.6.7a	$f(x) = x^n$	8
2.6.7b	$f\prime(x) = n * x^{n-1}$	14
2.6.8a	$f(x) = ex$	8
2.6.8b	$f'(x) = e^x$	9
2.6.9a	$f(x) = g^x$	8
2.6.9b	$f'(x) = g^x * \ln g$	14
2.6.10a	$f(x) = \ln x$	7
2.6.10b	$f'(x) = \frac{1}{x}$	11
2.6.11a	$f(x) = g\log x$	6
2.6.11b	$f'(x) = \frac{1}{x} * \frac{1}{\ln g}$	20

Bibliography

[1] V. Aldridge. *How is Chinese written in braille?* 2002. URL: http://www.braille.ch/pschin-e.htm.

[2] M.T. De Garmo and L.R. Smith. “Introduction to Braille music transcription”. In: Division for the Blind and Physically Handicapped, Library of Congress. 1970.

[3] A. Karshmer, G. Gupta, and E. Pontelli. “Mathematics and accessibility: a survey”. In: *Proc. 9th International Conference on Computers Helping People with Special Needs*. Vol. 3118. 2007, pp. 664–669.

[4] J. Dixon. “Size and Spacing of Braille Characters”. In: (2010).

[5] J.D. Allen et al. *The Unicode Standard*. Addison-Wesley, 2007.

[6] A. Karshmer and C. Bledsoe. “Access to mathematics by blind students”. In: *Computers Helping People with Special Needs* (2002), pp. 111–124.

[7] C. Simpson. *The Rules of Unified English Braille*. Round Table on Information Access for People with Print Disabilities, 2010.

[8] S. Duffy. “Grade III braille compared to other abbreviated forms of braille”. In: *Future Reflections* 9.2 (1990),

[9] G. Cooper. “Research into cognitive load theory and instructional design at UNSW”. In: *Retrieved October* 22 (1998), p. 2003.

[10] Fred Paas, Alexander Renkl, and John Sweller. “Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture”. In: *Instructional science* 32.1 (2004), pp. 1–8.

[11] R. Gouzman and A. Kozulin. “Enhancing cognitive skills in blind learners”. In: *The Educator* (2000), pp. 20–29.

[12] K. VanLehn. “Cognitive skill acquisition”. In: *Annual review of psychology* 47.1 (1996), pp. 513–539.

[13] D.H. Warren. *Blindness and children: An individual differences approach*. Cambridge University Press, 1994.

[14] R. Helwig et al. “Reading as an access to mathematics problem solving on multiple-choice tests for sixth-grade students”. In: *The Journal of Educational Research* 93.2 (1999), pp. 113–125.

[15] G.E. Legge, C.M. Madison, and J.S. Mansfield. "Measuring Braille reading speed with the MNREAD test". In: *Visual Impairment Research* 1.3 (1999), pp. 131–145.

[16] A. Ahlberg and E. Csocsn. "How children who are blind experience numbers." In: *Journal of Visual Impairment & Blindness* (1999).

[17] D. Gillan et al. "Cognitive analysis of equation reading: Application to the development of the math genie". In: *Computers Helping People with Special Needs* (2004), pp. 628–628.

[18] Beate Hermelin and N O'CONNOR. "Spatial coding in normal, autistic and blind children". In: *Perceptual and motor skills* 33.1 (1971), pp. 127–132.

[19] M. Whapples. "Obtaining Braille mathematical documents". In: *MSOR Connections* 7.3 (2007), pp. 18–21.

[20] T. Crammer et al. *The Second Debate on Numbers*. 1993. URL: `http://www.iceb.org/ueb_num_debate2.html`.

[21] C. Navy. "The History of the Nemeth Code: An Interview with Dr. Abraham Nemeth". In: *Future Reflections* (2009).

[22] S. Amato. "Standards for competence in braille literacy skills in teacher preparation programs". In: *Journal of Visual Impairment & Blindness (JVIB)* 96.03 (2002).

[23] S. Amato and L.P. Rosenblum. "Preparation in and use of the Nemeth braille code for mathematics by teachers of students with visual impairments". In: *Journal of Visual Impairment & Blindness (JVIB)* 98.08 (2004).

[24] W. Jolley. "Unified English Braille: A Literacy Bedrock in the Digital Age". In: *twelfth ICEVI world coference, Kuala Lumpur* (2006).

[25] H. Cryer and S. Home. "User research ito the Unified English Braille (UEB in the UK". In: ().

[26] J. Engelen and B. Simons. "Wiskunde in Braille: de stand van zaken in Vlaanderen". In: *Infovisie MagaZIEN* (2011).

[27] R. Vink. "Braillestandaard voor algemeen gebruik in het Nederlandse taalgebied". In: (2005).

[28] G. Dam. *Profielen in de bovenbouw havo-vwo*. 2011.

[29] H. W. Laan. *Syllabus centraal examen 2012*. 2010.

[30] H. W. Laan. *Syllabus centraal examen 2012*. 2010.

[31] A. Leendert. "Het juiste gevoel". In: *Nieuwe Wiskrant* (2008).

[32] A. van Streun. *Handboek vakdidactiek wiskunde*. Universiteit Utrecht, 2010.

[33] D.W. Smith and S.M. Smothers. "The Role and Characteristics of Tactile Graphics in Secondary Mathematics and Science Textbooks in Braille". In: *Journal of Visual Impairment & Blindness (JVIB)* 106.09 (2012).

[34] J. Gardner. “Access by blind students and professionals to mainstream math and science”. In: *Computers Helping People with Special Needs* (2002), pp. 207–212.

[35] Chandrika Jayant. *A survey of math accessibility for blind persons and an investigation on text/math separation*. 2006.

[36] V. Lvesque et al. “Display of virtual braille dots by lateral skin deformation: feasibility study”. In: *ACM Transactions on Applied Perception (TAP)* 2.2 (2005), pp. 132–149.

[37] Gerhard Weber et al. “Banning a Chimera with the Hyperbraille Display”. In: (2011).

[38] S. Morley. “Digital talking books on a PC: a usability evaluation of the prototype DAISY playback software”. In: *Proceedings of the third international ACM conference on Assistive technologies*. ACM. 1998, pp. 157–164.

[39] T. Dick and E. Kubiak. “Issues and aids for teaching mathematics to the blind”. In: *The Mathematics Teacher* (1997), pp. 344–349.

[40] A. Piernik. *Accessible Calculators*. 2007. URL: `http://www.ucdenver.edu/academics/colleges/medicalschool/programs/atp/Documents/Accessible%20Calculators.pdf`.

[41] J.L. Toennies et al. “Toward haptic/aural touchscreen display of graphical mathematics for the education of blind students”. In: *World Haptics Conference (WHC), 2011 IEEE*. IEEE. 2011, pp. 373–378.

[42] M.M. Carneiro and L. Velho. “Assistive Interfaces for the Visually Impaired Using Force Feedback Devices and Distance Transforms”. In: *Info. Tech. and Disab. E J* (2004).

[43] N. Soiffer. “MathPlayer: web-based math accessibility”. In: *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. ACM. 2005, pp. 204–205.

[44] S. Maddox. “Mathematical equations in Braille”. In: *MSOR Connections* 7.2 (2007), pp. 45–48.

[45] TV Raman. *Audio system for technical readings*. Vol. 1410. Springer, 1998.

[46] M. Suzuki et al. “An integrated ocr software for mathematical documents and its output with accessibility”. In: *Computers Helping People with Special Needs* (2004), pp. 624–624.

[47] D. Archambault. *Braille Mathematical Notations*. 2007. URL: `http://www.access2science.com/braille/chezdom.net.htm`.

[48] H. Ferreira and D. Freitas. “Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project”. In: *Computers Helping People with Special Needs* (2004), pp. 627–627.

[49] Toshihiro Kanahori and Masakazu Suzuki. “Scientific PDF document reader with simple interface for visually impaired people”. In: *Computers Helping People with Special Needs* (2006), pp. 48–52.

[50] Cary Supalo et al. “Talking tools to assist students who are blind in laboratory courses”. In: *Journal of Science Education for Students with Disabilities* 12.1 (2008).

[51] Robert Shelton. “Influence of Fundamental Waveform on Perceived Volume and Pitch in Audible Graphs”. In: *Proceedings of ICAD*, pp. 6–9.

[52] F. Allemeersch et al. *Handleiding Vlaamse Wiskundecode en Wiskunde braille 2012*. 2012.

[53] a subgroup of the UEB Rules Committee within the International Council in English Braille Maths Focus Group. *Guidelines for Technical Material*. 2008. URL: http://www.brailleauthority.org/ueb/Guidelines_for_Technical_Material_2008-10.pdf.

[54] J. A. Sticken and J. M. Hart. *The Nemeth Code Tutorial for the Braille Lite Print Version*. 2005.

[55] G. Notaert, M. Suy, and E. Vandekerkhove. *Handleiding braille symbolen wiskunde*. 1984.

[56] Wolters-Noordhoff. “Formulekaart VWO A”. Attachment of textbook ”Moderne Wiskunde”, 8th edition.

[57] Wolters-Noordhoff. “Formulekaart VWO B”. Attachment of textbook ”Moderne Wiskunde”, 8th edition.