Towards Accessible Technical Documents: Production of Speech and Braille Output from Formatted Documents

A THESIS SUBMITTED FOR THE DEGREE OF PHD

Donal Fitzpatrick BSc
School of Computer Applications
Dublin City University

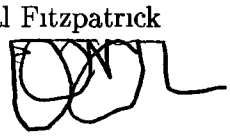
September, 1999

Supervisor Dr Alex Monaghan

This thesis is based on the candidate's own work, and has not previously been submitted for a degree at any academic institution

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Donal Fitzpatrick



September 21, 1999

Abstract

The primary objective of this research, was to devise methods for communicating highly technical material to blind people, through the medium of Braille or prosodically enhanced spoken output This solution necessitated devising strategies to both model the document internally, and to unambiguously produce the material in the two output media

The first phase in the generation of intelligible output was the transformation of the LaTeX source into well-formatted and accurate Braille Following on from this, methodologies were defined to convey the structure and textual content of documents using prosodic alterations to the synthetic voice We have devised mechamsms whereby mathematical content can be delivered in an intuitive manner, using the sole medium of prosodically enhanced spoken output This ensures that the listener will not have to learn specific non-speech auditory sound to gain access to this form of presentation

We have also devised a newer, and more flexible means for representing the structure and content of the document in the computer's memory This Directed Graph, is a radical departure from the traditional, tree-based approach of the past, and facilitates rapid and efficient browsing of the document's hierarchy

This thesis discusses the various aspects of the TechRead system, which will ultimately provide increased accessibility for blind people to technical documents We demonstrate how the methods used in TechRead differ from those previously employed to solve this problem

Acknowledgements

Though this work bears the name of a single author, I wish to acknowledge all those whose contributions made it possible First and foremost, I wish to sincerely thank my supervisor, Dr Alex Monaghan, who had to contend with

my crazy ideas, and missed deadlines Most of the good ideas expressed in this work are his, and he should not be held responsible for the remainder of the junk! More importantly, I wish to express my gratitude to Dr Monaghan, for access to one of the finest CD collections I have ever seen

I wish to express my gratitude to the two unfortunate people who had the tedious job of correcting this thesis. To Dr Lynn Killen, and Dr Serge Santi, all that I can say is thank you for both your thoroughness, and your especial fairness I look forward to working with both of you in the future, and research, and Brailling projects, academic or otherwise

I would like to thank the staff, and postgraduate students of the School of Computer Applications, at Dublin City University for their help over the last number of years Many of those still residing in the Postgraduate laboratory, will no doubt be relieved to know that they will not have to act as a *screenreader* when my computer crashes in future

To my parents and sister, I owe a debt which will probably never be repaid During my incarceration in special education, my family were always supportive, and never ceased to struggle in their attempt to ensure that I obtained the best education possible This thesis, finally marks the end, and indeed the vindication of their battle against the system Therefore, Mum, Dad and Niamh all that can be said is thanks

As all those who know me will agree, this thesis would not be in its present form without the inestimable contribution of Ms Orla Duffner, B Eng Ms Duffner proofread this thesis, and made all those visual alterations which ensured that I wouldn't be embarrassed when people looked at itt During the last, and most difficult few weeks, she bullied, persuaded and threatened this thesis into existence, when I would have preferred to be anywhere else So, O J , for the thesis, the music, and a list of other things too numerous to mention, thank you

I must conclude, by acknowledging the invaluable assistance of those mem-

bers of the DCU Folk Group, (or whatever its' name is this week) You are all too numberous to mention, but all that can be said is thanks for being in DCU at the same time as me In particular, however, I will single out two people for special mention Fr John Gilligan, (even though he's an Everton fan) and Sr Bernadette McDonagh were always around when I needed tea, beer or someone to give grief to To both of you, I wish to say a sincere "thank you"

Donal Fitzpatrick
September 21, 1999

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter is intended to outline the basic ideas underlying the design of the TechRead system It points out the origins and primary aims and considerations of the research This chapter also gives a brief overview of some of the key concepts used within the system, including a description of Braille, and an explanation of the **basic** notions associated with prosody

1.1 Principal objectives

This work was born out of the frustrations of many blind people in Ireland who, despite one of the best overall education systems in the world, could not gain access to material pertinent to higher mathematics It evolved, if such things can be said to evolve, from the notion that though Braille is in essence a primitive system it is the one most commonly in use by blind people throughout the world

Since its invention in the middle of the 19th century, Braille has been used to convey such diverse subjects as music and mathematics to blind people Its popularity has been mainly derived from its simplicity, and its ease of production Braille can be easily adapted to render almost anything which is available

m prınted form Thıs ıs shown by the fact that such languages as Japanese are produced ın Braılle Instead of attemptıng to represent each character ın Braılle a syllabıc character set ıs used Thus the word "Tokyo" would be produced ın two syllabıc Braılle characters However, there are many dısadvantages to thıs system Its ınnate sımplıcity does not lead to easy representatıon of graphıcal or mathematıcal materıal The tradıtıonal method of representıng mathematıcal materıal has been to produce ıt ın a lınear fashıon, akın to any other textual data Consequently, such materıal has proved dıfficult to read Another dısadvantage ınherent ın the Braılle system ıs that ıs not conducıve to the productıon of modern typesettıng By ıts very nature (ıt ıs, after all a dot-based system) ıt ıs not possıble to alter the sıze of the font ın Braılle The finger ıs not as sensıtıve as the human Eye, and therefore the ıncrease ın the sıze of the dots produced would have to be enormous to regıster By the same argument, ıt has proved very dıfficult to ıncorporate such features as emphasıs ınto Braılle documents In order to overcome some of the deficıencıes ın the Braılle system we decıded to harness the advances ın exıstıng technology such as synthetıc speech

Sınce the emergence of the personal computer ın the last decade, many companıes and research ınstıtutıons have ınvested countless man-hours and vast amounts of money ınto the development and productıon of both hardware and software to enable blınd people to utılıse the full benefits of computers Such technology ıncludes

Screenreader Software desıgned to convey anythıng appearıng on screen to the blınd computer user

Braille Translation software Software desıgned to take files of varıous ınput formats and to derıve Braılle output from them

Braille embosser A pıece of hardware used prımarıly by Braılle translatıon software Thıs ıs sımply a Braılle prınter, whıch embosses a hard copy of the Braılle generated by the translatıon software

Refreshable Braille display A piece of hardware consisting of a single line of Braille cells The cells comprise solenoids, which can be raised or not to produce the dot patterns required for Braille (See Section 1 2 for a description of Braille)

Speech Synthesiser A piece of hardware or software which artificially produces speech

Nearly all of the effort to date has been directed toward material of a literary[1] nature This, while being an essential development, still rendered material of a technical or scientific nature almost totally inaccessible to those who could not see it

A solution of sorts presented itself in the form of obtaining human readers to record technical material onto audio cassettes However, this apparently acceptable means of conveying such information has many drawbacks In order to appreciate these, it is first necessary to examine how sighted people in general read

The reading process itself is an active-passive process, where the sighted person is the active participant while the book they are perusing is the passive partner This process is reversed when using audio recordings The only means the blind person has to become active in the reading process is to move forward or backwards in a time-line That is, the tape can be advanced or re-wound by the blind "reader" but no more Instead of being active in the reading process, the book now becomes the active participant, while the blind person simply allows the information to flow past them This can be very difficult when technical material is being read It is widely accepted that the use of human readers in the provision of literary works on tape is perfectly acceptable However, the manner in which people read scientific, or technical documents is vastly different from how novels are read

[1] In this thesis, the word "literary" is used to denote documents of a non-technical, or unscientific nature

When one is reading a novel, the book can be read in a far more superficial manner than when mathematical material is being perused For example the two equations $3(x+y)^2$ and $3(x+y^2)$ are vastly different As can be seen from a glance, though the same symbols have been used in both cases, the semantic interpretation of these equations yields vastly different results A human reader speaking both these equations would have to say something like

"3 left paren x plus y right paren squared"
"3 left paren x plus y squared right paren "

The spoken approximation of these two simple equations is quite complex For example, it could be difficult for the blind reader to remember the positioning of the parentheses, or the quantities contained therein Imagine therefore, the mathematical expression such as that for finding the roots of a quadratic equation This equation, $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ would prove more than a little difficult to convey to a blind user using speech To use the same spoken notation as our previous examples, it could be rendered thus

"minus b plus or minus the square root of left paren b squared minus 4ac right paren over 2a"

From this rendering, it is unclear where the numerator of the fraction starts or concludes, i e the blind reader would be unsure whether the "-b" is contained in the numerator This is extremely ambiguous and could lead to extreme confusion when attempting to solve quadratic functions using this method As can be seen from these simple examples of spoken mathematics the length of the utterance is not comparable to the length of the equation, as even the least complex of expressions produce lengthier verbal versions The reason for this is that the eye and ear absorb information in vastly different ways

It is possible for the eye to rapidly scan over the equation and to distinguish the various components of a given equation, while the ear absorbs data in a serial

fashion Though an equation may appear visually compact, it could prove to be extremely complex to render in an audio format A feature of listening to spoken text is that there are many occasions when only part of the utterance is memorised by the listener This is **not acceptable** when dealing with what may be complex information

When listening to non-technical material, it is usually enough for only part of an utterance to be heard, syllabic interpretation and indeed intuition can usually be relied upon to fill in the gaps In direct contrast to this, from the two equations above missing the words "left paren" could yield entirely the wrong meaning to the mathematical material When reading normal running text, it is possible for the human eye to scan ahead and to rapidly peruse the document

Syntactically complex information is totally different The two sample equations above serve to illustrate this point A simple glance will show that exactly the same symbols are used in the two expressions, though their position relative to each other yields vastly different results The reader must examine in detail what the symbols actually mean, which can involve breaking the expressions into sub-expressions, until a point is reached when the underlying meaning of the material can be deduced When using standard audio recordings this is not possible As was stated earlier the listener can only control the movement of the tape through a time-line This does not cater for the perusal of sub-expressions or for a simple "glance" at an equation, facilities which sighted people can call upon without a second thought

Coupled with this is the notion that blind readers must rely on their memory extensively As those familiar with mathematics will agree, the permanence of the image on a page means that the reader is not compelled to retain a mental picture of the data , but can instead devote such faculties to its understanding When listening to such information, one cannot readily access such a permanent picture As a consequence, most blind people read any scientific or other technical information using Braille

Any system which attempts to provide access to mathematical material must take into account the need to reduce complex information to manageable blocks, thereby easing the burden of retention We therefore believe that document browsers must provide a non-linear representation of technical material which allows the user to choose between detailed examination, overview mode, or even skipping the material altogether TechRead is designed to facilitate the non-linear representation of complex objects, and also to provide different "views" of an object as the user chooses

The primary objectives of TechRead can be summarised as follows

- to build an off-screen model of the document reflecting its structure and content
- to produce intelligent and well-formatted Braille from LaTeX [Lam85] documents
- to use the document model to generate prosodically enhanced spoken output
- to provide an interface to documents making the reading process easier

To begin with, the TechRead system has been designed with modularity and expansion uppermost in mind Accordingly, it has been decided to implement the system in discrete stages, namely an input (or pre-processing) stage, and two different phases to control the output media of the final accessible version of the document In order to achieve this, it was felt that an accurately generated model, reflecting the document's structure and content should be derived Consequently the system will produce a Directed Graph of the document This graph is then used to produce both the Braille and spoken output (For more information on this internal representation of the document, see Section 3 1)

The phrase "accurate and well-formatted Braille", as used in the preceding paragraphs needs some explanation It has been decided to produce the Braille

from the TechRead system in accordance with the rules as laid down by the Braille Authority of the United Kingdom (BAUK) As such, the guidelines specified by this organisation will be adhered to in the following areas

Braille signs In terms of textual transcription, the symbols used in both Britain and North America are consistent However, the symbols used for mathematical and scientific transcription differ immensely and have no similarities in either the dot pattern used, or their conceptual basis

Formatting and Layout The Braille produced by the TechRead system will be accurately positioned on the page One aspect which will be considered is an attempt to improve the readability of mathematics At present, the typical form of presentation suggests that mathematical material should be laid down on the page in the same linear fashion as textual material This is a complete contrast to the approach taken when writing printed mathematics, where it is usual to use vertical as well as horizontal alignment to denote the relationship between components of formulae We believe that incorporating a degree of vertical positioning in the Braille transcription of the mathematical material will assist the blind reader to gain a more rapid view of the information A greater degree of spatial orientation will be introduced to produce as close a replica of the printed mathematics in Braille as possible

Currently it is not possible to convey the changes in visual formatting which make existing printed documents both attractive and easy to read As a consequence the Braille reader is unable to discern whether a passage of text is more important than surrounding information or is part of the general body of the document (For a more in depth discussion of this portion of the system, see Chapter 4)

When using current forms of access technology for the purposes of reading, difficulty arises when emphasised or other visually enhanced material is encountered Authors use changes in the visual attributes of their documents

to convey material which in their opinion, is more significant or important than the majority of their work When reading with standard screenreaders using synthetic speech devices, such altered characteristics are not spoken as the developers of the screen access software have not harnessed the full capabilities of the speech synthesisers In essence, the text is read in a tedious monotone, with occasional pausing when simple punctuation is encountered in running text As must appear obvious therefore, the content of the document **appears** to be of exactly the same importance from the perspective of the blind reader For example the following two sentences would be read in exactly the same fashion

"Throw that ball to me" "Throw **that** ball to me"

It is visually apparent from these two sentences that the importance of the word "that" is different in both cases By use of a simple typesetting technique, the visual appearance, and hence the importance of a passage of text is highlighted Using current screen access technology this would not be apparent to the blind reader TechRead aims to solve this problem by introducing prosodic enhancements into the spoken version of the text The addition of pausing, rate, pitch changes and amplitude variations will enhance the spoken output, hence improving the quality of the speech for the user It must be said at this juncture that although these capabilities will exist within the system to enhance the text, the resulting output will only be as good as the speech synthesiser will allow Some devices are more flexible and programmable than others, and contain more commands to modify their output For example, when a passage of emboldened text is found, the speaking voice would alter to reflect the importance which the author attributes to this portion of text The nature of the prosodic enhancement would depend entirely upon the material being presented, that is, if a section heading were being spoken, though it looked the same as a portion of running text, it would of necessity be spoken differently to convey to the blind user that it was a section heading and not an emphasised passage in running text

The perceived need for a flexible interface to a document is one of the key features of the TechRead system A prime consideration in the design of such an interface is the fact that blind people, unlike their sighted colleagues, cannot "skim" a passage to pick out only those portions of text which the author has designated as being of more importance than the rest A simple example will suffice to illustrate this Suppose that a sighted person picks up a newspaper, and wishes to find the latest football results The first thing the reader will do is to turn to the relevant section of the publication Then, they can simply "scan" the page until the headline is found Headlines are easily distinguishable from the majority of text in newspapers by the fact that they are typeset in a larger, emboldened font, causing them to stand out from the surrounding text Blind people, as things currently stand do not possess this facility All text in a newspaper would be spoken in the same monotonous voice, thus causing them to resemble, rather than contrast with the text of the articles they are describing TechRead aims to make this process easier

The intention is to examine the mark-up of a document in order to determine the structure and formatting attributes associated with the content An interface can be constructed to enable the blind reader to rapidly, easily and flexibly navigate through the structure of the document For example, the user could go from section to section until the correct one was reached Once this has been achieved, the section could be expanded to obtain the names of all sub-sectional units contained with the section, and when the required one was chosen, the text could be read

The interface to the TechRead system is based primarily upon the numeric keypad of a standard IBM compatible computer In accordance with trends in the design of current access technology, this aspect of the system is developed with the primary intention of making the learning curve as shallow as possible Many developers in the past associated keyboard commands with concepts which, though logical in theory, proved intrinsically difficult for users to grasp Recently, the notion of locating the command-set in one particular area of the

keyboard has assisted in making the product far easier to use There are also advantages to this approach which extend outside the realms of the blind user Though the system is designed with such users primarily in mind, there are no reasons why TechRead could not be used by others for whom reading technical material is a problem Consequently, for people with limited mobility, the use of an interface which can be operated with only one hand (or one finger) could render the system not only accessible but most useful

1.2 Braille: origins and overview

Until recently, the only means of reading and writing for blind people was through the medium of Braille, a system based on six raised dots Latterly, through the emergence of such media as audio tape and increased access to computerised material, Braille has faded somewhat and the emphasis has moved away from its instruction, but it is still the most widely used form of literary communication for the blind

Braille was devised by Louis Braille He was born in 1809 in Coupvray, a small town near Paris The son of the local harness maker, his blindness was directly attributable to his father's profession, as he severely injured one of his eyes while playing with an awl Later, infection set in and as a consequence total blindness ensued At the age of 10, he left home to attend the Royal Institution for Blind Youth in Paris, having received a scholarship to do so Here, most of the education was oral, though a primitive form of embossed writing was used This consisted of each individual letter being shaped in copper wire, which was then embedded in the backs of sheets of paper by a specially designed press, thus making it impossible for blind pupils to reproduce such a writing scheme themselves

What is not commonly known, is that Louis Braille did not actually invent the system which now bears his name Rather it evolved from a method of writing devised for the army Captain Charles Barbier de la Serre had invented

a means which, he believed would aid soldiers to pass messages along trenches quietly at night This system, known as "sonography" or "night-writing" was a clumsy twelve dot system, in which the dots could be combined to form dot patterns reflecting the sounds of the words, rather than the spelling However, as the system was incredibly complex and was not easily understood by the soldiers, it was rejected by the army Louis Braille, studied this method of reading and writing and soon realised the implications for blind people He spent the next several years simplifying the twelve dot cell to the six dot version now in common usage, until in 1827 the first Braille book was published

A Braille character or "cell" consists of six dot positions arranged in a rectangle comprising three rows and two columns The dots at any of these positions can be either raised or not, thus giving 2^6 total possible patterns For ease of reference, the dots are universally numbered 1-3 (from top to bottom on the left) and 4-6 (from top to bottom on the right)

As can be observed, the limitations on the number of possible combinations by its very nature implies that Braille is **highly** context dependent For example, let us assume that the dots 1, 3 and 4 were raised (i e the top and bottom dots of the lefthand column, and the top of the righthand one) This, in accordance with the original assignments of Louis Braille would be interpreted by a reader of English language material as the letter "m", whereas if the text were in Greek, the same pattern of dots would represent the symbol "μ" It is important to point out, that it is not considered relevant that the Braille symbol be similar in shape to its printed equivalent Instead, the arrangement of the dots follows a specific arrangement, the logic of which can be seen from Appendix A

The 64 possible Braille characters are insufficient to cater for the myriad print characters and their variants To surmount this, a shorthand, or "contracted" form of the Braille system was devised The notion of using a single Braille symbol to denote multi-character sequences is the essence of this contracted Braille There were several varied reasons why this form of representa-

tion was thought to be necessary Firstly, the introduction of a single symbol to indicate a multi-character sequence would assist in reducing the reading-time of documents Since the finger cannot scan as rapidly over the material as the eye, the time taken to read a page of Braille is slower than to peruse the comparable amount of material in print Also, the use of short-hand representations of commonly used words and character sequences ensures that the size of Braille books is reduced Unlike the printed representation, the Braille page can only accommodate between 25 and 28 lines, each comprising 42 characters, yielding a maximum of 1176 characters This compares with the printed page, which can contain up to 3500 characters, on paper which is smaller than that used for Braille Also, the fact that Braille must occupy three dimensions (that is, the dots themselves add height to each page) the Braille output is bulkier than the printed equivalnt Indeed, the paper itself adds its own extra thickness to documents, as in order to hold the dots, it must be card-like paper [2] As a consequence of this extra thickness, the Braille version of a book can occupy many volumes For example, a printed desk-dictionary could occupy an entire book-case when reproduced in Braille

The shorthand version of a word is most often achieved by using various Braille characters as "prefix" characters, which change the meaning of subsequent cells For example, a specific character, when placed prior to the letters "a-j" transforms their meaning into the Arabic numbers "0-9" Other forms of shorthand are used in this system, which is also known as "Grade II" Braille Thus a single character is used to represent the definite article, while combinations of characters are used to provide a means to compactly express common letter groupings Unlike print fonts, the Braille cell has a uniform height and width, meaning that such visual formatting as increased font size is unavailable to the Braille transcriber Also, such textual enhancement as emphasis must be conveyed to the Braille reader via similar indicators to those described above for the production of digits The use of centering in Braille is commonplace, and is applied in much the same manner and for the same reasons as in printed

[2] typically, paper of between 110G–130G in weight is used

material

The discussion until this point has been concerned with the representation of natural languages in Braille However, other systems also have their Braille equivalents Among these are included Mathematics, music, scientific notation and chess The key to these Braille systems is an association between the same 64 Braille characters, (or sequences thereof) and the symbols and other notational elements of interest However, it should be emphasised that all of these systems reproduce the same linear-based Braille, thus making reading and writing of complex mathematical or technical material both time-consuming and difficult

1.3 What is prosody

The prosodic component of speech is that set of features which lasts longer than a single speech sound The term *prosody* can be traced back to ancient Greek where it was used to "refer to features of speech which were not indicated in orthography, specifically to the tone or melodic accent which characterised full words in ancient Greek" [CK86] When the tonal units of ancient Greek disappeared, the use of prosody narrowed to refer to stress distinctions By the 15th century it became known as versification, one of its primary denotations today As the influence of the classical languages waned so also did the emphasis shift from a metric to a melodic view of prosody, though some scholars believed that the English language had no melody However, the term *melodic prosody* remained almost forgotten until the 1940s, when it was revived as an approach to the study of linguistic analysis Prosody is therefore defined as " those auditory components of an utterance which remain, once segmental as well as non-linguistic as well as paralinguistic vocal effects have been removed" or "sets of mutually defining phonological features which have an essentially variable relationship to the words selected" [CK86] Therefore, it may be said that prosody contains the following aspects of speech

- Loudness
- Duration
- Pitch
- Pausing

Acoustically, *speech* can be decomposed into three primary components, frequency, amplitude and time "Frequency is the term used to describe the vibration of air molecules caused by a vibrating object, which are set in motion by an egressive flow of air during phonatation" [CK86] The unit of measure used in the frequency domain is the Hertz (Hz) Speech is not as simple as other acoustic sounds, as it can contain many elements vibrating at different frequencies The frequency of repetition of the complex pattern is referred to as the *fundamental frequency*, and it is this frequency which is primarily responsible for the perception of pitch All other frequencies in any pattern are typically whole integer multiples of the fundamental frequency, and are known as the second, third, etc harmonics

Amplitude, (measured in decibels) is the acoustic component which gives the perception of loudness A common definition is "the maximal displacement of a particle from its place of rest" [CK86] The duration of a signal is the third component in the acoustic view This is simply the measurement, along the time-line of the speech signal If one considers the prosodic component of speech, then it can be reduced to a series of frequencies, a succession of intensity levels and a sequence of durations It is these components which yield our understanding of pitch modulation, relative loudness and/or the relative duration of syllables, words or phrases

The notion of the syllable is intrinsic to the understanding of prosody, however it is easier to state what the syllable is not, rather than what it is Native speakers of a language can always agree on the number of syllables any word contains, although non-native speakers can often be confused For the pur-

poses of this discussion, a syllable is defined (unscientifically) as a small unit into which the sounds of words are decomposed

The use of *stress* is a key prosodic feature of the English language What is perceived to be stress, is the increased prominence which is often added to a word or syllable This increased prominence seems to confer a degree of emphasis on the utterance, and is akin to the addition of visual emphasis in the written material Stress has been given a number of different definitions Some definitions relate stress to the force with which a speaker utters certain syllables, thereby relating it to a greater articulatory effort Other definitions equate the addition of stress to the perceptible loudness observed by the listener, "loudness being a perceptual dimension" [CK86]

There are two basic acoustic characteristics associated with *loudness* Firstly, it is dependent on an increased amplitude, signifying that more energy reaches the ear per unit of time Coupled with this is the increase in the rate of vibration of the vocal folds, implying that more pulses reach the ear per unit of time "In English at least, duration and pitch are also involved in perception of prominence" [CK86] There are at least three primary acoustic cues associated with a stressed syllable Higher intensity, greater duration and higher fundamental frequency are all perceptible However, these characteristics are also used for other purposes such as to convey emotions Moreover, the primary factors associated with stress are conditioned by the presence of surrounding consonants, the position of the stressed syllable in the phrase or sentence, and the rate of speech Consequently, stress cannot be identified acoustically in an unequivocal manner it must be judged relative to the context

If at least two, or possibly three primary components of stress can be identified, the question which arises is the means of their interaction, thereby producing the auditory impression of stress It is believed in some quarters that intensity plays no part in the perception of stress However, an alternative view is that it " is at least sufficient to cue a stress judgement although it may not be necessary" [CK86] The difference in stress between the two syllables of

insult is based in part on a perceived difference in the syllable length Kuhlen tells us that this perception can be expressed in musical terms, using the grouping of a quaver followed by a dotted-crotchet to represent the two syllables of the verb *inSULT* However, there is no equivalent difference in length between the two syllables of the corresponding noun *INsult*, which can be expressed as two crotchet beats Kuhlen states that "what seems to be more important is a distinctive rhythm pattern" [CK86, p22] Distinctive pitch patterns are also observable in stress To re-visit the example of *insult*, when the word is used as a noun, two differing static pitches can often be heard On the first syllable, a higher pitch is used, descending to a lower frequency for the second In its verbal usage, the first syllable is often observed to have a static pitch, after which a gliding pattern can be seen which descends gently

It would be wrong to assume that only pitch height is responsible for stress perception It has been shown that a sudden step downward in pitch also assists in its recognition Kuhlen says that Bollinger was convinced that pitch obtrusion was "a rapid and relatively wide departure from a smooth or undulating contour", which is responsible for the perception of prominence [CK86, p24] However, experimentation has shown that there is an observable hierarchy to the contributors to the perception of stress in English fundamental frequency, duration, intensity The fact that stress is still perceptible when fundamental frequency is absent (i e in a whispered conversation) would seem to confirm that other mechamsms are in operation

An important feature in the understanding of stress is an examination of the principles which determine the location of stress in the utterance One of the oldest theories of stress location is provided by Jones (1918) Couper-Kuhlen cites Jones as saying "as a general rule it may be said that the relative stress of the words in a sequence depends on their relative importance The more important a word is, the stronger it is stressed" [CK86, p35] Other theories however, associated importance with word categories One theory states that content words (such as nouns, main verbs, adjectives and pronouns) carry the

major semantic weight of the sentence, while function words (such as auxiliary verbs, prepositions and adverbs) do not This results in the former being stressed and the latter not However the principle of associating the stress of the utterance with word categories alone, cannot account for the location of all rhythmic stress beats in an English utterance Aspects of the rhythm, related to those of tempo, interact with the word-category membership to determine the location of the stresses within an utterance [CK86, p37] This is a very complex and difficult area of prosody, and TechRead does not attempt to tackle it, but leaves stress placement to the synthesis device (For an account of stress placement and prosody in speech synthesis, see [Mon91, Mon93, Mon99])

Stresses alternate regularly in English utterances Consequently, people will adjust the stress patterns which they impose on their speech to reflect this For example, if a sequence of three monosyllabic stresses occur in an utterance, then the middle stress will be reduced [Mon90] The rhythmic principal of alteration however, is intimately connected with considerations of speech tempo Therefore, several stressed syllables can be articulated successively, through the introduction of pseudo-pausing, or through the lengthening of some syllables By contrast, stress beats need not be added in a sequence of unstressed syllables if the rate of delivery is appropriately fast

Another aspect of the location of stresses is the speakers' own wish to utilise them to add alternate semantic meanings to what they are saying There are several possible, though equally correct stress patterns permissible in certain English utterances However, the emphasis of one particular portion over another is used to convey to the listener that the speaker wishes to impart greater importance to that particular portion of the utterance TechRead uses this fact to convey visual emphasis in document formatting (see Chapter 5)

One of the significant features of speech (used extensively in TechRead) is the rhythmic component There are two contrasting views of what rhythm actually means One view states that rhythm is the reoccurrence of an event at regular intervals of time, while another states that rhythm comprises a pattern

of events related to one another in terms of salience The former is known as the temporal view of rhythm, while the latter is a non-temporal view

In the temporal view of rhythm, the burst of gunfire from an automatic weapon, or the regular dripping of a water tap are rhythmic, as they constitute a reoccurrence of the same or a similar event at equal intervals in time Supporters of the non-temporal approach to rhythm would, on the other hand deny that the sound of water dripping or a burst of automatic gunfire is inherently rhythmic [CK86, p52] Rather they claim that rhythm is perceived in the mind They state that rhythm is perceived as an amalgum of sensory perceptions, rather than as a set of unrelated events This view of rhythm groups events according to salience, some having greater prominence than others In this less strictly temporal view, speech is said to demonstrate various types of rhythm, rhythm based on duration being just one form [CK86]

Patterns can also arise, in which other aspects of the audio spectrum can play a role The basic unit of rhythm used in English speech is the *foot* Abercrombie defined a foot as "the space in time from the incidence of one stress pulse up to, but not including, the next stress pulse" [CK86, p55] An example which illustrates the use of feet, is the first line from the well known nursery rhyme which states "This is the house that Jack built" There are four stresses in this utterance, and hence four feet The decomposition of this utterance into stressed and unstressed syllables yields the pattern "THIS is the HOUSE that JACK BUILT" As can be seen, the feet are different lengths, the first foot comprising three syllables, ("THIS is the"), while the third merely consists of the monosyllabic "JACK" Observation of the utterance, will also reveal that (with the exception of the penultimate foot) stressed syllables are followed by unstressed ones This causes a rhythmical pattern to emerge, and indeed if we consider the only instance where two stressed syllables follow one another, there is a slight pause between them, to maintain the rhythm pattern of the utterance

Silent syllables can also be included in feet, as is illustrated by lines within a Limerick. It should be noticed that since the duration of each foot is roughly equivalent, the syllables contained in them are often stretched or compressed to maintain the rhythmic metre of the phrase.

A feature of English prosody of extreme importance to TechRead is that of *intonation*. There are several definitions of this phenomenon:

1. the suprasegmental level of pitch, stress and pausing
2. the non-lexical manifestations of melody in speech
3. the gradient contrast due to pitch

From an auditory perspective, intonation is intrinsically related to pitch. It should be remembered however, that a perception of a pitch change does not occur at every alteration in the fundamental frequency. "The extent of fundamental frequency changes as such is no reliable measure for the perceptual relevance"[CK86, p63]. At an acoustic level, intonation can be seen as a succession of fundamental frequency curves in time. Alternatively, on a phonetic level, intonation can be viewed as a succession of perceivable pitch events. At a phonological level, pitch events are grouped together into meaningful categories. In the British tradition, the analysis of intonation related the pitch and stress of an utterance. Only the pitch of the stressed syllable is considered relevant for the characterisation of intonation patterns. The alternative approach is known as tune analysis. This form of analysis reduces the intonational system of a language to "a small set of holistic contours or tunes, with variations allowed for special circumstances" [CK86]. Variations for emphatic sentences included increased stress as well as widened or narrowed, raised or lowered pitch range. The tune approach to analysis imposes a rigid frame on the melodic line.

A third approach to the analysis of intonation is to use a so-called "features" approach. Several prosodic features are used in this model: tone, pitch range,

loudness, rhythmicality and pausing This leads to a discussion of intonation in its broadest setting

In the prosodic model used in the TechRead system, intonational features are used to convey structural and visual aspects of any document The TechRead model is based on the phonetic model developed by Ladd and Monaghan [LA87, MR91] and the phonological model outlined in [Mon91] For further discussion of the role of prosody in conveying information, see [Lad96, CK86]

1.4 Summary

This chapter has described the principle concepts and objectives of the TechRead system The fundamental concepts on which the research is based were introduced, followed by an illustration of the Braille character encoding The chapter concluded with a description of the fundamentals of prosody

The remainder of this document builds on what has been begun here Chapter 2 gives some information pertinent to previous research in the area of accessibility for the blind to technical documents, while Chapter 3 begins the discussion of TechRead itself by defining both the internal model and human interface for the system Chapters 4 and 5 describe the two output media used in the system, discussing the methods whereby meaningful output is obtained for each Chapter 6 outlines the implementation of the system, and the means employed to evaluate the prosodic component of the mathematical output The description of the research is brought to a close in Chapter 7 by some concluding remarks, and a brief discussion on the future prospects for the system

Chapter 2

Literature review

This chapter gives an overview of both current and previous research on which the TechRead system is based It outlines the underlying concepts on which previous systems have been designed, and introduces other ideas which will aid in the understanding of the subsequent chapters It is not intended as a complete reference on the means of producing more accessible technical documents, rather it is hoped that it will give useful background information

The chapter begins with a description of the mechanical actions needed to peruse documents from both the audio and visual perspectives Following from this, a description of the more traditional models used to encapsulate the document's content and structural information is presented Included in this Section is a discussion of the model used in ASTER [Ram94] which was among the first attempts to solve the problems relating to the accessibility of technical documents to blind people From this point the chapter discusses the means used in various systems to allow the user to browse technical documents using audio presentations, and concludes with the means employed by these systems to actually present the various types of content to the blind users

2.1 How do we read?

One of the key decisions which must be made when considering the design of a "document browser" is the means of presenting the information to the target user In order to implement this ideal, the notion of what information to present needs to be decided on first, followed by how to present this material It is therefore important to understand the reading process, in order to fulfill the dual purpose of determining both **what** and **how** to present the relevant information to the user

2.1.1 Audio or visual reading

A good starting point for the discussion on the methods of reading is the similarities and differences between visual and audio based reading It is these differences which will form the core of the design of both the interface, and the means of presenting the spoken and Braille output which forms the core of the TechRead system

A key feature which is present in the visual reading process is the role of the printed page This medium affords the reader not only the facility to act as an external memory, but also facilitates a highly refined control over the flow of information In his Ph D thesis, Stevens states that Raynor [RP89] describes reading as ' the ability to extract visual information from the page and comprehend the meaning of the text' [RP89, p23] Stevens [Ste96, ch2] also tells us that reading can be divided into three main domains

1 The input of information from a physical, external source, into the reader's memory via the visual system,

2 The recognition of words and their integration into higher level structures such as sentences,

3 The process of understanding what has been read

It would appear that there exists a point at which the process of listening and reading converge This would seem to indicate that, once the information has been absorbed by the reader, it is both syntactically and semantically decomposed in the same manner, though the processes of actually retaining the material are quite different depending on which means the reader uses to read It would appear that many readers hear a voice inside their head whilst reading This voice articulates what is being read, giving the reader both a phonological and sub-localised impression of the document It is the former impression which is of paramount importance to the TechRead system, and hence the discussion in the subsequent paragraphs will focus on this aspect

Stevens [Ste96] defines the *phonological code* as "the auditory image kept in working memory during reading" It can be said that the written text is converted to this phonological code, which contains all the features of natural speech, such as pitch, rhythm etc The notion of *inner speech* is quite speculative, but Rayner states that "Some proponents of inner speech have argued that reading is little more than speech made visible" [RP89, p190] The above appears to suggest that the visual component of reading is converted to an audio version, seeming to suggest a point where the two converge After this point, the comprehension of the information should be the same It is clear that the only differences in the reading process are the mechanical means of obtaining the information

One aspect in which listening and reading differ significantly is the role of paper as an external memory The manner in which the eye can relate to this external memory is a very powerful tool to aid in the retention and comprehension of written information It can rapidly scan over the printed words, and by virtue of the juxtaposition of characters or symbols on the printed page, semantically interpret those symbols to produce the underlying meaning Once the information is lost from the short term memory, it can be easily refreshed by the rapid movements of the eye

There are a number of steps involved in the visual reading of a document A

skilled reader will normally read at a rate of 250–300 words per minute The eye does not actually start at the top of a page and continue in a linear fashion until the end of the material is reached, rather the reading process consists of several distinct movements Stevens tells us that there are a number of tasks which the eye performs in order to gain informational input The reading process can be broken down into a series of **sacades** (jumps) and **fixations** He tells us that

> "The saccades move the point of fixation in accordance with how much information has been or can be apprehended Forty nine percent of reading time is taken up with fixations The rest of the time is taken up with the selection of which portion of the text to next fixate and the move to that location "

[Ste96]

Rayner[RP89] points out that as the informational density increases, the eye needs a series of regression steps to absorb the material Thus (as with spoken input) shorter, more commonly used words are not fixated, but are comprehended The following is an extract from Stevens thesis, and consists of a table produced by Rayner [RP89] which shows the decrease in the reading rate when informationally dense, or complex material is found As can be seen, the reading rate slows as the more complex (difficult) mathematical information is encountered

It is the absence of this external memory that is so different when the information is being read using audio The facility of the paper as an external memory source is not present, and as the speech signal is transitory, the listener cannot easily recapitulate over already read material Further, a vastly increased load is placed on the short term memory, thereby detracting from the ability to comprehend the material which could otherwise be easily understood

An important distinction which must be made between reading and listening is the method by which the information is assimilated by the reader When

Topic	Fixation (ms)	Saccade(mm)	Regressions	WPM
Light fiction	202	9.2	3	365
Mathematics	254	7.3	18	243

Table 2.1: Comparison of eye movements during the reading of light fiction and mathematical text. Saccade length is measured in character spaces. Regressions are measured as the percentage of fixations that were regressions. Extracted from [RP89].

listening, the speech signal flows past the reader [Ram94, Ste96]. The process can therefore be described as a serial exchange of information from the audio device to the passive listener. Visual reading, on the other hand involves rapid movements of the eye over a visual stimulus, and the acquisition of information through the visual cues which are inherently part of any document. Even the most rudimentarily written document consists of various forms of visual cue.

Firstly, the use of white space can determine the ends of both sentences, and other sectional units (such as paragraphs). This use of visual enhancement becomes more pronounced as the complexity of the material being presented increases. Finally, when a point is reached when mathematical (or other such technical information) is included, the use of the innate visual cues becomes more and more important to aid the visual reader in distinguishing the diverse types of content from the main body of the material. Consequently, material which is important can be easily recognised by virtue of the visual characteristics which have been imparted to it. The listener, on the other hand is reliant on the prosodic features which form a part of all spoken output.

Another distinction is that, unlike the visual processes described above which rely on speed and accuracy to control the input of information, the control over the speech signal is crude and inaccurate at best. Though it could be said that the speech signal on an audio cassette or a computer display forms a permanent record of the material, the control over this data is minimal. For example, the only control over an audio cassette is to advance or rewind the

recording through time In order to locate a specific passage both time and intense effort must be employed to locate it

Some organisations who record technical material specifically for use by blind people have introduced the technique of recording tones at various key points within the document This "tone indexing" involves the use of three tones to denote the start of a new chapter, two to denote the beginning of a sectional unit within that chapter, and one to indicate the transitions between pages These tones are audible on cassette players which have a "cue" or "review" facility Though this enhancement is effective in speeding up the location process, it does not enhance the control the listener has over this essentially crude form of audio presentation

Consequently, though the experienced blind user can listen to a speech synthesiser at up to 400 words per minute, this must compare with a speed reader, who can peruse material at thousands of words per minute It is this lack of close control over the information flow which reverses the normal active/passive nature of reading As was stated previously, the temporal nature of the speech signal means that the information flows past the passive reader, which is in direct contrast to the visual process

The absence of the external memory (paper) also implies that an increased mental work load is involved in the retention and comprehension of audio-based mathematical information Whereas the reader can use the printed page as the external memory and an aid to retention, the listener has only their memory of the spoken utterance, thus making the comprehension of syntactically rich data extremely difficult

Much of the investigation into the intelligibility of synthetic speech has been carried out using lists of single words, separated by pauses [Wat87] It was demonstrated that when the length of the pause was reduced, the retention was degraded far below that of natural speech Waterworth conjectures that the reason for this is that listeners are exhibiting a recency or primacy effect

[Wat87] It is inferred that the listener's working memory is concerned with either analysing and interpreting the acoustic input, or rehearsing material which is already present Coupled with this, Pisoni [RPL+91] has shown that the comprehension of synthetic speech depends on the quality of the system, and varies over a wide range, from 95 5% in the case of natural speech, to 75% when poor quality synthetic speech was in use It can be further inferred that the intelligibility of spoken output is determined by

- whether the spoken output is synthetic or natural
- the quality of synthetic speech
- the level of prosody contained in the spoken utterance

Stevens [Ste96] informs us that Smither [Smi93] conducted an experiment into the increased load synthetic speech placed on short-term memory He tested young and old adults using both natural and synthetically generated speech His results shows that older adults performed worse than younger, while both groups faired worse when using the synthetically generated spoken output A major factor in the understanding of synthetic speech is the *fatigue effect* which is primarily brought about by the monotonous quality of synthetic speech Over long periods of time, or in the course of lengthier, more syntactically complex utterances, the comprehension of the material being heard can decrease as the listener becomes either tired or bored with the vocal presentation It has been found that the introduction of prosodic cues into spoken output has increased intelligibility significantly One possible reason for this is the relieving effect that the inclusion of prosodic features, such as alterations in the pitch range, and changes in the rate introduce a rhythm more akin to natural speech, hence relieving the tedium of the monotonous voice

This fact has major implications for the presentation of syntactically complex material such as mathematical equations Two sets of rules are known to exist for the production of spoken mathematics The first is provided by

the Confederation of Taped Information Suppliers (COTIS), and the second is a set of guidelines written by Larry Chang [Cha83] (see Section 2.4.1 for a description of these rules). These rules attempt to alleviate the problem of syntactically rich material through the addition of lexical cues, adding to the mental workload of the listener. Also, both these sets of guidelines are aimed at the human reader. Consequently, they are flexible enough to permit the semantic interpretation of the material, or to read the various symbols as they occur. In addition, the fact that both sets of rules are intended for human use assumes that the human reader can employ all features of natural speech when speaking the material. Such semantic interpretation is not available to any automated system; necessitating the development of a tighter set of rules to unambiguously present the material.

The alternative to natural speech is the use of synthetically generated output. However, the problem with this form of access is that most generic screen access technology cannot present mathematical information, reducing the means of manipulating mathematical data to the use of linear programming languages such as LaTeX[Lam85], which can be accessed using a standard text editor and a piece of screen access technology. (For a more comprehensive discourse on the rendering of mathematical information, see Section 2.4).

2.2 Extracting Document Structure

The means of expressing the structure of a document can be viewed in two different ways, namely a layout based approach and a mark-up based approach. The interpretation of these diverse views is ultimately responsible for the amount of structure which can be extracted from the document. The approach taken when examining an Optical Character Recognition (OCR) based document will therefore be vastly different from that taken when examining a document which contains a high degree of mark-up. In both cases, the objective is the same: to produce a model which accurately reflects both the content and underlying

structure of documents As the fundamental aim of the TechRead system is to extract the structure and content of a document using a mark-up based approach, it is this particular area which will form the basis for the rest of this discussion

2.2.1 Extracting structure from mark-up

Such languages as LaTeX [Lam85] and SGML[MJ95] (Standard Generalised Mark-up Language) use a form of mark-up to represent documents in a manner independent to their layout The intention is to leave the author free to concentrate primarily on the content, while also encapsulating the structural and visual attributes in a means which is intuitive Moreover, the notion of *Document Type Definitions (DTD)* has evolved [MJ95] Here, a document template is defined, which encapsulates different classes of documents For example, the majority of information written for publication on the **Internet** is written in HTML, which is a DTD of SGML

The advantage of using languages such as SGML is that multiple applications can access the same documents, and more relevantly for the TechRead system, multiple views of the same data are not only possible, but extremely practical For example, it is possible to view the SGML document in a WYSIWYG ("What You See Is What You Get") form, which performs some analysis on the SGML to produce visual rendering of the mark-up Alternatively, it is feasible to view the SGML source using a standard text editor to display the actual mark-up

A significant amount of work has been carried out on editors which are sensitive to the structure of documents Indeed, it is common practice that integrated development environments for programming (such as the Visual C++[Kru97] environment) uses both colour and indentation to reflect different attributes of programming languages This method of displaying data has been extended to documents, to enable their structures to be displayed in a more user-oriented manner

Typically, a tree based architecture is used to encapsulate the logical structure of documents Using this type of structure, the hierarchical content of documents can be modeled both accurately and unambiguously Using this form of representation, it is possible to replicate the notion of the hierarchical nature of sectional units If we assume that the root node of the tree is the topmost level of the document (i e titular information etc), then the highest level sectional units can be located below this, while each subsectional unit can form the remainder of the branches of the tree, until the lowest level leaf nodes are defined

As the complexity of documents increases, it is becoming increasingly obvious that the nature of the constructs needed to contain both their structure and content need to alter to reflect this Whereas a plain ASCII document can be modeled using a standard tree-based structure consisting of homogeneous sub-trees, a highly technical document produced by LaTeXneeds more complex heterogeneous structural components For example, obvious distinctions exist between the textual content found in LaTeXdocuments, and the mathematical content also found therein

One such difference is that the textual content consists of purely horizontally juxtaposed characters which can be grouped together to form words, each of which are delimited by white space Mathematical expressions on the other hand, utilise vertical positioning, coupled with the horizontal juxtaposition of the symbols to give an underlying semantic meaning

Consequently, two (at least) different structures are needed to store this vastly different content, while each individual sub-structure must itself be capable of being contained within the overall superstructure of the model The former, (used to store the textual content) need only be capable of linking items in a horizontal manner, and contain data elements capable of storing the font attributes necessary to convey the enhancements which authors ascribe to indicate such aspects as emphasis The mathematical structure, on the other hand, needs to be an entirely different structural component It must be capable of

not only representing the horizontal juxtaposition of symbols, but also their vertical location relative to one another

Though the hierarchy of the document and all textual elements are contained in a tree based architecture, it is essential that the capability for the inclusion of alternative structures (such as linked lists) be present within any system Some work has been carried out on the transformation of documents prepared using one system of mark-up, into another such language Two approaches are commonly used to achieve this goal The first method involves the recognition of the high level structures in the form of an abstract syntax, followed by the conversion of this abstract syntax to any desired concrete syntax Alternatively, it has been traditional to produce an output form, which comprises the least common denominator of the various input sources, and to then exchange this

2.2.2 The document model used in ASTER

One of the most important attempts to produce accessible technical documents to date is the ASTER system [Ram94] ASTER (Audio System for Technical Reading) aims to produce accurate renderings of documents marked up in the TEX[Knu84] family of languages Unlike the TechRead system, it uses both spoken and non-speech audio to assist in this process

The portion of the ASTER system responsible for extracting the high-level document structures can cater for varying degrees of mark-up, from the plain ASCII file to the highly complex and technical LATEXdocument The essence of the document model used in ASTER is the `attributed tree` In this model, each hierarchical level is modeled as a node in the tree, where each node can have content, children and attributes In Chapter 2 of the description of ASTER, [Ram94] the discussion on the extraction of high-level structures uses the "article" document class found in LATEX, and so the description of the extraction process used here will also focus on this area

The discussion begins with the description of the class *article*, which has

attributes such as title, author, abstract and date The children of this object are the various sectional units of the document, while the *prologue* of the object consists of any text which occurs before the beginning of the first sectional unit The model described in the context of the *article* class, can be extended to cope with the other classes of document found in LaTeX, such as *report* or *book* The leaves of this model comprise the actual content As in the TechRead system (see Section 3 1), ASTER differentiates between the textual and mathematical content of a document, the former being represented by a set of ordered textual objects, while the latter is modeled using the `InlineMath` object Each node is linked to both its parent and siblings Some of the objects found in this model are [1]

sectional unit This object type has attributes consisting of title, Section number and sectional name This sectional name could be *section*, *subsection* etc The children of this node are the subsectional units, while the prologue contains a list of document objects containing the text found before the first subsectional unit

paragraph This object has no attributes, and no children Its content is the list of document objects

word The attributes of this object type are the footnote markers (if any), and has no children The contents are made up of the string, which is the actual word

lists The enumerated and bulleted lists are sub-types of this generic class There are no attributes, and the children consist of the items of the list There are no contents

item The attributes of this object are the item number or type of bullet etc There are naturally no children, and the contents of this node is a list of document objects

[1] These definitions are extracted and paraphrased from chapter 2 of Raman's Thesis

tables The attributes of this object are the footnote markers (if any) and captions There are no children, and the contents consist of the tabular contents, modeled as a doubly linked list

text block The font information forms the attributes of this object type There are no children and the contents consist of a list of document objects

math equation This object is a numbered mathematical equation, whose attributes are made up of the equation number and other cross-reference tags The content is a `MathObject`, and there are no children

A Quasi-Prefix notation has been devised to hold both the content and structure of mathematical formulae [Ram94] This notation is based on the prefix form of mathematical formulae, but is extended to cater for such aspects of mathematics as superscripts and subscripts A tree-like representation is used to store the quasi-prefix notation, in which the operator with highest precedence would form the root node , while those operands to which this base-line operator apply would be its children

Using this model, the simple equation $3x - 5 = 5$ is represented with the $=$ forming the root node of the tree The right-hand branch would simply contain the 5, as this is one of the operands to which the $=$ refers The left-hand side is slightly more complex, consisting as it does of an extra level Below the $=$ on the left branch is to be found the $-$ operator The leaf nodes of this portion of the tree consist of the two operands $3x$ and 5 respectively Raman defines the quasi prefix notation as one which " delays the assignment of semantic interpretation to instances of written mathematics The quasi-prefix form captures the mathematical notation itself leaving the assignment of semantic interpretation to a later step " [Ram94]

The order of the operators in the encoding need not be that which is actually spoken The author points out that the LaTeXdisplays the mathematical notation in one presentational form only, and leaves the interpretation of the

information to the reader Since the listener is the passive partner in the reading process, there can often be times when a different ordering is desirable As a simple example, imagine the two sentences "x equals a product b" and "x = the product of a and b" Both are semantically equivalent, but the latter could be preferred by some users

An interesting feature of the ASTER model is the ease with which it can be extended to cater for new object types [2] Raman states that the first step involved in the extension of the model is the incorporation of the new object types as part of the representation In Object Oriented terminology, the introduction of a new object can be reduced to the specification of the object, and the various attributes and methods which can be associated with it ASTER adds the new object by firstly defining the name and number of parameters of the object, the processing function associated with it and its precedence in the hierarchy of objects Other features are associated with the macro, which determine how it is rendered in audio

We can see that the model used in the ASTER system is capable of displaying the myriad document elements available in LaTeX The means how this representation is accessed by the user to permit highly structured browsing, is described below

2.3 Browsing documents using spoken output

As was stated in Section 1 1, the control over the audio view of a document is minimal at best Currently, the most common form of "audio reading" is to have a transient speech signal flowing past the passive listener, whereas with normal visual reading the roles are reversed, and the information remains passively on a page while the reader actively peruses it [Ram94, Ste96] The following sections describe some of the methods used by people browsing both structured and non-structured electronic information using audio devices of various types

[2] Indeed, the notion of incorporating such features into TechRead can be traced to this fact

2.3.1 Non-structured browsing

The two most common means of electronically accessing documents are by the use of either the audio cassette, or screen access technology The former is self-explanatory, whereas the latter form of access needs some explanation The means whereby blind people access computer based technology is primarily through the use of screen access software, synonymously known as screenreaders These applications simply act as a communication layer between the visual medium of the screen, and the blind user, conveying the information which appears on screen using either synthetic speech or refreshable Braille If a `dialog box` appears on screen, the screenreader should read the contents to the user

One major disadvantage of such generic software is that, though extremely flexible, it does not take into account the need to browse information in a structured manner Rather, documents are reduced to components related hierarchically through their relative location on screen This does not permit the user to browse the document in the same manner as their sighted colleagues who can navigate through the content using the structural units, and visual alterations to assist them Screen access technology permits the user to navigate through the document using alternative structural units, among which are lines, sentences, words or characters In an era of `Object Linking and Embedding (OLE)` this form of representation does not support complex document structures, such as mathematical content or tables

There are two basic strategies for reading a document using a screenreader Firstly there is the sequential *document read*, which simply commences at the current point within the document, and continues to read until the end is reached, or the speech is terminated by the user This is akin to an audio tape, as the information simply flows past the listener, however this alternative offers a little more control Each screenreader manufacturer has concluded that a so called "audio focus" exists As speech is a serial medium, the "audio focus" is deemed to be that item on which the user is currently located This focus

is extremely narrow; lacking the ability to look rapidly at other parts of the display and thereby determining the relative location of other items Consequently, the notion of *ReadCurrent*, *ReadNext* and *ReadPrevious* has evolved for each Section into which the document has been reduced In one particular screenreader, the user can hear the line on which the current focus is located by holding down the Insert key on the numeric key pad, whilst at the same time pressing the number 8 on the same portion of the keyboard, whereas to hear the current word the insert key is once again held down, while the number 5 key is pressed

This form of browsing yields a modicum of control to the audio-based user, though it is insufficiently flexible The sighted reader can use the changes in the presentational style to enable rapid perusal of the information, while the blind reader (using generic access technology) is restricted to those units which the developers use in their navigation model The degree of flexibility to simply flick a page and move to some point thereon through the use of visible attributes, is simply not present using this interactive process As was stated in the previous paragraph, the means of accessing the content of a document is based on a decomposition of that document into visually juxtaposed sections, and takes no account of the structural hierarchy of the document The innate ability of sighted readers to rapidly skim through large portions of text and to discern the salient features by their visual appearance is simply not present in these generic screenreaders

To return to the analogy of the newspaper presented in Section 1 1, it is not possible to rapidly and effectively glance down the document to discover the headlines or other sectional units Rather, a series of atomic actions must be performed to locate and read articles of interest The following would be an example of an algorithm used to locate a given article in the sports Section of a daily newspaper

1 load the document into a Word Processor or editor containing a searching facility

2 activate this search facility and enter a keyword such as "Sports Section"

3 continue to do this until cross-references have been bypassed, and the actual sectional unit with this title has been reached

4 activate the search facility once again, and enter as many unique key words as possible

5 continue until the article required has been found

As can be seen from the above algorithm, the means of actually finding required text is both time-consuming and tedious For example, suppose an article entitled "Last Minute Goal Wins Final" were being sought The user would have to trawl through all references such as "see article entitled " before the actual text of the article had been located

Such complex document objects as mathematical material are not catered for in any fashion by generic screen access technology The primary reason for this is that Word Processing software typically uses graphical symbols to denote the unique symbols used in this form of presentation The results of this are that the screenreader cannot access the graphical symbols, and hence cannot provide any output (meaningful or otherwise) to the listener The alternative is to use linear notations such as LaTeX [Lam85] to read the technical content Further, in syntactically complex material, it can often prove too mentally taxing to retain and comprehend the semantics of the mathematics, while simultaneously determining what the linear notation actually means The result is a need for alternative strategies to impart this type of material to the user

2.3.2 Structured textual browsing

As was illustrated by the examples in the previous section, the need for some means of browsing documents in a structured and ordered manner is paramount ASTER [Ram94] aimed to solve this by introducing the notion of tree-based

document browsing The reasons for this form of document browsing strategy are explained by the structural model outlined in Section 2 2 2 The author has reduced the browsing of the content and structural elements of documents to several atomic actions, based on tree-traversal techniques which employ to advantage the design of the document model The primary atomic actions catered for by ASTER are

1 go to next sibling

2 go to previous sibling

3 go to parent

4 go to leftmost child

5 go to right-most child

6 mark current node

7 return to marked node

The interface used in the ASTER system is based on the keyboard mnemonics used in the `Unix` based `VI` editor Though not explicitly stated, the ASTER system appears to be aimed at users who have attained a high degree of computer literacy and competence, as it is difficult to use It requires an in-depth knowledge of tree traversal, and complex computer systems in order to both operate and customise effectively

As an example of the browsing which is possible using this type of structure, let us assume that a footnote has been encountered in the running text The atomic actions required to read the footnote and return to the current point in the document are as follows

1 mark current node

2 go to parent (This gets us out of the current paragraph)

3 move across siblings until footnote is encountered

4 read footnote

5 return to marked node

[Ram94]

As this demonstrates, the model used in ASTER to represent both the content and structure of the document is extremely rich It is possible, though not efficiently so, to navigate successfully through the document hierarchy to read the information required

The key-mapping used in this system is based on the Unix editor VI Consequently, the keys used to achieve the atomic actions necessary for tree traversal, though intelligible to anyone familiar with this editor, are not intuitive, as Table 2 2 shows The keys do not immediately suggest their purpose, which makes the system extremely difficult to use

An interesting anomaly of this keyboard mapping is the use of "L" to denote a movement to the right, while "h" indicates a movement of the focus to the left There seems to be no apparent relationship between the keyboard mnemonic, and the action which it is intended to perform While accepting that the author has kept to the design principles inherent in the VI editor, it is assumed that such a keyboard mapping would be extremely difficult to memorise

2.3.3 Structured mathematical browsing

The discussion to date has focussed primarily on the means to successfully and efficiently browse the textual content of structured documents Another major consideration in the design of a system to present accessible technical documents to blind readers, is the more complex or mathematical content Existing screen access technology provides little or no meaningful access to this kind of material, and it is only through the advent of specialised document viewers, or systems

Keyboard Mnemonic	Action
t	Go To Top of Document
CTRL-U	Go to top of current math expression
h	Move left set current selection to previous sibling
l	Move right set current selection to next sibling
J	Move down set current selection to first child
k	Move up set current selection to parent

Table 2 2 Basic Keyboard Mnemonics for ASTER

designed for the sole purpose of mathematical access that this data has become even modestly accessible

Math browsing in ASTER

ASTER uses the same tree-based hierarchy to represent and browse the mathematical content of a document Once again, a system of keyboard mnemomcs based on the VI editor has been assigned to supplement those already described above The keyboard mnemomcs utilise the quasi-prefix representation described in Section 2 2 2 The assumption when using ASTER to browse even a simple equation, is that the user is **extremely** mathematical and computer literate The reasons for this assumption are

- the notation is not a traditional view of the equation
- the means of browsing equations dependent on an understanding of a notation which is not in common usage
- a combination of the knowledge of tree-traversal, and the quasi-prefix representation necessitates a high level of mathematical knowledge

The interface also does not lend itself to easy use, unless the person using the system is conversant with both LaTeXand the VI editor As Table 2 3 indicates,

Keyboard Mnemonic	Action
1	Set Selection to Current Content of Node
"	Go to superscript
	Move to subscript
	Move to Accent
#	Move to underbar
!	Move to left Subscript
%	Move to left superscript

Table 2 3 Supplemental keyboard Mnemomcs for Mathematical Browsing in ASTER

the mnemonics used to supplement those described previously are not intuitive, rather they are extremely dependent on the reader being an advanced computer user

To those familiar with LaTeX, it will be immediately apparent why some of the keys have been chosen However, others need some explanation the author informs us that

> "The above key-map for traversing the attributes was arrived at as follows The choice for superscript and subscript is automatic, since the keystrokes match the symbols used by TeXto markup these attributes Placing the fingers on the row of numerals on a standard keyboard, the actions necessary for typing " and are mimicked with the left hand to arrive at the key-bindings for the left superscript and subscript The middle finger of each hand is used to get to the accent/underbar "

[Ram94, ch4]

Though this explanation is extremely logical, there is still very little intuition about this interface It demands a very high mental work load to both

absorb and memorise the key mapping, and to comprehend the highly complex mathematical information which is being presented by the system

Browsing in MathTalk

MathTalk [Ste96], designed by Dr Robert Stevens at the University of York is another example of a system designed to render algebra more accessible to blind students Unlike the ASTER system described in the previous sections, this system is aimed at students who are still at school, and is particularly aimed at those between the ages of 16 and 18 Accordingly the form of presentation, and its interface are deliberately simpler and more intuitive to use The design principles on which this work are based, attempt to render the algebraic formulae in a non-interpretive fashion This contrasts totally with the approach taken in ASTER, where the mathematical content was rendered to make it more intelligible (See Section 2 4 for a more in-depth discussion on the ways in which both systems speak the mathematical content) The interface is also less intrusive The aim of the system is to present algebra in an unambiguous fashion, and to give the user complete access to all portions of the equations One feature which is incorporated into the system is the notion of *object folding* This involves presenting the equation through various levels Thus the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2 1}$$

would be spoken at its most superficial level as

"x = minus b Plus or minus the square root of, a fraction "

[Ste96, ch3]

It is immediately obvious what the equation consists of, namely a value x, which is related to some other formula via the relational operator = It is also

immediately apparent that the right-hand side of the equation consists of a fractional quantity

The notion of *object folding* essentially means that when the user wishes to browse through this formula, it is possible to look at the top-level of the equation first, to verify the nature of each individual term Then, if they so wish the terms can be decomposed into their sub-terms until no further decomposition is possible

The keyboard mapping which is used to achieve the browsing capabilities in the MathTalk system is based on a *meta-language*, consisting of a keyword, and a target action These keyword and action pairings combine to form a language which is extremely flexible The keywords which are used within the system revert back to the standard approach taken by the developers of all screen access technologies, namely the ideas of *current*, *next* and *previous* When one of these keywords is combined with a set of actions (or targets) they can produce an interface which is intuitive and easy to use

The targets which are available within the system vary greatly, depending on the type of formula which the reader is browsing The keyword "next" could always be combined with the target "term" to hear the next term of the equation This two key sequence would, if another term was present, then relay the description of the type of content found, or play an error if no other terms were encountered Another example of the target actions possible is the combination of "next" and "fraction" This option is only available if the type of material being perused actually consists of a fractional unit, otherwise an error message will be played

The keyboard mnemonics used within the MathTalk system appear to be more intuitive than those found in the ASTER system Evaluation of the system [DNA97] reveals that much effort went into the design of this portion of the system The keywords *next*, *current* and *previous* were mapped to the letters *n*, *c* and *p* respectively The target keywords also followed the practice of using

a mapping, which ensures that (insofar as is possible) they correspond to the first letter of the term they are supposed to represent The command *next term* would be executed by the key sequence *nt*

It should be noted at this point that MathTalk is in no way as comprehensive a tool as the ASTER system The author set out to design a system for the browsing of algebra, whereas Raman's[Ram94] main aim was to produce a system for the rendering of entire documents into Audio Stevens' objective was to deduce a set of design guidelines on which future systems could be built, to enable the manipulation and writing of mathematical formulae It should also be noted, that the system confined itself to a mathematical sub-set, namely that of algebra, whereas ASTER is unrestricted Moreover, the interface to the MathTalk does seem to offer scope for expansion, and appears from the data found in [Ste96] to work effectively

2.4 Producing audio output

As was stated in the previous section, the vendors of currently available screen access technology do not harness the prosodic capabilities of synthetic speech devices This makes listening to, and more importantly understanding output from these devices extremely difficult for those who are not used to using speech synthesisers Much of the research into the area of producing accessible technical documents has been directed towards the prosodic enhancement of textual and mathematical content in an attempt to derive spoken output which listeners find more appealing This Section discusses the significant contributions made in this area It is intended to outline firstly the rules which Larry Chang [Cha83] devised to aid human readers to describe mathematical equations when reading them onto audio cassette Proceeding from this the efforts of Raman [Ram94] and Stevens [Ste96] to devise alternative methods of producing spoken output are described

2.4.1 Chang's rules for spoken mathematics

In the 1980's, Larry Chang defined a set of rules for the verbalisation of mathematical expressions These guidelines were designed for use by human readers, though the author envisaged their inclusion in any system for the production and artificial utterance of mathematics The rules are based on the addition of lexical cues to an expression They did not rely on the prosodic component of speech, rather on the use of descriptive phrases to inform the listener as to the type of material to which they were listening Chang states that "mathematical material is primarily presented visually, and when this material is presented aurally, it can be ambiguous" [Cha83] The rules described below were defined in an attempt to reduce ambiguities found in complex mathematical content

Chang begins his definition of the guidelines by introducing the various alphabets and the means for their verbalisation He states that large letters should be preceded by the word capital" or "cap", and when visual enhancements are introduced, they should be uttered Therefore, if the following expression $a+B$ were encountered in a passage of text, the version recommended by Chang would sound like "italics a plus cap b" He gives an interesting definition of the methods for speaking alternative alphabets to the roman style of letters He outlines phonetic representation for the Greek alphabet, which attempts to inform the user as to the correct pronunciation for these letters As an example, α is represented as "al fuh", with a stress mark indicating that the "al" should be spoken in this manner

The basic symbols (such as operators) are simply spoken as they occur, though an alternative suggestion is provided which is more interpretive The + sign can be spoken as *plus* or *positive*, − said as *minus* or *negative*, etc Parentheses and all other brackets are preceded by the words *open* or *close* respectively, further indicating the nature of these delimiters

The actual description of the various forms of mathematical content now follows Chang begins with simple algebraic formulae and equations The first,

and simplest example (found in section 4), $a - (b + c)$ is spoken using this method of presentation as "a minus the sum b+c", "a minus the quantity b plus c end quantity" or "a minus open parenthesis b plus c close parenthesis" [Cha83] This example illustrates how the various lexical cues can be combined in both an interpretive or non-interpretive fashion A further example of this concatenation of cues is the expression $a[b+c-e(f-g)]$ The verbal rendering of this formula is extremely cumbersome and bears no relation to the length or complexity of the visual form The spoken utterance for this equation is

> "a times the quantity b plus c minus the product e times the the difference f minus g"

The alternative renderings simply speak each symbol as they occur, causing the inclusion of such cues as "open bracket", "close parenthesis" are substitutes for the cues just described

The guidelines defined by Chang seem to work better in the realms of trigonometric functions For example, $\sin\theta$ is spoken as either "sine of theta" or "sine theta" The cues seem to fit these types of functions, as they are more readily verbalised, however as the complexity increases, the utterance once again becomes inordinately long Another area where the set of guidelines works well is in the description of such mathematical concepts as summations, or bounded integrals The expression $\sum_{i=1}^{\infty} x^i$ is spoken as "the summation from i = 1 to infinity of x sub i" Here the concatenation of the various lexical cues works well, as they form an intelligible, relatively brief utterance

The final aspect of Chang's rules which is relevant to our discussion is the manner in which he describes the verbalisation of matrices and arrays This form of construct is highly visual in nature, though Chang outlines a solution which is novel and easy to understand Let us assume an array consisting of the following elements $\begin{vmatrix} 2 & 7 \\ 3 & 10 \end{vmatrix}$ This would be spoken in two alternative methods The first would be to describe the matrix in row order, the second in column

order The utterance "a two by two matrix, first row 2,7 second row 3 10" gives the listener a clear impression of what is contained in the matrix The version spoken in the order of the columns gives exactly the same information, save from a different perspective An obvious flaw with this mode of presentation is that as array elements become more complex, it will be difficult for the listener to maintain an accurate mental picture of their relation to the other elements in the array This is where the flow of control over the information is extremely important

Chang's rules were defined **before** the advent of widely accessible computer technology, and were intended primarily for use by human readers, a fact which should be borne in mind Consequently, though their verbal expression is often cumbersome it provides a beginning from which the TechRead system, and indeed those previous systems described below could continue

2.4.2 Audio output generated by ASTER

The tool used by the ASTER system to produce the audio output, is a language devised by the author known as `Audio Formatting Language`, or `AFL` This language, an extension to `LISP` can be described as " the audio analogue of visual formatting languages such as `Postscript`" [Ram94] The aim of this language is to provide mechanisms to control the multiple aspects of the audio presentation, such as speech-based, and non-speech sounds The output produced by the audio-formatter in ASTER represents the various components of the audio presentation using *audio space*, which is derived by taking the sum, or cross product of the various individual dimensions of the audio space Examples of these dimensions would be the spoken utterance, and the `earcoms`[3] which enhance the spoken output The output from the *audio formatter* is altered by adjusting the dimensions (parameters which may be changed) of each individual aspect of the audio space Such dimensions would include the pitch and rate of the voice, the means by which these characteristics change to reflect alterations

[3]audio equivalents of icons

in the visual appearance of the text, etc AFL rules usually introduce a new block, adjust the audio state and execute some actions [Ram94]

One of the most important functions used in ASTER is the `LISP` extension `ReadAloud` This function is then overridden to provide different renderings of the diverse objects contained in the ASTER document model However, this function only provides one view of the document object being rendered To surmount this, ASTER utilises the notion of styles, and rendering rules which can provide different, contextual renderings of the same object These rendering rules consist of `AFL` statements which alter the audio state in some manner Only one rule can be active for any given object at a specific time If the listener wished to produce a new rendering style, new rendering rules for the various default actions would have to be defined ASTER uses the styles in the order in which they were activated If a style which produced extremely descriptive renderings of the document were in use, and a simpler version of the rule for describing integral mathematics was required, the user would simply have to activate the rule which produced this version of this form of content

The structural components of the document are conveyed using " audio layout made up of extra textual speech and non-speech audio cues" [Ram94] Raman defines two forms of audio cue He says that fleeting cues are those which do not last, and whose duration is specified by the nature of the cue itself A persistent cue on the other hand, is one which lasts, and whose duration is affected by other audio-events An example of a fleeting cue, is the use of the word "section" which is spoken before the number, and title of a sectional unit An example of a persistent spoken cue, is the raising of the pitch of the voice to denote an itemised list, while a persistent non-speech audio cue is the use of a background sound, which continuously repeats Raman defines "audio layout" as follows "Audio layout is achieved by superimposing persistent, and fleeting cues on the renderings" [Ram94] In order to convey nesting, the alterations in the persistent cues " need to be monotonic in the mathematical sense" [Ram94] The author says that

"Let P represent a point in audio space Let f be a change of state function To convey nesting effectively, f should be monotonic, there should exist an ordering $P < f(P) < f^2(P) <$ where this order is perceptible "

[Ram94, ch4] Document structure is conveyed by playing fleeting, and persistent cues For example, the fleeting component would be the keyword "section", followed by another giving the number A persistent cue, could be a prosodic alteration to the voice, or a sound played in the background Specialised environments (such as lists) are conveyed in LaTeXusing indentation to visually distinguish then from the surrounding text This is analogously achieved in audio by ASTER's use of rising pitch to denote the same nesting Tabular data is another specialised environment ASTER uses stereo to convey the spatial separation found in tables Using this method, the leftmost column is heard only in the left-hand speaker, and the movement of the sound through the stereo space, is analogous to moving across each row in the table, until the rightmost column is heard solely in the right-hand speaker

The means used in ASTER to produce mathematical content is vastly different from that used in MathTalk (see below) Raman has observed that there is little in the way of similarity between the evolution of a written mathematical notation, and the audio counterpart He points out

"Any notational system is a combination of conventions and intuitive use of the various dimensions that are provided by the perceptual modality and the means available to produce appropriate output for that modality"

[Ram94, ch4] Raman also points out that the traditional mathematics notation uses a set of primative layout cues to achieve highly complex and often nested structures The main aim of ASTER's audio representation of mathematics is to produce a non-visual counterpart to the highly complex visual writing of

mathematics The system used in this particular software approach is to offer the listener the option to obtain highly descriptive renderings of the mathematics, or conversely purely notational The former can be used when new material is being perused, while the latter can be utilised when familiar content is encountered

The actual audio output aims to reduce the extra verbiage, which is an integral part of most audio documents produced by human readers The rendering also conveys nested structures, such as superscripts and subscripts, each of which can themselves contain arbitrarily complex sub-expressions The audio notation is achieved through the use of the fleeting, and persistent cues described above The technical means for rendering the mathematical expressions are by moving along various dimensions in the audio space `AFL` once again provides the functionality to achieve this movement

Unique dimensions are chosen to map the quasi-prefix notation described in Section 2 2 2 to dimensions in audio space This quasi-prefix form of the equation is an **`attributed tree`** (see Section 2 2 2) A dimension is chosen in audio space, and mathematically monotomc functions defined to control the movements along this dimension Next, a dimension orthogonal to that just outlined is chosen to denote the visual cues found in mathematical expressions The means that sub-expressions are rendered is to use a softer more animated voice, which produces a sense of "falling off into the distance" [Ram94] as the nesting deepens Superscripts are conveyed by a movement along the dimensions of audio space which cause a pitch increase in the utterance, while subscripts are produced using a decline in the pitch of the voice

The myriad delimiters found in printed mathematics, while being extremely useful to aid in the visual parsing of sub-expressions, can pose problems in an audio context To merely announce the delimiters would produce unnecessary extra verbal cues The rendering of the mathematical sub-expressions is highly subjective in ASTER The author has designed the system to distinguish between the simple and complex entities which make up the mathematical expres-

sion In the case of an expression like $a + \frac{b}{c} + d$ no new audio state is introduced to speak the fractional sub-expression, while in the expression $\frac{a+b}{c+d}$ a movement along the dimensions of audio space is used to convey both the numerator and denominator of the expression

Another means used in ASTER to convey the grouping of sub-expressions is to surround these expressional components with pauses The duration of the pause is given by a weighting which is assigned to each sub-expression in a given formula If the weight assigned is 1, no pause is introduced Alternatively, a scale factor is used to calculate the weighting of a sub-expression A complexity measure is utilised to calculate the weights of each given node in the tree-based structure of the mathematical expression The system is initially set up with a list of default special patterns to which the user can add, through method definitions which are based on a LISP function These extra patterns can then be activated, by calling another LISP function Such special patterns include the use of "squared" to indicate exponentiation by two, or "cubed" to indicate this form of exponentiation

The oral communication needed to convey complex material takes more time than the visual, and also places an extra burden on the short-term memory of the listener To aid in overcoming this problem, ASTER uses a system known as "variable substitution" to produce more meaningful output Raman conjectures that the experienced reader of mathematics will first look at the top-level structure of the complex mathematical equation and then progressively reads the sub-expressions

In an audio rendering of the formula, the listener does not possess the luxury of listening to both the high-level structure, and the basic sub-expressions in a linear rendering Consequently, ASTER takes on the responsibility for examining the grouping implicit in the sub-expressions and producing a meaningful abbreviated description This is known as *variable substitution* Therefore, given an arbitray fractional expression $\frac{e^1}{e^2}$, where both the numerator and denominator are themselves complex expressions, the spoken output would be

something like

"fraction e1 over e2 where e1 is (and so on) and e 2 is (and so on)"

[Ram94, ch4] The use of *variable substitution* is extremely useful when perusing complex mathematics The author points out, however that this form of rendering should be used extremely sparingly In the complex mathematical formula used as a demonstration of "variable substitution", he states that without "variable substitution" the equation takes 68 seconds to speak, while when it is incorporated the formula is spoken in 80 seconds The criteria for applying this rendering method have to be specific in nature A complexity measure has to be defined which determines both the nature, and the complexity of the substitution needed to produce more meaningful mathematical utterances One of the main outcomes of the application of "variable substitution" should be that of producing a succinct top-level description of the formula It also should convey as much information regarding the structure of the equation as possible, while not increasing the ambiguity of the spoken version of the expression

Regrettably, there is no data available for evaluation of the textual and mathematical audio produced by ASTER The author seems to have carried out some informal tests of the system, though it seems to be designed with a particular type of computer user in mind As will be apparent from the above discussion, the ability to produce highly customised renderings is built in to the program In order to achieve this customisation, a **highly** detailed knowledge of both the LISP programming language, and the algorithms used in ASTER is needed Some analysis of the output would be useful as a comparison between both the the future implementation of the TechRead system, and also with the output from the MathTalk system described in the next section

2.4.3 Audio output produced by MathTalk

The MathTalk program uses a vastly different prosodic model to convey both the content and structure of algebraic formulae This work focused on the core areas of algebraic notation, that is letters, numbers, radicals, superscripts and subscripts It should be kept in mind, that this system set out to solve a problem which formed a subset of that dealt with by the ASTER system The MathTalk program was designed to assist in tasks usually carried out using the paper or external memory [Ste96]

The first attempted presentation of the mathematical information involved the augmentation of the utterance with "lexical cues" in accordance with the rules devised by Chang (see Section 2 4 1) Accordingly, a subset of these rules was defined, incorporating alterations to some of the lexical cues added to the mathematical expression The first principle introduced into the MathTalk program was that of the necessity to reduce (as far as possible) the number of extra lexical information provided to the user It was realised that in accordance with the stated aim of the designers, to reduce the mental workload of the user, the length of the utterance should be kept as short as possible

Compromises were made from the position of total non-interpretation of the mathematical equations For example, a major dilemma for the designers of this system was, whether to verbally distinguish between the unary operators and their binary equivalents That is, should $-b$ be spoken as "minus b" or "negative B" It was ultimately decided to adopt the "minus" form of this operator, in an attempt to provide consistency between the binary and unary forms of the same operators Another factor in this decision was the notion that the use of different verbal naming conventions for symbols which visually appear the same, could mislead the blind listener into the misapprehension that there were two symbols used to denote the same operator

The methods for conveying the grouping of sub-expressions as outlined by Chang [Cha83] were described in Section 2 4 1 Stevens puts forward the theory

that some of these cues are interpretive

> "The phrases suggested by Chang 'begin quantity' and 'end quantity' are used to delimit the start and end of parenthesised sub-expressions An alternative to the word 'quantity' is 'group' The word 'quantity' might imply that the contents should be regarded as one mathematical entity, where the word 'group' might imply less, that the symbols are simply grouped together and the reader then has to decide that the group be treated as a 'quantity' "

[Ste96]

Another area in which Stevens attempts to reduce the number of lexical cues is in the depiction of fractions The example of the formula for finding the roots of quadratic equations is used to outline the means whereby the lexical cues can be reduced This equation demonstrates how, using lexical cues, it is apparent that certain types of operator terminate implicity the domain of others, thereby negating the requirement for explicit verbal ending of these domains The lexically augmented version of the equation following is given as

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2 2}$$

> 'x equals the fraction numerator minus b plus or minus the root of the quantity b super two minus four a c denominator two a '

[Ste96] It can be seen from this example that the lexical cue "denominator" ends both the scope of the root and numerator Consequently, there is no need to explicitly end these two terms by using additional lexical augmentation It becomes apparent that the author is forced to make compromises between the non-interpretive design of the MathTalk program, and the desire for increased ease of comprehension The point is made that a distinction exists between the

means to lexically enhance simple or complex fractions For example, the fraction $\frac{a}{b}$ is a simple fraction, so should it be spoken as *numerator A denominator B* or as the more typical *a over b*? The latter is more interpretive, but is more legible from the listener's point of view Consequently, the compromise was made and the more commonly used version describing the fractional content spoken

Unlike the printed Roman alphabet, which uses the vertical juxtaposition of characters coupled with various delimiters to convey the written form of a language, printed mathematics uses slight alterations in the vertical position of symbols to convey various aspects The superscript is usually used to convey exponentiation, although it is not solely confined to this role Consider x^2 as an example of a simple mathematical instance of a superscript Conversely in second order derivatives, the use of the superscript does not, in fact denote exponentiation The simple expression $\frac{d^2y}{dx^2}$ uses a superscript for vastly differing reasons As a consequence, the MathTalk program implemented a solution involving the use of the word "super" as a lexical cue denoting the presence of this element of mathematical syntax The reader is left to learn the meaning of the cue in various diverse contexts, as the visual reader must also do

Thus far, a description of some of the problems tackled by the MathTalk program have been described and the solutions which the designers implemented were outlined A summary is now required to clearly and accurately explain the general rules for the augmentation of mathematical expressions Stevens uses the following list to summarise the general rules implemented in MathTalk for the provision of lexical cues

- For Latin letters, MathTalk only prefixes uppercase with a tag
- For sub-expressions, MathTalk uses only the tag 'quantity'
- MathTalk speaks simple fractions (those with a single term in both numerator and denominator) with only the word 'over' between the two terms

- Complex fractions (those with more than one term in either numerator or denominator) are bounded with lexical cues
- Roots are enclosed with the lexical tags 'the root' and 'end root'
- Simple roots are spoken without an end tag
- Initially MathTalk used the cue 'to the' to indicate exponents Later this was replaced by 'super' (shortened from 'superscript') to comply with minimal interpretation
- The word 'all' can be used with the opening superscript cue, when the superscript governs a complex object

[Ste96, ch3]

We can see from these lexical cues that the utterances will be lengthened considerably by their presence Stevens also discovered this and, though the provision of lexical cues seemed an adequate solution, it emerged that an alternative, prosodic model could yield shorter expressions which would place less of a burden on the short term memory of the user Accordingly, the next phase of the MathTalk project was to devise a means of conveying algebraic content using prosodic structures

In order to devise a set of rules to prosodically enhance the algebraic utterances, Stevens conducted an experiment involving two experienced subjects, both of whom were native speakers of British English Each was presented with a random set of 24 equations, and two recordings on high quality tape was made for each speaker The participants were asked to speak the equations "as if they were addressing a class of sighted students" [Ste96] They were also asked to convey the information in as neutral a manner as possible, that is, not to "indicate any of the intentions of the mathematical notation" The recordings were analysed for pitch, timing and amplitude and these three characteristics were related to the syllabic content of the spoken versions The results provided

the designers of the system with the knowledge needed to construct a "simplistic prosodic model" [Ste96] which could be used to convey algebraic notation The three prosodic elements (pitch, pausing and timing) were measured in their traditional units, and the results were as follows

Only one recording was analysed in full, as the author informs us that all four were consistent, thereby reducing the need to analyse each 23 out of the 24 expressions were analysed for global changes in pitch It was discovered that the mean initial pitch was 159 Hz with a standard deviation of 20 Hz The mean ending pitch was 110 Hz with a standard deviation of 8 Hz All but one expression showed an overall decrease in pitch over the utterance Interestingly, the only expression which did not conform to this was that which included a subscript, but the author informs us that this formula was included for referential purposes, as the presentation of information contained in subscripts was outside the scope of the MathTalk system

The theory is presented, that as the spread in starting pitch is considerably greater than that found at the end of an utterance, the length of same could be determined by the pitch at which the speaking of the information finishes, as it concludes with an almost constant pitch Other trends outlined by Stevens are that if there are no baseline operators (such as =), then the pitch fall is roughly linear When baseline operators are included in expressions, the pitch fall occurs in a series of steps interrupted by pauses [Ste96] The final point to note about the effects of pitch on the speaking of an expression is that the pitch either remains level, or rises slightly when a baseline operator is encountered However, when the final such operator is spoken, the pitch falls dramatically, accounting for 34% of the total pitch fall within an expression [4]

A fundamental feature of prosodic enhancement is the addition of pausing

[4]The prosodic model used by Stevens is at odds in important ways with current views of prosodic and intonational phonology, and though it produces reasonable results, it is not the model we are using For a discussion of appropriate phonological models and their use in synthesis, see [Lad96, Mon91, Mon99]

to separate spoken utterances into units. This is particularly true of algebraic, or other syntactically complex material, where the pausing can indicate the grouping within an expression, assisting the listener to more easily understand the verbal presentation. Stevens [Ste96] discovered from the analysis of the recordings that a mean pause of 250 Ms was observable at baseline operators such as + or −. He conjectured that such a pause could be used in conjunction with an = operator. He also observed that the pause was related to the proceeding term, rather than that immediately preceding the operator. One approach which (he surmised) could aid the listener in anticipating the length of material on different sides of equations, was to adjust the pause length at the = operator, lengthening or shortening it to denote the length and complexity of the two sides of the formula.

The rules governing the pitch and timing variations within sub-expressions are different from those encountered at the baseline level. Stevens separates the types of sub expressions into those which follow a multiplication sign and those which follow printed base-line operators. Also, it is important to consider the position of the sub-expression within the overall expression; that is, whether it is a terminal sub-expression or one which occurs at the mid-point of an expression.

Interestingly, the recording analysed contained the lexical cue *times* before the sub-expression, preceded by a pause of 250 ms. Following this there was a pitch fall, which was greatest when the lexical cue "times" was omitted, resulting in a pitch fall of 81 Hz. Consequently, the lexical cue and pause were used in the MathTalk program.

If, on the other hand a sub-expression occurs as the first component of an equation, the pitch fall is observable on the first syllable. Stevens reports that pausing is not observable within sub-expressions, save where the length necessitates the taking of a breath. The grouping of the sub-expressions by pitch contour and pausing seems to combine them into a single unit, possibly making them easier to recognise by the listener.

The prosodic model defined by Stevens is, by his own admission, simplistic However, he formulates the theory that this means of prosodic enhancement could form the basis for a more concrete model It is pointed out that the range of examples used in his experiment were not wide enough to be certain of the exact nature of the rules required, but maintains that from the data gathered, there should be no problems extending the model to form the basis for a broader set of rules encompassing facets of mathematical content other than algebra

In order to evaluate the prosodic model which has just been outlined, Stevens conducted an evaluation, involving the comparison of mathematical expressions augmented with prosodic enhancement, with expressions supplemented with lexical cues The two hypotheses on which the evaluation was based were that the prosodic additions to the utterance, would produce comparable results to those expressions containing extra lexical cues Also, it was surmised that the lexical cues would disrupt the retention of the listener A simple recall task was used in this experiment A single utterance was used in this experiment, and while it did not reflect the real world, it was hoped that it would show the problems which the listener encountered It is pointed out that even if one form of expression was superior to the other then it was probable that not all the structure and content would be retained after a single utterance It was hoped that the transcripts would reveal the problems encountered by the subjects with regard to the types of structure with which there were difficulties, and the amount of information which could be retained after a single utterance It was further hoped that these errors would produce some design goals which could be implemented and incorporated into other aspects of the user interface

The experiment incorporated three distinctly different presentations of the algebraic expressions Firstly, the utterance was augmented by prosodic cues The alternative presentation was the use of lexical cues to provide the structural and contextual information needed for the successful intelligibility of the formula Finally the mathematical content was simply presented using the de-

fault speaking style of the synthesiser The use of this third form of utterance was to provide a bench-mark to assist in the annalysis of the other two groups The unenhanced version of the formula would indicate the nature of the errors encountered by the subjects when little or no structural information was given From this, it could be determined how the other two forms of presentation performed in comparison With this in mind, two different groups were presented with the same utterance The first heard the expression using both the lexically supplemented version and that which employed no enhancement whatsoever (known as the LN group), while the alternative subject group were presented with both the version containing the lexical cues, and that which had been prosodically enhanced The recall nature of the experiment produced a varied set of responses However, the author found it difficult to accurately mark, as the answers were rarely totally correct or totally erroneous To combat this, the questions were marked separately for the perception of the structure, and the retention of the content The author defines the correctness of an answer as one which retained the content of the formula, while also determining the structural elements A subjective marking scheme was used to determine the correctness of the answers The content had to be at least 75% correct, while in order to ascertain the correctness of the comprehension of structure, the key elements of the formula (such as base level operators, sub expressions etc) had to be present in the subjects' answers

The experiment designers used the NASA Task Load Index (TLX) to determine the workload which the subjects encountered in performing the individual questions The participants were asked to give a rating on various aspects of the presentation The five criteria used to measure the workload needed were mental demand, time pressure, effort expended, performance level achieved, and frustration experienced After the second condition in each group, the participants were requested to "quantify these measures relative to the first assessment" [Ste96] A final part of the experiment was that the users were asked to outline which of the two forms of presentation they preferred That those taking the experiment would prefer one form over another is undisputedly true,

	Condition	
Number	Non-prosodic	Prosodic
1	$(y+6)(y-6)$	$x = x^{n-2} + 7x$
2	$y + \frac{6}{y-4}$	$x + 5(x-5)$
3	$y = (x-9)^{k+3}$	$(x+2)^2$
4	$3^{y+6} = 5$	$\frac{x+3}{x-6}$
5	$(\frac{4y}{7(y+2)})(y+9)$	$5 + (x^2-9)^x + 5$
6	$9(y-6) + 3(y+7)^7 = 3$	$x = (y+5)^n - 2$
7	$y = 2(y^4 - 8(y+5)$	$(x+7)^{\frac{1}{2}} = y$
8	$\frac{2}{3}(3x+9) = x+3$	$4(x-9) + 5(x+7)^2 = 0$
9	$(x-9)^3$	$\frac{1}{2}(2x+4) = x+2$
10	$3 + (y^6 - 5)^{y+7}$	$\frac{3x}{(x+4)(x+7)}$
11	$(y+1)^{\frac{2}{6}} = x$	$x = 3(x^2 - 8x + 1)$
12	$y = y^n - 9 + 4y$	$3(x+4) = 7$

Table 2 4 Questions for the prosody evaluation experiment Both conditions are shown in the order of presentation The prosodic condition stimuli were those used for the *no-cues* condition

but it was hoped to gain a form of satisfaction rating from this data

As was stated above, a single utterance was used throughout this experiment Twelve expressions were used, and were augmented using the prosodic and lexical enhancements described earlier in this section What's interesting is the range of expressions used in this experiment (see Table 2 4) [Ste96]

It can be seen from this table that the expressions all contain either fractions (simple or complex), sub expressions or subscripts The author tells us that the expressions were chosen to conform to the standards of those found in algebra manipulation exercises found at A-level examinations [5] Before the commencement of the experiment, the 24 participants were allowed to gain some practical experience of listening to synthetic speech The synthesiser used in

[5]This level of examination constitutes a level of pre-university examinations

this experiment was `Berkley Systems' BEST` synthesiser. The prosodic and lexically supplemented versions of the equations were sent as text strings to the synthesiser, while the **no cues** version comprised the un-adorned textual representations of the equations, which were simply sent to the synthesiser.

The procedure followed in the course of this evaluation involved the participants becoming familiar with the presentation style of the synthetic speech device. They had the general design of the experiment outlined to them in a script, and some sample expressions were read, both by the experimenter and by the synthesiser. When the participants were satisfied that they were entirely comfortable with the presentation style of the synthesiser, the actual questions pertinent to the experiment were presented. The participant heard each expression once, and was asked not to write down their answers until the verbal presentation had finished. They were also asked to denote their uncertainty either by using question marks or ellipses.

The results of this evaluation proved without any doubt that the participants performed better when listening to the equations presented in a prosodically enhanced, rather than a lexically supplemented version. One fact which emerged was that those who partook in the experiment where a lexically enhanced, and **no cues** experiment were able to comprehend and retain more of the structure of the expressions when the lexically augmented version was played.

However, this process was reversed with regard to the actual mathematical content, more of which was recalled when no lexical cues were inserted into the expressions. A comparison between those listening to the prosodic version, and that which was sent unadorned was not carried out as the results for those listening to the lexically enhanced version was comparable between both groups, rendering this third comparison unnecessary. As a consequence of this experiment, the hypothesis that prosodic enhancement would prove at least as good as the addition of lexical cues was rejected in favour of another which found that the prosodic enhancement of mathematical expressions conveyed more structure, and permitted greater recall than the addition of lexical

cues.

The results of this experiment demonstrated that the prosodic component of speech added significantly to the recall of the expressions. What is interesting about the effects of the three conditions on mental workload, is that both the prosodically enhanced, and unadorned versions proved to be almost identical, whereas the lexically augmented version proved to be significantly more taxing mentally [Ste96]. While the mental workload associated with the prosodic version of the expressions is lower than the other two versions, it is still quite high, thereby leading to the notion that the listening process itself is quite difficult.

It can be said therefore, that the prosodic model of algebraic presentation aids the listener in the comprehension of the structure and content of the expressions in the same way as white space assists the print reader to parse the equations into meaningful sub-units. The addition of pausing and pitch variations around spoken sub-expressions can aid in the parsing of the expression and, while not perfect, it is believed that it will increase the accessibility to mathematical content greatly.

2.5 Summary

This chapter has focused on the research carried out to date, relative to the production of accessible technical documents. It has been shown how visual and audio-based reading differ, and also the point at which they converge. A description of the model used in ASTER [Ram94] to convey document structure was given, and the browsing strategies which this model permits were discussed. The interfaces used in several systems were described, and the chapter concluded with a description of the means that have been investigated to assist in the speaking of mathematical and technical content. The principal objectives of this chapter were to give the reader a perception of the current state of the research area, and a description of the problems which the TechRead

system addresses The remainder of this thesis outlines the methods employed by the TechRead system to improve on those which have preceded it

Chapter 3

Different views of the same data

In the previous chapter, a description of the work carried out to date into accessibility to technical material was discussed This was partly to give the reader some idea of the current state of accessibility to technical documents for blind people, but also as an introduction to the problems which the TechRead system overcomes This chapter begins the description of the TechRead system itself It shows how the document will be translated from the input LaTeX into a model which is conducive for browsing It is also intended to describe the *human interface* to this document, outlining the "look" and "feel" of the interface It will be evident how the browser will appear to both the sighted and blind users respectively In the course of the discussion on the *human interface* we describe the reading strategies which it is intended to solve

3.1 The document model

As was stated in Section 1 1, one of the principal goals of the TechRead system is to produce an accurate model of the document, which both reflects and encapsulates its content and structure As was pointed out in Section 2 2, there

have been previous systems which tackled the problems of technical accessibility for the blind in various ways However, the approach taken in this work is vastly different from that demonstrated in such systems as ASTER[Ram94] and MathTalk [Ste96] That system used an **attributed tree** which modeled the document structure as a hierarchy, and the content as attributes of nodes (see Section2 2 2 for a more complete description of this model) TechRead on the other hand, uses a graph-like structure to model the document For the sake of referential purposes, this graph will be described in the course of this discussion as a cross-linked tree, however the model stored internally does not in fact resemble a tree structure

The essence of the model used in the TechRead system is that of representing each document unit as a distinct object This notion sprang from an article submitted to **DL94**[1] by Richard Furuta (see [R 94]) This article expressed the belief that in a digital library, the capacity had to be included to cater for heterogeneous rather than purely homogeneous objects, and it is thus with modern documents Furuta expressed the belief that in the case of the digital library, material prepared using diverse composition systems, and using a range of structures to contain their information, would have to be inter-linked to form the whole information space He says "Heterogeneous data structures may be used to describe different elements of an information space When multiple structures are defined over a set of contents, the general question is whether they are interrelated in any way" [R 94] This paradigm can be applied to the universe of technical documents, where many diverse elements (textual, mathematical and tabular objects to list but three) are combined to present an author's material

The model used in the TechRead system, had to be designed with this view in mind The heterogeneous inter-linking described by Furuta is extremely relevant in this genre, where heterogeneous structures must not only interconnect, but also can be recursively defined For example, a mathematical expression

[1]The first anyal conference on the theory and practice of digital libraries

can contain sub-expressions, or a sectional unit can contain several subsections The model used in the TechRead system must have the following characteristics

1 be capable of containing the various distinct object types found in modern documents

2 be flexible enough to be extensible to cater for new objects

3 be robust enough to incorporate the facility to move to and from heterogeneous objects

4 be capable of use in conjunction with the browser as an aid to rapid perusal of the document

With these criteria in mind, a structure was sought which would fulfill all the needs of the system As was stated in Section 2 2, the most traditional means of internally representing documents is via a standard tree-like structure This was rejected on the basis that it did not offer sufficient scope for rapid browsing Raman pointed out that it is possible to explore a document in a series of atomic steps, which can be reduced to a sequence of tree-traversal techniques However, we believe that this method of representation would place an unnecessary extra burden on the user

Another method which was discarded early on was the notion of modeling the document as a series of related, interconnected linked lists This was also believed to be unnecessarily complex The paradigm used in this model, would have necessitated a list of elements which constituted a list of document objects at that hierarchic level For example, at the uppermost level, a list of sectional units would have been instantiated, the head of which contained the titular information of the document, and which pointed to the upper level sectional units These units would then point to the head of further lists of subordinate nodes, comprising subsectional units of some sort, until a final list of word nodes was generated

The complexity of programming this task was considered too great, and the resulting benefits too few for it to be used in the system The browsing strategies (such as navigating through the hierarchy) would have proved too cumbersome from the perspective of the user as they would be returned to the beginning of each list as they ascended or descended through the hierarchy

As a consequence, the notion of modeling the document as a graph arose We believe that this form of data structure offers many of the capabilities necessary to incorporate the features of modern typesetting

Connectivity Using this model, it is possible to successfully and easily interconnect elements of different data types in a pattern which can be used as an aid to successful and efficient browsing Unlike the tree, where one is confined to linkage downwards through various hierarchic levels, use of the graph structure has innate characteristics, that nodes can be linked by edges in many different directions

Browsing Strategies It is possible to devise browsing strategies, on the basis of which the user can easily peruse a document This use of the graph structure, by virtue of the use of edges to link nodes offers the scope to include such facilities as the rapid examination of footnotes, the following of cross-referential links etc

Extensible It is feasible to extend this model at a later date to include such elements as audio depictions of highly graphical information The representation of these objects is beyond the scope of this discussion As the technological advances proceed, more and more information on the methodologies necessary for the depiction of these specialised forms of material will become commonly available

For example, it is possible to link objects consisting of mathematical information, to both tabular and textual material, without loosing any of the unique characteristics of each Also, by modeling each component of the document as

a distinct entity, it is possible to use the features of the `Object-Oriented Paradigm` to assist in the generation of the internal representation of not only the data, but the logical structure which the author has placed in their document Through the use of inheritance, it has proved possible to model all sectional units based on a single base class Chapters have roughly the same characteristics as paragraphs, save that the former has a title, whereas the latter does not Using the `Object-Oriented Paradigm`, it was possible to produce a generic class `section` and to derive other classes from this, which represented the unique features needed to cater for the different forms of internal node necessary to fulfill the needs of the model

3.1.1 Model description

Before embarking on a description of each of the currently represented content types, it is first necessary to describe the overall structure of the model in terms of the actual technical details of the linkage As was stated above, it is intended to internally represent the content and structure of the document using a graph, though for the purposes of this discussion it will be described in terms of a `cross-linked tree`

In order to understand the nature of each of the different object types contained in the TechRead model, it is first necessary to show how the model is structured At the root of the tree, there exists a node known as *global document settings* node This node contains all definitions and assignments which are deemed to be global to the entire document This could include such items as the default font used, the default speaking voice used throughout the document etc At a logical level below this are found the nodes containing sectional units

Even the most rudimentarily marked-up document contains a degree of structure It is rare indeed to find an author who does not use paragraph units and it is impossible to produce *meaningful* output without the use of sen-

tence structures Therefore, no matter what the document, there will be at least one sectional unit at a lower hierarchic level than the "global settings" node

Further down the hierarchy than the sectional nodes, are found the actual *terminal nodes* of the model These nodes contain the content, and the associated formatting, both visual and audio Thus we have three types of node, namely the *global settings* node, the *sectional unit nodes*, (henceforth known as non-terminal nodes) and the *terminal nodes* which contain the actual text and attributes of the document

The links between these nodes form an integral part of the model used in the TechRead system In order to understand how the heterogeneous collection of entities are tied together to form the overall structure, it is first necessary to introduce the various operations which can be performed on each node Given a node N, then the following are possible

$p(N)$ returns the parent of node n This operator returns a single node

$c(N)$ returns the children (if any) of node N This operator can return [2]

1 a list of nodes at a lower hierarchic level than N

2 a single node at a lower hierarchic level than N

3 the empty set, implying that N has no links to nodes at a lower level

$next(N)$ returns the next node at the same hierarchic level to the current one

$prev(N)$ returns the previous node at the same hierarchic level to the current one

$s(N)$ returns a complete list of the siblings of node N That is, it yields a complet set of nodes at the same hierarchic level

[2]It should be noted that this operation is only possible for internal nodes, as terminal (leaf) nodes are at the lowest level possible in the model, and hence cannot return nodes at a still lower level

$l(N)$ returns a node pointed to by N, which my not be a sibling or a parent This operation is included to facilitate the heterogeneous linkage which forms an integral part of the overall internal model

$t(N)$ Returns the type of the given node This operation is used in determining browsing strategies

The terms *parent*, *children* and *sibling* used above are intended to denote objects at the respective levels The *parent* of node N is the sectional unit at the hierarchic level above, while the *siblings* are those sectional units at the same level within the document Thus, the $p(N)$ operation returns the parent of a given node This can be either a sectional unit, or the *global settings* node described previously The $c(N)$ operation returns the children of the given node These children do not need to be homogeneously defined nodes, but can be of all object types contained within the model The $next(N)$ and $prev(N)$ operations are as they suggest, simple actions which return the nodes immediately adjacent to the one on which the current focus is located This is useful for moving through a document in steps of individual units For example, should the user wish to proceed from section to section, then it would be possible to retrieve the sectional units immediately adjacent to the current node both easily and efficiently

The need for $s(N)$ may not be so readily apparent The reader may well ask "could the same information not be given by repeated calls to $n(N)$ and $p(N)$?" This is indeed the case However, it was felt that an operator yielding all of the current siblings of a given node would be useful The reason for its inclusion is to facilitate the continuous reading strategy discussed in Section 3 2 If a user is reading through a sectional unit, it could be extremely useful to obtain all the siblings of the paragraph units within that sectional unit Consequently, a list is ready at hand, which the software can utilise to offer more efficient browsing

Another instance of where this could be extremely useful is in the case where the reader executes a "read current paragraph" command Then the

point of focus could be established at the commencement of the text, and an $s(N)$ operation performed. Thus, the entire list of word nodes contained within the sectional unit could be retrieved.

Another application for the $s(N)$ operator is the situation where the reader has reached the end of the final paragraph in a given sectional unit, and wishes to continuously read through to the next sectional unit. Logically, the next paragraph is at the same level as the current one (though this might not be the case in tree traversal terms), so the $s(N)$ operator can retrieve this quickly. This facility was incorporated into the model to facilitate a *read whole document* command.

Unlike the tree-traversal techniques described in 2.2.2, the movements up and down the tree are redundant. The model can cater for rapid and easy navigation in all directions through the document structure and content. Therefore, it can be said that the TechRead system represents the document as a series of independent, though integrated objects, and contains the ability to navigate from any point to any point within the overall hierarchy. The reading strategies to enable such rapid movement will be discussed in Section 3.2.

In order to facilitate this type of browsing, the operator $l(N)$ was introduced. As was stated in Section 2.2.2, one of the necessary features of any document browsing system is the ability to navigate to such entities as footnotes, and portions of the document referenced by cross-referential links. LaTeX contains the syntactic elements to include such features as footnotes and cross- references in documents, and hence the model used in the TechRead system must cater for them. Accordingly, the $l(N)$ operator will enable the user to jump to a footnote or cross-referenced object and to return to the point from which they left. As was stated above, its purpose is to afford the user the facility to move to other portions of the document in a manner similar to that of the sighted user glancing at another part of the text, and then returning. It could be argued that the use of the $l(N)$ operator is introducing the notion of `hypertext` navigational capabilities into documents which were not intended for this purpose, though

this is not the case Though it is envisaged that the model can be extended to cater for **hypertext** based documents, at present the intention is simply to provide the means to navigate to any point of interest in the document

It is worth stating at this juncture that the links used in the TechRead document model are all bi-directional Therefore, the definition of the model as a graph should be qualified to add that it is a directed graph By virtue of this bi-directionality, the user can return to a given point after perusing other such links as cross-references or footnotes For example, let us suppose the user went to examine a footnote or margin note It is probable that they would wish to return to the point in the document from which they left, hence the use of bi-directional linkage

The final operator provided in the TechRead model is the $t(N)$ operator This, as was stated above, returns the type of any given node The operator examines a list of known object types, and if the current document component fits into these categories then it is a *known object* and hence will have browsing strategies associated with it If it is not found in the list of object types, then it is deemed to be a *new object* Several methodologies are available to incorporate this into the document model Firstly, the user could be shown a presentation of the object and asked how they wish it to be translated Alternatively, the actual mark-up of the object itself can be examined by the system, to verify that it can be included in the model For example, at the time of writing it will not be possible to include such objects as pictures in the model and hence they will be ignored The mark-up of the LaTeX documents would reveal the use of the commands necessary to produce pictures, and hence they could be discarded Alternatively, the user could be shown some form of rendering of the picture and asked how they wished it to be designated within the documents' spoken version

It is by use of this technique, that LaTeX macros will be incorporated into the model The macro capabilities of the LaTeX mark-up language allow user-defined types and presentations to be used instead of the defaults provided by

the package itself Instead of displaying sectional titles in their present form, an overriding command could be issued to alter the font, and display each Section title with the characters reversed TechRead must cater for such extensions, hence the use of $t(N)$ to allow the object types recognised by the system to be increased in number

An extremely important component of the internal representation, is the linkage which exists between various heterogeneous types of node ASTER [Ram94] used the traditional tree-based linkage, supplemented by minimal interconnection of siblings TechRead utilises the functionality provided by graphs to provide links not only from level to level within the hierarchy, but across the same level Also, using the $l(N)$ operator, it is convenient to inter-connect nodes not only at diverse levels, but located in different parts of the document The bi-directional edges connecting downward through the hierarchy provide links from the higher-order sectional units, to those subordinate units which are contained within them This affords easy movement in the vertical direction through the various levels of the document structure

Within each sectional unit, each non-terminal node is linked to its neighbours at the same hierarchic level This is to facilitate the use of the $n(N)$ and $p(N)$ operators which, as described previously, return the next and previous nodes to the one which currently has the focus The $s(N)$ operator also uses these sibling-links to return the list of sectional units at a given hierarchic level

Unlike the traditional tree-based depiction of document structure, TechRead's model utilises linkage between sections To place this notion in the context of a tree-like description, there will exist links within the document which interconnect various branches of the model The reasons for the inclusion of these extra links should be apparent If they were not incorporated, the user would have to perform the following actions to move from Section to Section within a document

1 proceed up the hierarchy until a common parent is reached

2 descend along the next branch

3 verify that this was the correct place

4 re-commence reading

The use of the $n(N)$, $p(N)$ and $s(N)$ operators to provide this continuity between sectional units, negates the need for the tree-traversal browsing actions Using the inter-connectivity provided for in the model, the user simply reads at the same level It should be stated that the final subordinate unit in one section, when connecting to the first in the next section, need not do so at the same absolute level Consequently, if the final paragraph in Section 1 actually occurs in Section 1 3 1, while the first in Section 2 is found to be in Section 2 1, then the link will exist between them as though they were on the same level It is with this type of link that the departure from the tree-based approach, so commonly found in document modeling becomes apparent The user is actually ascending several levels through the document structure, but in essence they are staying within the next available container unit

Another feature of the connectivity of the document model is that the edges do not need to join two nodes of the same type As each object in the document model is essentially an independent entity, it can be connected to other entities quite easily without loosing any of its own innate characteristics The $n(N)$ operator will return the next node to the current one, without distinguishing its type The browsing strategies pertaining to that node are derived from the $t(N)$ operator It is thus, that the heterogeneous needs outlined by Furuta [R 94] are catered for within this document representation

Thus far, the explanation has been confined to an overall description of the document model and the means used to interconnect the nodes of various heterogeneous types The following sections describe the characteristics of the various nodes found as default types within the system

3.1.2 Representing text

Though it is said that "a picture is better than a thousand words", the majority of authors use the written word rather than the image to convey the greater part of their ideas As a consequence, the first object type which was introduced to the internal representation used in TechRead was the *text object* Authors use various means of distinguishing the importance of portions of their documents such as ALTERATIONS IN THE VISUAL ATTRIBUTES and our object had to include both data members to store, and methods to retrieve this information

Authors also impose a structure on their documents, grouping those sentences and paragraphs together into larger container sectional units It emerged that it was possible to represent the textual content of documents in a similar manner Consequently, it was decided to store the actual text of each word in **terminal nodes** and to associate both visual and audio formatting with each textual node, hence keeping the content and formatting together in a single entity The example "Give me that spoon" would be stored as a series of terminal nodes, all homogeneously containing the same visual attributes Should the author, on the other hand, wish to convey a degree of emphasis, as in "give me **that** spoon" the word "that" would in all respects be similar to the rest except in that the visual attributes (and consequently the audio counterpart) would be different

Therefore, there are now three data members associated with the terminal node **text**, namely the actual content, and the audio and visual formatting The obvious question is the nature of the overhead required to store such redundant information, should a lengthy passage of text occur which in some manner stands out from the main body of material contained within the document Hence it evolved that in this event, the starting and ending points of the visually enhanced text would be marked, but the intervening words would be left unaltered This evolution proved extremely useful when deriving spoken output from the model, as it was possible to both efficiently and rapidly deduce

where to alter the vocal characteristics to reflect some visual change, and where to reset the same features to indicate a return to normally displayed textual material.

Comparisons can be made between the linear nature of the LaTeX mark-up and the prosodic changes used to convey aurally what is indicated visually by this syntax. Spoken output is a serial mode of presentation; and it can be argued that LaTeX offers a means to serially produce visual alterations which are inherently not serial in any respect. What this model facilitates therefore, is the ability to render one serial form of presentation, from another serial (linear) in nature. The use of delimiters such as {...} to indicate the beginning and end of altered text is useful, as it can also be utilised to mark places where the prosodic characteristics of the speech must be altered to reflect these visual changes.

The means of holding the formatting information is by using `C` type `struct` data types. Investigation revealed that the visual attributes used by `MS Windows` programmers to store and manipulate font information is via a `LOGFONT` structure. Table 3.1 introduces the data members of this structure. Proceeding from this, it seemed logical to construct a similar `C` type structure to hold the vocal attributes for the text. Table 3.2 shows the various features which the TechRead system alters to reflect audibly the visual changes in the document.

At present, not all of the fields described in Table 3.1 are utilised, as LaTeX does not use as many font attributes as are to be found in this structure. As will also be observed, there are fewer elements in Table 3.2. The reason for this is that the number of prosodic alterations to produce the audio equivalent to a visual font change are fewer than those necessary for controlling the appearance of the text. There are myriad ways in which visual appearance can be altered, but only a small subset of changes are made to the vocal characteristics to convey such alterations. The reason for this is that individual aspects of the visual representation can be combined in an almost infinite manner, to produce new and (sometimes) interesting effects. Audio-based views of the same information

Field	Description
`lfHeight`	Weight of font in logical units
`lfWidth`	Width of font in logical units
`lfEscapement`	Angle at which to draw the text
`lfOrientation`	Character tilt in tenths of a degree
`lfWeight`	Font weight
`lfItalic`	A nonzero value indicates italics
`lfUnderline`	A nonzero value indicates an underlined font
`lfStrikeOut`	A nonzero value indicates a strikethrough font
`lfCharSet`	Font character set
`lfOutPrecision`	How to match requested font to actual font
`lfClipPrecision`	How to clip characters that run over clip area
`lfQuality`	Print quality of the font
`lfPitchAndFamily`	Pitch and font family
`lfFaceName`	Typeface name

Table 3 1 `LOGFONT` fields and their descriptions

Characteristic	Explanation
Rate	The number of words spoken in 1 minute
Average Pitch f_0	The fundamental frequency of the voice
Pitch Range	The percentage range which the voice can vary around the fundamental frequency

Table 3 2 The vocal characteristics altered by TechRead Includes a brief, and unscientific definition of each one

do not work in the same manner. For example, if some aspects of spoken output are altered by too great an extent, the synthesiser can become overloaded, resulting in strange, unintelligible squawking noises.

Figure 3.1 demonstrates the means whereby homogeneously formatted words are catered for in the TechRead document model. The method used to depict the insertion is based on the state of the structure containing a given word; showing its attributes and links. Figure 3.2 shows how the model is altered to reflect the changes in visual formatting. Both Figures, 3.1 and 3.2, are based on a mixture of `C` syntax, and an English-like pseudo-code to convey the state of the textual content of the model. In Figure 3.2, it can be seen how the audio and visual formatting are altered at the point where a change occurs, while if viewed in conjunction with Figure 3.1, it can be inferred how the formatting is re-set to the defaults when the enhanced portion of material has been completed. In both Figures, `C`-style comments are used to improve the readability of the code. These comments are indicated by the character sequence '//".

3.1.3 Representing mathematical content

The means whereby mathematical content is presented in printed documents, is by using the juxtaposition of various symbols, coupled with their spatial orientation relative to one another to convey the semantics of the expression. In order to produce **meaningful** audio representations of the equations, the internal representation used in the TechRead system must offer facilities for the user to be able to browse through both the vertical and horizontal dimensions of the equation.

As an example, consider the simple equation:

$$x = \frac{a+b}{c+d} \tag{3.1}$$

which demonstrates both horizontal and vertical alignment. Here we have a simple quantity x a relational operator $=$ and a fraction $\frac{a+b}{c+d}$. The user needs

```
start
    WordStruct word = "hello!", //place the string in the word
    node
    WordStruct Font = Par Struct;  //use default formatting
    WordStruct Audio = Par Audio;  //Use the default paragraph voice
    WordStruct NextWord -> NextWordStruct;  //Point the current word
                                            //to the next one
    if (WordStruct ISNOT FirstWordStruct) //If there are no prior words,
        WordStruct PrevWord -> PrevWordStruct,
        //Point word to previous one in list
    EndIf
    Par AddToChildren(WordStruct),
    //Add the current word to the list of children of the paragraph
    WordStruct ParPointer ->Par,  //Point word up to its parent
end
```

Figure 3 1 The attributes and linkage of a structure containing a word which utilises the default formatting of the document It can be seen how the word inherits the font information from the paragraph in which it is contained

```
start
    WordStruct.word = "hello!"; //place the string
                                //in the word node
    WordStruct Font SetEmphasis ("Italic", OR "Underline" OR
    "Bold");  // Set the emphasis
    WordStruct Audio SetVoice ("Emphasis");  // Adjust audio
              //parameters for the word to reflect emphasis
    WordStruct NextWord -> NextWordStruct;  //Point the current word
                                            //to the next one
    if (WordStruct ISNOT FirstWordStruct) //If there are no prior words,
        WordStruct PrevWord -> PrevWordStruct,
        //Point word to previous one in list
    EndIf
    Par AddToChildren(WordStruct),
    //Add the current word to the list of children of the paragraph
    Par HasChange = TRUE;  //Set a flag in paraph node to indicate
                           //change has occurred at word level
    WordStruct ParPointer ->Par,  //Point word up to its parent
end
```

Figure 3 2 A structure which contains the alterations in formatting to convey emphasis It can be seen how the word uses its own formatting, and does not inherit from the default formating of the document

to have in-built capabilities to browse each of these three components of the equation Further, the fractional component can be reduced into a numerator and denominator, essentially four individual terms to this simple formula

Consequently, the mathematical object (designated `MathObject` in this discussion) had to be designed with the following criteria in mind

1 the facility to allow rapid browsing

2 the flexibility to be extended

3 the characteristic of allowing each term, or combinations of terms to be broken down easily

For these reasons, the `attributed tree` and `quasi-prefix` notation [Ram94] (see Section 2 2 2) were rejected As stated previously, it was believed that though the model used in ASTER can be learned, it is not considered intuitive Hence, the notion of modeling the mathematical formulae as a sub-graph of the overall document was decided By using this form of representation, it proved easy to arrive at keyboard mnemonics to enable the user to rapidly traverse the equations

The mathematical object is connected to the main body of the document by two links Since it is unusual for mathematical content to form a paragraph in, and of itself, it was decided to place the edges interconnecting the formulae with the main body of the document within the paragraph to which they naturally belong Hence, should a paragraph contain the LaTeX commands to denote the start of mathematical material, a link is placed from the paragraph unit to a `MathObject` This is done by simply adding the `MathObject` to the list of children of the given paragraph In order to maintain the standards for the model described in Section 3 1 1, a link must exist from the children of the container paragraph, to the `MathObject` The $l(N)$ operator is adjusted in the terminal text node immediately preceding and following the `MathObject`, to point at this entity The user can thereby employ the reading strategies defined

for this technical object, or decide to skip the object and simply continue to read the actual text of the document

The `MathObject` itself is simply the root node of a sub-tree, which contains the spoken string "Math Equation", and the associated list of vocal characteristics to distinguish it from the surrounding text This `MathObject` also contains pointers to the various terms of the formula The `MathObject` can be said to be a form of sectional unit, which is at a level somewhere between a paragraph and a terminal node [3]

It is intended to provide several levels of equation reading within the system These are

1 announce presence of equation

2 give an overview of the equation

3 give a running paraphrase of the equation

4 permit the fully fledged browsing of the equation

As a result, the model provides all of the functionality to permit these diverse levels of access The `MathObject` is connected downward to a node giving a textual overview of the equation This is a simple node, containing merely the spoken utterance and associated audio formatting, coupled with the necessary links The `OverviewObject` is in turn linked (in a similar manner to the interconnection of the textual content) to the nodes below, which gives the running paraphrase of each term within the equation These `ParaphraseObjects` are similar in nature to the previously defined `OverviewObject` and contain the textual content, and audio formatting of the paraphrased equation The lowest level, consists of standard terminal nodes containing the textual version of each term, the visual formatting to enable its display in standard printed notation, and the audio formatting necessary for spoken output The standard operations

[3] If sentences were included in the model then it would probably be at this level

described above are always possible on each of these nodes, thus enabling the user to rapidly return to the textual content of the document

As can be seen from the above description, the `MathObject` consists of a sub-graph within the overall document model The reason for this is that different reading strategies apply when reading mathematical, as opposed to textual content As was discussed in Section 2 1, the visual reading process slows down when the absorption of highly technical or syntactically complex material is encountered It is similar with listening A greater degree of control is required over the information flow to enable both the comprehension and perusal of this highly technical material The mathematical graph described in this Section offers this control, as will be explored in section 3 2

In order to appreciate how the various levels of the mathematical sub-graph interconnect with the remainder of the document, an example is required A sample equation $b + \frac{x=c}{d} - e$ is used in the following Figures to illustrate how an equation is added to the model Figure 3 3 announces the presence of the equation, and illustrates the linkage between the mathematical sub-graph and the remainder of the document Figure 3 4 gives a running paraphrase of the equation, while Figure 3 5 shows how terms are added to the model Once again, a pseudo-code syntax is used to represent the instantiation of the various attributes, and the interconnection of the various nodes of the graph Comments are indicated by C-like syntax

3.1.4 Representing tabular data

Though the mathematical content of technical documents poses significant problems to blind users, tabular data, being highly visual in its organisation can pose equally frustrating, though different problems The fact that related information is arranged in vertical columns set in rows related to each other by some common feature, necessitates a totally different set of browsing criteria Consequently, the model of the document used by TechRead emulates

```
start
    OverView_struct string = "Equation present   ",  //Set up the
                                                    //announcement
    OverviewStruct PrevPointer -> PrevWordStruct;  //Link math
                                    //object with preceding word
    OverviewStruct Nextptr -> NextWordStruct, //Link to next word
                                                    //struct
    Par AddToChildren (OverviewStruct),  //Add overview to list of
                                        //paragraph children
    OverviewStruct Parptr -> Par,  //Link upwards to parent of
                                    //OverviewStruct
end
```

Figure 3 3 Announces the presence of a mathematical equation to the listener Also demonstrates the linkage to the textual objects before and after

```
start
    ParaphraseStruct string = "x equals b plus a fraction minus
    d";  //Store the running paraphrase text
    while (not EndOfTerms) //process all terms
        ParaphraseStruct AddToChildren (term),  // Add each term
                                            //to the list of children
        next term,  //Process the next term
    EndWhile,
    ParaphraseStruct Parent -> OverviewStruct;  //Link upwards to
                                                    //parent node
    OverviewStruct child -> ParaphraseStruct, //Link downward to
                                                    //lower level
end
```

Figure 3 4 An overview of a mathematical equation Gives a running paraphrase of the equation

```
start
    while (NOT EndOfTerms) //While there are still terms to
                           //process
        TermStruct Word = GetTerm(),  // Get the term and store it in
                                      //the string attribute
        TermStruct Audio = SetAudio(term),  //Set the audio
                                            //formatting for the term
        TermStruct Font = SetFont(term),  //Alter font family and
                           //appearance to depict the term
        TermStruct Nextptr = NexttermStruct,  //Point to next term
        if (TermStruct ISNOT FirstTermStruct) //If it is not the
                                         //first term in equation
            TermStruct Prevptr -> PrevTermStruct;  //Point to
                                                   //previous term
        EndIf
        TermStruct Parent -> ParaPhraseStruct;  //Point upwards to
                                                //parent
        next term;  //Process next term
    EndWhile
end
```

Figure 3 5 Shows how one term of the equation is stored in the model This can be extended to all terms

the structure of the table, by constructing a series of nodes arranged in the same pattern as the visual table An overview level is included, which simply announces the presence of tabular data, and indicates the number of rows and columns which can be found therein

As with the mathematical object `MathObject`, the tabular information is linked to the paragraph in which it is contained However, it is "floated" to the end of this sectional unit, and is linked to the text at the point of the final word node within the paragraph The reason for this is simply that LaTeX, when visually typesetting documents, does not permit the insertion of a page break in tabular data, unless absolutely necessary The audio equivalent is to offer the reader the chance to peruse the textual data prior to reading the table, and then to present the tabular information when the reading of the text is completed It should be stated that this is in keeping with the means LaTeX uses to deal with several diverse environments

It is customary to visually "float" various types of information to the ends of sectional units within a document It is not unusual for example, to float a Figure or table onto the top of an adjoining page when there is insufficient space for its inclusion on the one to which it applies Consequently, the object containing the overview utterance "table containing x rows and Y columns" is linked both upward in the hierarchy to the paragraph node, and horizontally the last terminal node contained therein Should more than one table be present in a paragraph, then they are linked in order of appearance

Below this `TableObject` is found the actual contents of the table The mark-up of the tabular environment can be examined and, by virtue of various characteristics attributed to the first row, it can be determined whether this is a *header row* The means to deduce this fact is that it is customary for authors to separate the header information from the main body of the tabular information by using vertical lines, drawn to segregate this explanatory information from the rest This has significance when browsing the table, as will be seen in Section 3 2 A simple sub-graph is produced, linking each node to its neighbours in

as many of the cardinal directions as is appropriate Consequently, in a table containing three rows and three columns, (nine cells in all) the center cell (row 2 column two) is linked to the two cells adjacent to it in the same row, while also being connected to those immediately above and below it

It is believed that the connection of each cell to those adjacent to it, will enable the user of the TechRead system to quickly gain an impression of the contents of the data, and the relationship between the various cells will be unambiguous Also, since the mark-up can be examined and an effort made to determine the nature of any headings which apply to the various columns, this information will also be made available With this information, it should prove possible for the user to determine extremely rapidly what the contents of any given column are, and more importantly, appreciate their semantic meaning in the context of the other elements within the table

3.2 Reading strategies

Before embarking on a description of the actual interface which the user will employ to gain the maximum information from the TechRead system, it is first necessary to describe the forms of reading which **must** be catered for by the system The physical processes involved in both visual and audio-based reading were described in Section 2 1 This portion of the description of the TechRead system introduces the various reading strategies built into the system

3.2.1 Continuous reading

The most obvious form of reading is to simply start at the beginning of a document, and proceed in a sequential manner to the end The first strategy, therefore, which must be catered for in the design of any document reading system is the ability to provide the user with the functionality to read continuously from a starting point to an ending point, without interruption or over

several discrete periods of time The user could begin reading through a technical document, and proceed to Section 3 2 At this juncture, they could stop reading and, at some other time, return to the document and take up reading from the point at which they ceased providing this functionality necessitates two distinct components, namely the ability to send text and formatting commands to the output device and also the ability to mark positions within the document The first of these facilities is relatively trivial

The functionality required to communicate with a synthetic speech device is an inherent part of the system, hence all that is necessary to fulfill the technical aspects of this strategy is the ability to ensure that the same text is not sent twice, and also that no text is lost (see Chapter 6 for the techmcal details of communicating with speech synthesisers) Consequently, we arrive at the notion of book-marks

The TechRead system will incorporate the ability to store, and return to marked locations within the document This is akin to the facility in various `Internet` browsers, where users can store the address of their favourite Internet sites and return to them at a later date The mechamsm used to store the book-marks will be to use the idea of *labels* in the same manner as they are used in LaTeX, though they are used for entirely different purposes Should the user wish to add a bookmark, a simple `Dialog box` will appear, permitting them to enter the label which they wish to associate with a given point in the document This facility, coupled with the ability to continuously send text to the speech synthesiser (see chapter 6 for the methodologies employed in communicating with the synthetic speech devices), allows the user to read the document continuously in the same manner as their sighted colleagues

The internal model of the document caters for this form of reading admirably The graph-based linkage throughout the document allows the user to simply press the *start reading* key, and, through the various links described in the previous section, the information can be rapidly and easily conveyed to the listener An example will suffice, to illustrate the efficiency of the model in

terms of the continuous reading strategy Let us assume that a relatively simple document is being perused by the listener, consisting of a title, abstract, and three sectional units The two forms of continuous reading can be illustrated by the following description of the technicalities of proceeding through the document model, and passing the content and formatting to the synthetic speech device

The user initiates a *start reading* command at the top of the document At this point, the current focus is on the title of the document, which is duly read, followed by the author's name, and any other titular information, all of which is found in the series of terminal nodes pointed to by the *global settings* node (see Section 3 1) The first branch of the graph, leads to the abstract This is a sectional unit in its own right, though the formatting may be different The vocal characteristics are altered to reflect the visual changes in the text. Following from this, the focus drops down a level, to the paragraph units, and thence to the actual words of the abstract

A link also exists from the paragraph units to their siblings contained in alternative sectional umts Therefore, when the abstract has been completed, the focus simply moves via the $l(N)$ operator described above, to the title of the first sectional unit The same process is then repeated until the document has been fully read Should the user wish to cease reading at any time, the *stop reading* key is pressed, at which point the option to store a bookmark is available Should the user wish to do so, a key press can cause the `dialog box` to appear, into which the label associated with the book-marked location can be stored This is then saved, along with the point in the document to which it relates, thereby permitting the user to return to the position at which they stopped reading at a later date It should also be pointed out, that the ability to store many book-marks pertinent to the same document will be incorporated into the system

3.2.2 Skim-reading

One of the alternative means of proceeding through the contents of documents to continuous reading, is that of perusing the titles of the various sectional units, to determine the relevance of each to the interests of the reader Another is the so-called *speed reading*, whereby the sighted reader scans rapidly through the text, absorbing a superficial knowledge of the content, without concentrating on the finer points The following paragraphs describe the means whereby both these reading strategies are achieved by the TechRead system

The first, and most basic means of *skim reading* is to examine the Table of Contents to determine the name and page location of the various sectional units This can be easily achieved within the TechRead system, by use of the internal representation The user can, through various key presses proceed from sectional unit to sectional unit and, if they desire, read this Alternatively, they can proceed to the next Section until they reach the one they wish to read, when they can then employ the reading strategies contained in the interface to extract the content The visual analogy for this form of reading is to simply scan down the page listing such entries, and proceed from thence to the page number on which the relevant material is stored

The notion of *flicking the pages* is immensely more difficult to incorporate into an audio-based representation of technical documents This form of superficial reading, is based on the reader's ability to scan through the printed material, and to pick out the salient portions of information which interest them The features which these portions of text all have in common, is a visual attribute, or a word grouping which stands out by virtue of the fact that the reader is actually seeking information pertinent to this word grouping The visual appearance of the content of the document is a useful aid to those reading at high speeds, as the salient points of interest can be apprehended extremely easily Thus, a form of *summary* can be introduced One method for achieving this, is to examine the mark-up of the document, and by definition of the

visual attribution caused by the mark-up language, produce some summary information relative to each sectional unit. However, this method is fraught with problems. Let us suppose the following example constitutes a portion of a document:

"He broke **that** glass, throwing it through *this* window"

Here, examination of the mark-up would simply summarise the above as "that" and "this", which does not provide a meaningfully accurate superficial view of the paragraph. Thus, an alternative approach had to be found. A means of producing a summary of the document, which provided an overall view of the content was needed. The commonly used approach often found in linguistic analysis software, was incorporated into the system. Here, the document can be examined, and the most commonly used trigrams extracted and given a weighting. Using this weighting, the sentences containing these trigrams can be rendered as a summary version of the document. This method is refined in TechRead to incorporate aspects of the mark-up. A combination of those sentences containing the most common trigrams, and those containing alterations in the mark-up are combined to form an accurate summary of the document. Using this joint approach, it is feasible to extract summary information based on commonly occurring word-groupings within the document, and also those salient features which are so readily observable when read by a sighted reader.

It is realised that this method is not perfect, though it is believed that (certainly when applied to longer documents) it will prove both efficient and easy to use. Through the use of the Table of Contents, and this summary data, the listener can gain a rapid overview of the document structure and content in a similar manner to their sighted colleagues. As was described in Chapter 1, one of the most frustrating and time-consuming aspects of non-visual reading is the need to trawl through often unwanted content. Using the methods described, this will be eliminated, as when the listener reaches either the sectional unit they wish to read, or when a portion of summary text is read, they can commence

reading from that point, using all other strategies contained within the system This summary view is also extremely useful when mathematical material is incorporated into a document

As was described in Section 3 1, the internal representation of document structure and content used in the TechRead system is conducive to the perusal of mathematical material, and it was also shown how the model catered for different *views* of the mathematical expressions In the document summary,if an equation is contained within a sectional unit, its presence is declared This enables the listener to read or ignore the syntactically complex material In a paragraph of text which contains an equation, the phrase "equation found" is spoken This has no bearing on the fact that the same phrase is utilised in continuous reading Rather, it is incorporated into the summary view, to present the listener with the same information as the sighted reader can gain from simply glancing at the content and layout of the document Just as the sighted reader can distinguish the presence of mathematical content by virtue of its spatial location and alternative symbols, so the listener can, through the use of the summary information, deduce the presence of this type of material

3.2.3 Non-sequential reading

An alternative reading strategy to the continuous reading described above, is the possibility for a reader to jump to and from various portions of the document they are reading This requires the ability to note a starting point, find an end point, and (if desired) read from this point on, before returning to the point from which the jump was initiated In LaTeX, the means to incorporate such jumps into documents is by use of the `\ref` command Using this mechanism, the author can direct the reader to other, related portions of their work Thus if a passage of text such as "see Section " occurs within a document, and the "section " is referenced using the command described previously, a link can be established between the section currently being read, and that being referenced

In essence, the incorporation of these jumps, is akin to the notion of introducing a degree of the features of hypertext navigation into the document As a consequence, the facilities used in `Internet Browsers` to cache and retrieve previously referenced material can be utilised Though the visual appearance of the document in the TechRead system will not contain the formatting usually associated with hypertext links, the facility to follow what are essentially *within document* links can be employed The reasons for the introduction of this strategy are best summed up by the fact that it is extremely difficult to manually follow these cross-references, and to return to the point at which the reference is placed in the text Using existing access technology, the easiest, and most efficient means of achieving this is to follow the following algorithm

1 mark the starting point with a unique set of characters (for example //)

2 use the word Processor's "search" facility to find the portion of text referenced

3 read the relevant passage

4 use the "search" facility to find the unique character string used to mark the starting position of the jump, and then to return to the point at which the reference was made

The introduction of the mechanism for following these cross-referential links will help to short-circuit this time-consuming process The availability of this jumping mechanism will, it is hoped, enable the listener to move easily and efficiently from portion to portion of related content within a document If the LaTeX commands designed to allow the inclusion of cross-references are not used, the availability of this type of reading will be seriously diminished Though the two sentences "see Section " and "see section `\ref{ }`" produce the same output on the printed page, it should be noted that, as no semantic examination of the text is performed, the movement from portion to portion will only be possible if the latter method of mark-up is used

Cross-references are not the only means of moving to and from portions of related text Another important mechanism which authors use to impart extra information is *footnotes* The same technique described in the context of cross-references is used to afford the listener the facility to read footnotes if they so wish These notes form a written version of the spoken "aside" That is, they are portions of text which can be read, or ignored at will Therefore, the hypertext like features of linkage can afford the listener the ability to peruse the notes, or ignore them These two examples of non-contiguous reading are included merely as illustrations of the power of this flexible strategy

The model used to internally represent the document's content and structure is flexible enough that, in the future, other such jumps will be possible An obvious addition could be the inclusion of features enabling the user to jump to and from bibliographic entries The LaTeX command `\cite` (used to include citations within a document) could form the starting point of the link, while the bibliographic entry would constitute the arrival point The reason why this feature is not presently included is that a separate program is responsible for the incorporation of the bibliographic entries into the document Consequently, it is necessary to examine the `dvi file` in order to ascertain what entries are contained in the document The TechRead system does not presently examine this file, rather the source is examined

3.2.4 Reading mathematical material

As was shown in Section 2 1, the strategies employed by visual readers to peruse syntactically complex material is quite different from those used when reading textual material The reading strategies used within the TechRead system must cater for this increased level of control needed to gain a full understanding of the technical content Consequently, the need to incorporate features into the system which offer the user the degree of control over the information flowing past them was imperative

The ability to reduce mathematical expressions to their component sub-expressions is a fundamental part of the reading process used in the TechRead system To begin with a simple example, the formula $\frac{(a+b)}{(c+d)}$ can be reduced to two components initially, namely the numerator and denominator of the fraction The listener who hears this simple expression flowing past them, may need to peruse each individual sub-term of the expression This decomposition necessitates mechanisms which enable the user to peruse each component of the expression at will

In Section 3 1, the mathematical objects contained in the model were described, and how these objects were essentially sub-graphs within the overall representation of the document was illustrated The strategies used in TechRead utilise this representation, by permitting the user to navigate through all levels of the mathematical content At a superficial level, the reader can firstly deduce that the expression described above may be decomposed into the numerator and denominator of a fraction, while at a lower, more in-depth viewpoint it can be observed that the sub-expressions themselves can be further reduced to two quantities, separated by the relational operator +

As the vertical alignment, and use of white space can be used in printed mathematics to delimit sub expressions, so the mark-up of LaTeX can be utilised to deduce the same relationships It can be observed from this simple expression, that the need for decomposition of the overall content is paramount The reading strategies used in the system to peruse mathematical material must therefore contain the following attributes

- be capable of allowing the user to gain access to the different views of mathematics described in Section 3 1 3
- the ability to decompose mathematical expressions into sub-expressions
- be capable of affording the user the flexibility to jump to and from any component of the expression

- afford the user the facility to ignore the expression entirely

The ability to gain access to the various views of mathematical information is perhaps the most important strategy catered for in this work Sighted users can distinguish the presence of highly complex material by virtue of layout, and other presentational cues which form an inherent part of the printed mathematics Their perusal of the document can be determined by the type of content they wish to read For example, some readers of a highly specialised nature will simply wish to peruse the document and extract the mathematical content found therein, while others could wish to ignore this type of material entirely It is not envisaged that the ability to peruse merely the mathematical content of a document be a feature of the first version of TechRead, however the ability to ignore mathematical content will be an integral part of the system Alternatively, should the reader wish to peruse this type of material then the strategies of decomposition discussed in the previous paragraphs can be employed

The browsing strategy which caters for mathematical content is based on the various views of the data described previously The first, or overview level is akin to a sighted reader glancing at a passage of text and determining the presence of mathematical material The next level (known as the running paraphrase) describes the equation in terms of the significant operators contained within it This is incorporated within TechRead to provide the listener with the basic idea of what the expression contains, and is intended to simulate the sighted reader's ability to rapidly determine the key points of the mathematics The third, and most comprehensive view of the formulae is that which enables the listener to browse the information in detail This view is present to facilitate the decomposition of the expression into the constituent parts for the purposes of in-depth analysis

The notion of navigating from term to term within an expression is an important one An example of where this can be utilised is in the following

expression

$$\frac{a+\sqrt{c+d}}{x^2yz} \tag{3 2}$$

At an overview level, the system merely reports that the equation is present The running paraphrase of this expression would inform the listener as to the nature of the material viz *fractional expression found Numerator contains a square root, denominator contains a superscripted quantity* At the lowest level the importance of the mathematical browsing strategies is demonstrated The numerator of this expression can be further reduced to two terms, namely a quantity "a" followed by the remainder of the sub-expression which is encompassed by the $\sqrt{}$ By contrast, the denominator consists of a superscripted element, followed by another juxtaposed quantity Using traditional methods of presentation, this expression (though simple in nature) would prove quite difficult to remember when presented aurally A sample textual version of this formula could sound like

> "paren a plus the square root of b+c paren over paren x squared y z paren"

The addition of lexical cues to this sample text does not remove the ambiguities which are an innate part of even this level of expression It is not readily observable, for example, whether the radical symbol encompasses both quantities in the numerator of the fraction, nor is it clear whether the "z" forms part of the superscript Such ambiguities must be removed from the utterance to present the listener with spoken mathematics (see Section 5 3) However, the interface plays an important part in clarifying the semantics of the expression

The system must incorporate features to allow the user to determine what elements a superscript refers to, or which components are contained with a radical Combined with this notion, the facilities must be present to navigate to and from any component in the expression For example, if the reader were

perusing the contents of the square root, they should be able, at the press of a button, to jump immediately to the superscripted elements contained within another part of the equation

The reasons for this strategy are that visually it is easily discernible what elements are related to each other Through the use of vertical juxtaposition, and other spatial orientation, the reader can immediately deduce that the radical symbol encapsulates the latter two elements of the above example, while the "y" is the only quantity which forms a superscript The vertical alignment of the fraction also reveals which is the numerator and which is the denomenator, thus providing the visual cues needed to parse the expressions Luckily, LaTeX contains the syntax to unambiguously mark-up the mathematical content found in most technical documents Examination of the source of the documents can reveal the points at which to parse the expression, and thereby provide the listener with the means to gain a rapid and efficient presentation of this complex type of content

3.2.5 Reading tables

One of the most difficult forms of visual presentation to convey through the medium of speech is the table where data is presented in both vertical and horizontal alignment, relating each element by some common factor It is immediately observable that column N in row M is related to column $N-1$ in row m by some commonly held attribute, while column N in rows M and $M-1$ are comparable by virtue of some other common feature The vertical and horizontal juxtaposition of the rows and columns within the table, provide the visual cues necessary for their interpretation

It is these visual associations which must be conveyed to the listener, hence the need for a vastly different reading strategy The user must be able to

1 view the current cell in the table

2 examine the cells which comprise the neighbours of the currently selected element

3 be able to deduce the common feature which links the various rows and columns

4 where possible, be in a position to discern the header information to deduce the semantic interpretation to place on the data

It is only recently that generic screen access technology has begun to incorporate methods to enable the reading of tables The preferred method, is to use the *tab* key to move from cell to cell, thereby offering a linear mode of access This can often ensure that the relationship between the various tabular elements is unclear As will be seen in Section 3 3 the use of the *numeric keypad* provides several features which can be harnessed to provide solutions to these problems

The most important cells relative to any given tabular element are those to its left and right, and those immediately above and below Moreover, a common practice in the presentation of tabular data is to use a "header row" or "description column" These particular tabular elements are extremely useful when trying to form a common link between the various items Therefore, while the reading strategy must incorporate the ability to present the user with the cells immediately adjacent to a currently selected one, it has also been decided to incorporate the ability to determine what the topmost element in any given column consists of, coupled with the data in the leftmost column of each row

It is fully realised that this solution is not perfect Firstly, there is no means of determining that the top row, or left column of any given tabular object consists of so called *header* information, save through the examination of the mark-up It is realised that the provision of these facilities will often be redundant, though the use of the *nearest neighbour*, and the *header information* should make the comprehension of tabular data easier

3.2.6 Reading newly defined objects

As was described in Section 3 1 1, the LaTeX language provides the facilities to add newly defined commands and environments to the existing set Consequently, the TechRead system must also cater for, and where possible, incorporate these facilities With this in mind, the flexibility to add the keyboard mnemomcs for newly defined objects must be built in to the software [4] We envisage that some form of script language be utilised to relate the keyboard mapping to the internal representation of the document Using this mechamsm, TechRead could be constantly expanded by its users to cater for their own specific needs The strategies for newly defined objects **may** not be known On the other hand, if the mark-up of the newly defined objects is examined by the TechRead system, it could be possible to deduce the reading strategy which best suits the newly defined command or environment This is the method employed in the initial phase of design Through perusal of the LaTeX source, it can be determined what type of object is being introduced Without much semantic analysis of the document, in some instances it will simply not be feasible to determine the appropriate strategy to apply to these user-defined objects

3.3 The human interface

To date in this chapter, it has been demonstrated how the document will be modeled internally Also, the strategies built in to TechRead to enable rapid perusal of both the content and structure have been introduced The following sections describe how the system will appear to those who actually utilise its many features It is intended to show how the software can be used by blind listeners, in collaboration with their sighted colleagues As was stated in Section 1 1, one of the primary target groups for the TechRead system are students, thus making the incorporation of visual, as well as audible information extremely

[4] It is not envisaged that this feature will be present in the first incarnation of the TechRead system

important As the poet says, "No man is an island", and experience has shown that more can be achieved through the collaborative process than when working alone

3.3.1 The feel of TechRead

The fundamental aim of the keyboard interface to TechRead was to enable the blind user to gain rapid access to both the structural hierarchy and content of technical documents Consequently, the underlying principle was to minimise the need to perform multiple atomic operations to determine facts which are obvious to sighted readers In terms of generic screen access technology, the blind user must learn extra mnemonics in order to gain information which can be deduced from a brief glance at the screen For example, if a warning, or error notification message appears, the user of such access technology must perform several steps to deduce the nature of the message Then, and only then can they respond in an appropriate manner

It should be realised, that the apprehension of extra mnemomcs will not be totally eradicated in the TechRead system Rather, it is aimed to keep the need to learn extra features to a minimum It is also hoped to automate responses to the extent that the atomic actions needed in other screen access software to ascertain easily apparent material, will be superfluous An example of such automation is that, when `dialog` or `message` boxes appear on screen, their contents will be spoken automatically The user will not have to perform extra actions to gain the information, which is so easily visible to the sighted user

With these criteria in mind, it has been decided to locate the majority of the interface on the numeric keypad, found on most keyboards currently available The reasons for this are numerous

central location All commands can be centrally located One of the major drawbacks with some access products is the fact that their mnemomcs

are dispersed over unrelated portions of the keyboard The use of the numeric keyboard provides a central location for the command set used in the system Also, the use of this portion of the keyboard is becoming standard in the design of access technology

ease of learning The dispersal of keyboard mnemomcs can render the extra atomic actions needed to comprehend aspects of visual presentation extremely difficult If the command needed to discern the font in use at a given location is unrelated to that which provides information pertinent to the alignment of text, it will prove extremely taxing to remember both commands without confusion This lessening of the cogmtive load is a key design goal in the keyboard interface used in the TechRead system The principal desire is to reduce the effort needed to gain access to the information so readily apparent to sighted readers By lightening the mental workload, it is expected that the main effort expended by the user will be in the areas of content comprehension, rather than in the remembrance of the extra mnemomcs needed to assimilate the required information

easy expansion The numeric keyboard provides great scope for future expansion The number of overlays[5] are theoretically infinite Realistically the overlays which are possible are limited to those based around the addition of combinations of various pre-defined, and logically related control keys For example, the *5* key on the numeric keypad could be the *read current* key When pressed in combination with the *Control* key, the meaning of this key could be *Read Current Sentence* while when pressed singly it could be *Read Current Document*

As was stated previously, the "feel" of this application will conform to the standards laid down in the *Look & Feel Guide* produce by `Microsoft Corporation` for all `MS Windows` software As a consequence, though the actual manipulation of the document will be centered around the numeric keypad, it will

[5]extra meanings which can be attributed to keys, through the addition of extra control keys

also be possible for users who so prefer, to utilise menu systems to achieve the same goal Though a shortcut will exist to read the current section, a menu hierarchy will also exist to permit the user to fulfill the same objective The relationship between the keypad interface, and the standard `Windows`-oriented approach will be described in the subsequent paragraphs Another point of note is the fact that, ultimately, the TechRead system will (it is hoped) be platform-independent This has serious consequences for the design goals of the interface The objective is to produce a system which follows the visual design strategies of the operating system on which it is mounted, while ensuring that the blind user will not have to learn new keyboard mnemonics to cater for vastly diverse platforms These, and other implementation details are discussed more fully in Chapter 6

The numeric keypad interface

The means of human interaction with various previously implemented systems were described in Section 2 3 It was seen how in ASTER, a series of `VI` commands were utilised to give the listener access to the content and structure of the document, while the MathTalk system used in `EMACS`-based system of double keystrokes, consisting of targets and actions to offer the same functionality The description of the numeric interface begins with the simple, though essential reading commands All aspects of this discussion are based around Figure 3 6

The basic notions of *previous item*, *current item* and *next item* are utilised in this keyboard interface The reasons for this are, as described earlier, that the speech signal, being transient, offers a narrower degree of focus than the visual stimulus This narrower focus results in the listener being unaware of those elements which are adjacent to the one on which they are currently located, unless an action is performed to determine what it is

This is in direct contrast to the degree of focus offered by Refreshable Braille

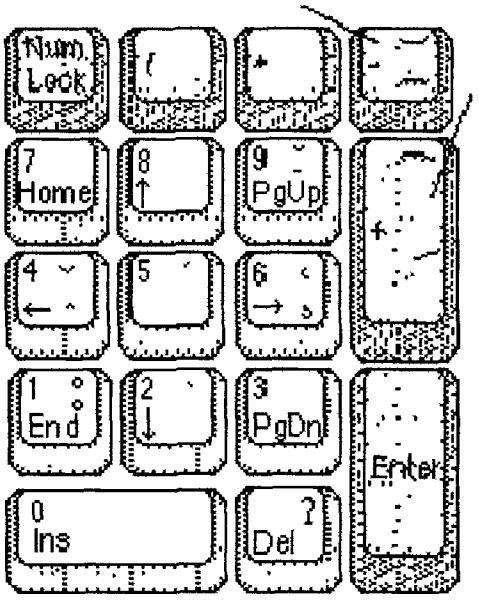


Figure 3 6 The Standard Numeric Keypad as found on *most* modern computer keyboards

displays, where the focus is far wider As an illustration of this, let us assume that the user is using `MS Windows`, and is located on the desktop When any given icon is highlighted, this is the location of audio focus If, on the other hand, Refreshable Braille were in use it could be easily determined whether any other icons were positioned horizontally adjacent to the current one, by moving the hand across the display The *5* (or centre) key on the numeric keypad has been mapped to a *read current item* task This item can range from the entire document, through the various sectional units incorporated into the model, to the atomic units of the representation, namely the word, mathematical element, or tabular item

There are two alternative approaches for reading the current items The first, is to have a cyclic approach Using this method, a single key press could read the document, a second could read the current section, and so forth down to the base level This approach is not considered to be optimal, however The cyclic approach would necessitate the user maintaining a mental image of not

only what was being read, but the current point in the cycle at which they were If they were at the point in the cycle when a key press would read the sectional unit, and they wished to read the paragraph, the number of key-presses to obtain the required information would have to be determined, and then the information could be heard

The alternative approach is to utilise the *read current item* key in conjunction with various other keys to read different hierarchic levels contained within the document To fulfill this role, it has been decided to use the *read current item* key alone to read the lowest hierarchic unit at which the listener is currently located For example, if the user is in a paragraph of running text, then this operation will cause the current word to be read, whereas if they are in a mathematical expression or tabular environment, then the current atomic element is spoken

The application of the various control keys will cause differing information to be read depending on which environment the listener is in In text mode, the use of the *Control* key will cause the next level up the hierarchy to be read This will customarily mean that the paragraph level will be spoken This key is said to be a *relative assignment*, as it is dependent on the user's location in the document hierarchy for what level is spoken when this combination is pressed For example, if the user is browsing the Table of Contents (see below), and this key is pressed, then the sectional unit could be read In mathematical mode, the addition of overlays to the *read current item* can cause vastly different responses For example, the `Control` key is pressed, then the view of the equation above the level at which the user is located is read

The functions of navigation within the TechRead system are combined somewhat with the reading keys with a view to minimising the number of mnemonics which the user must remember As can be seen from Figure 3 6, the *5* or *read current item* key is located at the centre of the numeric keypad The keys occupying the cardinal positions (north, south, east and west) from this key, serve as the navigational keys of the system The concepts behind their choice

are quite simple in nature, but their incorporation results in a lessening of the cognitive load on the part of the user In order to move through the document at the same hierarchic level, the *next item* (*6* key) and *previous item* (*4* key) are utilised When these keys are used without any additional control keys they simply move through the document at the same level If the user is reading a paragraph of text, then the *previous* and *next* item keys will move the user through the text word-by-word Alternatively, then they are utilised in conjunction with the *control* key they will navigate at a paragraph level

It can be seen, that using this method, the user never actually moves up the hierarchy, instead, the subordinate unit, and the unit at the present level are immediately accessible The other keys mentioned above, are responsible for the movement upwards or downwards through the document structure If the user is at a sectional level, and wishes to descend into that unit then the *2* key is pressed, while the converse is achieved by use of the *8* key

In order to appreciate the purposes of this assignment, it is recommended that the reader imagine themselves at a computer keyboard, with the middle finger of their right hand on the *5* key of the numeric keypad Using this key, the current item can be read, while the remaining fingers can easily and rapidly utilise the other features described above However, there is often a wish to simply start reading and not stop till the document had been concluded The *zero* key on the numeric keypad fulfills this purpose [6] Here, the document is read from the current location point, until the end has been reached, or the same key is depressed once again to terminate reading Therefore, this key acts as a toggle, that is, it can be turned on, or off

As was also alluded to in the previous section, the necessity to incorporate features for the storage and retrieval of bookmarks is essential in this form of reading strategy The ability to insert small pieces of paper at various convenient locations in a document is often employed by sighted readers to maintain a record of either information important to them, or places in a document to

[6] In some circumstances, the "read current item" can also fulfill this purpose

which they wish to return This facility is performed in TechRead in two ways, either through the menu system (see below) or through the " " key Either approach leads the user to a simple `Dialog Box` which gives them the options to add, or re-locate to bookmarked information The user can thereby interrupt their reading of a document, in the sure knowledge that they can return to the point at which they left off at a later time

Another **extremely** important key in the TechRead system is the 'What is this" key This key serves several purposes

1 it can be used to gain access to an interactive help system

2 it can be used to gain a rapid insight into the purposes of each key

3 it can be used to determine the nature of the unit at which the current focus is located

This key offers useful information in various contextual settings By using this key on its own, the user is taken to an online help system which gives information relative to the overall use and functions of the system When pressed twice in rapid succession it yields the nature of the object at the current location Therefore, it is not only possible, but easily so for the user to determine whether they are perusing a textual object, a mathematical expression or a table One aspect of the help system is that if the key is pressed, and held down for more than a second, it offers the user the opportunity to quickly determine the purposes of the remaining keys in the interface The means to obtain this information is

1 press the key

2 hold down for (roughly) one second

3 release, and press the key whose purpose is required

It is thus easy to obtain help information. As was stated in the previous section, one of the most difficult strategies to implement in a document browsing system, is the ability to afford the user the facilities to jump from location to location within a document, in the same manner as sighted readers. The interface is expanded to cater for this form of reading. The first means of "flicking the pages" described above, involved the sighted user scanning through the document seeking text which contained various visual attributes, which distinguished it from the surrounding text. Using these cues, sectional titles (or other important text) could be determined. This form of browsing is implemented in TechRead through the use of the Table of Contents.

The user is presented with a tree-like representation of the hierarchic sectional units within a document. Using the reading/navigation keys described in the preceding paragraphs, the user can then traverse the Table of Contents, until the required Section is reached. Through the use of the level adjustment keys, they can then descend into, or simply read the information they require. The summary information, is achieved through the use of the *Read Summary* key. This key (the $\star$ key prepares, and reads a summary of the document contents. It will be possible, in future versions of the system to augment this feature to provide, for example, a rendition of all the mathematical expressions in a document, or other features which will become apparent from empirical feedback.

Another form of "jump-reading" is the notion of following cross-references, or the examination of material contained in various types of footnote, or marginal note. This facility is not presently available using existing technology, apart from mainstream hypertext oriented software. This is in essence, what is being suggested; that a degree of the properties of hypertext system be introduced into the LaTeX document, to render their navigation more conducive to blind readers. Consequently, the *follow link* key has been provided in the interface, and assigned to the $+$ and $-$ keys respectively. The underlying meaning of these keys will be obvious. The $+$ key will follow the link in one particular direction,

while the alternative will return the reader to the point of departure For example, should the reader encounter a cross-reference in the course of running text, then the + key can be pressed to follow the reference to an alternative location in the document Once the required information has been read, the − key will return the reader to the beginning of the link

An obvious question at this juncture must be whether the interface can cope with multiple jumps For example, suppose the reader followed a link from Section 1 of a given document to Section 3, and from thence to Section 5 A simple stack can contain the points of departure and the arrival points, thus permitting the user to return to the point at which the original leap was made This is akin to the **Caching** performed by **Internet Browsers** to maintain lists of sites visited in a given session Just as the inter-document linkage supported in **Internet Browsers** is not supported in TechRead, so the sophistication of the **Caching** mechamsms is unnecessary and a simple stack will suffice

The discussion to date has focused entirely on the textual and structural components of the technical material However, the interface design incorporates the features necessary to peruse mathematical and tabular data The reading keys described above are extremely useful in the mathematical environments, though they are not sufficient to gain an accurate view of the mathematical data, where a finer degree of control over the information flow is required To facilitate this, the remaining *number* keys are introduced to give the user the flexibility to peruse this highly syntactically complex form of presentation Just as the atom of the textual content is the word, the atomic elements which comprise the mathematical expression are defined as the smallest unit contained within an expression, which can be deemed to have any bearing on the comprehension of that expression

A term, on the other hand, is deemed to be the juxtaposition of various mathematical atoms which give semantic meaning to the expression The means in which these terms are defined, determines the degree of access which it is possible to attain As a consequence, the degree of control needed to determine

the various atomic parts of the expression is far finer; hence the use of the remaining keys. If the user is reading an expression like $(a + b^2)$, then such information as which terms are governed by the subscript is imperative. This information will be conveyed using prosody (see Section 5.3). However the ability to scan and re-examine the content of this form of material is essential. If the user is in a mathematical expression, the keys described above take on new significance. The "1", "2" and "3" keys are utilised to discern which atoms in a given term are governed by subscripts and superscripts respectively. The reading and navigation keys described earlier (and in a textual context) can be used to:

- read the whole expression
- read the current term
- jump to the next term
- jump to the previous term
- give an overview of the equation
- give a running paraphrase of the equation

These facilities are provided through the use of the various hierarchic levels of the mathematical expressions. In order to ascend from the lowest and most comprehensive view of the equation, the user need only use the *move upward* (*8*) key. As the focus ascends through the expression, the user will have access to the summary information presented at that point. Consequently, the reading strategy (outlined in the previous section) to cater for the mathematical content, is implemented.

Tabular data, as has already been mentioned, causes many headaches for the blind reader; hence the use of the numeric keypad which is ideally suited to the provision of an audio view of this unique form of presentation. If the reader can imagine placing their middle finger on the *5* key of the numeric

keypad, then the remaining keys in this row are covered by the two adjacent fingers. The *read current*, *read previous* and *read next* can be applied to the cells in the tabular presentation, just as easily as they were applied to the mathematical data and textual content. Further information is often required from tabular presentation, namely the contents of columns above and below the presently selected one. If the middle finger is moved upwards (to the *8* key) or downwards to the *2* key, then these columns can be rapidly deduced. Other keys used in this object type are the *7* key which reveals the topmost cell in the column, and the *1*, which reveals the leftmost column in any row. As has been already pointed out, the author often uses these tabular locations to convey information which explains the remaining entries in the table.

The keyboard assignments used in this interface are arbitrarily assigned to keys which are thought to be most intuitive. They have a logical relationship based on their proximity to each other, thus deriving a logical relationship in the actions they perform. This mapping will not be conducive to use by all, hence the possibility of alternatives. For example, though the *4* and *6* keys are currently used as *read previous item* and *read next item* respectively, there is no reason why (for example) *2* and *8* could not be used, if the user found this mapping more conducive to their needs. Other alterations would, quite naturally follow from this change in the basic key assignments. However, the only restriction would be that they be kept in a central portion of the keyboard. The reason for the incorporation of this flexibility is the more frequent use of `laptop` or other portable computers which do not possess numeric keyboards. Before any distribution of the system, examination of various `laptop` keyboards will have to be undertaken and a default strategy defined for the location of the keyboard mnemonics on this type of keyboard. At the time of writing, it is uncertain as to the outcome of this examination, as such a diverse range of `laptop` keyboards exist.

To illustrate why the default keyboard mapping was chosen let us assume that the reader is currently at Section 1 of a document, which contains 2 sub-

sections If the user wishes to proceed to Section 2, then they must use the *2* key whereas to actually open this particular branch of the hierarchy they must use the *6* key Therefore, the key which appears to be taking them down a level in the hierarchy is actually maintaining their level at the current state, while the converse is also true that the key which appears to keep the position at the same hierarchic level within the document is actually descending

3.3.2 The menu system

In order to maintain a degree of consistency with other MS Windows applications, it has been decided to implement the features described above, using a series of menu system commands Though all of the document navigation commands have dual shortcuts, such aspects as file management are handled by menu-based commands The reason for this is simple Though this application has been designed primarily for the use of blind people, such users do not operate in a vacuum It is customary for a blind student to work with either a sighted teacher, or colleagues who are familiar with this form of interaction The menus will act as an aid to the sighted, as well as the blind user Personal experience, coupled with some (albeit basic) empirical testing has revealed that the means many blind people seem to gain familiarity with new software, is to explore menu systems and derive a mental model of their hierarchy Once this has been achieved, the association between the hierarchic levels of commands can be reduced to a series of shortcut keyboard mnemomics, namely the numeric interface described in the previous section To aid in their retention, the various commands will be located in menus which group the various types together Proposed inclusions in this menu system are

File menu This menu contains all the commands pertinent to the file management Such aspects as saving, loading, translation and embossing[7] will be contained in this menu

[7]the physical Braille printing of a file

View menu This menu is not duplicated in the numeric interface. Here, the visible aspects of the application are controlled.

Settings Menu This is an important group of commands. The various commands which can alter the appearance, and sound of TechRead can be found here. Such options include:

1. voice alteration
2. keyboard mapping
3. Braille Translation Tables

Read menu This menu contains many of the features described in the context of the numeric interface. Such aspects as *read current item* and *follow link* are found here.

Help menu As the title suggests, the online help system is accessed via this command set. It is tied to the numeric interface by the *What is this* key?

Some of these brief descriptions need further explanation. As has already been stated the *file* menu will contain the commands necessary for file management, such as saving, embossing and the like. Where possible, the commonly utilised procedures are adhered to in all these situations. For example, when the user wishes to open or save a file, the standard `dialog boxes` appear. Most users familiar with `Windows` applications will have encountered these entities previously. Consequently, to minimise the new features with which they must become familiar, it has been decided to incorporate many of the standard aspects of software into the TechRead system. Some new `dialog boxes` are unavoidable.

The `view menu`, will contain the standard settings which are normally found in `Windows` based applications. An additional command found in this menu is one which alters the view from a `WSYWYG` view, to one which displays the raw LaTeX. The reasons for this are simple. Firstly, the user may wish to use the LaTeX view to gain an insight into the means for producing the content they are

perusing Just as many **Internet Browsers** contain features which permit the user to view the source of the HTML, so TechRead will enable the user to view the LaTeX Also, this feature can have uses in deciding what reading strategies best suit new, user defined objects For example, a quick examination of the LaTeX source could convince an experienced user, that a macro can be denoted as a form of emphasis, and treated accordingly Alternatively, a quick view of the source could inform the user that this particular macro is best ignored and can be discarded Other features which can be altered in the **view menu**, are the presence or absence of **Toolbars** or **Status bars** These elements of the visual appearance of the application are more relevant to the sighted user, as they can provide useful features to aid in *their* use of the system

The **settings menu** is one of the most important aspects of the TechRead system Here, it is possible to modify the sound, (and in the case of the Braille output, the appearance) of the system For example, if the **Speech settings Dialog Box** is chosen, the user has the option to alter the default speaking rate, average pitch and pitch range of the voice It is also possible to change the means of presenting the types of content where the characteristics of the voice are altered For example, the default set-up of the system instantiates that a generic category *emphasis* contains any text which is underlined, emboldened or italicised Further, it is possible (should the user require) to define a different speaking voice for each of these environments Using this **dialog box**, it is possible to define speaking styles for hitherto undefined objects Should a macro be encountered in the source document, it is possible to introduce this command into the command dictionary for future reference

The help menu contains the features which have been described in relation to the "what is" key previously discussed, though there are also some additional features not associated with that key Examples of this include a searchable help index, and some on-line manuals for which the standard modes of keyboard access found in **Windows** programs are utilised

3.4 The visual interface

In keeping with the principle of following the **Look & Feel** of other `Windows` applications, the visual appearance of TechRead will assume all the characteristics found in other software That is to say, the buttons, `dialog boxes` and all other visual components will inherit those visual attributes which have been associated with the commonly used controls The implementation details to physically produce this interface will be described in Chapter 6 However at present, it is suffice to say that the interface will conform to all standards

The document itself will be displayed in two panels on-screen The leftmost panel, will consist of the Table of Contents containing all of the sectional units which have titular information associated with them A standard `tree control` is used to hold these elements The reasons for this are

1 the `tree control` can cater for the hierarchic nature of the sectional units

2 the control lends itself to use by the numeric keypad based interface described previously

The alternative mappings described in the previous section, will need to be incorporated into the keyboard interface for this control By default, the *6* key opens up each branch of the tree, while the *4* key closes the sub-level, and returns to the preceding one Though there is nothing theoretically wrong with this keyboard (indeed it was suggested as a possible alternative previously), it is not considered to be as intuitive as that used as the default set-up of the system The keyboard events pertaining to this control will therefore have to be overridden to ensure compatibility with the interface As can be seen, the control facilitates the easy use of the interface

The right-hand panel consists of a simple display area, where the actual content of the document is presented A `RichEdit Control` is used to achieve this, with its editing capabilities disabled to ensure that no content can be

altered Future versions of the system will incorporate the editing features needed to permit the manipulation of technical content It is foreseen that the guidelines described in this document can be extended to incorporate the extra needs of those who wish to manipulate the technical information in some way The reason for the use of this control, is that it incorporates features which can be harnessed to display the LaTeX in a more accurate manner For example, the `RichEdit Control` has built-in functions which aid in the alignment, and line spacing of the text Coupled with the `LOGFONT` structure (see 3 1), the LaTeX can be displayed in a manner closely resembling the output presented on the printed page

The fonts used in LaTeX are not as precisely defined as those found in more recent Word Processing packages For example, the degree of control which the user has over such aspects as point size is not as refined in LaTeX as it is in `MS Word` In the former, a global setting is used to give the font size used in the greater proportion of the text The modifications which can be made to this are through use of keywords such as *large*, *Large* etc whereas in the latter the alterations in the font size are expressed in absolute terms This presents a problem when displaying the material To do so, an algorithm has been defined, which seeks to relate the alternative relative font sizes, to their absolute equivalents While this may not be entirely correct, it is a simple matter to alter the assignments

A slight difficulty arose, when characters other than those in the ASCII set were needed Such characters as mathematical symbols, and Greek letters were quite difficult to display accurately Two possible means of overcoming this problem were discovered On some systems, fonts such as `symbol` are included for the specific purposes of drawing these specialised characters They employ the ASCII character set, and map these characters to the unusual representations required for display purposes As an example, the letter "S" under a normal textual font, is displayed as a σ when using the `symbol` representation What is needed is a mapping between the ASCII character set, and

the corresponding associated characters in this unique font style In order to cater for those systems which do not possess such fonts, it has been decided to maintain a small database containing the characters, along with their bitmap representation This is not particularly memory intensive, as there is only a finite character set which can be drawn upon to produce mathematical content An 8 by 8 bitmap is stored along with a code which denotes each of the special characters which are incorporated into the visual interface

Another extremely good reason for the use of the `RichEdit Control` is that the addition of those features which are expected in `Windows` applications, such as scroll-bars and the like is extremely easy This functionality means that, where the blind user can manipulate the document using the keyboard, their sighted colleague can utilise the mode of interaction with which they are most familiar, namely the mouse The addition of these features ensures that the user can simply scroll down the right-hand panel, reading the text as they proceed Alternatively, should they so wish, then the functionality provided for the blind user should make the reading of the document infinitely easier For example, where the blind user can implement a key-stroke to take them from point to point within a document, the visual reader can implement the same actions using the mouse

The use of the `toolbars` and `Status Bars` can be extremely useful in any interface, and hence will be a part of the TechRead one The `Toolbars` will include the buttons useful for sighted users to enable mouse interaction with the system However, it will be possible (as in other `Windows` software) to disable them entirely The `Status Bar` will contain the information needed to give the sighted user their orientation within a document The *what is* key will provide the listener with this type of information For example, it will be apparent what type of object currently has the focus Also displayed on this portion of the screen, will be the page, line and column number at which the reader is currently located

Thus far, the visual appearance of the TechRead system has been described

in terms of the **Windows** operating system There is however a vast and diverse range of both platforms and visual appearances to which the software can be ported These range from the highly graphical in nature, to those purely driven by command-line interfaces A different display strategy will be needed for these types of interfaces, and indeed, it may prove impossible to display some of the highly complex mathematical material in some of these more visually limited platforms

As was stated previously, though the *feel* of the system will be exactly the same, the visual appearance of the displayed material will be vastly different An example will suffice to illustrate these differences The overall method of display, in which the structural material was abstracted from the content into two different panels on the screen was described above Under **X-Windows**[8] such items as **Tree controls** are not in common usage As a consequence, the visual display will have to be adjusted to allow for these alterations in the standards of presentation The screen will be split horizontally, with the structural information above the content As a consequence, only one Section will be apparent at any given time To explain this alteration in visual appearance, it is necessary to envisage a file-list directory

Under **Windows** it is possible to view the list of directories on the left, and the files they contain in a panel on the right of the screen Under **X-Windows** the contents of a single directory are the only items visible at any given time Therefore, at the highest level, (the root, if you will) the sectional units are visible When the user selects one of these, then the sub-sectional units contained therein are visible, with the actual text in the panel below It should be re-iterated that the keyboard interface will not be affected by these changes in the visual appearance

The overall interaction between the user and the TechRead system has been modeled on the various methods of displaying directory listings, as they mirror somewhat the objects which this software is trying to represent Originally,

[8] a platform designed for UNIX systems

it was hoped to produce a version of the TechRead system which would function on non-graphical platforms Indeed, the keyboard interface to the system for these platforms proves no difficulty The visual component, on the other hand presents many problems The pixel-based drawing capabilities inherent in graphical-oriented systems are not supported to the same extent on the older, command-line Operating Systems Many of the characters which can be easily displayed under graphical environments, cannot be rendered in those which are command-line based It is becoming clear that only graphical systems can cater for the complexity of the objects which the system needs to draw It is therefore proposed to limit the implementation of the TechRead prototype to those systems which can readily support the mathematical symbols, and other special characters needed to visually present the material displayed

3.5 Summary

This chapter has discussed the various components of both the internal representation of the LaTeX document, and the means by which the user can interact both visually and through the keyboard with it It was shown how the source is transformed into a model consisting of a document graph, which can then be used by the external human interface to facilitate efficient browsing The various reading strategies incorporated within the system were also included, and the reasons for their inclusion explored Subsequent chapters will describe the methods which will utilise this internal representation, and derive the physical output The method of Braille translation, and the algorithms needed for the prosodic enhancement of the spoken version of the content will be discussed

Chapter 4

Producing Braille

The means to electronically produce Braille have been known for quite some time, though the greater effort has been directed to the production of literary, rather than highly technical material. Consequently, there is only a limited number of features incorporated into existing commercially available Braille translation packages to facilitate the production of mathematical material. TechRead, using LaTeX as the input source, aims to rectify this problem. This chapter discusses the means used in the system to produce accurate, and well-formatted Braille output. It also highlights the problems with the existing formatting standards, and proposes some improvements to these methods of Braille presentation.

4.1 Braille translation

As was described in Section 1.2, Braille is a system based on various dot-patterns which are utilised to emboss characters and abbreviated words onto heavy paper. As was also described in that section, the context-sensitive nature of the character encoding used in the system often needs semantic interpretation on the part of the user, to determine what the correct symbol to use in any given situation actually is. For example, the word *these* has an abbreviation of its

own, as does the word *the* Consequently, when the computer is automatically translating some electronic text into Braille, guidelines must be built into the system to preclude the use of the "*the*" symbol, followed by *se* to represent a word which has its own contraction Other such semantic problems occur when different meanings are attached to the same symbol, depending on their location in a letter grouping For example, a certain symbol when found at the beginning of a word, denotes the letter grouping *dis*, when found in the middle of a word indicates the grouping *dd* while when encountered at the end of a letter grouping means " "

Another significant problem with the automated translation of Braille is the non-linear representation of mathematical and other graphical material Many word processing packages use alternative fonts, or graphical symbols, to represent the unusual character set utilised in mathematical material As a consequence, it has proved difficult for developers of Braille translation software to derive meaningful output from this complex form of presentation In order to overcome this problem, the LaTeX representation is used in the TechRead system (see Section 1 1) The linear representation of the non-textual content facilitates the derivation of Braille output, through the examination of the mark-up and the interpretation thereof

However, other issues remain to be overcome Arrabito [Arr90] states that without a degree of semantic mark-up, the production of a TeX based Braille translator, and indeed a universal Braille translator, is impossible He points out that using the TeX primitives, authors can control the visual modality of their document, with no regard for the overall structure of their material They could, for example, define various characters or symbols in terms of line segments, or use environments to achieve display styles for which they were not intended Our experience has shown the conjecture outlined in Arrabito's thesis [Arr90] to be true Using the TeX language, some authors often use the "display" environment to display textual material, as opposed to the mathematical content for which it was designed

There are two important phases to the production of Braille:

Deriving the Characters This involves the use of an algorithm to determine the correct Braille symbol to use in a given situation. The input text is examined to arrive at this information.

Text formatting Unlike printed text, Braille (being substantially bulkier) is predominantly linear. Also, the spatial location of textual, and mathematical content on the physical page is significantly different, hence necessitating the application of different formatting rules.

4.1.1 Obtaining the characters

As outlined above, the first objective in the generation of Braille output is to deduce which symbol should be used in a given situation. To achieve this, algorithms must be devised, which examine the word groupings within a document to determine whether Braille contractions exist to represent them in their entirety. If such abbreviations exist, then further analysis must be undertaken to ensure that the short-hand symbols designed to abbreviate the word can be used in the context in which the word occurs. For example, if the word *to* occurs in the middle of a line of text, then a symbol is used and conjoined to the proceeding word. If, on the other hand the same word occurs at the end of a line of text then the word is written in full.

It should be noted at this juncture that all references to linear position refer to the Braille line, and not the printed version. The reason for this, is that there are only 40 characters available on the Braille line, compared to the traditional 65–80 found in printed documents. If no contraction is found which can be used to represent a whole word, then the letter groupings within the word are examined. Two alternative approaches can be used to do this.

Firstly, the system can begin by taking the longest letter-grouping (which is not a whole word) and attempting to find a Braille symbol to represent

this If none is found, then the next longest is tried, until a symbol is found To illustrate how this method operates, let us trace through an example An appropriate word for the discussion of this approach is *thesis* The algorithm would firstly determine that there was no abbreviation which catered for this word in its entirety Consequently, an iterative process would be undertaken, commencing with the grouping *thes*, and continuing until a point was reached at the grouping *the* where a symbol was found The letter grouping *sis* remains, for which there is no contraction, hence the Braille equivalent of the word is determined

This method ensures that the longest form of contraction will be used in a given situation For example, an alternative Braille representation of the word *thesis* would be to use a symbol to indicate the letter-grouping *th* and spelling the remainder of the word out in full This is an inferior representation as more Braille characters are used in the word In terms of the rules of English Braille, the former symbol-grouping is in fact the correct version

Another, more complex example of where this method of character-generation is effective is in the example of words such as *station* No Braille symbol is available to produce the entire word, so the algorithm described above is applied The resulting letter grouping from the first parse through the word is the *st* symbol, followed by the letters *ation* A second examination of these remaining letters, results in the use of the symbol designed to represent exactly this letter grouping, hence reducing the word *station* to three Braille characters [1]

An alternative method of generating the characters needed to produce the Braille output, is to start with the shortest letter grouping, and to work upwards, i e , to apply the algorithm outlined above in reverse Either solution is acceptable The former is used in TechRead

Though the algorithm just described sounds simple in nature (a stochastic process trawling through the character grouping), there have to be subtle re-

[1] the *ation* contraction is a compound abbreviation, using two Braille characters to denote this letter grouping

finements built in to cater for the context sensitivity of the Braille character encoding As was alluded to earlier, the same symbol can be interpreted differently, depending on its location in a letter grouping Also, certain contractions can only be applied to letter groupings if they appear as whole words, or in the middle of words As a consequence, the character generation portion of the algorithm must be adapted

There are two approaches which can be adopted The first, is to devise a dictionary of rules, which encompass the exceptional circumstances provided by these character groupings This method is employed in the **Cypher** translator produced by **Dolphin Access Systems** It requires the definition of large numbers of rules to ensure that all possibilities are catered for An example of such a rule is as follows Let us assume that the word *ornament* is being translated The correct Braille representation is to use a contraction for the *ment* grouping, and to use the letter symbols to indicate the remainder of the word However, problems arise, when one considers that a contraction exists to represent the *name* grouping This contraction **cannot** be used to indicate this grouping within a word It may be used in the context of *names* or *namely*, where it constitutes the root of a word, but not within another word Hence, the **cypher** system implemented a rule to the effect that the *nament* letter grouping was translated in such a manner as to use the *ment* letter symbol, while using *na* to complete the grouping Having spoken to the developers, it appears that there are over 900 rules in this system, which seems an extremely high number for what is essentially a pattern recognition problem

The other method is to use a **regular expression grammar** Regular expressions are used extensively in the **Unix** Operating System when general patterns are known, rather than specific instances of letter or number combinations For example, using this method it is possible to express such things as

- ensure that three characters are followed by four numbers in a given combination;

- ensure that there are one or more occurrences of a letter;
- ensure that no letters are present in a symbol grouping;

Using this method, the actual location of the pattern can be determined, and the correct contraction chosen. To return to the example of *ornament*, constraints could be built into the regular expressions to ensure that the *name*, when occurring in the middle of a word, would be ignored, while if it constituted the root of a word then it could be translated correctly. This is achieved by specifying that the *name* grouping must be preceded by a white-space character, thereby indicating that the *n* is located at the start of the word. The same applies in reverse to the *ment* grouping. If this is encountered at the beginning of a word (as in *mentor*), then it should be ignored, and alternative abbreviations used instead.

A variation on the regular expression grammar was used in a prototype Braille translator, implemented as a module of the third year B.Sc in Computer Applications course in Dublin City University [Wal99]. Here, the translator was provided with a pre-formatted text file and the ASCII examined for both whole word, and partial contractions. The method used is a simple, though effective one. The algorithm is as follows:

1. determine whether a whole word contraction exists
2. if not, perform an alphabetic, incremental search for the first partial contraction to be found
3. continue until the entire word has been translated into Braille

This algorithm is analogous to the *shortest-first* method described above. An example of how it operates is as follows. Let us assume that the word *shadow* is to be transformed into Braille. The first letter being *s*, the algorithm moves to the portion in the list of contractions at the beginning of those listed under this letter. Next, the position of the final contraction beginning with *s* is

determined, thereby providing a start and end-point for the examination of the contraction list Incremental comparisons, reveals that the *sh* contraction is the only contraction which can be used in this situation, therefore it is extracted from the word, and the process begins once again with the letter *a* Repetition results in the word being rendered successfully into Braille, in the form of a symbol to indicate the *sh* letter grouping, the single letters *a* and *d*, followed by the final *ow* symbol

Exceptions are handled in the prototype through the use of an *exceptions dictionary* This file consists of those words and letter groupings which yield unusual contracted forms An example of a part-word contraction would be the letter grouping *ghouse*, which occurs in words like *doghouse* The syllabic nature of Braille translation would indicate to the human transcriber that this grouping, occuring as it does in the middle of a word, would be translated as the letter *g*, while the *house* would be treated as though it were independent of any preceding letters Without intense semantic examination, (which TechRead does not purport to do) it would not be apparent to the automated process that this letter grouping should be treated in this fashion, and hence erroneous contractions would be used The translation process would produce the *gh* symbol, followed by the remainder of the word The exceptions dictionary is designed to avoid such problems

4.1.2 Obtaining mathematical symbols

As was described in Section 1 2, the same Braille characters are used to indicate different characters or symbols depending on the context in which they are used Unlike the printed mathematical alphabet which supplements the Roman letters and Arabic numbering system with other symbols taken from many diverse sources, the Braille mathematics encoding is based on the same 2^6 symbols used in the writing of textual material Accordingly, a mapping from the visually rich printed presentation to the impoverished Braille equivalent, loses much in the translation

There are two primary Braille mathematical encodings in common usage at present The first is the Braille Mathematics Notation [otUK87] produced by the Braille Authority of the United Kingdom (BAUK) This notation uses a linear presentation style The second is the Nemeth Code [Nem72], which is primarily used in North America and Canada This character representation attempts to produce Braille symbols which are analogous in shape to their printed counterparts There are disadvantages to such an approach as new symbols are integrated into mathematics, Braille equivalents must be found which conform in shape to maintain consistency with the policy of the Nemeth system As there are only a limited number of Braille characters, anomalies cannot be avoided

The British notation has attempted to use combinations of symbols to represent the printed notation This is best illustrated by an example of Greek letters In this form of presentation, all Greek letters are based on the closest Roman equivalent The closest Roman equivalent to the σ is the s, and hence this is the root of the Braille equivalent The addition of one particular dot-pattern immediately preceding the s letter sign, transforms it into a σ, while another prefix produces the Σ

The British approach benefits from a relatively intuitive mapping, but the small number of Braille symbols available necessitates the use of several prefixes to indicate the intended meaning of each dot-pattern One obvious disadvantage of such a mapping, is the increase in the complexity of the Braille representation As more symbols are used to convey even the simplest equations, the readability of Braille mathematics is extremely low It is often unclear where the scope of a summation or an integral ends, and it can sometimes be difficult to deduce which set of parentheses contains which symbols The linear nature of Braille mathematics makes the manipulation and perusal of highly technical material even more difficult

It is not proposed to devise and implement a new mathematics notation in this thesis Rather, it is intended to utilise the existing British mathematics

notation [otUK87], since we believe that this form of presentation contains a more concrete, and reasoned mapping than other codes (such as the Nemeth character encoding [Nem72]) which enables the production of technical Braille Where necessary, the rules of transcription found in the British Scientific Notation [otUK90a], and British Computer notation [otUK90b] will be introduced in to the system

Another important reason for the decision to make use of these character systems is that they are used in Irish schools Presently, the Nemeth system is used in North America and Canada almost exclusively, and consequently its inclusion would be almost totally meaningless to the majority of Irish students

The translation algorithm

The translation algorithm for the production of mathematical Braille is not unlike that described for text, though a different set of rules is required to generate this vastly different type of information The internal document model (see Chapter 3) determines the rule-set to apply to a given object within the document The $t(N)$ operation returns the type of a given node in the model, and this is used in the decision process whereby the correct translation algorithms are applied to the correct portions of the document

The mathematical algorithm examines the structure of the equation to determine where sub-expressions begin and end,which groupings are contained in base-level operators such as summations or integrals, and the correct symbols to use The context-sensitivity of Braille is much in evidence in mathematical material In this form of Braille, the individual symbol looses its relevance, but rather the group of which it is a part takes precedence An example of such an occurrence is the Braille mathematical symbol used to indicate a $\star$ When found in another context, (that is, when proceeding another symbol) the same dot pattern indicates the $=$ operator

The lack of a 1–1 mapping between printed mathematical notation and the

Braille equivalent, has necessitated the development of a dedicated translation strategy The LaTeX mark-up is examined to produce the Braille symbols required As there are over 300 rules indicating the correct symbol to use at given instances, the individual rules of symbol-selection will not be described in any depth here, as their inclusion would constitute a book of their own It is sufficient to say that the LaTeX mark-up is extremely conducive to the production of Braille mathematics The reason for this is that it is itself a linear representation, and provides the syntax to present complex visual structures in a linear fashion

Consequently, what is needed is a mapping which transforms one linear sequence of mark-up into another different, though comparable, linear notation As will be discussed below, one of the principal aims of this portion of our work is to derive an alternative Braille layout which will make the reading of mathematical information easier and more rapid Important distinctions which must be remembered when producing Braille output are the following

parentheses Parentheses play an extremely important role in the translation and intelligibility of mathematical content into Braille To date, vertical alignment has not been utilised as a mechamsm for assisting the blind reader in the understanding of formulae, and hence the use of parenthesised sub-expressions can often be the only means to determine what elements are related to what Consider the following example $a + \frac{b}{c} + d$ It is immediately clear to the visual reader, that the fractional component of the expression consists of $\frac{b}{c}$ However, if this were written as $a+b/c+d$ (as it is in Braille) then it would not be easily determined which elements comprised the numerator and denominator Hence, the true linear Braille representation of this equation would be $a+(b/c)+d$ The introduction of the parentheses ensures that the semantics of the formula are easily observable It is apparent, for example that the a does not form part of the numerator, and the d is separate from the denominator

grouping Coupled with the introduction of parentheses, the grouping of terms

is vital to the comprehension of Braille mathematics. The printed notation uses both horizontal juxtaposition and vertical positioning to convey such notions as superscripts and subscripts. Such a facility is not available in the Braille modality. Hence, extra symbols must be introduced to indicate to the reader where such items as exponents end. The expression $x^y z$ serves to illustrate the need for the symbol grouping encountered in Braille. The sighted reader can readily observe that y is the only element which forms the superscript to the x. As only horizontal juxtaposing is utilised in Braille, confusion could be introduced if a special character were not used to indicate that the z was not apart of the superscript. The means whereby this is achieved is to introduce a symbol after the y, hence indicating to the reader that the grouping is ended.

The extra symbols place a heavy burden on the student, as they must learn not only a large array of different symbols, but their use in contrasting situations. The tightly-controlled nature of symbolic usage ensures that automated output of Braille mathematics is quite feasible. However, it is only since the adoption of LaTeX and other semantic mark-up languages that technical Braille has become a reality.

The explicit mark-up of complex mathematical formulae in LaTeX can be examined closely, and accurate Braille representations of the material derived. For example, it can be discerned from a perusal of the mark-up, what elements comprise the lower limit of a summation, and which are intended as its upper bound. This is not based on semantic interpretation; rather it is through the syntactic analysis of the mark-up. Consider two brief fragments of LaTeX mark-up. The first example `$\sigma_i=1^\infty$` produces the output $\sum_i = 1^{\infty}$. This is a specific usage of superscripting and subscripts to indicate the upper and lower bounds of a summation. As an alternative, consider the example `$x^2$` which produces x^2. In the latter brief extract, the superscript is used to indicate exponentiation. Both can be extracted from this linear representation and reproduced in the analogous Braille form through an examination of the

syntax of one, and an awareness of the rules of the other.

It is beyond the scope of this research to recommend new symbolic representations of printed mathematical characters, though other research has been directed towards this end [Sch98]. However, it is our belief that improvements will require the direct co-operation of the major producers of Braille throughout the world and those authorities who are charged with the responsibility for the development and maintenance of Braille standards.

4.2 Braille layout

The previous section described how the actual Braille symbols are produced from the electronic input. What is even more important, is the means of spatially orienting those symbols on the physical, two-dimensional page. As any computer programmer will no doubt agree, there is no practical hindrance to the writing of a piece of software in one large function, located on one continuous line, though to anyone wishing to read and understand their work, this form of programming would be almost totally unintelligible.

As the resolution of the finger is lower than that of the eye, it is important to utilise the spatial location on a page, to compensate for the lack of visual enhancement which forms an integral part of any printed document. The following paragraphs discuss the current standards specified by the British Authority of the United Kingdom (BAUK) which specify the methods for locating textual and mathematical material on the physical page. We also describe some suggested improvements, which will be incorporated into the TechRead system. These suggestions are included in the system as alternatives to, rather than replacements for, the existing standards. Our belief is that if this system is to be used, then it must adhere to the accepted standards. However this does not mean that additional methods of layout cannot be included in translation systems. An example of commercial vendors, including their own representations of document objects can be found in the `Megadots` system, where a *stair-step*

representation of a table was provided as an alternative (see below for a more complete description of this mode of tabular presentation)

4.2.1 Simple layout

The layout of simply formatted documents is a relatively trivial matter However, it still has its problems What is important to note is that the Braille page has different dimensions to the standard paper sizes used in normal printing The traditional Braille page can accommodate only 25–28 lines of information, each line containing a maximum of 42 characters [2] As a consequence the printed material must be re-formatted before it can be finally output onto the paper The resulting Braille output is more bulky than the printed equivalent It is estimated that the average page of printed output results in 2–2 5 Braille pages [3] This necessitates the splitting of larger documents into more than one Braille volume An example of the increase in size of Braille books, is found in the Little Oxford Dictionary When printed, this book is extremely portable while the Braille version comprises 14–16 volumes (depending on the edition) and hence is not *little* in the common sense of this word!

The first step in the formatting of simple documents therefore, is to split the Braille output to ensure that it fits properly in the space available This is not merely a matter of moving characters around, or inserting or removing white space There are instances when certain Braille contractions cannot be used if they occur at the ends of lines Such an occasion is when the word *to* is used This word is an unusual Braille contraction, as it must be attached to the following word without any white space between When it occurs at the end of a line, the letters themselves are used instead of the whole-word contraction

[2] 38–40 is more normal, particularly when the Braille is automatically produced

[3] The difference between 1000 and 3500 characters per page would suggest an even greater increase However, conventional Braille formatting avoids most of the white space to be found in printed documents It can be assumed that technical documents in Braille will actually result in a greater increase, perhaps a factor of 4, since more characters are required in the printed version and a greater use of blank space may also be necessary

Accordingly, what is needed is not only a determination of the location of the Braille symbols on the given lines, but a re-examination of the contractions used to determine the propriety of their use in that given instance This may seem over-zealous, and indeed some translators do not manage to complete this task However, in order to produce well-formatted and correct Braille, it is extremely necessary The reason why it is so important, is due to the context-sensitivity of Braille If the symbolic representation for *to* were erroneously used at the end of a line, it would be misconstrued as an exclamation mark and an incorrect translation would result

Another problem which frequently occurs with automated Braille production, is the misuse of space on a page A common occurrence, is to take an input file, and to maintain the line/page breaks in accordance with the printed material Consequently, the resulting output is needlessly large, since the Braille line can end after only 1 or two words This *run-over* effect can make the reading of such material extremely difficult The commonly employed reading strategy for Braille material is to run the index finger of the right-hand across the line, while using the left index finger to locate the beginning of the next one The right hand then returns to the point at which the left index finger is located, and continues reading from that point As can be imagined, the use of shorter lines breaks up the flow of reading It is akin to the eye having to make rapid and needless regressions after only a small portion of material has been read

To overcome this problem, it has been decided to ignore the line-breaks in ordinary running text Instead of assuming that the input is a *well-formatted document*, TechRead assumes that it merely contains paragraphs and sentences which must be placed onto the Braille page with as much regard for the conventions of Braille formatting as possible The basic units of a document in TechRead's representation are the word, sentence and paragraph, although they differ in their Braille presentation from the printed documents

There are several means to denote paragraph breaks in printed documents

1. leave a blank line between the end of one paragraph and the start of another

2. commence paragraphs with a small degree of indentation

3. combine the first two methods; that is leave a blank line between paragraphs, and to employ a little indentation at the commencement of each

The rule in British Braille is to start each new paragraph in cell 3 of the line on which it commences. **No** blank lines are left between paragraphs. It is by dint of the use of such indentation that the reader can distinguish the beginning and ends of paragraphs. This mode of denoting the starting and ending of paragraphs is performed irrespective of the methods used in the printed documents. Another common feature often found in printed documents, is to leave a little extra space at the ends of sentences, and after certain punctuation characters. Once again, this is not done in Braille. The reasoning behind this, is that the medium takes up enough space as it is, and the introduction of such a device is visually useful, but not particularly so in a situation where the finger is employed to read. Indeed, one experienced Braille transcriber pointed out in some discussions, that the use of such extra space between sentences could be a hindrance to the inexperienced Braille reader. The belief is that if extra space is introduced, the inexperienced reader will have difficulty locating the beginning of new sentences. There is no substantive evidence to back up this assertion, but since there is no valid reason to include this extra space (save to give a certain consistency with printed presentation) it is not done in the TechRead system.

4.2.2 Headings, titles and page numbering

The means to visually enhance the content of a document is not readily available in Braille. Accordingly, the relative spatial locations of document objects is often used as a device to indicate that which is achieved by font alterations or

other such print-based methods This is particularly the case with headings Authors traditionally ascribe various visual cues to indicate to those reading their material, that the portion of text they are reading is a heading, or other structural document component For example, some authors will write chapter headings on a line of their own, using a larger, emboldened font Some authors also centre this type of material to give it more weight or emphasis

At a level below this, is a sectional title Once again a larger font than the main body of textual content is employed Alternatively, this title could be written at the left-hand margin, thereby indicating that it is at a lower level than the chapter heading, but is still an important item As the nesting of sectional units deepens, so the visual attribution associated with their importance also decreases It is commonplace to indicate lower-order sectional titles by simply writing their title on a blank line, with no alterations in the visual attributes, thereby indicating that, though the text is a titular passage it is not an important one

Such visual alterations are not available in the medium of Braille As was stated above, it is only through the use of spatial location that the transcriber can inform the reader that the passage they are reading in some way stands out from the surrounding text Though italicisation is possible (see below), it does not convey the same weight as is the case when it is used in the printed modality

There are two means of producing headings in Braille, to locate them centrally on a line of their own, or to place them at the left-hand margin, once again separated from the surrounding text by line-breaks As a consequence, the methods employed in the printed form of presentation are adopted to a degree Unlike the ability afforded by the visual alterations, the techmque used in the Braille context does not permit the user to rapidly scan down the page to find the portions of text which the author has designated as headings In order to find the beginning of a chapter (which is not the first) in a given book, the reader must employ the following strategy

1 locate the correct volume

2 use the Table of Contents for that particular part of the document to determine the page number on which the chapter commences

3 locate the page

4 skim the finger down the left-hand margin until white space is found

5 move the finger across until text is encountered

The same technique can be applied to the side-headings described previously As there is a blank line immediately above these headings, the reader can navigate downwards through the document until white space is encountered However instead of locating a centrally placed heading, they will move further down the page until the subsectional indicator is found There does not seem to be any means of surmounting this problem of the time-consuming nature of the location of unit headings The finger cannot locate material as rapidly as the eye, resulting in a slower rate of search As will be seen below, we propose some alterations to the overall document layout which could assist in the more rapid perusal of Braille material However as the eye can scan at far higher speeds than the finger the use of Braille as a reading medium will always be slower

An important aspect of the Braille document is the title line There are two forms of presenting Braille The first is to simply use single-sided printing, while the second is to use both sides of the paper to emboss the material The first is self-explanatory, however the second needs some explanation As Braille is produced by the insertion of small holes into the paper, the use of double-sided printing can cause problems In order to ensure that readability is not impaired, a slight offset is introduced, thereby ensuring that lines of text on opposite sides of the page are embossed in an overlapping manner For example, the text of alternating odd and even pages is reproduced on opposite sides of the same page Their lines are embossed in such a manner as to ensure that

they occupy the spaces between those of the material on the alternate side of the page. The introduction of this offset ensures that the holes thus produced do not impair the readability of the material on either side of the paper. In single sided transcription, the title of the work is located centrally at the top of each page, while in the case of the alternative method only those odd numbered (or right-hand) pages contain the titular information.

The title line should consist of the following information:

- The print page number (which should occupy the first cells of the line)
- the lowest ranking meaningful title, abbreviated as necessary, which may be that of the book itself or a section of it
- The lowest ranking meaningful divisional number, or number group
- The Braille page number (which should occupy the last cells of the line)

Experience has revealed that in many cases, the first of the items listed above are often omitted from the title information line. This is primarily the case when the material being read pertains to the literary genre, and not technical material. Accordingly, it has been decided to incorporate the full title line into TechRead's translator. The second two items need some explanation. In the case where no chapter titles are available, then the book title itself is presented as the titular information, followed by the chapter number. Where lower-level information is available, it is usual to precede the sectional title with its number. Both these alternatives are catered for in the TechRead system.

There are cases where no titular information is provided. Examples of this are in prefatory information, such as title pages, Tables of Contents and the like. Here, the convention is to use Roman Numerals rather than Arabic numbering, as is the case with printed material. Unlike the conventions used in printed documents, Braille page numbering **must** be located at the upper right-hand

corner of the page [4]

Title pages are presented in a similar manner to that found in printed documents All items are centered horizontally, and an adequate amount of space is inserted between the various items found on this page This is primarily for presentation purposes As is also the case in printed material, distinctions arise between the different types of document For example, it is considered acceptable to locate titular information above other prefatory information (such as tables of contents) in shorter documents such as reports, whereas in text-books or other longer documents separate pages are used for this type of information

4.2.3 Conveying Emphasis

One of the more significant deficiencies in the Braille system, is the inability to indicate emphasis in a comparable manner to the visual alterations used in print At present, the only means to convey emphasis using Braille, is to precede the material to which it pertains with a prefix symbol This is merely present to inform the Braille reader that the material is italicised in the print version

Regrettably, there are very few ways in which this lack of emphasis indicators can be improved upon One possible alternative is to devise newer, comparable symbols to indicate to the reader that the material is emboldened, underlined or otherwise enhanced The lack of means to ensure that the prefixes stand out from the text to which they apply will not be improved by these additions However, their inclusion will yield a more accurate depiction of what the printed version looks like

[4]this is in the case of British Braille The Braille Authority of North America (BANA) prescribes that page numbers be located at the bottom right of all odd-numbered pages

4.2.4 Miscellaneous layout considerations

As is the case in printed material, there are occasions when the standard forms of paragraph layout cannot be employed One such instance is when lists of various types are encountered There are specific standards defined, which clearly specify the means whereby these lists should be laid out in Braille It is traditional to start each new point in a list on a new line, as is the case in print However, unlike the printed version, indentation is not used as effectively, as, should a point encompass more than one line, the run-over begins in cell 1 of the subsequent lines Nested lists begin in cell 7, with, once again, the run-over occupying cell 1, if necessary As will be discussed below, we propose an alternative layout for this form of presentation, which uses the layout conventions prevalent in setting out computer programs

One of the most difficult forms of presentation to represent in Braille, is the tabular based layout As was stated previously the dimensions of the Braille page are such that a significant amount of discrepancy exists between the amount of material which can be displayed on the printed page and the Braille one Consequently, alternative strategies have been devised to present tabular information

Where possible, the procedure adopted in the print version is adhered to in Braille However, there are instances where it is necessary to adjust the layout to ensure that the material is either more readable, or to conserve space Two alternative layouts are possible The first of these is to represent each row of the table as a paragraph, each column being separated by a semicolon The alternative is to use indentation where the lines in a table are too long to permit the normal tabular or paragraph-based layouts This method, could be used in legal schedules, and involves the first column of each row beginning in cell 1 of the line, with all run-overs in cell 7 This mode of presentation would be used in cases where each column contained lengthy passages of text

It is also permissible to use facing pages to depict tables This doubles the

width of the row, and permits wide tables to be represented in the same manner as in print However, this method is only applicable in two-sided printing

4.2.5 Braille mathematical layout

The information surrounding the layout of mathematical material is sketchy and superfluous at best Paragraph 1 of Braille Maths Notation states that

> Set out mathematical expressions are generally Brailled beginning in cell 5 with runovers in cell 7, whatever the setting in print

[otUK87]

As Braille Mathematics Notation is a purely linear setting, there are complex rules governing the locations where line breaks can or cannot be inserted Dot 5 (row 2 column 2 of the Braille cell) is used as the mathematical hyphen when it is necessary to divide an expression at the end of a Braille line, whether or not a division is made at that point in the print [otUK87] Stringent rules apply as to when the hyphen should not be used Consequently, it is often necessary to introduce line breaks at points before the right margin

As was described in Section 1 2 the numbers are written in Braille by preceding the letters a-j with a specific pattern of dots As was also discussed in this section, these letters themselves constitute whole word contractions, hence a further sign must be placed immediately before these (and indeed all) letters to indicate that their meaning is in fact the letter rather than the whole word contraction The first rule, therefore, is that lines cannot be broken in such a manner as to separate the numbers from the numeral sign, or letters from their letter sign, which would make the meaning of the Braille notation extremely ambiguous and unclear

The second instance of where the line cannot be broken is in the case where indices, dashes etc would be separated from the terms to which they apply This

also applies to functions and their arguments, unless the function is unusually long An obvious point at which the lines comprising mathematical expressions cannot be broken, is that after an operator which is joined to the following operand, but separated from those preceding it Many operators are written in Braille joined to the following operands, but separated from those before Consequently, the line is broken before such operators The same rule applies to the opening parenthesis Expressions should not be split immediately after an opening bracket of any kind, as it can render the following sub-expressions ambiguous The converse of the previously described rule applies, namely that the line should not be broken **before** a closing bracket of any kind "It is generally not good practice to divide a short expression (e g $x = 1$) which could be convemently Brailled complete on a new line " [otUK87]

Spacing is extremely important in Braille mathematics As only the horizontal direction is used, it is used both to group, and to separate terms of an expression It is also used to reduce the ambiguity of the various signs found in this form of presentation, as the highly context-sensitive nature of Braille ensures that the same character can be used in many different locations, to signify many diverse concepts Many of the operation and relational signs in Braille are preceded by a dot pattern (dots 5 and 6) All such operators are placed adjoining the proceeding operands, but separated from those before by a space The reason for this is that the prefix just described, is used as an indicator of small Latin letters Therefore, if no preceding space was inserted, the meaning would not be clear

Mathematical punctuation is generally spaced according to the rules followed both in textual Braille and ordinary print presentation It is not customary to leave spacing before, as well as after these signs The introduction of spacing in mathematics presents interesting problems to the automatic production of Braille To a human transcriber, applying their intuition to this difficult task, the spacing follows the natural understanding of the meaning of each individual sign It is obvious, for example, that should the letter sign (also used as

the prefix to certain operators) be conjoined to preceding operands, ambiguity would be introduced However, this is not so to the computer Accordingly, it is necessary to specify explicitly the method of introducing spacing before and after each and every mathematical symbol where ambiguity is possible Consequently, it is only through the evolution of the system, that all anomalies will be eradicated and the ambiguities resolved

Matrices and other arrays pose an equally interesting though different problem in terms of their layout The preferred form of Braille matrix presentation is to use a similar approach to that taken in the printed version, that is to separate the various columns of data by white space, and to place each row in vertical alignment Columns are separated from each other by **at least** one clear cell running down between the columns [otUK87] Elements within a column should be Brailled with their left-most cell in alignment unless they are prefixed by a + or − sign, which is Brailled to stand out Should a column contain elements comprising two or more terms, then they are united with the appropriate letter or numeral symbols being used Spacing is avoided where possible

As has already been stated, the Braille page cannot contain the same amount of material as can be displayed on the printed page Hence, if a matrix is too large to fit across a page, a different strategy is required than that found in printed mathematics In section 10, paragraph 6 of Braille Math Notation we are told that

> A matrix which is too wide for the Braille page may be split between columns with the remaining part of the matrix placed beneath and indented two cells from the start of the matrix Facing pages may also be used for wide matrices if convenient

[otUK87]

Both methods have their disadvantages From the perspective of the automated translator, the former method (the use of indentation) is superior, as

it is easier to format the matrix in this fashion The use of facing pages can prove difficult, as it must be determined which portions of the matrix must go on which page The use of indented columns can make the reading of matrices extremely difficult It can be unclear which material is located in a given column, when the matrices are laid out in this fashion A linear representation of matrices is also possible, though this form is primarily used to present binomial coefficients and other symbols where the particular types of brackets used in the printed version can be represented in Braille Using this method, each row is written in a linear fashion, starting at the topmost and working downwards until the final row has been reached Each row is separated by a designated sign, and the line breaking may occur at any convement point The mathematical hyphen is not used between spaced elements This method will not be used in the TechRead system, as perusal of text books, and conversation with other Braille transcribers has revealed that this method is slipping from common usage and the former non-linear method of displaying this form of information is preferable

The particular instances of how best to display various forms of mathematics have been outlined in the previous paragraphs The following remarks are paraphrased from Section 13 of Braille Math Notation [otUK87] Centred headings should not begin before cell 9 of the line on which they occur The reason for this is to avoid " confusion with set out mathematics" The use of reference numbering in mathematical contexts is a common feature in highly technical documents Braille caters for their inclusion by setting them down (in brackets) in cell 5 of the line on which the equation begins If other references are assigned to other lines of the equation, then they are also set out on the lines to which they pertain, again located in cell 5

There are cases when it is advantageous to use alternative starting points (such as cell 1) rather than the recommended cell 5 Such instances are

- wide arrays

- worked calculations
- to enable simultaneous equations to all start in the same cell if the first equation does not begin in cell 5, or is otherwise preceded by an equation number

A common feature of printed mathematics is to use larger brackets which encompass wide ranging, vertically aligned portions of an equation This is not possible in Braille, and hence the following strategies for the layout of this form of presentation If the right-hand side of an equation contains multiple options as would be located within large brackets in print, then the following possibilities can be adopted

1 The lefthand side, and first option may be Brailled as a complete equation, and the second and further options placed beneath with their equals signs aligned in the same cell as that of the first option if the options are all short enough to fit without runovers

2 The lefthand side of the equation (if it is reasonably short) may be repeated for each option so that the options are each Brailled as complete equations

3 the lefthand side may be Brailled first, followed by the first option, the subsequent options each beginning on a separate line with their equals signs in cell 5, and with all runovers in cell 7

[otUK87]

Other standards exist for the presentation of mathematical material It is hoped that, in future incarnations of this system, it will prove possible to offer the user the opportunity to select which form of presentation they require For example, it should prove possible to switch between the Nemeth presentation

[Nem72], and that specified by BAUK It is also hoped to include some alternative layout standards which will take advantage of newly evolved Braille representations

4.3 Improvements to existing standards

Though it is imperative to include the existing standards in any translation system, it is also extremely important to devise newer means of presenting the same information For example, a new form of tabular presentation was introduced into a version of `MegaDots` translation software, which used a linear form of arrangement to present this highly visual type of material The form of presentation consisted of each row being arranged in a vertical pattern, with each column of the same row being indented slightly below the one above The effect was a *stair-step* pattern, hence the name associated with this style of tabular representation This arrangement was extremely space-intensive, as only a single tabular element was placed on each line, which meant that even a small table could run to more than a page of Braille output This section describes the direction which we believe that Braille should turn towards It outlines the belief that until 8-dot Braille is in common usage, much of those visual features so adored by authors will simply not be available to the blind reader

4.3.1 Textual layout

In the main, the standards described previously cater for the simpler forms of document layout However, they are not designed for the more complex document entities such as itemised lists, and the like They also do not take into account that though visual formatting has advanced significantly over the last number of years, Braille has essentially stood still There is still only one form of emphasis in the Braille encoding, and this is used sparingly A dilemma now presents itself as to whether new signs should be added to the Braille

encoding which could confuse those persons who have been using this form of presentation for their entire lives? The answer, we believe is yes If those who developed newer forms of visual presentation did not do so on the grounds that they might not be used by all, then they would never emerge

It is at this point that we believe the newer 8-dot Braille scores over its more antiquated 6-dot predecessor The increased number of possible characters (256 in all) ensures that the vast array of characters needed to facilitate the incorporation of font attribution information can be incorporated It should be noted, that an ASCII-Braille representation has been devised, so the concept of using these extra two dots is not entirely original Both refreshable Braille displays and embossers are designed to display and print this form of Braille, the only problem is that it is deemed too complicated The need is paramount for a re-examination of this form of Braille to ensure that (though ASCII equivalents should be adhered to) alternative meanings can be attributed to these dot-patterns For example, there is no reason why the existing alphabet could not be extended to include extra font information which could be contained in the lower row of dots

If the 8-dot Braille is brought into more common usage, the scope for increased spatial re-organisation of the textual material is extremely wide Through the use of the font information, a simple side-heading (as described previously) could be transformed into a sectional heading, a subsectional heading, or a subsubsectional heading, in the same way as is apparent from simple visual perusal

More importantly, we believe that further re-examination of the methods used to display such document components as bulleted or enumerated lists, tables and other objects, is long overdue A means to easily distinguish the nesting of list elements, is to employ the strategies often found in computer programming If a programmer wishes to convey the fact that their statements are related to each other at a given hierarchic level, then they will align them directly beneath each other, while a further indentation implies a deeper nest-

ing There is **no** reason why this paradigm could not be applied to the display of Braille lists The bullet point could begin in cell 3 of a given line, and the run-over from this could be set down in cell 4 When the text returned to cell three, the reader would immediately know that the next point had been reached Further, if the next line commenced in cell 5, then it would also be obvious that a sub-list had been encountered This is an extremely simple innovation, however it would make the reading of lists extremely efficient, as the reader would merely have to scan their finger down a particular column on the page to determine their location

An extension of this notion, is that of actually indenting all paragraphs below their heading If one assumes that a given section heading is located in cell 1 of a line, then the beginnings of paragraphs could begin on cell 5, with the main body of their text being located in cell three Again, this would make the location of relevant sectional units in a given document extremely rapid The user would merely have to skim their finger downward through the document outside the boundaries of the running text, until the next heading were encountered Coupled with the font enhancements described above, we believe that these improvements to the Braille standards will make reading of technical material a far more efficient process

4.3.2 Enhancements to mathematical Braille

It is beyond the scope of this discussion to suggest new symbolic representations for the depiction of mathematical symbols in Braille Rather, it is intended here to outline some suggested solutions to the problems of readability often encountered when attempting to apprehend syntactically complex material which has been written in Braille

Personal experience, coupled with that of several colleagues, has revealed that to undertake complex mathematical tasks through the sole medium of Braille takes approximately twice as long as to attempt the same task using the

print medium. The reason for this is primarily that it is extremely difficult and inefficient to re-trace prior calculations, as each particular line must be read until the correct one is encountered. A simple solution to this is the forced introduction of line numbering. Using this method, it would be easy for the blind mathematician to keep a mental record of which number was associated with which particular set of calculations.

One aspect of the standards laid down in British Math Notation [otUK87] is the use of short lines to segregate different lines of calculations. It is **vital** that this obscure portion of the standards be placed more in the common usage.

We believe that the transition to 8-dot Braille will prove as beneficial to the production of clear mathematical content as it will for the improvements to the textual content described above. Currently, the linear representation implicit in the Braille encoding ensures that the readability of mathematical content is quite difficult. Also, we believe that, using the 8-dot versions of the characters, the vertical position can be combined with the horizontal juxtaposition of characters to produce a translation which is closer in appearance to that found in the printed form. For example, if the numbers and letters were re-defined to be located in the middle two rows of the 8-dot cell, then superscripts and subscripts could be represented both above and below the quantities to which they pertained, as is the case in standard notation. It is our belief that the Braille notation should converge with, rather than diverge from the printed notation insofar as is possible.

As blind students become more integrated into mainstream education, the more they will be involved in collaborative work with sighted colleagues. Consequently, if the two forms of presentation are related (if not alike) then all can converse using the same lingua franca.

Another obvious alteration to the mathematical presentation is the manner in which fractional expressions are displayed. The equation $\frac{a+b}{c}$ can be immediately visualised as a fraction, as the numerator and denominator are placed

vertically relative to each other Braille does not presently offer this facility, as fractional expressions conform to the remainder of the mathematical notation and are presented in a linear fashion An obvious alteration would be to write the numerator over the denominator, as is the case in printed notation, thereby enabling the rest of the formula to be expressed relative to this quantity, hence removing any ambiguities as to which sub-expressions were part of the fraction and which were not

It is not possible to increase the font size of the symbols used in Braille As a consequence, the notion of the large brackets described above is not particularly familiar to blind mathematicians One obvious means to convey the nesting of parenthesised sub-expressions is to use indentation Base level parenthesis could be located in cell 1 of the line on which the mathematical expression commences, with the remainder of the content being indented to signify that it is all contained therein The drawback with this use of space is that it will consume more paper than is otherwise used Also, as Braille is more bulky than the corresponding printed output (even in its present form) the increase in paper usage could make both the cost and size of the documents prohibitive

The changes to the presentations described above are essentially simple in nature, though their use in combination is quite unique TechRead, being the first program to integrate the spoken and Braille modalities is attempting to introduce newer and more experimental layouts for already existing standards However, it is only through the empirical testing and the evolution of the system that more innovations will become clear and can be included

4.4 Summary

This chapter has discussed the means of translating LaTeX into Braille Both the methods for deriving the characters, and the standards used in the layout of textual and mathematical material were described The chapter concluded with the description of some innovative standards which can be included in the

system The following chapters describe the alternative medium used in this system, and the details relevant to the implementation of various prototype systems

Chapter 5

Producing spoken output

Though Braille is the most common means whereby blind people can access information of any kind, it is rapidly being superseded by spoken versions of the same material. Owing to the bulky nature of Braille, the ability to transport a small portable computer, rather than multiple volumes of a book has far greater appeal. Consequently, the TechRead system will take advantage of this newer technology. This Chapter discusses the methods for transforming LaTeX input into spoken output. It will be shown how both textual and mathematical content are produced, and how our methods differ from those previously attempted.

5.1 Introduction

The structure of a document is the method whereby an author groups together those portions of work which they deem to be related, or relevant to each other. This structure can be further decomposed into two separate types: implicit and explicit. *Implicit structure* is that portion of structure which conveys to the reader the explicit structure which the author has placed on their work. *Explicit structure*, on the other hand, is the means whereby a document is organised to ensure that those portions of the material are grouped together.

The explicit structure is achieved in the electronic medium, through the use of mark-up languages (such as LaTeX) or through the use of WYSIWYG systems like modern Word Processors The implicit structure therefore, can be said to be those visual attributes which convey to the reader the distinctions between the various hierarchic levels of the document For example, the explicit structural entity marked up with a *\section command* in LaTeX can have the effect of causing implicit structural characteristics to be placed on a certain portion of text

The job of the TechRead system, is to produce a different form of presentation to ensure that the listener can perceive the implicit structure of the document Therefore, the objective in this portion of the system is to devise a new form of implicit structure which is comparable to, but not the same as that visual structure which the author conveys through the use of explicit mark-up The explicit structure of the document is catered for by the internal model outlined in Chapter 3 However, those sectional units also have visual attributes which impart to the visual reader that they are at various hierarchic levels of the document For example, the Chapter titles could be centered, written in a larger emboldened font, thereby conveying to the user the hierarchic importance of this particular unit heading Alternatively, the sectional units which are contained within a chapter could be presented with a slightly smaller font with a different form of visual attribution As the reader descends through the hierarchy, the implicit structure can convey this fact by presenting the structural elements in fonts closer to those used in the main body of running text

It is this form of implicit structure, which must be conveyed through the medium of speech We described in the previous chapter how headings and other structural elements could be conveyed using Braille The spoken utterance is more flexible than the dot-based character representation The prosodic elements of speech can be used in the system to indicate the structure which the author has placed on their work Non-speech sounds will not be used in

the auditory output produced by TechRead The reasons for this are that the non-speech audio (known synonymously as **earcons**) adds distractions to the listener For example in ASTER [Ram94], a fleeting high-pitch beep is used to indicate the commencement of bullet-pointed lists This is not a particularly intuitive mode of presentation, as the user must first learn the meaning of each of the earcons used in the system, and as there is as yet no defined standard for the use of these non-speech sounds, each system employs a different set of noises which they themselves find meaningful

The prosodic features of speech (such as pitch contour, duration, amplitude, rate and rhythm) are all aspects which the users of the TechRead system will find familiar As synthetic speech does not use all the prosodic features found in natural speech, it will prove necessary to emphasise those prosodic elements found in synthesisers to compensate for those which are lacking The objective is not to mirror natural speech, but to achieve a close replica which will be intuitively understood by the listener It is preferable, we believe, to enhance some prosodic elements for the sake of intelligibility, than to achieve natural-sounding, though completely incomprehensible synthetic speech

What is DEC-Talk?

The synthesiser used in the development of this portion of the TechRead system is **DEC-Talk**, [Kla80, AHK87] a device produced by the Digital Corporation The reasons this synthesiser was chosen were as follows

widely used The DEC-Talk synthesiser is among the most widely used in the world Until the emergence of the cheaper, and hence more popular software synthesisers, it was the preferred synthesiser of many people, as it was intelligible

flexibile The primary consideration when choosing a synthesiser, was that it be capable of supporting the prosodic features which we wished to

modify in the system All synthetic speech devices contain the ability to alter their rate, average pitch (f^0) and pitch range However, DEC-Talk contained other prosodic features (described below), which could be altered to produce more varied output

programmable From the perspective of development, it was essential that the synthesiser of choice was easily programmable It was important that text could be sent easily and rapidly to the device, without the necessity to experiment with vast sets of rules, or hardware programming

The means whereby the prosody of a speech synthesiser is altered, is by the inclusion of various control character sequences in the course of the ASCII text which it receives as input For example, the sequence *[ra 250]* adjusts the rate of speech of DEC-Talk to 250 words per minute Table 5 1 shows some of the control sequences used in TechRead Those which are delimited by "[—]" are self-contained sequences, while those which are not surrounded by brackets are used in conjunction with the *define voice* or [dv] command It may be seen from Table 5 1 that the control the programmer has over the synthesiser is quite extensive Indeed, there are other parameters which are not used at this juncture

Coupled with the parameters listed in Table 5 1 to alter the characteristics of the voice, are those which introduce pausing DEC-Talk operates on individual clauses, and once the clause boundaries have been encountered in the text it receives, it alters the vocal characteristics to reflect that boundary, and introduces pauses of various length For example, if a " " is encountered, then DEC-Talk assumes that the end of a sentence has been reached, and adjusts the voice in such a manner as to reflect this The duration of these pauses can be adjusted by various commands Thus, instead of the default length of the pause inserted at a comma, the developer can specify their own length, causing DEC-Talk to cease speaking for that specific length of time

Ctrl Seq.	Meaning
[ra xxx]	Adjusts the speech to "xxx" words per minute
ap xxx	Adjusts the average pitch (f^0) to "xxx" Hz
pr xxx	Adjusts the pitch range on a scale of 0–100 A value of 0 yields a monotone
hr xxx	Indicates the nominal height in Hertz (Hz) of a pitch rise to a plateau on the first stress of a phrase A corresponding pitch fall is placed by rule on the last stress of the phrase
sr xxx	Indicates the nominal height, in Hz, of a local pitch rise and fall on each stressed syllable
bf xxx	Indicates the level of f^0 at which a sentence begins and ends In DEC-Talk, a baseline fall of 0 starts a sentence at 115Hz, and ends at the same level, while a Baseline fall of 20 commences at 125Hz, and falls at a rate of 16Hz per second until the final value has been reached

Table 5 1 DEC-Talk Parameters modified by TechRead All occurrences of "xxx" represent integers

TechRead will not perform any semantic analysis on the input Consequently, the pronunciation of certain words will not sound as well as it would if a human speaker were uttering the same material The system merely sends the output to the synthetic speech device, which is responsible for the overall pronunciation of the text Rather, the pre-defined prosodic parameters of the synthesiser are altered to yield more meaningful-sounding output Accordingly, certain aspects of the spoken utterance will be different depending on which synthetic speech device is in use

5.2 Conveying structure

In Chapter 2 we discussed the fact that the developers of screen access technology for the blind do not harness the prosodic features of speech synthesisers This means that the listener can gain no idea as to the nature of the hierarchic level, or the visual attributes of the text they are reading TechRead will indicate the explicit structure placed on the document using a new model of implicit structure The aim is to produce a representation, which displays aurally what is conveyed visually through the alterations in font, and the spatial location of portions of text on the printed page The following paragraphs describe how the explicit structure is transformed into an alternative model For the purposes of the discussion, the default vocal characteristics are taken to be

Rate Set at 250 words per minute Many blind listeners familiar with listening to the output from synthesisers, can listen at far higher speeds, however this value was taken as a good point, as it is neither too fast or too slow

Average pitch This is set to 120Hz The clearest of the DEC-Talk voices (mnemomcally known as "Perfect Paul") uses an average pitch set to this value

Pitch range The pitch range used as the default is 50 considered a good point from which to start, lying as it does in the centre of the permitted scale

Hat rise The default hat rise is set to 40Hz This ensures that the frequency rises a nominal 40Hz on the first stressed syllable, and descends by the same amount on the last stressed syllable of the clause

Stress rise The Stress rise parameter indicates the nominal height, in Hz, of a local pitch rise and fall on each stressed syllable This rise-fall is added to any hat rise or fall that may also be present

Baseline fall The baseline fall of the default voice is set to 80Hz The result of this setting is that the fundamental frequency begins at a level 40Hz above, and ends 40Hz below this frequency

The default vocal characteristics are flexible to the extent that they can be altered by the user to produce output which they find more pleasing Consequently, all modifications described in the remainder of this chapter are relative to any starting voice For example, `DEC-Talk` provides a voice which sounds like a child If a listener prefers to use this voice, then the modifications will take effect relative to this starting point No absolute alterations are built into the system Also, the vocal characteristics used as the defaults, will be adjusted by TechRead to ensure that modifications are possible For example, should a user choose the default voice of "Perfect Paul", then the characteristics will be adjusted to ensure that alterations can be made Like all other pre-defined voices provided by `DEC-Talk`, the parameters have been set by the developers to ensure maximum performance TechRead will need to adjust these to ensure that the relative changes incorporated in the system will be observable to the listener For example, the volume of the default voice is lowered, to permit an increased amplitude to convey amplitude which can be used to convey emphasis

In the discussion which follows, the various changes which TechRead makes to the vocal presentation are given in terms of the specific alterations which are made to `DEC-Talk` It should be noted, that these alterations are based on general principles of prosodic alteration For example, if a rate change of 10% is presented here, then it is a relative change, which produces a desired

effect when applied to DEC-Talk's output When this system caters for more than one synthesiser a mapping will be needed between the prosodic model used, and the specific alterations needed to produce similar results for different synthesisers

The discussion which follows is also given in terms of the *article* document class, which is one of the default options found in LaTeX In this document type, Chapters are not permissible, and the first sectional unit allowed is designated as "section"

5.2.1 Sectional units

As is the case with the visual modality, the prosodic alterations needed to convey the structure of a document, are made relative to a starting (or general) set of speech characteristics Therefore, in terms of the internal model described in Section 3 1, the voice used to convey the majority of the running text (known as running-voice) would be stored in the *global settings* node, and all prosodic changes would be relative to this starting point There are no specific default values for this running-voice, as each listener has their own preferences which will always be slightly different from those of other people The alterations described below, which affect the verbalising of structural elements, are all relative to this running-voice

In many documents, the first section of note is the title page TechRead has unique characteristics designed to impart this specific content to the user If we assume that the *global settings* node in the internal model is hierarchic level 0, then the title page could nominally be said to be at this level Accordingly, the vocal characteristics have been modified to ensure that the user can gain an insight into the different components of the page

To speak the titular information, both the rate and average pitch are slowed by 10% to 225 words per minute and 108Hz respectively Coupled with this, the pitch range is narrowed to 40% of the average pitch, while the baseline fall

is reduced to 60Hz This has the effect of producing a weightier, less excited voice, and imparts to the reader the fact that the material they are reading stands out from the main body of the text The name(s) of authors are spoken, using a lower average pitch than the title A further reduction of 10% is used to present the fact that this information is at a lower level than the title The reduction in the size of the font which is often used to convey this information visually, is imparted to the listener by the perceptible deepening of the voice A further slight reduction is used to indicate that the date (if present) is even less important than the author However this is a purely nominal amount, of between 3 and 5%

It is customary to leave some white space between the various components of the title page This is achieved in the spoken version, through the introduction of longer pauses between the various elements The equivalent of a double-paragraph pause is placed between the titular information, and the name(s) of the authors Smaller pauses are introduced as necessary to indicate the spacing found on this page

The next section of note in most technical documents, is the abstract If such a level existed, this particular Section would be placed at a point between the title page, (designated as level 0) and the first sectional unit, typically given a place in the hierarchy at Level 1 Therefore, for the purposes of this discussion, let us assume that the abstract is to be found at level 0 1 To convey the fact that the reader has descended somewhat in the hierarchy, the vocal parameters are adjusted accordingly Firstly, the average pitch is reset to 10% below that of the default voice Coupled with this, the pitch range is widened to 50%, and the baseline fall is increased to 80Hz This produces a slightly more animated voice than is evident, when the overall document title is being read After this has been read, the normal inter-paragraph pause is introduced, before the rate and average pitch are reset to their default settings at which point the actual text of the abstract is spoken

The methods described above are also used to convey the sectional titles

However, these particular headings also have various other attributes associated with them A dilemma in the design of the prosodic model was how much information should be spoken to describe the hierarchic level of the sectional heading For example, should the heading be preceded by the phrase "section " or simply by the section number? The former mode of expression was chosen The reason for this, is that if only the section numbers were used, the utterance would be extremely short. Imagine the heading "1, Introduction", which is simply 5 syllables in length and might be missed by the listener

The solution was to introduce the keyword *section* into the utterance, thereby lengthening it slightly, and also conveying the fact of the hierarchic level accurately to the listener Another reason for the inclusion of the *section* keyword, is to indicate to the listener that they are moving to a new and more significant portion of the document Lengthier pausing is also introduced around this type of heading The sectional headings stand out visually, by virtue of the increased font size, and other alterations applied to it The pausing (coupled with the prosodic alterations described below) inform the user that the text which is about to be spoken, is significant in some way, and stands out from the surrounding body of the document

In order to convey the fact that the hierarchic level is dropping, the prosodic aspects of the voice are altered Once again, the rate is slowed by 10% to yield a slower, more measured utterance The average pitch of the voice is decreased by 25% to distinguish the sectional title from the remainder of the text The pitch range is maintained at 50% of the average pitch, however the various parameters relating to the stress rise, hat rise, and baseline fall are increased The effect is to produce a slower, though more ammated voice This is akin to increasing the font size in such headings If the synthetic device supports it, the ideal inclusion for this type of material is to also increase the amplitude of the voice A common prosodic feature for emphasising portions of the utterance is to also increase the amplitude slightly. This is primarily why the various stresses are increased in the representation of the sectional title

Just as the degree of visual emphasis decreases as the hierarchic level deepens, so also the vocal characteristics converge with the running-voice Subsectional headings are spoken as "x x", where "x" is the sectional number An obvious question is why the keyword "section" is not included in this utterance The belief is that, just as its absence would make the utterance of the sectional heading too short, its inclusion in subsectional headings would render their utterance too long The number of syllables in the phrase "section 2" is exactly equal to the number in "2 dot 2" Also, one of the objectives of TechRead is to reduce the amount of extra, and often superfluous information which is relayed to the listener If the phrase "2 dot 2" is heard, it can be assumed that the user is capable of inferring that they have reached Subsection Two of Section Two Thus, by omitting the redundant information, the need for extra recall is also reduced, and the listener can concentrate on assimilating their whereabouts in the document, rather than redundant keywords

The prosodic alterations used in these subsectional headings are also less pronounced than those outlined in the context of sectional headings For example, the average pitch is not reduced by the same amount, but rather is kept closer to that of the running-voice Also, though the animation is lessened, the enhancements are adjusted to convey that, though this heading is not quite as important as others, it is still a sectional divider

An important feature of all the prosodic alterations used to convey structure is the lengths of the pausing In order to represent the subsectional level, the pause is noticeably shorter than that used in the conveyance of the sectional title The reason for this is that, as it does not stand out in as striking a manner visually, so should it not be afforded the same vocal separation from the viewpoint of the listener The intention is to show, through the convergence with the majority of the content, that the user is descending through the hierarchy, and that the heading being spoken is subordinate to that of the container section

The subsectional headings below those already discussed, are shown to diminish in significance by the lack of visual enhancements attributed to them

Indeed, it is often the case that lower-level headings are only visually distinguishable by virtue of their location on separate lines Accordingly, the spoken representation of these subordinate structural elements rapidly approaches the prosodic feature of the normal running-voice Subsubsectional titles, for example, are spoken as "x x x" for similar reasons to those described in the context of subsectional numbering In these lower-level cases especially, the inclusion of redundant keywords would make the pronouncement of the sectional indicators inordinately lengthy The addition of the extra level to the utterance, adds two syllables to the speaking of the section number, which is considered to be long enough There are only minor changes to the vocal characteristics As these are extremely minor headings, paragraph pausing is used to separate the heading from the text preceding and following it No alteration is made to the voice itself, just as no alteration is made to the visual appearance of the heading

Some miscellaneous aspects of the prosody found in the conveyance of structural information should be discussed at this juncture Firstly, one of the keystones on which the model is based, is that of balance This is best described in terms of the pausing It is extremely important that the pauses inserted both before and after the information to which they relate, be of the same duration This ensures that the user can rapidly learn to anticipate the type of structural element which is about to be read, by virtue of the pause which precedes it In another context, it is important that, should the vocal characteristics be altered by a certain amount, this amount be relative to the reference point from which they were calculated As was described previously, the default average pitch value used for the purposes of this discussion is set at 120Hz As a consequence, the 20% reduction in average pitch used to convey Section headings causes the new value to be set at 96Hz This is extremely distinguishable, as the two frequencies are a relative distance apart Also, if the pitch was altered by the user of the system, then by using relative changes the effects would be comparable, if not the same

A final point on the conveyance of structural information is that the pausing

is not additive. Consequently, if the end of one Section of a document is actually located at the subsectional level, the pause inserted **before** the announcement of the next Section heading would be that required to separate this element from the surrounding text, and not that achieved by adding the subsectional pause, and the sectional pause together.

5.2.2 Textual alterations

The previous section described how the structural elements of documents were conveyed to the user, using prosodic alterations. However, there are other, equally important aspects of presentation which can be conveyed using changes in the characteristics of the voice. These presentational features are obvious to the visual reader, but using existing access technology, they are not so readily ascertained. For example, the use of emphasis is not currently conveyed to the blind reader, causing all material to sound alike. The following paragraphs describe how different aspects of visual presentation are presented to the listener using changes in the prosodic features of the voice.

The characteristics of the running-voice were described above, and were instantiated to ensure that it did not sound too monotonous. One of the major drawbacks when listening to lengthy passages of synthetic speech, is the tiring effect brought about by the lack of alteration in the sound of the output. Consequently, one of the first considerations was to determine a set of parameters which, when applied to the voice, made listening to it for long periods of time bearable. To achieve this, the running voice contains a pitch range, and other features which ensures that it modulates (insofar as is possible), thereby relieving the tedium of listening to it. What this software attempts to do, is to provide the listener with an overall improvement to the general sound of the default vocal characteristics. Therefore, it was decided to leave the actual processing of the individual words, and their concatenation into meaningful clausal structures to the synthesiser itself.

Default text, therefore, is spoken using the running-voice The clause boundaries cause the insertion of pausing, as it is considered unrealistic to hear the punctuation spoken, rather than interpreted For example, when using some screen access technology, pausing is not used when clause boundaries are encountered in the running text, rather their names are incorporated into the spoken output The only alteration which is made to the manner in which DEC-Talk treats the clause boundaries, is to (where necessary) modify the pausing which it inserts

Three levels of pausing are to be found in the course of running text, paragraph pauses, sentence pauses, and clausal pauses Each of these diminish in length, in accordance with the hierarchic level with which they are associated It is possible to directly control the amount of silence which each of the clause boundaries causes to be inserted, by specifying these amounts in milliseconds using DEC-Talk-specific control sequences However, this level of control is avoided, as such facilities are not generally offered by commercially available synthesisers This, in fact is a feature of the TechRead ideal, to keep the modifications as generalised, and as minimalistic as possible

There are many different ways in which an author can alter the visual appearance of their text, to ensure that material they consider more important than the surrounding body of material stands out For example, in many textbooks in which computer programming languages are taught, it has been traditional to use a typewriter font (`like this`) to indicate that the material presented thus is actually the syntax which should be entered into the computer

Alternatively, for emphasis, the same author could use emboldening, underlining, or italicisation to enhance their document It is only since the development of systems such as ASTER [Ram94], that the notion of reflecting these visual changes in speech has evolved The use of different forms of emphasis is justifiable, by virtue of the fact that they are visually striking However, as the speech signal is transitory, and only the memory remains after the utterance

has been completed, a decision as to whether to use different-sounding voices for different forms of emphasis had to be made

It was decided to use a single alteration in the vocal parameters to reflect all types of emphasis The reason for this is primarily that it is believed that to keep track of many different and distinctive voices would be extremely difficult and confusing for the listener For example, if the average pitch and pitch range were altered to one set of values to reflect emboldened text, and still another when underlined material were encountered, the belief is that, over the course of the document, the listener would not be able to easily recall which was designed to signify which Another reason for the choice of a single voice to indicate emphasis, is that in the visual modality, it is used to ensure that the material written thus, stands out Consequently, the primary consideration when producing an aural equivalent was that the material spoken in an emphasised manner was discernable from the majority of the utterance

The means by which the voice is modified to reflect this emphasis is quite novel In previous systems, the voice was either left unaltered, or an entirely different, and unrelated voice was used, that is, a different person (in the synthetic sense) is used to indicate text written using different forms of emphasis) For example, ASTER [Ram94] uses a more robotic and monotonous voice than normal to convey the fact that typewriter (and by inferrence computer-based) material had been encountred TechRead, on the other hand uses the same "person" to represent the entire gammit of the vocal spectrum The alterations used to convey the emphasis are

1 widen pitch range and set to 80%

2 increase the accent height by increasing the stress rise, and hat rise respectively

3 minimise smoothness, by setting it to 0

4 maximise the vocal richness, setting it to 90

5 where possible, increase the amplitude

The objective in adjusting the parameters thus, is to derive a voice which sounds both animated, and weighty The widening of the pitch range, ensures that the contour fluctuates over a broader range of frequencies than occurs in normal text Also, the increase in both the hat rise, and stress rise respectively, ensures that those syllables which DEC-Talk emphasises in the course of the speech, become even more pronounced than is the case when the default running-voice is in use The concept behind this voice is to produce output which informs the listener that differently formatted, more striking material had been encountered It is not intended to produce an entirely different vocal sound As a final caveat to the discussion on whether different voices should be used to convey diverse forms of emphasis, it should be pointed out that among the alterations permissible in the system, is the ability to alter the sound of the voice to reflect different forms of emphasis While some users will prefer to use a single voice to indicate emphasis, others may prefer to use several different voices, and hence the need for this facility

The discussion of prosodic alterations to date has focused on those changes which are visually striking, by virtue of the differences in the appearance of the material However, there are also other, more subtle methods whereby an author can ensure that portions of their material be noticed One such technique is to place the content in parentheses The best analogy to describe this form of presentation, is to state that is the written equivalent of the spoken *aside* Usually, the speaker will increase the rate of speech, and reduce the overall amplitude of the utterance when communicating thus The use of parentheses, is usually reserved for the interjection of some extra material, or to insert a comment regarding some aspect of the text

It has been decided to convey the parenthesised material aurally by increasing the rate by 20%, decreasing average pitch by a comparable amount, and compressing the pitch range The stress rise and hat rise are also reduced, however the baseline fall is increased The effect is to produce a rapid, low-pitched

voice which *mutters* (though clearly and intelligibly) the appropriate material The conceptual basis for this technique is to simulate the visual reader's eye skimming over the parenthesised information, and then returning to the main body of the document The vocal effect is achieved by the alterations just outlined, as the transitory speech signal flows more rapidly past the listener, resulting in a rapid return to the main body of information

A similar technique is used in the presentation of both footnotes, margin notes, and cross-references It has already been demonstrated how the interface will cater for the movement to and from these document objects The vocal equivalent of glancing at these types of elements, is to speak more rapidly than the running-voice The same set of parameters discussed in the context of parenthesised information is used to convey these types of note The reason for the comparable treatment of notes and parenthesised information, is that footnotes and marginal notes are merely alternative ways of displaying the *aside* discussed above Though the material found in notes of various kinds is often longer than parenthesised content, the techniques for dealing with them should be comparable, as they are essentially the same thing

5.2.3 Lists, and other miscellaneous environments

The means for transforming the generic paragraph of running text were discussed in a previous section However, there are other specialised forms of the paragraph which should be considered separately from the generalised case The first of these are lists There are several forms of list available in LaTeX, the bullet pointed list (produced by the *itemize* environment), the numbered list (produced by the *enumerate* environment), and the descriptive list which is generated by the *description* environment The first of these types precedes each element in the list with a bullet mark The second numbers each element of a list sequentially, while the third permits the author to insert arguments to the items

A descriptive list is used in this document in several locations, the most recent being at the point where the default running-voice was described The descriptive list, therefore, can be said to substitute the users' own argument in place of the standard bullet point All lists are also offset from the main body of the text, and indentation is used extensively to relate the text of the various elements For example, in a multi-line list item, if the list element begins at position x on the page, then all subsequent lines will be located at position $x + y$, where y is some pre-defined indentation length Another feature of lists, is the fact that nesting is permitted LaTeX permits up to four levels of nesting [Lam85], which is usually sufficient for most needs The use of indentation, and different number indicators, conveys to the reader the nesting of the various lists For example, in a nested enumerated list, the outer elements are numbered using a numerical sequence, while deeper levels can be marked with alphabetic sequences, or alternatively Roman numerals

The means whereby this distinctive form of presentation can be conveyed auditarily, is highly dependent on the type of list For example, there is a difference between the bullet-point list and the enumerated list, in that one has a visual indicator which separates the various elements, whereas the other uses a feature which can be easily spoken Consequently, they must be treated differently Other systems have attempted to indicate bullet points by inserting fleeting musical tones to indicate to the listener that the subsequent material constitutes a list element [Ram94] TechRead uses a different model, based entirely on the prosodic component of speech to convey this information

The key to the understanding of the method of presentation, is to alter one's perspective of the list from an indented, visually striking portion of material to the sound of the voice when one speaks a list of items Let us assume, for example that a list of ingredients is being read from a book Either the readers' inner voice [RP89], or their actual voice will interpret the list as a series of points A rising fundamental frequency is observable at the conclusion of all but the final elements of the list For the last ingredient, a falling average pitch

is discernible, thus indicating to the listener that the verbalisation of the list has been completed The rising average pitch, is akin to that found when the exclamation-mark is inserted as a clause boundary in normal running text

With this spoken model in mind, it was necessary to experiment with the parameters available for use with DEC-Talk, in order to produce a synthesised sound which closely replicates the human model of speaking lists of elements The first aspect resolved, was the fact that a degree of audible separation should be inserted between the list, and preceding and subsequent material However, as the list is itself a specific instance of the paragraph, this pause should not be overly long Accordingly, a pause is inserted both before and after the list of the same length as separates paragraphs of normal, running text

In order to impart a more measured delivery, the rate of speech is slowed for the purposes of reading lists This decrease in the reading rate is a nominal amount, of between 5–7% The reason for this is that the list is not as visually striking, nor as hierarchically significant as a sectional heading Consequently, the normal running-voice is not altered significantly to convey this type of paragraph At the end of each list item (save the last) a `[']` control sequence is inserted This command has the effect of causing DEC-Talk to treat the text as a clause ended by an exclamation mark, and results in a perceptable rise of average pitch to reflect it The final list item is treated like a sentence, and is passed to DEC-Talk which handles it using its own internal sentence-processing rules

The techniques described for the verbalisation of subsectional units, are employed with nested lists That is, the rate is once again slowed, and the same inter-paragraph pausing inserted As is the case with the nesting of sectional units, the notion of balance is extremely important However the pausing is not cumulative, and should a nested list end a nesting one, only one inter-paragraph pause will be inserted before a return to the normal running text

In order to differentiate between the bullet pointed list, and both the enu-

merated and descriptive forms, an alternative strategy is employed Both the enumerated and descriptive list are treated similarly, and are even more specific instances of the generalised approach, which is applied to the bullet pointed version The essential difference between the generalised case and the two specific instances, is that the latter contain material which can be spoken, rather than a graphical symbol to distinguish different elements Consequently, a greater amount of information can be aurally presented to the listener when these types of list are included in a document

The overall strategy for dealing with both enumerated and descriptive lists is the same, that is, the same inter-paragraph pause surrounds the entire list, and the same decrease in the speaking rate is used However, the addition of the non-graphical parameter is spoken first, with a short pause inserted between the descriptor and the main body of the item The pause used is approximated to that used when a comma is encountered in running text The intention is to separate the element descriptor from the main body of the list, but to maintain the connection between it and the material to which it relates An increase in stress rise and hat rise are also used when the descriptive parameter is being spoken, thereby conveying a little more emphasis than the main body of the list, and to ensure that it stands out in the audio sense Another form of specialised paragraph layout is the use of centering This is primarily used in quoted passages of text, though there are many other instances where it can occur The main reason why this portion of content stands out, is because of its physical location on the page

Typically, all the visual enhancements used in regular paragraphs can be present in centered paragraphs Consequently, there are very few alterations made to the spoken representation All that is intended is to indicate to the listener that some form of alteration has occurred in the layout, but that the text of the paragraph is essentially the same style as the surrounding material As is typical in the audio presentation, a paragraph pause surrounds the centered text This is merely to indicate that it is offset from the main body of the

document Also, the rate of speech is slowed, which conveys to the user that centered material is being heard

Quoted passages of text provide an interesting problem In many cases, the quotation is attributed to an individual, and in the case of non-technical material such as literary works, it is often the case that dialogue is encountered The dilemma thus caused, is whether to attempt to speak the quoted material in a voice, whose gender matches that of the speaker, or to merely modify the vocal aspect to indicate that a passage of quoted material is being read Without semantic analysis of the text, and a comparison of the name with a database of existing names and their gender, it would be impossible to consider matching the speaker with a gender-specific voice Therefore, it has been decided to merely use the alterations described in the previous paragraph to convey quoted material Coupled with the decrease in the rate, and the other alterations permissible to convey the font alterations, it is believed that this will be sufficient for the present Future versions of TechRead may incorporate some semantic analysis, or the comparison of the names associated with quoted passages to produce a clearer and more accurate representation of this type of material

5.3 Mathematical prosody

The previous sections demonstrated how the prosodic component of speech is utilised to best effect in TechRead to present both the textual, and structural content of technical documents Another, equally significant portion of the material often found in these types of document, consists of syntactically complex data which is not readily conveyed using spoken output What is needed is a strategy which transforms the complex mathematical structures into a rich, linear rendering which expresses the visual cues in the printed equivalent There are two issues which must be addressed in audio-based version of the material

Firstly, is the notion of how to actually verbalise the syntactically complex

material which is only readily understood by interpretation of the visual cues associated with it Combined with this, is the need to incorporate superficial views of the data into any system, which involves the examination of the structures of the material and their depiction using the spoken modality

The following paragraphs outline the means employed in TechRead which will accomplish these tasks We show how our conclusions of what should be best presented in order to gain an overview of the material were formulated, and also describe the prosodic model used to verbalise the mathematical expression as a whole

5.3.1 What to speak

The methods used in TechRead for speaking the constituent parts of a mathematical expression will be discussed in the next section However, before embarking on this discussion, it is necessary to describe the means whereby the more superficial views of the formulae are imparted By dint of its unique form of presentation, mathematics contains the inherent visual cues which readily distinguished it from the surrounding textual content This fact is not so observable when one is listening to technical material, particularly when existing forms of screen access technology are being used To improve this, is one of the key design objectives in the TechRead system, and we believe that, when implemented, our solutions will ensure that mathematics and other syntactically complex matter can be read as easily when using spoken output, as with the visual modality In order to determine the exact nature of what should be imparted at these diverse levels of perusal, we conducted a pilot study, which, it was felt, would aid in establishing what sighted people saw when reading mathematical expressions

The nature of the experiment

In order to establish what sighted people saw when perusing mathematical expressions for various fixed lengths of time, an experimental strategy was devised, with forced them to examine the expression for pre-determined, fixed periods of time, after which they were requested to answer questions pertinent to what they had perceived Fifteen mathematical expressions were used in the experiment, and displayed three times to the subjects, for a period of 5 seconds, 15 seconds and 40 seconds respectively These figures were chosen to yield a total viewing time of 1 minute In between each equation, the subjects were given 40 seconds to write down the answers to their questions The reason why different time intervals were utilised, was to simulate a reader glancing at an equation, looking briefly at the same material, and finally studying it in depth It is realised that the 40 second period could be considered insufficient for the sighted reader to gain an accurate impression of the material However, coupled with the previous viewings, we believed that a total viewing time of 1 minute would be tantamount to allowing them to view the material until they were satisfied

The hypothesis we wished to test, was that, at more superficial views of the material, the larger operators, (such as $\sum$, $\int$ or lim) would stand out visually, and hence be recogmsed We also conjectured that, as the time interval increased, the smaller constituents of the expression would become more recognisable to the subjects The objective is to use the knowledge of what readers using the visual modality recognise, and to apply this to the production of accurate portrayals of the more superficial views of the expressions

The materials and subjects

The materials for this pilot study were chosen from various mathematical textbooks, which are in common usage at first-year Undergraduate level at Dublin City University Though these expressions are above the level of mathematical knowledge expected at pre-university levels, it was felt that they offered

a challenge to the subjects Also, examination of the textbooks used in pre-university syllabi revealed a distinct lack of usable expressions Included also in the materials used, were two expressions used by Raman in ASTER [Ram94], as examples of his methods of artificially synthesising spoken mathematics The complete list of equations used can be found in Appendix B, along with the sample paper used in the experiment

The subjects were drawn from all parts of the University, and no stipulation was made that they have any mathematical experience The reason for this, was that TechRead will be used by students of varying abilities As a consequence, the importance of perceiving how sighted people of equally varying mathematical literacy viewed the material would prove interesting A total of 32 people participated in the experiment, consisting of undergraduate engineers, physicists, postgraduates in mathematics and computer science, and some staff members with a background in statistics or management science The exact break-down of the subjects is given in Table 5 2

As can be seen, the mathematical experience of the participants ranged from highly competent, to those who did not use this type of material regularly For example, it can be assumed that while the Engineers present would be highly mathematically literate, the Business Studies student would not have the same exposure to this form of technical data, and consequently would not be as familiar with it

The procedure

The procedure used was in essence a recall-based one, involving, as it did the subjects viewing the material, and then writing their answers on the papers provided, after the viewing time had elapsed All those present were requested not to write anything, **until** the viewing time had ended The equations were presented on slides, which were projected onto a screen using standard technology The viewing time, and the time allowed between the presentation of each new

Category	Quantity	Percentage
Engineering	9	25 %
Mathematics	8	25 %
Computing	3	9 %
Physics	3	9 %
Science	3	9 %
Business	1	3 %
Not Given	5	15 %

Table 5 2 Experiment 1 Subjects' Area of Mathematical Expertise by Category

equation was carefully controlled, to ensure that consistency was maintained Finally, in order to preclude the affects of fatigue, a gap of approximately 5 minutes was left between each presentation of the complete set of equations Before beginning the experiment, the exact nature of the proposed format was described to the subjects **No** sample equations were used in this pilot study

The participants were given prepared answer papers, and asked four questions on each of the 15 equations An extract from a sample answer paper can be seen in Appendix B The same questions were used for fourteen of the equations, the only exception being the fifteenth, which consisted of a matrix In summary, the participants were asked to

1 rate the equation according to how difficult they perceived it to be

2 specify which mathematical objects were present in the displayed equations These mathematical objects were presented using their English, descriptive names

3 Indicate which mathematical symbols were present in the equations

4 rewrite the equation in their own words

The various results, and their impact on the form of delivery used in TechRead are given below

Results obtained

In general, the results obtained from this experiment verified that, as the length of time increased, people retained more of the material Some interesting anomalies were discovered which, though having no impact on the spoken presentation of this material are unusual phenomena in themselves

Fourteen out of the fifteen expressions were analysed, and the results obtained from three of the four questions asked The equation omitted from the analysis was equation fourteen of appendix B, as it was deemed too complex to both analyse and present Indeed in hindsight, this formula should have been excluded from the presentation, as the time afforded by the various presentations of the material was inadequate to absorb an equation of this complexity

The reason that Question 4 of appendix Bwas not considered in the analysis, was that, unexpectedly, the participants did not use English words to reproduce the equations Rather, mathematical symbols were used to express the remembered view of what the various formulae consisted of The objective when including this question, was to attempt to discern how people would describe the equations in English, not how they remembered them in terms of mathematical representation, as a consequence of which it was not included in our analysis

The first, and most readily observable fact is that the difficulty rating which the participants ascribed to the equations dropped significantly over the course of the three viewings As can be observed from question one of appendix B the participants had four possible levels of difficulty to choose from Accordingly, a scoring scheme of 1–4 was employed to quantify the difficulty of each expression, where 1 represented the participant's belief that the material was simple, while four implied that the equation was extremely difficult The average difficulty for the equations at 5 seconds was 2 662, with a standard deviation of 2 163, while at 40 seconds the average difficulty was 2 079, with a standard deviation of 0 859 These difficulty ratings are relatively insignificant in and of themselves,

Mathematical Notation	5 seconds	15 seconds	40 seconds
Differential	86 %	88 %	55 %
Fraction	48 %	55 %	61 %
Polynomial	15 %	24 %	16 %
Matrix	91 %	94 %	88 %
Function	55 %	63 %	62 %
Subscripts	14 %	23 %	29 %
Superscripts	14 %	29 %	43 %
Trigonometry	39 %	36 %	36 %
Summation	54 %	68 %	63 %

Table 5 3 Descriptions correctly recognised in equations over time

however their use can explain (at least in part) the reasons why more of the material is not always observed, even though the time interval is increased If the participants still viewed the formulae as complicated at the 40 second time period, then it is doubtful whether they would observe the finer points as readily, being still engrossed in discerning what was actually presented

At the five second interval, the elements which were noticed by people were those larger scale, and more prominent aspects of the material This fact is best illustrated by the fact that, as can be seen in Table 5 3, 91% of those participating in the pilot study observed that a given matrix was present in an expression This is hardly surprising, as the visual presentation of a matrix is quite striking When this fact is extended to include such terms as differentials, summations and functions, which were recognised in quite large numbers Table 5 3 shows that 55% of the subjects categorised the equations as differentials, while 54% noticed the presence of summation signs On the other hand, very few of those who answered the questions noticed the presence of smaller components when viewing the material for such a brief period of time Only 14% perceived the presence of both superscripts and subscripts, thereby suggesting that they do not stand out when only a brief glance at an equation was permitted

It should be pointed out, that the figures just presented were obtained when the participants were asked to indicate the presence of these items, based on the English terms which have been associated with them When presented with the mathematical symbols (question three of appendix B) they fared better As Table 5 4 illustrates, the integral symbol ($\int$) was the most prominent, as 91% of the participants observed its presence at the five second time interval This is followed by the derivative symbol at a recognition level of 76%, again at the five second interval As the time interval increased, most recognised symbols increased However, the perception of the derivative symbol dropped to 52% at 40 seconds We can only speculate that at this time interval, this particular symbol had lost its significance and was lost in the remainder of the expression's content

An interesting observation from our analysis of these results is that the percentage of those who perceived various symbols remained quite constant over the first two time periods For example, Table 5 4 shows that 1% of the participants observed a summation sign at the 5 second interval, while at 15 this had risen only to 8% At the 40 second time period, a final figure of 84% of participants found the summation sign to be present Though the results are not as staggeringly dispersed for other symbols, the same observable trend can be found In the case of functions, the first two time periods account for 16% of those present observing their presence, while at the final viewing, 48% discerned that they were part of the expressions

Producing superficial views

The results of this experiment proved extremely useful in determining what information should be conveyed to the listener at the higher, more superficial views of the equations It is realised that the interface will play an important part in the presentation of mathematical expressions This is because speech is a serial form of communication and so it cannot afford the listener the same level of control over the information flow as is available to the visual reader,

Mathematical Symbols	5 seconds	15 seconds	40 seconds
Integral	91 %	95 %	86 %
Derivative	76 %	76 %	52 %
Summation	1 %	8 %	84 %
Square root	5 %	5 %	48 %
Function	16 %	16 %	58 %
Limit	20 %	26 %	88 %

Table 5 4 Mathematical symbols recognised in equations over time

using an essentially parallel form of information access The intention when presenting higher-level views of the equations, is to indicate to the listener the approximate nature of the content

At the highest level, TechRead's spoken representation of the mathematical expression will be to announce its presence in a document Consequently, the phrase *equation found* will be heard by a listener when pursuing a continuous reading strategy or moving through the document using any of the methods outlined in Section 3 2 At the next, and lower hierarchic level, a running paraphrase of the formula will be presented It was for this purpose that the pilot study was undertaken, that is, to determine what information should be presented to the listener at this level of perception

Using the knowledge gained from analysing the results of this experiment, it is clear that the minute details of the equation (such as superscripts or subscripts) is not recognised by sighted readers, and as a consequence it will not be incorporated into the superficial spoken "glance" What will be presented here, is a general description of the material, in terms of the terms present For example, if the $\int$ symbol is part of an expression, then it will be included in the running paraphrase, as it is one of those mathematical elements which sighted readers perceive to be important at this level of perusal

A simple formula will illustrate the means of verbally presenting the mathe-

matical content at this level. The formula $\int \frac{a+b}{c+d}$ contains two terms; namely an integral sign, and a fraction. Accordingly, the superficial running paraphrase must reflect this fact, producing a verbal rendering of this type of material such as *integral of a fraction.* The means whereby the paraphrase is determined, is to examine the nesting of the expression, and to incorporate an amalgamation of the outermost terms into a brief, descriptive phrase which informs the listener as to the nature of the mathematical content. The means whereby the actual content of the expression is delivered, is described in the next section.

The presentation of superficial views of the mathematical content of technical documents is a difficult process. Previous work [Ste96] has tried such diverse methods as playing sequences of musical notes, in a particular rhythm pattern to affford the user of their systems the ability to deduce the content of mathematical expressions from a superficial "glance". The model used in TechRead is a totally new approach to the presentation of higher-level views of mathematical material. It was demonstrated in Chapter 2 how ASTER [Ram94] utilised the notion of *variable substitution* to convey this higher level audio depiction to the listener. TechRead's mode of doing the same thing is, we believe, a more intuitive process, as it involves the examination of the LaTeX mark-up to verify the existence of higher-level structures which are apparent to sighted readers at a glance.

The use of the experimental data was a key factor in determining the optimal strategy to employ in incorporating this facility into the system. The consistency between the presentation of the higher level views of the material, and the complete rendering which is described below, is a useful tool in ensuring that the users of the system do not have to learn an entire new system of representing higher level structures. The derivation of a brief, descriptive phrase can be used by the listener in determining whether the material is relevant or not, and based on this, they can either descend hierarchically and read the material at a more detailed level, or proceed to the next passage they deem relevant.

5.3.2 How to speak mathematical expressions

The prosodic model used in this system [Mon91] is based on an application of that already discussed in the context of producing spoken output from English-based material The keystone on which the presentation of syntactically complex material is based, is that of conveying the structure, and grouping of an expression using pausing and alterations in the speaking rate Before embarking on a description of the methods used to convey diverse mathematical constructs and LaTeX environments, it is first necessary to discuss in some depth the theoretical basis for choosing this form of presentation

The paradigm on which the spoken mathematical presentation is based, is that of converting a sequence of juxtaposed symbols, delimited by both white space and other visual cues (such as parentheses) into a serially transmitted linguistic approximation In order to achieve this, a parallel was drawn between the structure found in mathematical expressions and the inherent composition of English sentences

There is often a nesting, and grouping to be found in English language, whether spoken or written This can be seen in the sentence, "the goal, which was scored in the last minute, won the match" The clause "which was scored in the last minute" could be left out, without a loss in intelligibility in the sentence However, its inclusion endows a more descriptive appearance to the sentence, and conveys more information to the reader (or listener) than would otherwise be the case

This clausal break-down of a sentence structure is done automatically when the utterance is being spoken by a native-speaker of a language, and coupled with the inflection of the voice (that is, the use of pitch, amplitude and other prosodic features) it can convey a variety of meaning, depending on what timbre the speaker imparts to it For example, the sentence "'the goal which was scored in the last minute,", when said with a falling pitch at the end of the utterance, can convey to the listener that a statement is being made, whereas "the goal

which was scored in the last minute?"' implies that the speaker is asking a question

The nesting of clauses within sentence structures, therefore can be seen as a means to impart more descriptive semantics to the information being spoken The subtle use of pauses and changes in prosody assist the listener in interpreting what the speaker is trying to convey Coupled with the localised nesting of clauses within sentences, is the combination of unrelated, though logically sequenced sentences to form passages of spoken material Each sentence forms a unique entity, which contains both verb and noun phrases, thereby enabling its understanding However, when placed in the context of surrounding material, the semantics of an individual sentence can be altered radically

If we extend this to the written form of presentation, then the sequence of sentence structures can be brought together into paragraph units, which themselves contain an underlying message, of which each individual sentence forms a part The nesting, therefore, can be seen as the overall paragraph unit, which contains sentence structures, which contains words, which themselves contain characters, both alphabetic and punctuational By inferrence, therefore, it can be said that the atom of the paragraph is the character, as it is the smallest unit into which the complex structures of English can be reduced [1]

It is not just the characters in and of themselves which facilitate the understanding of the written, or spoken information, rather it is the means in which they are concatenated to form the superstructures defined above Imagine the set of characters "thankyouforreadmgthisdocument" This is merely the concatenation of character symbols, to form the words of an utterance, which are all syntactically legal, but the lack of delimiters (in this instance white space) can render their meaning ambiguous The addition of such delimiters turns the above sequence of character symbols into the intelligible sentence 'thank

[1] Linguists have developed smaller units, such as phonemes into which languages can be decomposed For the purposes of our discussion, the atom of the sentence is perceived as the character, as it is the smallest unit into which we need descend

you for reading this document" This is an extremely important fact to be born in mind when an analogy is drawn between the presentation of English, and the transformation of the visually oriented mathematical notation into a linear, transitory speech signal Just as the visual cues which form an implicit part of the parsing of mathematics, so also does the addition of pausing, and other prosodic features assist greatly in the disambiguation of the syntactically complex mathematical material

The function of clausal grouping was illustrated above In a system where not only the horizontal juxtaposition of characters, delimiters and white space is permissible, the sub-grouping of component parts of the whole becomes more important It is exactly this problem which printed mathematics presents to the systems' developers, who wish to produce a verbalised approximation of the material Unlike the methods used in English, mathematics uses horizontal offsets, coupled with the horizontal arrangement of symbols and white space to infer different meamngs to material Consider the two examples, x^{y+z} and $x^{y} + z$, where exactly the same symbols are used In the first representation, the three symbols $y + z$ form a superscript to the x, while in the second, only the y is part of the superscript

The use of such visual cues, as can be perceived from the preceding two examples, is one form of grouping found in printed mathematics Others involve the use of parentheses, brackets and white space to assist the reader to parse the material and to gain an understanding of the underlying meamng It is this form of ambiguity which must be removed from the spoken rendering of this syntactically rich data It is imperative that an intuitive presentation be incorporated into any system which is responsible for speaking this form of information, hence the use of spoken utterances alone, and the lack of non-speech audio to convey the information

The principle objective of TechRead's prosodic mathematical model is to disambiguate the grouping which is so easily perceptible to the visual reader Another factor in the design of the model used, is to ensure the brevity of the

utterance This necessitates the minimalisation of the form of lexical cues defined by Chang [Cha83] and a reliance on the *inflection* of the voice This means that a reliance is placed on the use of pitch contour, pausing and alterations in the speaking rate to impart the information

5.3.3 The prosodic model

The previous sections discussed the means whereby the higher-level views of equations and other formulae would be spoken by TechRead Also described were the cardinal ideas on which the model is based This Section describes the practicalities of including verbal renderings of various commonly seen mathematical constructions The syntax used here is not intended as a full guide to the manner in which TechRead treats each and every mathematical object which is encountered in the course of technical material Rather, it is designed to demonstrate the manner in which the general principles of verbalising syntactically rich material can be applied to other unspecified areas The examples of environments have been chosen to meet a standard of those which can be found in Irish pre-University exams[2]

Symbols and basic operators

The core of any mathematical expression is the symbols and operators which are used to syntactically represent the information Typically, these consist of Roman and Greek letters, Arabic numbers and various diverse operators which indicate their interrelationship This is not to say that these are the only symbols used in mathematics, as there are occasions where others (such as Hebrew letters) can be encountered in the course of various types of expression TechRead is designed to cater for the Roman and Greek letters, the Arabic numerals and the various relational operators which indicate how they should be combined The reason for this, is that, as has been already stated, the first

[2]Honours Leaving Certificate

version of this system is intended to serve students who have not yet reached university standard mathematics, and consequently the more obscure constructs will not be built-in to the system

The verbalisation of the Roman letters is a trivial matter, as `DEC-Talk` is designed to handle this form of character set Also, Arabic numerals, forming as they do a part of the lower-order `ASCII` set, prove no difficulty to this synthesiser Other character sets, on the other hand can pose problems Though it is true that certain of the Greek letters are contained in the `ASCII` character encoding, `DEC-Talk` is not designed to incorporate them into its spoken output Accordingly, a translation must be made, which verbalises each of these letters This is performed by assigning a Roman spelling to various LaTeX keywords For example, the letter α is verbalised as "alpha", while θ is spoken as "theta" There is no need to employ those phonetic representations defined by Chang [Cha83] which were discussed in section 2 4 1, as `DEC-Talk` pronounces each of the Greek letters in a clear and distinct manner

One dilemma which did occur when considering the verbalisation of these basic symbols, was the manner in which to distinguish between upper and lower-case letters There are two possible approaches which can be taken to ensure their differentiation The first possibility is to precede each upper-case character by the word "cap", (as Chang suggests), while the second involves a raising of the average pitch to convey that an upper-case character has been encountered Both these methods are used to good effect in different screenreading packages, and either will remove the ambiguity surrounding the type of symbol being read

The method finally decided upon, was the latter just described The reason is that, in accordance with the wish to minimise the number of extra lexical cues, the alteration of the average pitch ensures that the utterance is not lengthened by the inclusion of this form of information For example, the simple expression $\sum_{n=1}^{N} e^n$ demonstrates the use of the capitalised, and lower-case form of the letter If the word "cap" were included, then the utterance would be made

longer than needed, while by altering the average pitch, the same information is conveyed, but in the same amount of time as would be the case if only lower-case letters were used.

The first step towards the derivation of spoken mathematical formulae is to devise a means whereby the operators, which form an intrinsic part of this type of material, can be uttered. There are many different uses to which certain operators can be used, and their interpretation transforms them from binary, to unary interpretation. For example, in the expression $b - c$, the $-$ operator is a binary, in-fix operator which indicates that the difference of b and c is required, while in the example $-bc$, it acts as a unary operator, indicating that a negative amount should be computed. Stevens [Ste96, DNA97] points out that in the course of developing MathTalk, (see Chapter 2), a decision had to be made whether to distinguish these two different views of the same operator. The approach taken in TechRead, is to use the same keywords, irrespective of the in-fix, pre-fix or post-fix nature of the operator. That is, if the unary or binary form is used in an expression, the same verbal cue wil be used.

Another problem was the nature of the verbal depiction of the operator to use. Should, for example the $*$ operator be designated as "times", or "multiplied by"? To comply with the stated objective of minimising the length of the utterance, the shortest form of acceptable verbal cue is incorporated into the system. A compromise is made, in the case of $-$, where the word "minus" is used instead of "dash", as the former is in more common usage in a mathematical context. Table 5.5 shows some examples of the operators and their verbal equivalents. As can be seen, they follow a pattern of depicting the symbol in as close a manner as possible to their usage in natural speech, while still maintaining their mathematical context.

LaTeX provides the mechanisms to unambiguously distinguish the symbolic nature of mathematical expressions. For example, a distinction is made in the mark-up between $\pm$ and $\mp$, thereby facilitating the unambiguous audio rendering of the material. As will be seen below, this facilitates the incorporation of

Symbol	Pronunciation
$+$	plus
$-$	minus
$*$	times
$/$	over
$=$	equals
$\pm$	plus or minus
$<$	less than
$>$	greater than
$\leq$	less than or equals
$\geq$	greater than or equals

Table 5.5: Mathematical operators and their pronunciation. This table demonstrates the means whereby some operators are pronounced in TechRead.

alterations in the vocal prosody to clearly represent the division between subgroups within the overall scope of the expression. In addition to the standard mathematical operators, we have defined a "verbal operator", which is used in the case of fractions to indicate auditorily, what the white space makes so visually clear. This operator has the same precedence in our model as the four basic operators (+, −, * and /), and is treated in exactly the same manner. This operator constitutes one of the few lexical additions to the spoken equations. We feel that it is necessary to define such an operator, as merely using pausing and rate changes would not convey the vertical relationship between the numerator and denominator of the fractional element.

Other operators

At the level of mathematics for which this system is intended, there is only a minimal set of complex operators in common usage. Their inclusion, therefore, is intended as a basis from which the system can ultimately be extended to permit the verbalisation of more complex expressions. The discussion which

follows, therefore, is intended to demonstrate some general guidelines which can be extended at a later time, and applied to more complex structures, for which the operators currently included form the basis

Trigonometric functions

The trigonometric functions included in TechRead's initial command-dictionary are those primatives which make up any expression of this genre They include sin, cos and tan, coupled with their respective inverses As Chang's rules [Cha83] of Section 2 4 1 demonstrate, the pronunciation of this type of construct can be achieved quite accurately The standard English representations are used to depict these trigonometric operators, that is "sine", 'cos" and "tan" respectiviely The reason "cosine" is not employed is to avoid confusion with the "sin" operator We believe that, if the listener happened to miss-hear this operator, then confusion could be caused by the similarity in the sound of these two functions, hence the use of "cos"

The typical means of expressing the inverse of the various trigonometric functions just described is to use a negative exponent, as in $\sin^{-1}$ There are two methods of speaking this type of expression The first is non-interpretive, and involves the examination of the LaTeX mark-up, deducing the presence of the sin function, followed by a superscript, and speaking it thus The alternative is more interpretive, and is the method used in the TechRead system This involves the perusal of the entire scope of the sin function and the translation of the negative superscript into the keyword *inverse*, which is placed before the trigonometric function to which it applies

The reason that non-interpretive rendering is not used, is that it makes the utterance too long and unwieldy Also, the means whereby the inverse of trigonometric functions is depicted by TechRead follows those principles employed by teachers who customarily speak mathematics to blind students An addendum to this discussion is the reason why *arc* was not used in place of

inverse The reason is that this term is not in common usage among the target user-group, and we believe that it would introduce a degree of ambiguity into the rendering

Superscripts and subscripts

Both superscripts and subscripts play an important role in the presentation of mathematical material, irrespective of the medium used for its depiction This being the case, it was necessary to deduce strategies for their inclusion in the verbalisation of the material Typically, superscripts are utilised to convey exponentiation, although they can often be used in other, more specific contexts This discussion will focus on their use in the context of exponentiation, though it is readily understood that their alternative usage will have to be catered for in future versions of the software

Chang [Cha83] has defined various lexical cues for the production of spoken superscripts In Section 2 4 1, they were shown to be effective in the presentation of this type of structured information Stevens [Ste96] has defined a subset of these rules, which are included in the MathTalk program TechRead uses a similar approach, albeit using different lexical cues, to indicate the exponentiation implied by the use of a superscript The phrase *to the* is added to impart to the user that the material which follows is a constituent part of a superscript As will be discussed below, the use of the pausing strategies used to convey grouping are combined with this to indicate the scope of the information contained within a superscript

Subscripts are not encountered extensively in the types of mathematics for which the system is designed Their inclusion is primarily confined to co-ordinate geometry, where they are typically followed by a superscript, as in x_1^1, which is used to specify the x co-ordinate in the first of two points This form of representation, is merely sent as an unadorned text string to `DEC-Talk` It's verbal representation would be "x 1 1", and it is left to the user to inter-

pret this as appropriate As will be seen in the next section, superscripts play an important part in LaTeX mark-up when the bounds of limits, integrals or summations are specified However, these are specific instances of the use in a syntactic context, and do not apply to their description here

Large-scale operators

There are several large-scale operators to be found in printed mathematics, for which LaTeX has included commands Among these are lim, $\int$, $\surd$ and $\sum$ Their inclusion facilitates the translation into intelligible utterances Each of these larger operators have several forms, for which TechRead caters admirably The simplest of these operators is the $\surd$ sign, which is spoken as "square root of " The cube-root is also supported, as these are in common usage Above this level, the number n is substituted, where n is any number For example, if the fourth-root of an equation were to be spoken, then pronunciation would sound like "root 4 of " The grouping of symbols to indicate their scope and relationship applies to this symbol, and will be discussed in a subsequent section

There are two forms of the lim command supported in TechRead The first of these, is demonstrated by the example $\lim_{i \to 0}$ This is spoken as "limit as i tends to zero of " This rendering was chosen as it conveys without ambiguity the nature of the limit, and the nature of the constant to which it tends There is also another version of this operator, which indicates that a variable approaches a given constant *from below* This is shown by $\lim_{x \to 0^-}$, where the differences will be apparent TechRead handles this type of mark-up by inserting the words *from below*, as a qualifier to the verbalisation of the limit Integrals and summations also have similar variations to their general presentation For example, the integral can be bounded, or not The utterance which conveys this fact, is based on the principle described in the context of the lim command Using these methodologies, a general form of the verbal presentation required is established, and variations devised as necessary For

example, the general form of expressing an integral is to say *integral of* , while if it is bounded then the form "integral between x and y of " is used instead

These large-scale operators demonstrate the unusual use to which subscripts and superscripts can be put in LaTeX mark-up The code fragment `\sum_{i=1}^{N}` produces the output $\sum_{i=1}^{N}$, while the code fragment `x_{1}^{1}` produces x_1^1 It can be seen that the two extracts look remarkable similar, in that they both use a subscript, and superscript construction LaTeX however interprets their usage to mean different things, depending on what command they follow, and this is the approach taken by TechRead In order to ascertain the appropriate translation of the subscript and superscript to an element, then a command dictionary is consulted in order to verify that no special usage is intended If a specialised application of these forms of presentation is found, then the verbalisation proceeds using these renderings, otherwise, the approach taken is that previously discussed for dealing with this type of presentation

5.3.4 Combining the terms

We have seen how TechRead presents each individual component of a mathematical expression However, their meaning in isolation can be entirely different when combined with other adjacent items Accordingly, a strategy was needed to cater for the diverse combinations permissible in mathematical presentation Unlike previous systems [Ram94, Ste96], TechRead does not employ any non-speech audio to convey the material, rather it relies on the vocal representation only to impart the information Therefore, it proved necessary to examine the grouping of the material, to deduce a method whereby it could be delivered in an unambiguous manner as possible Another stipulation was that the utterance be maintained at a length which would not impair the listener's ability to apprehend the material Coupled with this, was the knowledge that the interface to TechRead would provide the functionality to enable the user to peruse the material in detail, thus alleviating some of the problems encountered with

the flow of information past the passive listener (see Chapter 3) The objective was to produce a system which relied on the prosodic component of speech to produce a rendering of the mathematical material in an unambiguous and clearly understood manner

Previously, it was discussed how the prosodic model used in TechRead is based on an application of those paradigms found in the decomposition of English, namely clauses If one assumes, that a mathematical expression is a structure in and of itself, and that it can contain an arbitrary degree of nested sub-expressions, then it can be resolved into either a sentence structure which contains various clauses, or a paragraph (i e , a set of sentences which may themselves be unrelated, but which when logically combined make up a superstructure) This being the case, the need was perceived to examine the LaTeX mark-up in an attempt to deduce the points at which the delimiters could be interpreted as the equivalents of clause boundaries in English This proved to be successful, as it emerged that, in order to produce well-formatted equations, it is necessary to mark documents up using unambiguous syntactic representations Using this as a basis, it could be inferred that, should these delimiters be present in the mark-up, it should prove possible to equate them with clause boundaries in an English verbalisation of the material

As a consequence of this examination, it emerged that it would be possible to construct three levels of nesting [3] Using `DEC-Talk`, it is extremely difficult to accurately control the length, and placement of pausing For example, one observable phenomenon is that when the characteristics of the voice are altered by the inclusion of a control sequence in a string of text, an indeterminate pause is introduced into the utterance Also, above certain speaking rates (240 words per minute, approximately) certain minor pauses are ignored by `DEC-Talk`, as they would not be perceptible to the listener Ideally, the pausing would be set at predetermined values, from which the same relative changes made to

[3]This can be extended However, for the purposes of our investigation, the number of levels of indentation was kept to three, as it was consistent with the paradigm of paragraph, sentence and clause boundaries observable in English

the other prosodic features of the speech could be applied to the pausing, thus maintaining compatibility between this element and the remainder of the voice contour

Due to the lack of consistency in the pausing, the accurate portrayal of the predetermined grouping can be extremely difficult The two methods used in TechRead to imply the grouping of terms into sub-expressions, (and by extension of sub-expressions into the whole formulae) is to insert pausing, and alter the speaking rate at strategic points within the presentation This is not an arbitrary process, as the subsequent paragraphs will reveal, but is based on both the mathematical components preceding and following the point at which the pause is needed For example, the expression $a + b + c$ is a simple, linear concatenation of three quantities, separated by two relational operators, irrespective of whether the material is being read visually or auditorily However, the expression $\frac{a}{b} + c$ is non-linear in the print medium, but linear in the audio environment

Accordingly, something must be introduced to indicate to the listener that the fractional component of the expression merely encompasses the first two elements, and that the final one is not contained therein The method whereby this is achieved, is to speak the expression as follows "a over b, plus c" In order to imagine the effectiveness of this strategy, we recommend reading the expression aloud, following the grammatical clause boundary which has been inserted Using this method, it can be clearly seen that the distinction between the fractional component of the expression and the remaining relational operator and quantity is evidenced If one also adds a slight quickening of the speaking rate of the utterance of the fraction, then the distinction becomes even more apparent to the listener

The example of the previous paragraph is an extremely simple demonstration of the capabilities of the methods used in the system to produce spoken output from mathematical material This method works equally well, when complex entities are included in the expressions As an example of such a more

complex example consider the following equation

$$\sum_{i=1}^{n-1} a^i + \frac{i+1}{i-1} \tag{5 1}$$

This example is more complex than the preceding expression, as it contains several operators in combination to produce a formula which could prove quite difficult to read Using unadorned text strings, it would not be discernible where the scope of the various superscripts ended, or which items were contained in the fractional component of the formula Also, there could be confusion as to whether the summation encompassed the entire scope of the remaining elements or whether it was merely confined to that component which immediately followed it Using the prosodic enhancements, it is possible to produce an audio rendering which can clearly and intelligibly inform the user of the means whereby this formula can be decomposed

The first aspect in producing any meaningful output from mathematical material is to determine the degree of nesting in the formula Once this has been established, the levels, and indeed lengths of pausing required to speak the content can be determined In the preceding example, the summation encompasses the entire gamit of the remaimng material, thereby yielding one level of nesting The material itself consists of one superscripted element, followed by a fractional component However, the fractional component itself comprises three complex terms, as the numerator consists of $i + 1$, while the denominator consists of $i - 1$ Accordingly, there are several degrees of nesting observable in this expression There must be a lengthy pause to indicate that the entire formula is contained within the scope fo the summation, followed by various pauses to indicate to the listener to the scope of the superscripts, and sub-expressions of the fractional element

The approximate verbal rendering of this equation would be as follows For simplicity, the symbol " " has been used to indicate a longer pause, more akin to that found between paragraphs

"sum from i equals 1 to n minus 1 of a to the i plus 1 plus 1 plus i plus +1 over, i minus 1 "

The pausing in this example illustrates an approximation of the nesting used in TechRead to indicate the scope of various operators Combined with increases, and decreases in the speaking between the various clause boundaries, it has proved possible to use this mode of presentation to determine the scope of operators in expressions even as complex as this Moreover, the use of pausing **after** operators is a useful aid in assimilating the fact that the operator is itself followed by a complex term

In the quoted passage above, the reader can observe the pause after the lexical operator "over" Once the listener has become familiar with the system, this pausing strategy can be used to anticipate the fact that a complex term is about to be spoken Conversely, in a simple equation, the pausing would not be placed after the operator but before This indicates the scope of operators (such as fractions), and also informs the user, without the addition of extra lexical information, that the term they are about to hear is a simple one

The alterations in the speaking rate have been carefully calibrated to ensure that, even at the deepest level of nesting, the utterance is not delivered too rapidly With this in mind, a value of 6% has been chosen to assist in informing the listener of the subtleties of the grouping which are so easily observable to the visual reader The reason that an increase of 10% were not chosen, is that if three or more levels of nesting were encountered in an expression, then the ultimate speaking rate would be too fast, and as a consequence, the listener would be unable to ascertain what was being spoken

The alterations in the speaking rate, and the pausing which is introduced into the verbal presentation, is always relative to the starting rate from which the alterations are made This caused many problems when experimenting with various `DEC-Talk`-specific settings, as this particular synthesiser uses internal rules to calibrate the pausing and pitch contour of the voice Consequently,

there are occasions when the utterance can be stilted in its delivery This fact does not impair the intelligibility of the utterance however it proves quite frustrating to any developer who wishes to produce a standardised approach to rate and pausing changes

5.4 Summary

This chapter has described the methods used in TechRead for the presentation of various types of content using the medium of spoken output The basis on which the prosodic model is constructed was discussed, and the precise methods for conveying both the structure and textual content of documents was described The discussion of the prosodic model used to convey mathematical material included the description of a pilot study, conducted in order to ascertain the nature of what is apparent to sighted readers when superficially glancing at equations The results of this small-scale experiment were discussed, and their impact on the methods used in TechRead to convey the same type of information auditorily was explained

The chapter concluded with a description of the means whereby the finer details of mathematics are conveyed prosodically in this system The remainder of this thesis discusses the various implementation details, and an evaluation of the mathematical prosody used in TechRead

Chapter 6

Implementation and evaluation

The discussion thus far has focused on an explanation of the aspects of the TechRead system, which differentiates it from those previous systems described in Chapter 2 This chapter describes the practicalities of producing the internal model, the Braille output and the prosodically enhanced synthetic speech from LaTeX Also described in the following sections, are the methods used to implement the interface, and the problems encountered in developing various prototype systems The chapter concludes with a description of a small-scale evaluation of the prosodic model used to convey mathematical expressions

6.1 The search for a language

Over the course of our investigation into the production of more accessible technical documents various prototypes were developed using a variety of platforms and programming languages The first such system, involved a simple *regular expression grammar*, which proved that `ASCII` could be quickly and efficiently transformed into Braille This system took either keyboard, or file-based input, and produced an output file consisting of Braille characters Later this was

expanded to send the output to the embosser This primitive forerunner of the TechRead system, gave interesting incites into the means of translating ASCII into Braille, as it demonstrated the problems posed by the context-sensitive nature of this unique output medium This first system was written using a package known as Lex, [LTD92] which transformed an input file consisting of *regular expressions* into C syntax, which can in turn be compiled in the normal manner The output produced executes quite quickly, though the C derived from Lex is almost totally unreadable, as it does not conform to the conventions of good programming practice

A sample Lex program is given in Figure 6 1 It can be seen that a rule consists of two parts, a specification of a pattern which must be present in the text being searched, and an output portion which is activated if the rule is fired This latter portion of the rule consists of C syntax which is incorporated into the output program produced by this package

The next phase in the implementation process was to examine methodologies for representing the internal model described in Chapter 3 At this point, it became clear that the most efficient means of implementing the system was to follow the Object-Oriented Paradigm Using this programming model, it proved possible to represent the nodes of the model using *generic classes*, and specific instances of those classes to contain unique features which differentiate the various hierarchic levels of internal node described in section 3 1 Once the programming model had been decided, it proved necessary to seek a development environment which would cater for the various needs of the system The criteria which the environment had to fulfill were

1 be capable of incorporating Object-Oriented programming constructs such as classes

2 be capable of affording the low-level control necessary to communicate with hardware devices such as speech synthesisers or Braille embossers

```
%{
     /*  Lexer for the translation of LaTeX into Braille
     version 2 0
6/1/97

*/
#include "printers h"
#include "stdarg h"
#include "string h"
void make_brl ( int num,    ),
char *brl_text;
int last_tok_seen,
int brl_index = 0,
%}
alpha [a-zA-Z]
num [0-9]
%%
[ \t]+     { make_brl (1,SPC_INDEX), }
about   { if (last_tok_seen == SPC_INDEX)
              make_brl(2,A_INDEX, B_INDEX), }
above  { make_brl(3, A_INDEX, B_INDEX, V_INDEX), }
according  {make_brl (2,A_INDEX,C_INDEX); }
across  { make_brl (3, A_INDEX, C_INDEX, R_INDEX),}
after  { make_brl (2,A_INDEX, F_INDEX), }
as  { make_brl (1,Z_INDEX), }
%%

void main ( ) {
int i;

yylex ();
}
```



```
void make_brl (int num,    )
{
```

3 be capable of expressing an interface which was usable by both blind and sighted people alike

4 be rich enough to express the complex structures necessary to contain TechRead's internal document model

With these criteria in mind, the first language which was used to develop prototype systems was `Java` At first glance, this language seemed ideal for the purposes of developing TechRead, as it offered all the syntactic constructs necessary for completing the task Firstly, it is an `Object-Oriented` language, and all tasks must be expressed using the constructs of this paradigm Secondly, all interface design in `Java` is performed using an `API (Application Programming Interface)` This ensures that if a button is required, the programmer need only call a function which will place the button on-screen, at the specified co-ordinates This `API` was very attractive, since as a blind developer, the easy-to-use visually-based programming tools so prevalent today can be extremely cumbersome to use The use of the `API` would, it was believed, reduce the development of an interface to something like command-line programming, where everything could be expressed in terms of absolute co-ordinates

Another advantage of using `JAVA`, was its platform-independent nature `Java` is not compiled into machine-dependent binary code, rather it is interpreted by a the `Java Virtual Machine`, which uses a bit-code which can be interpreted across all platforms In accordance with the wish to render TechRead as system-independent as possible, the compatibility of `Java` seemed absolute, until we discovered that it was virtually impossible to access hardware devices using this programming language The methods for performing such low-level operations, were to integrate *native code* with the `Java` program This form of native code can be written in languages such as `C` or `C++`, and compiled into whatever form of library is utilised on the particular system Using this method, it would have proved necessary to develop specific sub-routines to communicate with the hardware devices, which could be used on every system for which TechRead was developed Also, using the integration of native code would

reduce the platform-independent nature of TechRead, thereby minimising the advantage Java had over other Object-Oriented languages such as C++

As a consequence, the emphasis moved away from producing a system which would work on **all** platforms, to the development of a piece of software which would function properly on one We decided to focus on Ms Windows 95, as it is the most commonly used platform by blind people in Ireland Proceeding from this, a language was chosen which would be compatible with this particular platform, and which would facilitate the production of various components of the system These components would be used to verify that the techmques described in earlier parts of this discussion were effective

Further, it was necessary for this language to have innate characteristics which would facilitate the specification of the interface in terms of a syntax-based approach (similar to that used in Java) rather than a visually-oriented system, such as is found in Visual Basic Coupled with this, the development environment had to contain features which ensured that it could be used in conjunction with a screenreader

With all these particular requirements in mind, the eventual language of choice for the production of TechRead was Visual C++ [Kru97] This particular language contained all the features necessary to ensure that it could be used effectively by a blind developer, as it permits the development of system interfaces using a resource language, which it interprets and integrates with the main portion of the programming project This development environment also contains facilities to perform the visual development of systems (as its name would suggest), which meant that collaboration with sighted colleagues was quite feasible

6.2 Producing a system

It should be stated, that at the time of writing there is no concrete implementation of the entire TechRead system. Rather, there are several components which have been developed, with a view to ascertaining that the programming techniques used in their production were effective, and produced the desired results. Therefore, those details of the implementation of TechRead are presented here to indicate that the development of those aspects of the output described in previous chapters are not only feasible, but fall well within the programming ability of any competent programmer.

6.2.1 Hardware communication

The methods used in TechRead for the purposes of communicating with the various hardware devices employed by the system, follow the recommended practices of `Windows 95`, and all other similar Operating Systems. Unlike older `DOS` based programs, the developer does not need to access the hardware port directly; rather an intervening layer is placed between them and the low-level memory-addresses used for the purposes of actually interfacing with the devices. Using `Windows 95`, the programmer treats the communications port as a file, and reads and writes data using exactly the same methods as are used in standard disk-base file handling. Figure 6.2 demonstrates the manner in which a serial port is opened, data written to it, and then closed once more. As can be seen, there is no need to specify the exact memory address as was the case in older, command-driven programs. Using this file-based approach, the programmer can be sure that the data they require to be sent **will** be sent, and that the Operating System will take care of all those minute details which previously were the job of the application developer.

Both hardware-based speech synthesisers, and Braille embossers can be communicated with, using the previously described means; the only difference is that, typically, the parallel port is used by the embosser, and the serial port is

```
// declare file handle
HANDLE token;

// associate file handle with COM1 serial port and initialise
token=CreateFile( "COM1", GENERIC_WRITE|GENERIC_READ, NULL, 0,
OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

// write len bytes from buffer to port

WriteFile(token,buffer,len,&len,NULL),

// deallocate handle
CloseHandle(token);
```

Figure 6 2 Sample code for communicating with a serial port under Windows 95 It demonstrates the file-based approach

employed by the synthesiser There are, however, some extra details needed to communicate effectively with a synthetic speech device, the most important of which is *indexing* It is a well-known fact that computers produce data at speeds which far exceed those at which the peripheral device with which it is communicating can utilise it Consequently, fail-safe mechanisms must be incorporated into any communication library, to ensure that when data is transmitted to the device, it is all consumed, and none is lost This is achieved through the use of indexing, when information is being relayed to speech synthesisers

The algorithm used in TechRead to ensure no data-loss is

1 extract a word

2 attach an index marker

3 send word

4. receive the index marker of the last word spoken

5. if an error has occurred, re-send those words subsequent to the last correctly spoken one

An index marker is a numeric value which is attached to a word, which forms a computable sequence with those preceding, and following it. Moreover, it must be contained in a control sequence which is specific to each synthesiser. This value is **not** stored in the internal representation of the document, as it will alter every time a communication session begins or ends. For example, a word in position x in a document will have index-value y, when sent as part of a continuous reading process, while when sent as part of a *read current word* operation, it would have value z. Index markers must be computed as the word is being sent, as data can be lost due to break-downs in the communication process. Because of this, the value can alter dynamically, thereby precluding the static storage of these values in the model. Their use ensures that the listener will not be subjected to sporadic data loss, causing the utterance to become unintelligible.

6.2.2 Parsing the LaTeX

There are two phases involved in the derivation of Braille or spoken output from the source LaTeX document. The first of these is responsible for the transformation of the input into the internal representation of the document, while the second renders this model auditorily, or in Braille. The second phase will be described in the next section, while the subsequent paragraphs detail the methods whereby the LaTeX mark-up is transformed into the internal model discussed in Chapter 3.

The first operation involved in producing the graph of the document, is to ascertain the global settings. This is achieved by examination of the type of document being translated. This information is to be found in the `\documentclass`,

located at the top of all LaTeX documents The document class, specifies what form of logical structure is at the author's disposal For example, if the *report* class is used, the author may sub-divide a document into hierarchic units consisting of Chapters, sections, subsections, paragraphs, sentences and words If, alternatively, the *article* class is specified, then the hierarchy can only contain sectional units at its highest level, since chapters are not permitted

Various options are allowable in this class, such as the font used, the line spacing etc Accordingly, this is the first LaTeX construct examined to ensure that the font used in the main body of the work, coupled with the associated default speaking voice can be instantiated within the model

There are two vastly different parsing techniques used in the input phase of the system The first of these is responsible for the extraction of the textual content of the document, while the second ensures that mathematical material is rendered unambiguously The textual translation is a far less complex task than the rendering of mathematical data, as the latter utilises both vertical and horizontal juxtaposition to convey the semantics of the material

Extracting the Text

The extraction of the textual portion of the document can be further reduced to two sub-categories, namely the sub-division of the input into words, and the placement of these words in their correct location in the internal model For the purposes of our discussion, a word is deemed to be any character-based unit of arbitrary length, which is delimited by white space, and may contain a punctuation symbol Accordingly, the job of the parser is to separate the mass of content contained in the input file (or files) into the words, which are used in the terminal nodes of the model (see Section 3 1) This is achieved by examination of each character in the input, until either a special character (a \ or a {) is discovered, or a word can be extracted Any character between the delimiters just described, is considered to be a constituent part of a word, and hence is

```
start
    while (NOT EndOfWords) //while there are still words
    GetCurrentChar();  //get current character
    while (NextChar NOT = Dilimiter, NextChar)
    if (CurrentChar = '\' OR '{') //start of token
        push({};  //add to stack
        MakeToken(word);  //pass keyword to special function
        EndIf
        if (LastChar NOT = '}'
            AddWord(word),  //insert word in the model
        else
            Pop (});  //end scope
        EndIf
    EndWhile
End
```

Figure 6 3 Some sample pseudo-code which illustrates the algorithm used in textual extraction This also demonstrates how the *MakeToken* function is called when the LaTeXcommand-indicators are encountered

stored in a terminal node of the model Figure 6 3 shows some sample pseudo-code which is responsible for extracting the various words of a document It can be seen how, once the beginning of a word has been encountered, the end of the word is then sought, at which point it is placed in a terminal node It should also be clear from this fragment of code, that the function *MakeToken* is called when the \ or } symbols are encountered in the input Since these symbols are used in LaTeX as command-indicators, the reason for their specialised treatment will be obvious

The *MakeToken* function is possibly one of the most important sections of the TechRead system, as it is responsible for the construction of both the internal nodes of the model, and the discerning of those formatting alterations which,

when translated into an audio rendering, give the listener the understanding of the visual appearance of the document There are various off-shoots to *MakeToken* The first of these is responsible for constructing the actual LaTeX keyword, and operates in a similar manner to that seen above in the context of Figure 6 3, save that it uses different delimiters to indicate the starting and ending of keywords

Typically, commands in LaTeX files are prefixed by a \ or { character, and usually end with a } character An exception to this is the *inline-math* environment, which is normally used to place brief expressions in the main body of running text This environment is denoted by a $ symbol, which marks the beginning and end of the brief mathematical expression

The purpose of *MakeToken* is to extract the LaTeXcommands, and ascertain whether they are structural commands, or those which cause visual alterations, such as emphasis If they are the former, then the model must be extended to introduce new hierarchic levels, while if the tokens constitute the latter form of commands, (such as font alterations, centering etc) then the terminal nodes must be adjusted accordingly *MakeToken* is responsible for determining the beginning and end points for this form of alteration in the model

The transformation of LaTeX into the output media preferred by TechRead, is designed to follow the guidelines set down for compiler implementation The translation of LaTeX into Braille or spoken output, is deemed to be akin to the production of C++ code from Java, the only difference being that there is very little type-checking possible The only form of granularity permissible, is the verification that sectional units do not occur within units of lower hierarchic importance, or that symbols reserved for mathematical mark-up are not used outside that particular environment As a consequence, *MakeToken* must ensure that the LaTeXcommands used in the input, conform to the guidelines for document preparation, as outlined by Lamport [Lam85]

What is being suggested here, is in essence a compiler which, instead of

transforming the LaTeX into a DVI (device independent) file, the TechRead system transforms the input source into a representation of its own We saw in Figure 6 3, how each word was extracted from the general body of a document What should also be pointed out here is that, coupled with *MakeToken*, it is this process which places the various elements in their correct locations in the model For example, if an emboldened font is currently in use, this fact will be revealed by the *MakeToken* function, and used by the word extraction process to inter-link the various terminal, and internal nodes of the representation

A final point of note, is the manner in which the nesting of various objects is determined A simple stack is used to store the command delimiters described above When they are encountered by *MakeToken*, they are *pushed*[1] onto those which are already on the stack, and when their scope is ended they are *popped*[2] once again This technique is standard procedure in compiler design, where nested constructs are an integral part of the content with which they must cope

Mathematical Translation

The methods used to transform the mathematical content of LaTeX into the TechRead internal representation are significantly different from those just described for textual extraction Instead of a linear approach to the derivation of the material, a *parse tree* is necessary to indicate the grouping of sub-expressions, and hence to verbally depict the material A parse tree is merely a tree-based representation of a mathematical expression This technique is extensively used in compiler design, for the transformation of one input programming language, into another output syntax, hence its use in the TechRead system

The key to understanding this technique, is to envisage an equation as a tree-like structure, whose root node is an arbitrary baseline operator Exam-

[1] placed on top of the stack

[2] removed from the top of the stack

ples of these operators are =, + (on occasion), and the verbalisation of the distinction between the numerator and denominator of a fraction in the form of the keyword *over*. Once the baseline operator has been chosen, the component sub-expressions are recursively examined to produce sub-trees of the overall expression, until a point is reached when the mathematical atoms are determined, and no further decomposition is possible. When this point has been reached, the tree-like representation of the equation is traversed (by the system as opposed to the user) and the mathematical sub-graph of Section 3.1.3 produced.

The parsing algorithm for the mathematical material is straightforward, in that it is a recursive-descent parser. This parser is called by *MakeToken* when a mathematical environment is entered. The algorithm is as follows:

1. traverse the mathematical expression to verify the presence of a baseline operator (such as =)

2. create a node which contains this baseline operator, with pointers descending from it

3. examine the content to either side of the pre-determined baseline operator and decompose each into sub-expressions

4. link each sub-tree to the baseline operator

It would be reasonable to assume that, as LaTeX is a linear notation, the Braille or spoken output could be easily derived from a linear-based parsing technique. This is not the case. As was demonstrated in Section 5.3, the addition of pausing, and the alterations in rate are extremely dependent on the grouping of terms within a sub-expression, and the further grouping of those sub-expressions in the context of the equation as a whole. The use of a tree-based parsing technique facilitates the deduction of the optimal verbal presentation of the material.

For example, a simple formula such as $\frac{a+b}{c}$ yields a tree, which immediately demonstrates the ideal locations for the placing of pausing The reasoning behind the insertion of these rate changes and pauses was outlined previously, this section is concerned only with their derivation Accordingly, the algorithm examines the overall scope of the formula, and determines that there is no immediately perceptible baseline operator Pursuant from this, the juxtaposition of the various sub-expressions must be determined, and by inference this used as a pseudo-baseline operator Since the expression consists of a fraction, the perceived baseline operator is the verbal cue *over*, thereby facilitating the decomposition of the mathematical content into the numerator and denominator of the fraction Examination of the high-level structure of the expression, therefore, has yielded a baseline operator which can form the root node of the parse tree for this expression

Following on this, the numerator is first perused, and it is immediately obvious that another relational operator (+) is present, and is hence linked to the root of the tree The remaining nodes are subsequently linked to the + node, completing the construction of the first branch of the tree The denominator consists merely of one element, and this is simply linked to the root node of the tree The fact that only a single element is present on the right-hand branch of the tree can be used to determine the fact that no pause is necessary after the *over* keyword, on the other hand, if more than one node were present, a pause would have to be introduced, and the right-hand side of the tree further decomposed

It is undisputedly true that a linear-based approach could have been used to fulfill the same task as that just presented in a tree-based context However, this data structure is in common usage in all forms of programming, and its use greatly eases the task of devising the utterance which corresponds to the equation Another important reason for the use of a tree to represent the parsing of the mathematical material, is the potential it offers to give overviews of the formulae It was demonstrated previously how, through the use of a pilot study,

it was determined what information should be spoken to simulate diverse *views* of the material. Using the tree-based representation, the different audio views can be rapidly determined.

Returning to our previous expression, it is apparent that the expression consists solely of a fraction. Superficial perusal of the tree reveals this fact, and thus it can be incorporated into the internal representation of the formula. As this is a simple expression, the next level would be somewhat redundant, however an accurate running paraphrase could be *numerator consists of 1 relational operator and denominator has only one quantity.* At the lowest and most comprehensive level, the use of a tree-based structure can be used to incorporate the pausing and rate changes needed to convey this expression accurately and intelligibly.

Constructing Tables

The tabular components of documents differ from the remainder of the components only in their vastly different layout. Individual tabular elements are made up of both mathematical and textual content, thereby necessitating the incorporation of both parsing strategies previously discussed. A second step is required to produce the internal links which can be used by the interface of Section 3.3 to enable rapid perusal of this type of content.

The presence of a tabular environment is first determined by *MakeToken*, and the appropriate sub-routine called. This sub-routine is responsible for determining which parsing strategy (mathematical or textual) should be applied to each cell, and once this has been determined, for linking the cells such that their cardinal directions are all inter-connected.

The responsibility for producing the actual output, is still the sole preserve of those previously discussed elements of the translator. The sub-routine responsible for incorporating the tabular material into the overall model of the

document merely slots the various elements into their appropriate places, and ensures that the whole sub-graph of Section 3.1.4 is produced.

6.2.3 Incorporating audio formatting

Though this facet of the implementation is described separately it actually occurs at the same phase as that of extracting the text, structure and other diverse document elements discussed previously. The essence of this portion of the implementation is to ensure that the audio formatting corresponds to the visual equivalents. In Chapter 5, the alterations in the vocal characteristics were demonstrated, the results of which make the output more bearable over longer durations of time. The means whereby the audio-formatting is associated with the visual equivalent, is discussed in the following paragraphs.

As was stated in Chapter 5, the core ideas on which TechRead's spoken output is based are those of balance, and relative change. For example, if a pause ends a sub-sectional unit, **and** a sectional unit, then the longer is used, since the process of pause selection is not additive. Also, if the alterations in the vocal aspect are needed, they should be relative to the default voice, whatever value is associated with it. Accordingly, as the hierarchic model of Chapter 3 is being constructed, the voice alterations are computed relative to the global settings which have been instantiated in the *global settings* node of the document graph.

There are two phases to the incorporation of the audio-formatting into the model; that of deducing the alteration needed, and the location where it starts to take effect and returns to the default voice. The vocal alterations are computed relative to a starting set of characteristics, which are defined in the *global settings* node of the document. These settings are commensurate with the font specified in this node, though a specific set of auditory parameters is not specified for each of the three global font styles permitted in LaTeX. That is, the `\documentclass` command in LaTeX permits the specification of 10, 11 or 12

point fonts as the normal default font sizes used in the document There is not an alteration in the vocal aspect to distinguish each of these, rather a default voice is utilised from which the relative changes can be made

The specific characteristics of the voice, which are used to convey the various LaTeX environments were discussed in Chapter 5 and consequently will not be reiterated here Suffice it to say that, when the *MakeToken* function encounters an environment, it passes control to the sub-routines dedicated to the production of the audio output, which determine the correct changes appropriate to the situation The algorithm for performing this task is as follows

1 compare LaTeX command with existing database entries

2 if found, alter characteristics to reflect new state

3 if not found, alert user **or** examine mark-up

4 create new vocal structure (see Table 3 2)

5 return

The details of the calculations for the desired alterations are not given here, as they comprise simple arithmetic operations to obtain the desired result As was shown in Chapter 5, the vocal changes are not specified in absolute terms, but are obtained by applying percentage alterations to the default speaking voice Such computations are trivial, as they involve merely the use of the four basic operators, +, −, * and / Once they have been computed, they are instantiated in the structure of Table 3 2, and the control returns to the *MakeToken* function

Once the parameters for the presentation of the material have been determined, it is a matter of ensuring that they coincide with their visual counterparts This is achieved, by associating the visual and audio-formatting information in the same node, as was illustrated in Section 3 1 1 The text, visual

and audio formatting are then slotted into the model, and the linkage completed This is a fail-safe mechanism for ensuring that alterations in the visual appearance of the material are compatible with their spoken equivalents The basis for the success of this method is that the *MakeToken* function caters for the scope rules which are an integral part of any document written using LaTeX mark-up For example, should an author wish to embed a portion of emboldened text, within an italicized part of the document, then all that is needed is to delimit the new block by braces, and the nesting is achieved `MakeToken`, being responsible for the deduction of such things, ensures that when the scope of a declaration ends, (i e when a passage of altered text has been completed) the vocal characteristics should be re-set to their defaults This process is not as complex as the first adjustment, as no relative alteration is needed Instead, the *global settings* node is examined, and those parameters found therein applied to the point in the text where they are needed

6.2.4 Producing Braille

Although the LaTeX source provides an excellent input from which to produce highly accurate and well-formatted Braille, we have decided to produce the Braille output by using TechRead's internal document representation The reason for this is that, in the future, it is hoped to add more filters to the program, thereby enabling the conversion of many different file formats Consequently, if the model is used, there will be no need to write individual translators, which can transform diverse inputs into the single output

For example, if `HTML` were added as a file format, then if the source were used, a new translator would have to be written and incorporated into the TechRead system Using this approach, all that will be needed is to produce a conversion routine, which maps those structures which are built-in to `HTML`, to those used in the hierarchic document graph

Using the model also ensures that, should it be extended or altered, all that

would have to be done is to adjust the translator to reflect the changes in the representation This modular approach has been a key feature in the design of TechRead, as it is envisaged that, due to the advances in technology, it will prove necessary to alter the system to keep pace with the ever-changing world of computers

The model also facilitates the production of accurate and well-formatted Braille The use of the terminal nodes provides the actual content of the document, coupled with the formatting associated with it, while the internal nodes can be used to provide the layout information needed to physically locate the Braille on the page For example, it was described in Section 3 1 1 how each word node is contained within a paragraph unit This fact can be used to advantage, in determining the locations to insert paragraph breaks in the output Unlike the spoken output, the Braille is not produced simultaneously with the document model, rather it is left until it is required The algorithms for the Braille translation were described in Chapter 4, and these are used to generate the Braille symbols which are then output to either a file, which may be printed later, or directly to the embosser

The formatting of the output is also handled by this portion of the system Using the internal nodes of the model, it is possible to ascertain where the Braille should be placed on the page This is particularly so in the case of headings, as their hierarchic place in the representation ensures that they can be correctly formatted in Braille For example, it is possible to rapidly deduce whether an internal node comprises a sectional, or subsectional unit Using such knowledge, it is feasible to determine whether the hierarchic location warrants the instantiation of a centered heading, or one which is located at the side of the page, separated from the main body of the document by virtue of the fact that it is placed alone on a blank line

The model also facilitates the incorporation of mathematical, and tabular data into Braille documents, something which has hitherto been impossible At the time of writing, there is **no** electronic Braille translator which contains the

facilities to produce technical documents made up of textual and non-textual content This is one aspect in which TechRead is stretching the boundaries of Automated Braille Translation Using the document model, it is possible to deduce where to alter the rule-set in use, to that which is designed to cater for mathematical material, and more importantly, where to return to textual production

Naturally, the formatting of mathematical material is different to running text, and with this in mind, a totally different formatter is dedicated to ensuring that the layout of the mathematical data is in accordance with the standards specified by the Braille Authority of the United Kingdom, (BAUK) in Braille Maths Notation [otUK87] The means whereby mathematical material should be formatted is discussed in Chapter 4

6.3 Implementation problems, and recommendations

This section focuses on those problems which arose in the implementation of various small prototype systems Some are technical problems, while others are problems which I, as a blind programmer, have experienced Also included are some recommendations which, if taken on board, would improve the productivity of blind programmers in the workplace and all other areas where computers are in use

It was pointed out previously in this chapter how `Java` would have been the preferred language to develop TechRead, were it not for the difficulties in communicating with peripheral hardware devices There is on-going research into developing an `API` which will enable `Java` applications and applets[3] to be used in conjunction with speech synthesisers There is also quite a lot of on-going work in the areas of producing a universal mark-up language which will enable the specification of alterations to the vocal characteristics of the

[3]Programs written in `Java` which can be embedded in `HTML` documents Used extensively on the Internet

synthesiser in abstract terms, after which this abstract syntax is mapped to the concrete control codes used in each individual synthesiser If this proves successful, it will enable systems like TechRead to function effectively with many different synthesisers

One of the major problems which I experienced, was the lack of co-operation between many screenreaders, and the visually-oriented programming environments which are in such common usage at present Such development environments include `Visual Basic` and `Visual C++` These programming environments have been developed to ensure that (if one can see them) they provide ideal options for the rapid production of interfaces, and the linkage between these interfaces and the code which actually manipulates the data. The fact that non-standard approaches were taken in displaying various aspects of the development environment, ensures that most screenreaders have serious problems building interfaces

The normal, visually based strategy for building front-ends is to merely click a button, drag it to the required location on screen, and drop it there This is simply not possible when using the keyboard in conjunction with a screenreader Accordingly it proved necessary to devise an alternative strategy to design the interface Luckily, `Windows 95` bases its interface specifications on a resource script language This language syntactically represents those aspects of interface design, which the user normally indicated by placing items on-screen in the manner described above However, it proved possible to write such scripts in a standard text editor, as the script-file is `ASCII` based, and as such can be manipulated by any editor which has the capabilities to save in this format

Figure 6 4 shows some sample resource syntax This code fragment (which is part of any standard resource, found in applications developed using `Visual C++`) illustrates the means whereby a menu can be placed on-screen This small piece of code demonstrates how the `file` menu for any standard `Windows` application can be created

```
POPUP "&File"
BEGIN
    MENUITEM "&New\tCtrl+N",                ID_FILE_NEW
    MENUITEM "&Open   \tCtrl+O",            ID_FILE_OPEN
    MENUITEM "&Save\tCtrl+S",               ID_FILE_SAVE
    MENUITEM "Save &As   ",                 ID_FILE_SAVE_AS
    MENUITEM SEPARATOR
    MENUITEM "&Print   \tCtrl+P",           ID_FILE_PRINT
    MENUITEM "Print Pre&view",              ID_FILE_PRINT_PREVIEW
    MENUITEM "P&rint Setup   ",             ID_FILE_PRINT_SETUP
    MENUITEM SEPARATOR
    MENUITEM "Recent File",                 ID_FILE_MRU_FILE1,GRAYED
    MENUITEM SEPARATOR
    MENUITEM "E&xit",                       ID_APP_EXIT
END
```

Figure 6 4 A sample resource script This script illustrates the syntax which causes the creation of the `file` menu in any standard `Windows` application

As can be seen from Figure 6.4, the nature of the input is immensely more complex than merely moving buttons from one area of the screen to another. The consequences of an inability to use the same visual features as other programmers, are that it takes significantly longer to produce software than those who are not forced to use such methods. The solution to this problem is twofold. Firstly, it involves the vendors of commercially available screenreaders taking a more active interest in those who are using computers as programming tools. Secondly, the developers of these visually-based environments **must** use standardised approaches to their screen presentations. Until this occurs, there will be no advances in the productivity of blind programmers in this area.

6.4 Prosodic evaluation

In order to evaluate the prosodic component of TechRead, a small-scale experiment was undertaken. The principle objective of this experiment was to determine whether the prosodic model used in the system was a noticeable improvement on the existing forms of synthetic presentation. The only means to access mathematics currently, is to use a linear-based language such as LaTeX, and a text editor to peruse the material, and we felt that the model used in TechRead posed a significant enough improvement to warrant investigation.

With this in mind, a strategy was sought which would enable the comparison of the prosodic model used in TechRead with the existing forms of presentation. Also, since the TechRead model attempts to replicate the rhythms of natural speech, it was considered advisable to also include a comparison between natural speech, and the prosodically enhanced synthetic speech.

Several options were considered at this point. Firstly, the notion of applying digital filters to the natural voice, thereby removing much of the non-prosodic information from the speech signal, was investigated. However, as the prosodic model does not in fact alter the non-prosodic aspects of the synthesiser, this notion was rejected. Also, the methods used by the human speaker in their

presentation had to be considered. Should, for example, the speaker be allowed to include words like *parenthesis* or *bracket* in their verbalisation of the material? It was ultimately decided that the human reader should as far as possible be restricted to the same form of lexical presentation as is to be found in mathematical equations presented by TechRead. However, they were to be left entirely free to decide the prosodic interpretation to place on the material. Combined with this, the human reader was not instructed in the nature of the keywords which they were permitted to use. For example, they were not specifically instructed to use the word *over* to differentiate between the numerator and denominator of a fraction.

The next phase in the design of the evaluation, was to establish the best means whereby the participants could answer the questions. The first means investigated, was to request the listeners to rewrite the material, having heard the utterance played once. It was surmised, that this would yield an accurate idea of what the listeners actually perceived on hearing an audio version of the expression. A significant problem with this form of answering, was the lack of any means to deduce what information the subjects had omitted from their answers, in the event that they were unable to completely comprehend the nature of the material being spoken. Two methods were proposed to incorporate this facility into the evaluation.

The first involved the participants using question-marks to indicate the number of terms they had missed, while a second was based on an application of *Magnitude Estimation*. The latter involved the listeners drawing a line, of a suitable length to indicate the amount of material which they had failed to apprehend. The means whereby the amount of material lost could then be calculated, was to obtain a mean line-length for each participant, and to use this as a bench-mark, from which the interpretation of each particular line could be calculated.

Ultimately, a simpler experimental procedure emerged. This involved the use of multiple choice answers to obtain the participants' feedback on the clos-

est visual match to a vocal presentation. Both the materials used, and the procedure are described in more detail in the subsequent sections.

6.4.1 Materials used

The materials used in this evaluation were significantly different from those employed in the pilot study, discussed in the previous chapter. Our objective here was to present the listener with brief utterances, which encapsulated the types of material which can often be found, grouped together into sub-expressions.

Consequently, the difficulty of the material was reduced to a level where it could be spoken succinctly, without losing any of the expressiveness afforded by TechRead's prosodic representation. In order to ensure completeness, equations were chosen from diverse parts of the mathematical spectrum; ranging from simple fractions and linear expressions encompassed by radicals, to more complex expressions involving summations, limits, integrals and trigonometric functions. The exact equations can be seen in Appendix C. These expressions demonstrate the power of expression which is incorporated into the prosodic model used in TechRead.

The synthesiser used to speak the materials was `DEC-Talk`, as it is the synthesiser for which the system is currently designed. Also, the person who read the equations was an Engineering Graduate, with a high mathematical competence and familiarity. The only restriction placed on the human speaker was that the words parenthesis, bracket, and the phrase *all over* could not be included in the utterance. With these strictures, the method used in presenting the material was left entirely to her own discretion.

6.4.2 Procedure

The procedure employed in this evaluation was a simple one. Twenty subjects were requested to participate, all of whom had a good working knowledge of

the mathematical constructs which would be in the materials presented Each participant was given three different answer booklets, containing four possible options for each spoken equation An extract from an answer booklet may be seen in Appendix C

Before starting the experiment, the nature of the evaluation, and the procedure to be adopted was explained to those present Each answer booklet had a series of instructions, which informed the participants what they were required to do They were asked not to make their selection until they had heard the full equation Once they had made their selection, each subject was requested to indicate their choice, by placing a mark next to their chosen equation

The same set of equations was played three times to the participating group The first playing consisted of the natural voice, the second was the unadorned text strings sent to `DEC-Talk`, while the final playing was based on the prosodic model used in TechRead Each equation was heard once only

Three sample equations were also played before each re-iteration of the experimental equations These equations were of a comparable standard to those found in the experimental materials and the correct answers were given to these sample questions The reason for their inclusion, was to facilitate those who had never listened to a synthetic voice before There is no way that an uninitiated participant could be expected to partake in an evaluation, and obtain viable results without a knowledge of the methods used to present the materials

There was no time-limit placed on the period between each equation, though in the main this proved to be between five and seven seconds Our objective was not to place any undue pressure on the participants, rather we were more interested in ascertaining that they could determine the nature of the equations from their spoken representations

6.4.3 Experimental results and discussion

The results of this experiment proved better than expected The worst case, as hypothesised, was the second recording of the equations, where those participating got an average of 25% of the answers correct This is exactly as expected, as the probability of obtaining correct answers from multiple-choice questionnaires, is one in four, or 25% The answers to this section of the evaluation provide some interesting insights into people's perception of spoken mathematics, as will be discussed below Before embarking on a detailed analysis of a comparison of the various results, it is first necessary to discuss in detail the break-down of each of the individual recordings

As was stated in the previous section, the first recording of the materials comprised the naturally spoken versions On average, the participants obtained 86% of the questions correct This ranged from a maximum where 100% of the participants correctly answered Question 12, to 60% for Question 3 Both these scores have implications for the prosodic model used in TechRead, as will be discussed below With these exceptions, the results lay roughly in the range 70%–95% correct This was expected, as we conjectured that the prosody used in natural speech, would contain more features which would aid in the decomposition of the mathematical expressions It should be stated at this juncture that the figures quoted here are averages only For a comparison of the overall average figures obtained from the three recordings, see Table 6 1

The version comprising unadorned textual strings proved to be the worst of the three recordings As `DEC-Talk` was permitted to interpret the textual mathematical material as it saw fit, it was surmised that the laws of probability would play a significant part in the averages obtained from this portion of the evaluation As was previously alluded to, the average number of correct answers obtained by those participating was 25% However, there are some interesting outcomes from this particular recording In Question 2 (see Appendix C), 75% of those participating obtained the correct answer Conversely, the remainder of

the equations provided relatively consistent answers; lying between 0% and 40% correct. Once again, the prosodic implications for the correctness of Question 2 are interesting, and will be described in the context of the other results below.

In the case of the third version (the prosodically enhanced model used in TechRead), the subjects obtained an average of 79% correct answers. The closeness between the average of correct answers for the naturally spoken and prosodically enhanced versions was quite surprising, as although it was expected that the third recording would produce significantly better results than the unenhanced synthetic version, it was not anticipated that the performance would be so high. The highest number of correct answers was obtained for Question 4, where 95% of the subjects were correct, while the minimum was yielded by Question 9 at 45%. The remainder of the answers were within the range of 75%–90%. There are reasons why this closeness has occurred, and these will be expounded in the following paragraphs.

As Table 6.1 demonstrates, although the general trend is clear there were several anomalies in each of the three versions. In terms of the natural speech, the most significant of these was the 100% correctness achieved by the participants for Question 12. In this particular recording, the speaker inserted the lexical cue *multiplied by* to indicate that the first quantity should be multiplied by the two contained in parentheses (see Appendix C). As expected, the unadorned synthetic speech faired badly in this instance,as only 10% of the participants correctly answered. The prosodically enhanced version achieved a correctness level of 85%, which was an enormous improvement on the unadorned version, but was not quite as effective as the presentational style of the human speaker. This fact has implications for the accurate depiction of mathematical content using synthetic speech. The expression $b(c+d)$ can be intuitively recognised as "b multiplied by, c+d". Alternatively, a construct such as $f(x+y)$ appears syntactically comparable to the previous example, but is semantically different, as the latter is intended to denote a function. The issue arises, as to how to convey these differences using prosody. It would appear,

that without a great deal of analysis of the material it would prove impossible to devise alternative presentational strategies to convey these syntactically comparable items, and hence the model will be left unaltered at this time

As can be seen in table 6 1 the minimum level of correctness obtained by the naturally spoken vversion was 60% for Question 3 Interestingly, 90% of the participants correctly identified the equation from the the prosodically enhanced representation The reasons for this can only be attributed to the fact that it is possible (and indeed desirable) to enhance the prosodic alterations of the voice to ensure intelligibility By virtue of this fact, the more regular pausing of which the speech synthesiser was capable, ensured greater clarity of presentation, thereby ensuring a higher degree of intelligibility

Table 6 1 shows that the best observable performance in the case of the unadorned textsstrings was found in Question 2, where a total of 75% of those participating obtained the correct answer This compared favourably with the other two versions, as the natural speech scored a total of 90% correct, while the prosodically enhanced synthetic version obtained a score of 80% The fact that the two synthetic versions of the material were so close in the numbers of correct answers highlights the ambiguities which prosody can remove It also presents a case for excluding prosodic alterations from certain forms of presentations For example in the case of linear formulae (such as $a + b + c$) there is a strong case for maintaining an unaltered prosody This could also apply to a prosodic equation, where both sides of the parse tree were equally balanced, as is the case in Question 2 (see Appendix C) Here, the lexical operator *over* forms the root node, with the equally balanced numerator and denominator forming the right and left branches As the sample size in the experiment is quite small, it is not possible to say with complete certainty that the lack of prosody can be used in balanced, or otherwise linear equations Future, more large-scale evaluations will have to take this aspect of the synthetic presentation into account

The worst case performance level for the unadorned synthetic version of the material arose in Question 9, where none of the participants obtained the

correct answer (see table 6 1) This equation demonstrates a fractional element, the numerator of which also contains one element which is encompassed by a radical This equation $\frac{\sqrt{b}+c}{d}$ also proved difficult to convey using prosodically enhanced speech, as only 45% of the subjects obtained the correct answer, while 85% obtained the correct answer when listening to the natural speech The erroneous answers were predominantly given as $\frac{\sqrt{b+c}}{d}$ when the enhanced version was being listened to, while those wrong answers for the unenhanced version were mainly $\sqrt{\frac{b+c}{d}}$

It is obvious, that the grouping of the discrete elements of this type of presentation poses a significant problem for synthetic speech The extra prosodic features of natural speech seem to aid in discerning the nature of this type of ambiguous equation It is just this type of problem for which the interface of Chapter 3 has been designed to solve Further investigation will be needed to determine a prosodic strategy for catering for this form of equation For example, instead of incorporating a baseline fall at the end of the numerator, it may be necessary to determine the methods of incorporating a rising f_0 contour at the ends of such utterances

The main observable problem with the prosodically enhanced version of the equations presented, seems to be the deduction of where the scope of such elements as radicals and superscripts end In most of those equations where a significant number of erroneous answers were given, the problem would appear to suggest that the subjects were not able to ascertain the scope of the superscripts, or which elements were encompassed by the radical symbol This would seem to suggest the incorporation of some form of pitch-contour variation into the presentation of the mathematical expressions This could be achieved by the use of rising (question-mark) clause boundaries to indicate the fact that there were elements remaining to be spoken as part of the existing sub-expression We believe that the use of nested pausing, and the rate changes described in the previous chapter assist the listener in ascertaining the correct structure of the mathematical expressions, and with the further introduction of some increased

Equation	Natural (%)	Unenhanced (%)	Enhanced (%)
1	85	20	80
2	90	75	80
3	60	30	90
4	90	40	95
5	95	20	90
6	90	50	85
7	95	5	75
8	70	15	80
9	85	0	45
10	95	30	70
11	80	15	75
12	100	10	85
13	85	10	75
Ave Correct	86	25	79

Table 6 1 Table of average correct answers for each equation

pitch contour modifications, the results should tend even more towards natural speech

6.5 Summary

This chapter has discussed the details pertinent to the implementation of the TechRead system We have shown those strategies which are relevant to the production of the document model, the Braille output, and the textual and mathematical spoken output We have also demonstrated how the mathematical prosody of the system has been evaluated, and the results obtained were analysed and discussed The next chapter completes the discussion of this research, and describes the proposed future of the TechRead system.

Chapter 7

Conclusion

The primary objective of this research, was to devise methods for communicating highly technical material to blind people, through the medium of Braille or prosodically enhanced spoken output This solution necessitated devising strategies to both model the document internally, and to unambiguously produce the material in the two output media

The first phase in the generation of intelligible output was the transformation of the LaTeX source into well-formatted and accurate Braille Following on from this, methodologies were defined to convey the structure and textual content of documents using prosodic alterations to the synthetic voice We have devised mechanisms whereby mathematical content can be delivered in an intuitive manner, using the sole medium of prosodically enhanced spoken output This ensures that the listener will not have to learn specific, and non-intuitive non-speech auditory sound to gain access to this form of presentation

We have also devised a newer, and more flexible means for representing the structure and content of the document in the computer's memory This Directed Graph is a radical departure from the traditional, tree-based approach of the past, and facilitates rapid and efficient browsing of the document's hierarchy

This thesis was motivated by the fact that an increased accessibility for blind

people to technical documents is essential In order to ensure that students can actively participate in integrated educational programs, the present state of inaccessibility must be redressed Previous systems such as ASTER [Ram94] and MathTalk [Ste96] have attempted to solve this problem, with varying degrees of success Despite their efforts, the realms of higher mathematics, and other scientific areas are still unattainable for many blind people

Previous systems have employed a multi-modal approach to solving the problems of technical accessibility ASTER uses both spoken and non-speech audio to convey the document to the user, while MathTalk uses spoken output to convey the equations, but *earcons* to provide an overview of the material TechRead is the first system to use solely prosodic alterations to impart the structure and content of documents to the listener The reason for this is that the prosodic component of speech is used intuitively by everyone in the course of their natural speech Its application to the production of mathematical output will, we believe, result in a more intuitively understood rendering of the content

Based on our research, it can be concluded that the methodologies described in this system will be effective in providing more accessible technical documents Firstly, the document model described in Section 3 1 permits the rapid and efficient browsing of both the structure, and content of the document The Directed Graph of the document lends itself to the incorporation of heterogeneous objects, each with its own innate characteristics This ensures that each object can be successfully interconnected and browsed using alternative strategies as is appropriate For example, a mathematical object will require a different browsing strategy from a paragraph of text, but using the representation provided by TechRead it will prove possible to browse each successfully We believe that this model will offer more intuitive browsing, as it more accurately reflects the overall structure of a document ASTER [Ram94] used a tree-like architecture to represent the document structure, and a *quasi-prefix* notation to depict the mathematical content of the document (see section 2 2 2) Though these representations obviously work, they require an advanced knowledge of

both computing and ASTER to ensure successful navigation through technical material As we demonstrated previously, the interface of ASTER uses this model effectively, once the user has an in-depth knowledge of the algorithms involved in producing the output, and the mnemonics involved in navigating through the document

TechRead's interface on the other hand, is primarily based on the numeric keypad This ensures both a centrally located command set, and a shallower learning curve for the user The use of the arrow keys on this portion of the keyboard (see Figure 3 6 on page 105) ensures that a user can move through the document hierarchy in an efficient and intuitive manner

The most notable aspect of TechRead is the combination of output media it provides Facilities are incorporated into the system to enable the more traditional Braille to be produced, while it also harnesses the prosodic features of speech synthesisers to produce more intelligible and natural-sounding spoken output The choice of output medium enables a multi-modal interaction with the system, as the user can listen to the material while sitting at the computer, or take a Brailled "hard copy" away with them There is also another reason for the inclusion of Braille as an output medium In future versions of TechRead, we envisage the incorporation of *Refreshable Braille Displays* as output devices in the system Many blind people do not like the artificially produced synthetic voice, and prefer to use Braille to access their computers Indeed, it is more common in Europe to find blind people using this form of interaction than synthetic speech Consequently, those methods of browsing and Braille production will need to be adjusted to cater for the newer output device For example, most *Braille Displays* have push-buttons, which the user can activate using their thumbs These buttons facilitate the scrolling of the display, without the necessity for the user to remove their hands, press one of the arrow keys, and then re-locate back to the display The presence of these extra push-buttons can be incorporated into the system, and they could theoretically be used as an alternative to the numeric keypad interface of Section 3 3

The Braille output itself contains two alternative output strategies The first of these is in strict accordance with those standards devised by the Braille Authority of the United Kingdom (BAUK), while the second includes alternatives which we believe, will increase the readability of the material For example, Chapter 4 described some of the layout standards which form a part of TechRead's Braille output These standards are concerned primarily with optimising the use of the available space, and as a consequence some sacrifices are made in areas of clarity and readability The improvements suggested in Section 1 2 will not use the available space in the same manner Instead, the improvements included in TechRead will place less actual content in the same area, but use the space more effectively

It is our belief that (certainly in the context of mathematical or other syntactically complex material) these alterations to the layout will ensure that the reader can peruse the content at greater speeds than is currently the case For example, the improvements which we have suggested, ensure that the Braille versions of itemised lists follow the same principles of layout as are used when producing the printed output This will consume more paper, but will, we believe make the material more readable

Another concrete example of where the changes of layout will have a greater impact is where they are applied to mathematical output As was discussed in Chapter 4, we recommend the convergence of Braillle with the printed version, that is, the greater use of relative vertical position to impart the meaning of the expression The use of these alternatives, makes the reading of mathematical material a more enjoyable, and productive experience However, the BAUK standards are included in order to produce output which is consistent with recognised guidelines, and which can be generally used by those who do not wish to follow our recommendations

Though the production of accurate and well-formatted Braille is an extremely important part of the TechRead system, the main focus of our research has been the production of more intelligible spoken output We set

out to achieve two goals to devise a means of conveying alterations in the visual appearance of documents using changes in the vocal aspect, and to produce intelligent-sounding verbalisations of mathematical expressions The former objective can be further reduced to sub-goals to produce spoken output which demonstrated the structure of a document, and also informed the user of changes in the visual attributes of running text The prosodic model used in TechRead has been designed with these needs uppermost in mind

The primary objective of this part of the system is to convey as much as possible, using as little as possible This is exemplified by the lack of lexical cues in mathematical expressions Chang [Cha83] devised a set of rules for the inclusion of extra descriptive information to supplement mathematical expressions These lexical cues were discussed in Section 2 4 1 The spoken output produced by TechRead, on the other hand, ensures that such additions to the utterance are kept to a minimum The reason for this is to minimise the load which is placed on the listener's short-term memory By keeping the utterance as brief as possible, the amount of information which the listener needs to remember is lessened considerably Consequently, their focus can be directed towards the comprehension of the material, rather than the retention of vast amounts of superfluous spoken output

Experimental evidence, coupled with some informal empirical testing, has revealed that the methods used in TechRead are eminently successful in conveying the underlying structure of equations In Chapter 6 the nature, and results of a pilot study were discussed It can be seen from the results obtained, that the listeners, though unfamiliar with synthetic speech, could accurately determine the structure and content of mathematical equations It can be inferred, that using this model in conjunction with the interface of Section 3 3, the users of TechRead will be able to peruse the mathematical content of documents as readily as their sighted colleagues The prosodic enhancements, combined with the keypad-based interface will provide both the structural and contextual views of the material

The use of a *running paraphrase* as an overview of the material is a novel approach to giving superficial views of the material Previous systems used a diverse range of methods to offer this information, varying from the *Variable Substitution* of ASTER [Ram94] to the musical *glance* of MathTalk [Ste96] TechRead's use of pure spoken audio, with prosodic enhancements, is a new direction in providing such an overview We believe that it is more intuitive than the methods employed in MathTalk, and less cumbersome than those used in ASTER The use of a brief passage of text which accurately describes the content of an equation is, we believe, equivalent to a sighted reader merely glancing at the material The overall structure of the expression is observable, but not the finer detail

Our work on TechRead has concluded that the need for such a system is overwhelming As anecdotal evidence of this fact, it has proved extremely difficult to proof-read this thesis using the traditional screenreader and speech synthesiser Consequently, much sighted assistance was required in order to ascertain that fonts were used consistently, paragraphs were not too large, and many other aspects of visual presentation were correct TechRead would assist greatly in this type of undertaking The use of either prosodic enhancement, or Braille output would reveal the visual aspects of the document, which are not presently accessible to blind readers

7.1 Future work

Up to this point, the development of TechRead has consisted of several small-scale prototype systems to demonstrate that various theoretical aspects worked in practice As a consequence, there are various disparate programs in existence, each of which performs a specific function The immediate future, is to attempt to link these subsystems together into a cohesive software package, which can then be submitted for testing We propose examining the use of other speech synthesisers, to determine the appropriate mappings between the theoretical

model of prosody used in TechRead and the physical constraints imposed by each individual synthesiser Ultimately TechRead will function with a wide range of synthetic speech devices, and Braille displays Finally, additions to the document model are being investigated, with a view to incorporating such features as audio depictions of pictures, or other highly graphical material

7.2 Final Remarks

The role of the computer in the next number of years will be vital in enabling blind people to play an equal part in academic and working environments TechRead addresses the needs of those people who cannot read technical material in the regular manner, but must rely on alternative means of presentation The models and methodologies described in this thesis prove that such access is theoretically, and practically possible What remains is to implement the ideas and to increase the availability of technical material to blind people

Appendix A

Table of Braille Signs

Braille is a method of reading by touch Braille letters are made of raised dots, like the six dots on a domino Devised by a Frenchman, Braille originally had no W K - T are like A - J but with one more dot, U - Z have two more Table 1 1 demonstrates the Braille alphabet

A	B	C	D	E	F	G	H	I	J
a	b	C	d	e	f	g	H	i	J
K	L	M	N	O	P	Q	R	S	T
k	l	M	n	o	P	q	R	s	T
U	V	W	X	Y	Z	for	With	of	and
u	v	W	x	y	Z	=	)	(	&

Table 1 1 The alphabet and some simple contraction in Braille

As can also be seen from Table 1 1, there are short forms of some common words Numbers are made by using the numeral symbol #, followed by the letter A - J to represent the numbers 1 - 10 So 6 looks like #6 Braille punctuation marks can also be seen in Table 1 2

,		,	!	()	?"	"	hyphen
;	:	.	,	!	( )	?"	"	-

Table 1 2 Punctuation symbols used in Braille

Appendix B

Visualising Mathematics

B.1 Extracts from the answer paper

Visualizing Maths: a Pilot Study

You will be presented with 15 different equations Each equation will be shown for a period of 5 seconds Remember that **this is not a test of your mathematical knowledge** Rather we want merely to see what you remember Please attempt all questions In all cases where multiple options are given, please place a ring around the appropriate letter to indicate your choice(s)

Before beginning, please complete the details below **Do not** include your name

1 Please place a ring around the appropriate letter

- A— UnderGraduate student
- B— PostGraduate student
- C— Staff Member
- D— other

2 If you are an undergraduate student, please state your course and present year of study

3 If you are a graduate, please indicate the nature of your primary degree

Equation 1

1 How would you describe this equation?

- A— simple
- B— Fairly simple
- C— Complicated
- D— Very complicated

2 Which term(s) do you think best describe this equation

- A— This is a differential equation
- B— This equation contains a fraction
- C— This equation is a polynomial
- D— This equation is a matrix
- E—contains functions
- F— Contains subscripts
- G— Contains Superscripts
- H— contains trig functions
- I— Contains a summation
- J— contains a partial derivative

- K— None of the above

3 Please indicate the operator(s) you think this equation contains

- A— $\int$
- B— $\frac{d}{dx}$
- C— ∂
- D— $\sum$
- E— $\sqrt{}$
- F— $f(x)$
- G— lim

4 Please write as much of the equation as possible in your own words

B.2 Equations used

#	Equation
1	$\lim_{x \to 0^-} \left(\frac{1}{x} + \frac{1}{x^2} \right)$
2	$\lim_{x \to 3} \frac{1-\sqrt{x-2}}{x-3}$
3	$\lim_{\delta x \to 0} \frac{f(x_0+\delta x)-f(x_0)}{\delta x}$
4	$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{g(x)\frac{d}{dx}[f(x)]-f(x)\frac{d}{dx}[g(x)]}{[g(x)]^2}$
5	$\int \left[\frac{cos\sqrt{x}}{2\sqrt{x}}\right] dx$
6	$\sum_{k=11}^{28} (k-10) sin\left[\frac{\pi}{k-10}\right]$
7	$A = \lim_{n \to +\infty} \sum_{k=1}^{n} f(d_k)\Delta x$
8	$S = \int_a^b 2\pi f(x)\sqrt{1+[f'(x)]^2}dx$
9	$\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{(k+1)\ln(k+1)}$
10	$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \quad + \frac{f^{(n)}(a)}{n}(x-a)^n + \frac{f^{(n+1)}(c)}{(n+1)}(x-a)^{n+1}$
11	$\frac{\partial^2 u}{\partial x \partial y} = (x^2 - y^2)\{t\, f''(t) + 3f'(t)\}$
12	$\sigma = \sqrt{\left[\frac{\sum(x-\overline{x})^2}{n}\right]}$
13	$\begin{bmatrix} S \\ \gamma \\ \tau(\gamma) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \zeta^2 & \zeta \\ 1 & \zeta & \zeta^2 \end{bmatrix} \begin{bmatrix} \theta \\ \sigma(\theta) \\ \sigma^2(\theta) \end{bmatrix}$
14	$\sum_{1 \le i \le n} a_i = 1$
15	$D_x^n w = \sum_{0 \le j \le n} \sum_{\substack{k_1+k_2+\ +k_n=j \\ k_1+2k_2+\ +nk_n=n \\ k_1,k_2,\ ,k_n \ge 0}} D_u^j w \frac{n!(D_x^1 u)^{k_1}}{k_1!(1!)^{k_1}} \frac{(D_x^n u)^{k_n}}{k_n!(n!)^{k_n}}$

Table B 1 Equations used in Pilot Study

B.3 Results

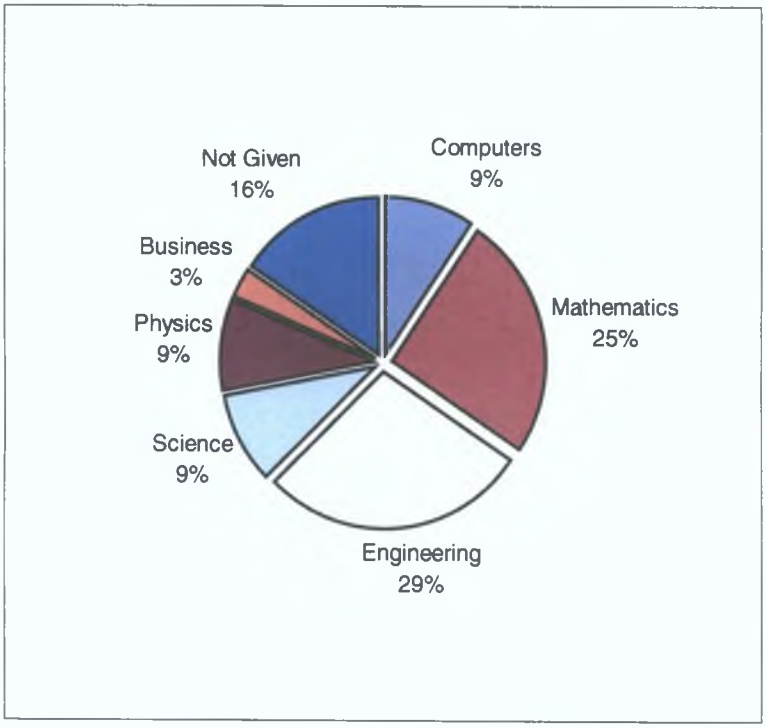


Figure B.1: Mathematical Background

Category	Computers	Mathematics	Engineering	Science	Physics	Business	Not Given
Quantity	3	8	9	3	3	1	5
Percentage	9%	25%	28%	9%	9%	3%	16%

Table B.2: Mathematical Background of participants

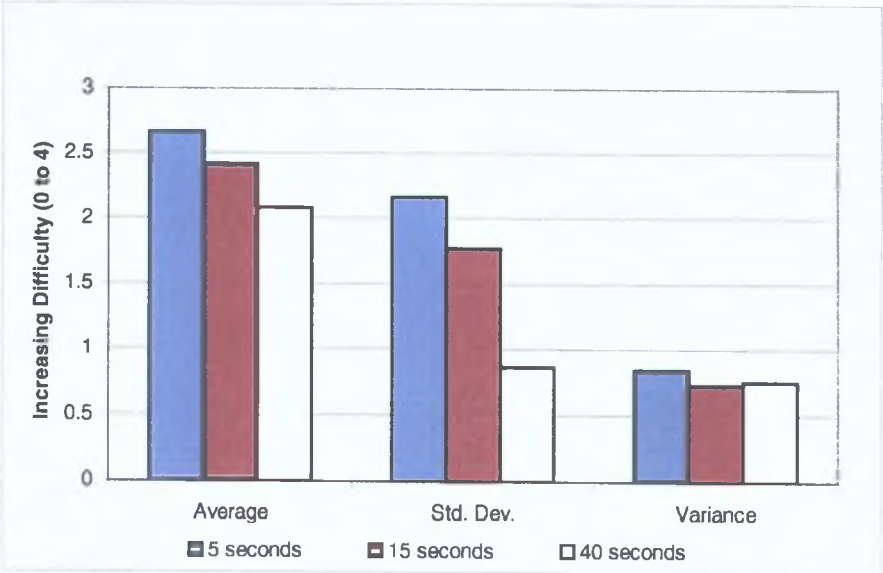


Figure B.2: Perceived difficulty of identical equations over time

Increasing Difficulty	Time	Average	Std. Dev.	Variance	Min	Max
(from 0 to 4)	5	2.662	2.162	0.838	1	4
	15	2.415	1.767	0.724	1	4
	40	2.078	0.858	0.750	0	4

Table B.3: Perceived difficulty of identical equations over time

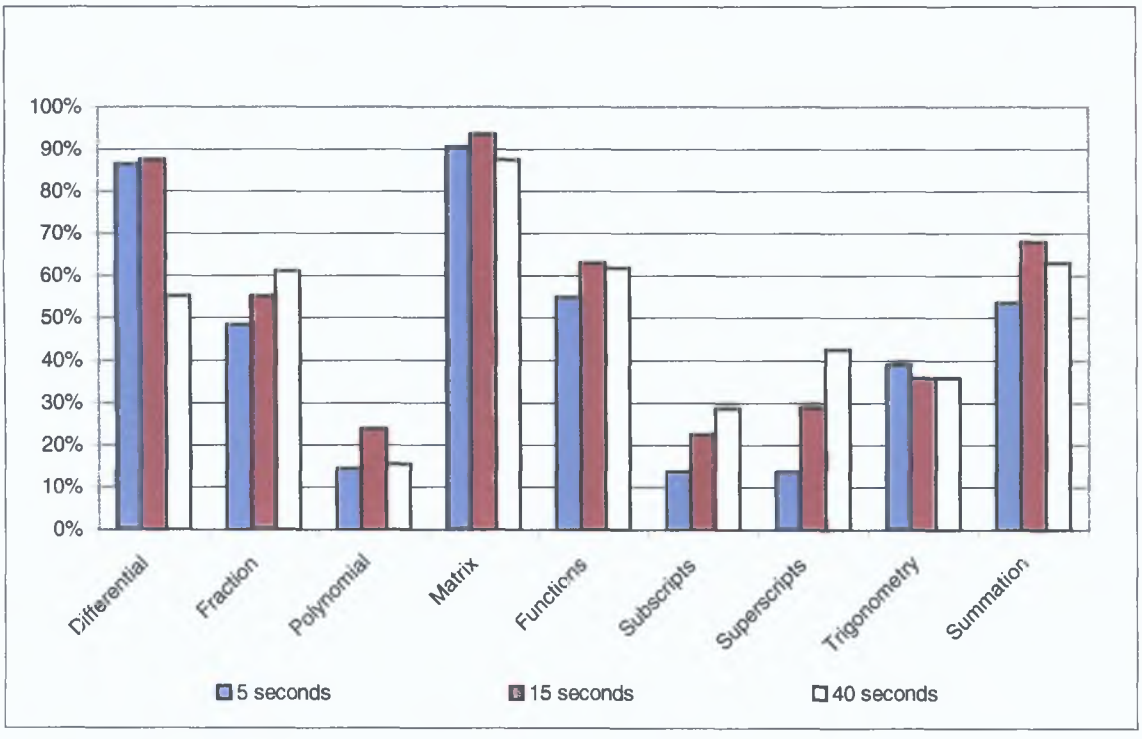


Figure B.3: Results obtained from Question 2

Time	Differential	Fraction	Polynomial	Matrix	Functions	Subscripts	Superscripts	Trigonometry	Summation
5	86%	48%	15%	91%	55%	14%	14%	39%	54%
15	88%	55%	24%	94%	63%	23%	29%	36%	68%
40	55%	61%	16%	88%	62%	29%	43%	36%	63%

Table B.4: Results obtained from Question 2

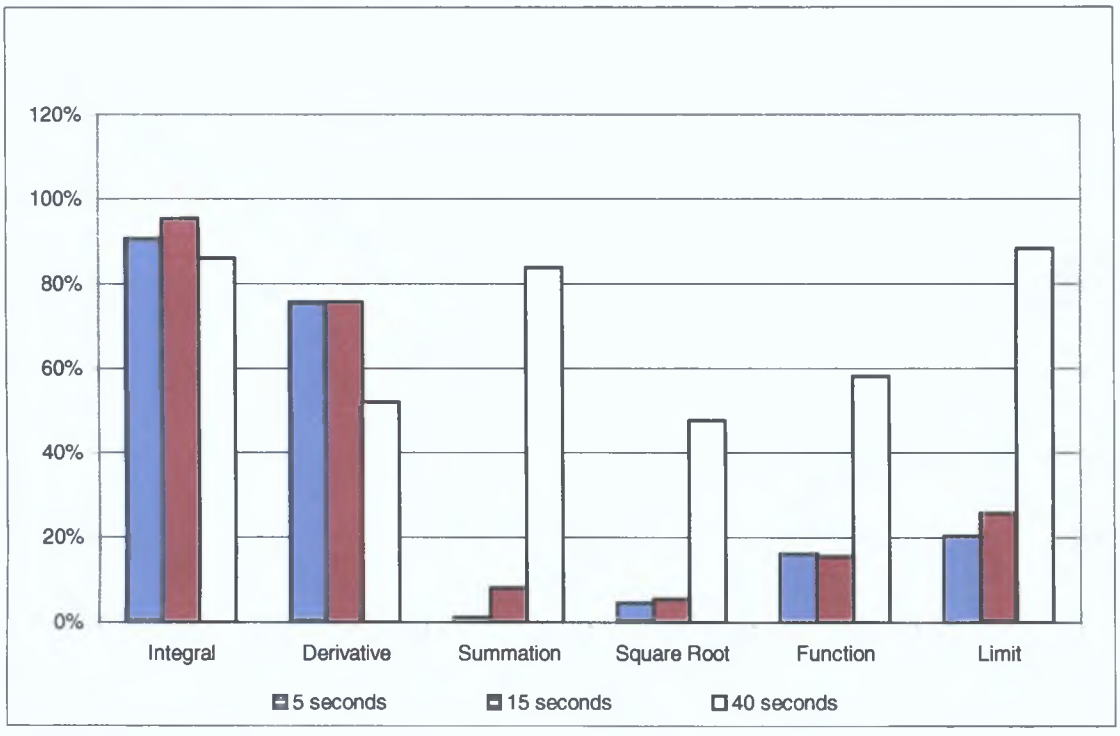


Figure B.4: Results obtained from Question 3

Time (seconds)	Integral	Derivative	Summation	Square Root	Function	Limit
5	91%	76%	1%	5%	16%	20%
15	95%	76%	8%	5%	16%	26%
40	86%	52%	84%	48%	58%	88%

Table B.5: Results obtained from Question 3

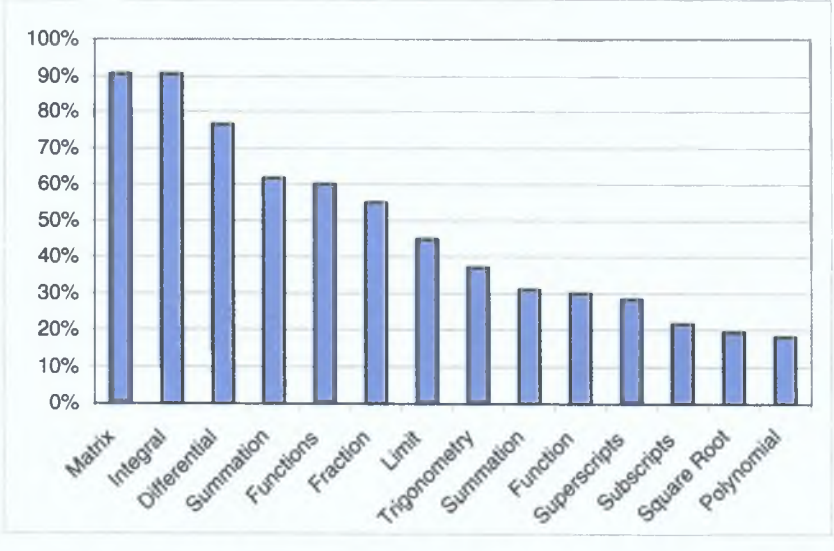


Figure B.5: Order of recognition of mathematical objects

Mathematical Object	%
Matrix	91%
Integral	91%
Differential	76%
Summation	62%
Functions	60%
Fraction	55%
Limit	45%
Trigonometry	37%
Summation	31%
Function	30%
Superscripts	28%
Subscripts	22%
Square Root	19%
Polynomial	18%

Table B 6 Order of recognition of mathematical objects

Appendix C

Evaluation of the prosodic model

C.1 Extract from the answer paper

Instructions

Please indicate your choice below by placing a mark opposite the equation you think most appropriate You will hear each equation **once** Please do not make your selection until the equation has been completed There is no time limit You will hear a set of formulae spoken once using human speech, once using unenhanced synthetic speech, and once using prosodically enhanced synthetic speech

Question 1

- $(x + y + z)^2$
- $(x + y)^2 + z$
- $x + (y + z)^2$
- $x + y + z^2$

Question 2

- $\sqrt{\frac{b}{c}} + d - e$
- $\sqrt{\frac{b}{c+d}} - e$
- $\sqrt{\frac{b}{c} - e}$
- $\sqrt{\frac{b}{c+d}} - e$

Question 3

- $\cos(xy) + \tan^2 x + y$
- $\cos(xy) + \tan^2(x + y)$
- $\cos x\, y + \tan^2 x + y$
- $\cos x\, y + \tan^2(x + y)$

C.2 Equations used

Equation No.	Equation	Equation no.	Equation
Sample 1	$(x+y+z)^2$	Sample 2	$\sqrt{\frac{b}{c+d}} - e$
Sample 3	$\cos(xy) + \tan^2(x+y)$	Question 1	$b + \frac{c}{d} - e$
Question 2	$\frac{b+c}{d+e}$	Question 3	$\lim_{b \to 1} \frac{b}{(c-d)^2}$
Question 4	$\tan(x+2y)^2$	Question 5	$\frac{b-c}{d} + e$
Question 6	$\int_0^1 (x^y - \frac{z}{10}) dz$	Question 7	$\sqrt{b+c} + d^2$
Question 8	$\frac{f'(x) + f(x)y + z}{xy}$	Question 9	$\frac{\sqrt{b}+c}{d}$
Question 10	$\sum_{i=1}^{N} a^i + b + ci$	Question 11	$\log 3(x+y)^3$
Question 12	$\frac{b(c+d)}{e^2}$	Question 13	$\sin^2(x+y) + \cos x + y^2$

Table C 1 Equations used in the evaluation of prosodic model

C.3 Results

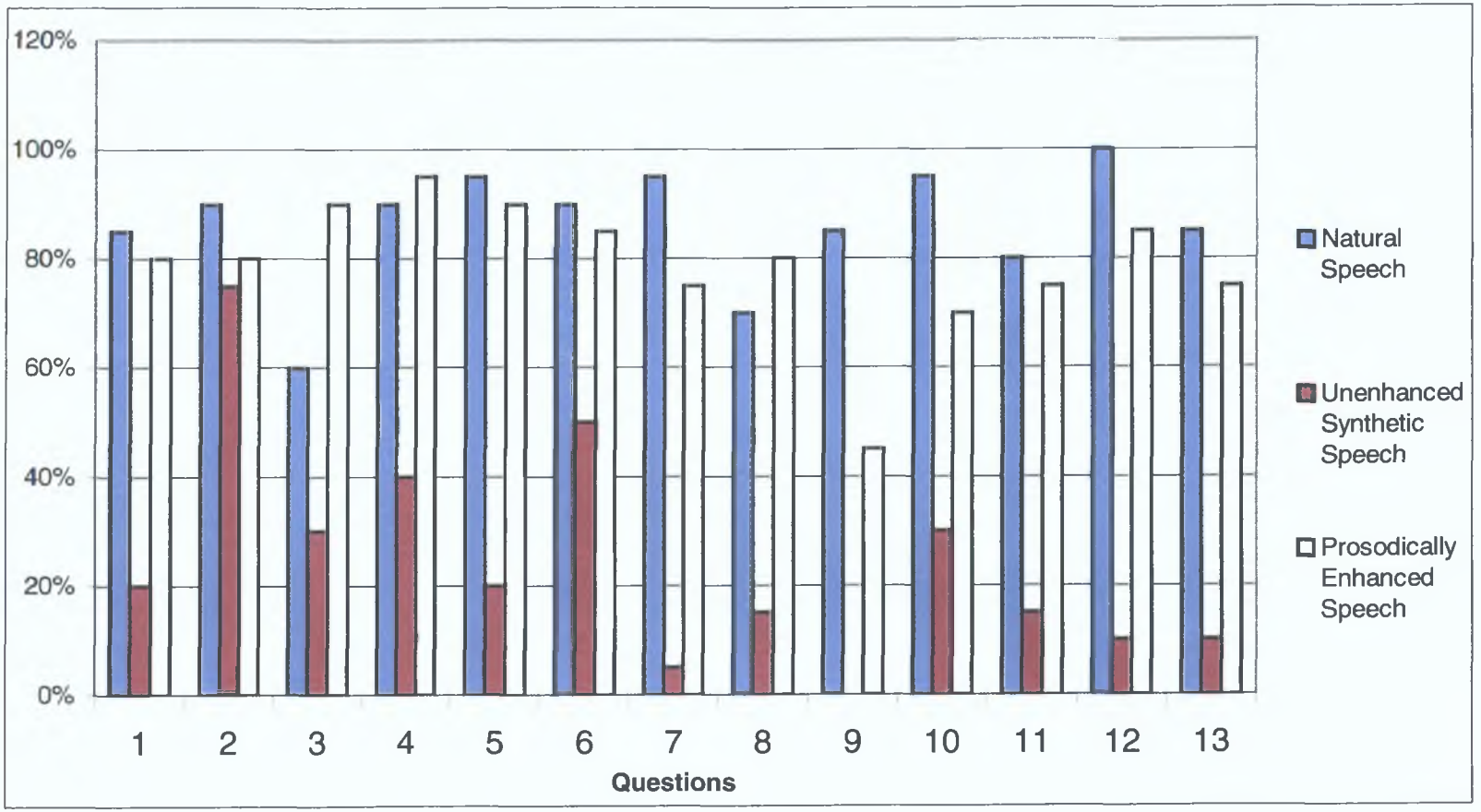


Figure C.1: A comparison of the performance of three different types of spoken output used in the prosodic evaluation.

Question	1	2	3	4	5	6	7	8	9	10	11	12	13	Average
Natural Speech (%)	85	90	60	90	95	90	95	70	85	95	80	100	85	86
Unenhanced Speech (%)	20	75	30	40	20	50	5	15	0	30	15	10	10	25
Prosodically enhanced speech (%)	80	80	90	95	90	85	75	80	45	70	75	85	75	79

Table C.2: A comparison of the performance of three different types of spoken output used in the prosodic evaluat

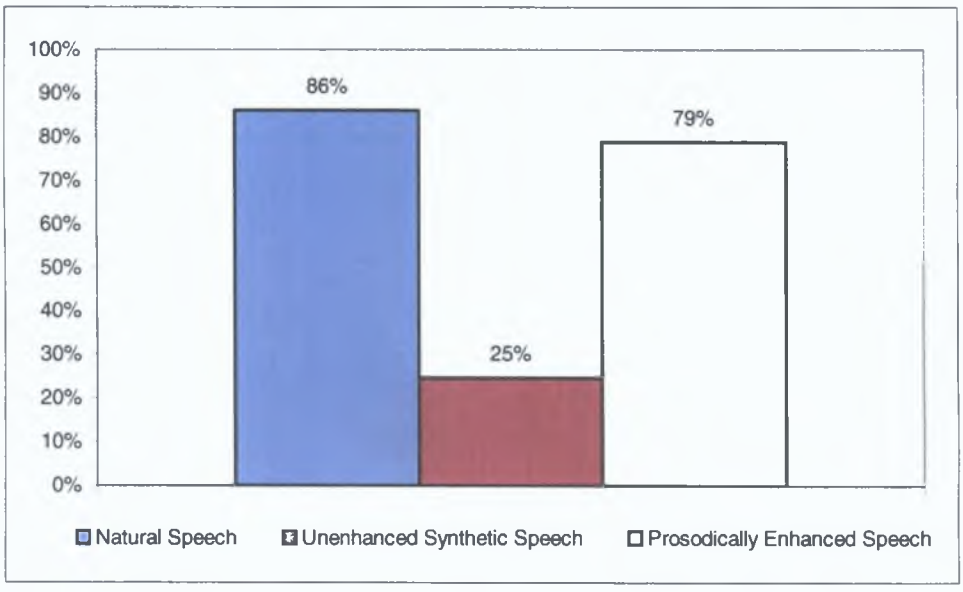


Figure C.2: Average performance of the three different types of spoken output used in the prosodic evaluation

Type	Average
Natural Speech	86%
Unenhanced Speech	25%
Prosodically enhanced speech	79%

Table C.3: Average performance of the three different types of spoken output used in the prosodic evaluation

References

[AHK87] J Allen, S Hunnicutt, and D H Klatt *From Text to Speech The MITalkSystem* Cambridge CUP, 1987

[Arr90] R Arrabito Computerized braille typesetting Some recomendations on mark-up and braille standard Technical report, The University of Western Ontario, August 1990

[Cha83] L A Chang *Handbook for Spoken Mathematics (Larry's Speakeasy)* Lawrence Livermore Laboratory, The Regents of the University of California, 1983

[CK86] E Couper-Kuhlen *An Introduction To English Prosody* Edward Arnold Ltd , 1986

[DNA97] Stevens R D , Edwards A D N , and Harling P A Access to mathematics for visually disabled students through multi-modal interaction *Human-Computer Interaction*, 12(1 and 2) 47–92, 1997

[Kla80] D H Klatt Software for a cascade/parallel synthesiser *JASA*, 67 971–995, 1980

[Knu84] D E Knuth *The TEX Book* Addison Wesley, 1984

[Kru97] D J Kruglinski *Inside Visual C++ Updated for Version 5 0 and Internet Development.* Microsoft Programming Series Microsoft Press, 1997

[LA87] D R Ladd and Monaghan A Modelling rhythmic and syntactic effects on accent in long noun phrases *proceedings of Eurospeech*, 2 29–32, 1987

[Lad96] D R Ladd *Intonational Phonology* Cambridge CUP, 1996

[Lam85] L Lamport *Latex - A Document Preparation System - Users Guide and reference manual* Addison Wesley, Reading, 1985

[LTD92] J R Levine, Mason T , and Brown D *Lex and yacc* O'Reilly and Associates, 2 edition, 1992

[MJ95] E Maler and El Andaloussi J *Developing Sgml Dtds From Text to Model to Markup* Prentice Hall, 1995

[Mon90] A Monaghan Rhythm and stress shift in speech synthesis *Computer Speech and Language*, 4(1) 71–78, 1990

[Mon91] A Monaghan *Intonation in a Text-to-Speech Conversion System* PhD thesis, University of Edinburgh, 1991

[Mon93] A Monaghan What determines accentuation? *Journal of Pragmatics*, 19 559–584, 1993 Also available from `http //www compapp dcu ie/~alex/PUB/prepositions ps`

[Mon99] A Monaghan State-of-the-art summary of european synthetic prosody r & d To appear in working papers of COST 258 Available from `http //www compapp dcu ie/~alex/PUB/soap html`, 1999

[MR91] A Monaghan and Ladd D R Manipulating synthetic intonation for speaker characterisation *ICASSP*, 1 453–456, 1991

[Nem72] A Nemeth *Nemeth code of braille mathematics and scientific notation* American Printing House for the Blind, 1972

[otUK87] Braille Authority of the United Kingdom *Braille Math Notation* Royal National Institute for the Blind, 224 Great Portland street London w1n 6aa, 1987

[otUK90a] Braille Authority of the United Kingdom *Braille Science Notation* Royal National Institute for the Blind, 224 Great Portland street London w1n 6aa, 1990

[otUK90b] Braille Authority of the United Kingdom *Computer Braille Code* Royal National Institute for the Blind, 1990

[R 94] Furuta R Defining and using structure in digital documents *Proceedings of the First Anual Conference on the Theory and Practice of digital libraries*, June 1994 Available at `http //www csdl tamu edu/DL94/`

[Ram94] T V Raman *Audio Systems for Technical Reading* PhD thesis, Department of Computer Science, Cornell University, NY, USA, May 1994

[RP89] K Rayner and A Pollatsek *The Psychology of Reading* Prentice Hall, 1989

[RPL+91] J V Ralsten, D Pisoni, S E Lively, B. G Green, and J B Moulinix Comprehension of synthetic speech produced by rule *Human Factors*, 33(4) 471–491, 1991

[Sch98] W Schweikhardt 8-dot braille for writing, reading and printing texts which include mathematical characters *in proceedings of IC-CHP*, pages 324–333, September 1998

[Smi93] J Smither Short term memory demands in processing synthetic speech by old and young adults *Behaviour and Information Technology*, 12(6) 330–335, 1993

[Ste96] Robert David Stevens *Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People* PhD thesis, Department of Computer Science, January 1996

[Wal99] E Walsh Brailler version 1 0 Technical manual Available on request from the School of Computer Applications, Dublin City University, Glasnevin Dublin 9, Ireland , April 1999

[Wat87] J A Waterworth The psychology and technology of speech, symbiotic relationship In J A Waterworth, editor, *Speech and Language Based Interaction with Machines Towards the Conversational Computer* John Wiley, 1987