Reflection and Its Use

from Science to Meditation

9.3.2004:1019

Henk Barendregt
NIII, Nijmegen University
The Netherlands

Contents

1. Overview

The phenomenon of *reflection* will be introduced and clarified by examples. Reflection plays in several ways a fundamental rôle for our existence. Among other places the phenomenon occurs in life, in language, in computing and in mathematical reasoning. A fifth place in which reflection occurs is our spiritual development. In all of these cases the effects of reflection are powerful, even downright dramatic. We should be aware of these effects and use them in a responsible way.

Reflection: domain, coding and interaction

Reflection occurs in situations in which there is a *domain* of objects that all have *active meaning*, i.e. specific functions within the right context. Before turning to the definition itself, let us present the domains relevant for the four examples. The first domain is the class of proteins. These have indeed specific functions within a living organism, from bacterium to *homo sapiens*. The second domain consists of sentences in natural language. These are intended, among other things, to make statements, to ask questions, or to influence others. The third domain consists of (implemented) computable functions. These perform computations—sometimes stand alone, sometimes interactively with the user—so that an output results that usually serves us in one way or another. The fourth domain consists of mathematical theorems. These express valid phenomena about numbers, geometric figures or other abstract entities. When interpreted in the right way, these will enable us to make correct predictions.

Now let us turn to reflection itself. Besides having a domain of meaningful objects it needs *coding* and *interaction*. Coding means that for every object of the domain there is another object, the (not necessarily unique) *code*, from which the original object can be reconstructed exactly. This process of reconstruction is called *decoding*. A code C of an object O does not directly possess the active meaning of O itself. This happens only after decoding. Therefore the codes are outside the domain, and form the so-called *code set*. Finally, the interaction needed for reflection consists of the encounter of the objects and their codes. Hereby some objects may change the codes, after decoding giving rise to modified objects. This process of *global* feedback (in principle on the *whole* domain via the codes) is the essence of reflection.

It should be emphasized that just the coding of elements of a domain is not sufficient for reflection. A music score may code for a symphony, but the two are on different levels: playing a symphony usually does not alter the written music[1].

[1] However, in aleatory music—the deliberate inclusion of chance elements as part of a composition—the performance depends on dice that the players throw. In most cases, the score (the grand plan of the composition) will not alter. But music in which it really does alter is a slight extension of this idea.

Examples of reflection

Having given this definition, four examples of reflection will be presented.

1. **Proteins.** The first example has as domain the collection of proteins. A typical protein is shown in the following picture. Its three dimensional structure can be perceived by looking at the picture with crossed eyes such that the left and right images overlap.

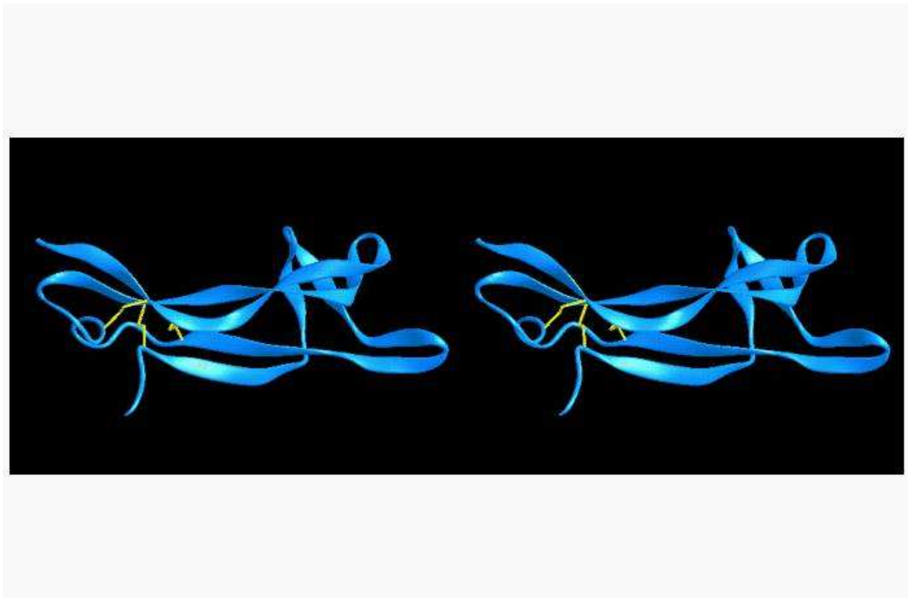


Figure 1: A schematic display of the protein NGF_Homo_Sapiens, a nerve growth factor. Courtesy of the Swiss Institute of Bioinformatics, Peitsch et al. [1995]. `ftp://ftp.expasy.org/databases/swiss-3dimage/IMAGES/JPEG/S3D00467.jpg`

Each protein is essentially a linear sequence of elements of a set of 20 amino acids. Because some of these amino acids attract one another, the protein assumes a three dimensional shape that provides its specific chemical meaning. The sequence of amino-acids for the NGF protein is shown in Fig.2.

Protein: 241 amino acids; molecular weight 26987 Da.
`www.ebi.ac.uk/cgi-bin/expasyfetch?X52599`

MSMLFYTLIT	AFLIGIQAEP	HSESNVPAGH	TIPQVHWTKL	QHSLDTALRR	ARSAPAAAIA	60
ARVAGQTRNI	TVDPRLFKKR	RLRSPRVLFS	TQPPREAADT	QDLDFEVGGA	APFNRTHRSK	120
RSSSHPIFHR	GEFSVCDSVS	VWVGDKTTAT	DIKGKEVMVL	GEVNINNSVF	KQYFFETKCR	180
DPNPVDSGCR	GIDSKHWNSY	CTTTHTFVKA	LTMDGKQAAW	RFIRIDTACV	CVLSRKAVRR	240
A						241

Figure 2: Amino acid sequence of NGF Homo Sapiens.

To mention just two possibilities, some proteins may be building blocks for structures within or between cells, while other ones may be enzymes that enable life-sustaining reactions. The code-set of the proteins consists of pieces of DNA, a string of elements from a set of four ‘chemical letters’ (*nucleotides*). Three such letters uniquely determine a specific amino acid and hence a string of amino acids is uniquely determined by a sequence of nucleotides, see Alberts et al. [1993]. A DNA string does not have the meaning that the protein counterparts have, for one thing because it has not the specific three dimensional folding.

The first advantage of coding is that DNA is much easier to store and duplicate than the protein itself. The interaction in this example is caused by a modifying effect of the proteins upon the DNA. This is also a second advantage of the protein coding, providing the possibility of change, to be described later.

ACGT-chain: length 1047 base pairs.
`www.ebi.ac.uk/cgi-bin/expasyfetch?X52599`

```
agagagcgct   gggagccgga   ggggagcgca   gcgagttttg   gccagtggtc   gtgcagtcca     60
aggggctgga   tggcatgctg   gacccaagct   cagctcagcg   tccggaccca   ataacagttt    120
taccaaggga   gcagctttct   atcctggcca   cactgaggtg   catagcgtaa   tgtccatgtt    180
gttctacact   ctgatcacag   cttttctgat   cggcatacag   gcggaaccac   actcagagag    240
caatgtccct   gcaggacaca   ccatccccca   agtccactgg   actaaacttc   agcattccct    300
tgacactgcc   cttcgcagag   cccgcagcgc   cccggcagcg   gcgatagctg   cacgcgtggc    360
ggggcagacc   cgcaacatta   ctgtggaccc   caggctgttt   aaaaagcggc   gactccgttc    420
accccgtgtg   ctgtttagca   cccagcctcc   ccgtgaagct   gcagacactc   aggatctgga    480
cttcgaggtc   ggtggtgctg   ccccccttcaa   caggactcac   aggagcaagc   ggtcatcatc    540
ccatcccatc   ttccacaggg   gcgaattctc   ggtgtgtgac   agtgtcagcg   tgtgggttgg    600
ggataagacc   accgccacag   acatcaaggg   caaggaggtg   atggtgttgg   gagaggtgaa    660
cattaacaac   agtgtattca   aacagtactt   ttttgagacc   aagtgccggg   acccaaatcc    720
cgttgacagc   gggtgccggg   gcattgactc   aaagcactgg   aactcatatt   gtaccacgac    780
tcacaccttt   gtcaaggcgc   tgaccatgga   tggcaagcag   gctgcctggc   ggtttatccg    840
gatagatacg   gcctgtgtgt   gtgtgctcag   caggaaggct   gtgagaagag   cctgacctgc    900
cgacacgctc   cctccccctg   cccccttctac   actctcctgg   gcccctccct   acctcaacct    960
gtaaattatt   ttaaattata   aggactgcat   ggtaatttat   agtttataca   gttttaaaga   1020
atcattattt   attaaatttt   tggaagc                                           1047
```

Figure 3: DNA code of NGF_Homo_Sapiens.

A simple calculation ($1047/3 \neq 241$) shows that not all the letters in the DNA sequence are used. In fact, some proteins (RNA splicing complex) make a selection as to what substring should be used in the decoding toward a new protein.

2. **Natural language.** The domain of the English language is well-known. It consists of strings of elements of the Roman alphabet extended by the numerals and punctuation marks. This domain has a mechanism of coding, called *quoting* in this context, that is so simple that it may seem superfluous. A string in English, for example

Maria

has as code the quote of that string, i.e.

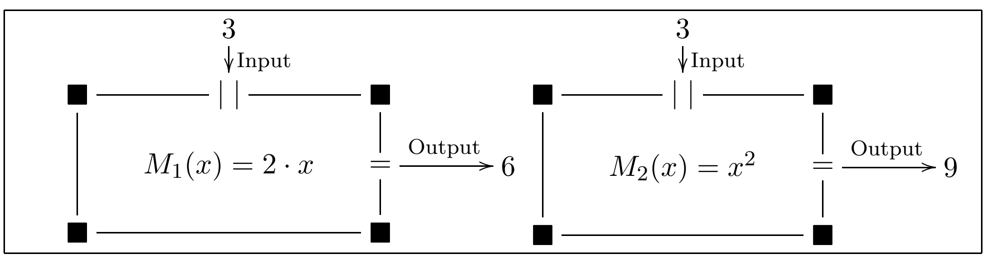
‘Maria’.

In Tarski [1933/1995] it is explained that of the following sentences

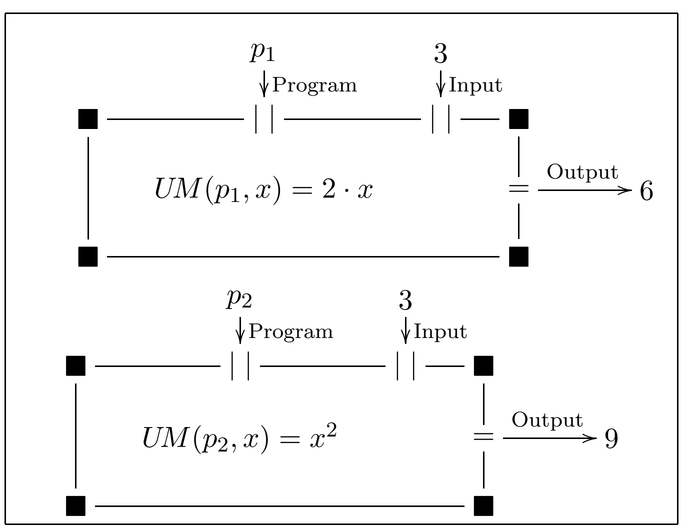
1. Maria is a nice girl
2. Maria consists of five letters
3. 'Maria' is a nice girl
4. 'Maria' consists of five letters

the first and last one are meaningful and possibly valid, whereas the second and third are always incorrect, because a confusion of categories has been made (Maria consist of cells, not of letters; 'Maria' is not a girl, but a proper name). We see the simple mechanism of coding, and its interaction with ordinary language. Again, we see that the codes of the words do not possess the meaning that the words themselves do.

3. **Computable functions.** A third example of reflection comes from computing. The first computers made during WW2 were *ad hoc* machines, each built for a specific use. Since hardware at that time was a huge investment, it was recycled by rewiring the parts after each completed job. Based on ideas of Turing, this procedure was changed. One particular computer was constructed, the *universal machine*, and for each particular computing job one had to provide two inputs: the instructions (the software) and the data that this recipe acts upon. This has become the standard for all subsequent computers.



Tabel 3. Two *ad hoc* machines: M_1 for doubling and M_2 for squaring a number.



Tabel 4. Universal machine UM with programs p_1, p_2 simulating M_1, M_2 respectively.

So p_1 is a code for M_1 and p_2 for M_2. Since we can consider $M_1(p_2)$ and $M_2(p_2)$, there is interaction: agents acting on a code, in the second case even their own code.

The domain in this case consists of implemented computable functions, i.e. machines ready for a specific computing job to be performed. A code for an element of this domain consists of a program that simulates the job on a universal machine. The program of a computable function is not yet active, not yet executable in computer science terminology. Only after decoding does a program come into action. Besides coding, interaction is also present. In the universal machine the program and the data are usually kept strictly separate. But this is not obligatory. One can make the program and the input data overlap so that after running for a while on the universal computer, the initial program is modified.

4. **Mathematical theorems.** A final example in this section is concerned with mathematics. A mathematical theorem is usually about numbers or other abstract entities. Gödel introduced codes for mathematical statements and used as code-set the collection $\{0, 1, 2, 3, \ldots\}$ of natural numbers, that do not have any assertive power. As a consequence, one can formulate in mathematics not only statements about numbers, but via coding also about other such statements. There are even statements that speak about themselves. Again we see that both the coding and interaction aspects of reflection are present.

The power of reflection

The mentioned examples of reflection all have quite powerful consequences.

We know how dramatically life has transformed our planet. Life essentially depends on the DNA coding of proteins and the fact that these proteins can modify DNA. This modification is necessary in order to replicate DNA or to proof-read it preventing fatal errors.

One particular species, *homo sapiens*, possesses language. We know its dramatic effects. Reflection using quoting is an essential element in language acquisition. It enables a child to ask questions like: "Mother, what is the meaning of the word 'curious'?"

Reflection in computing has given us the universal machine. Just one design[2] with a range of possibilities through software. This has had a multi-trillion US$ impact on the present stage of the industrial revolution of which we cannot yet see all the consequences.

The effects of reflection in mathematics are less well-known. In this discipline there are statements of which one can see intuitively that they are true,

[2]That there are several kinds of computers on the market is a minor detail: this has to do with speed and user-friendliness.

but a formal proof is not immediate. Using reflection, however, proofs using intuition can be replaced by formal proofs[3], see Howe [1995] and Barendregt [1997], pp. 21-23. Formal provability is important for the emerging technology of interactive (human-computer) theorem proving and proof verification. Such formal and machine-checked proofs are already changing the way hardware is being constructed[4] and in the future probably also on the way one will develop software. As to the art of mathematics itself, it will bring the technology of Computer Algebra (dealing exactly with equations between symbolic expressions involving elements like $\sqrt{2}$ and π) to the level of arbitrary mathematical statements (involving more complex relations than just equalities between arbitrary mathematical concepts).

The other side of reflection

Anything that is useful and powerful (like fire), can also have a different usage (such as arson). Similarly the power of reflection in the four given examples can be used in different ways.

Reflection in the chemistry of life has produced the species, but also it has as consequence the existence of viruses. Within natural language reflection gives rise to learning a language, but also to paradoxes[5]. The universal computer has as a consequence that there are unsolvable problems, notably the ones we are most interested in[6]. Reflection within mathematics has as a consequence that for almost all interesting consistent axiomatic theories, there are statements that cannot be settled (proved or refuted) within that theory (Gödel's incompleteness result mentioned above).

We see that reflection may be compared to the forbidden fruit: it is powerful, but at the same time, it entails dangers and limitations as well. A proper view of these limitations will make us more modest.

Reflection in spirituality

Insight (*vipassana*) meditation, which stems from classical Buddhism, concerns itself with our consciousness. When impressions come to us through our senses, we obtain a mental representation (e.g. an object in front of us). Now this mental image may be *recollected*: this means that we obtain the awareness of the awareness, also called *mindfulness*. In order to develop the right mindfulness

[3] Often an opposite claim is based on Gödel's incompleteness result. Given a mathematical theory $\mathcal{T}$ containing at least arithmetic that is consistent (expressed as $\mathrm{Con}(\mathcal{T})$), incompleteness states the following. There is a statement G (equivalent to 'G is not provable') within the language of $\mathcal{T}$ that is neither provable nor refutable in $\mathcal{T}$, but nevertheless valid, see Smullyan [1992]. It is easy to show that G is unprovable if $\mathcal{T}$ is consistent, hence by construction G is true. So we have informally proved that G follows from $\mathrm{Con}(\mathcal{T})$. Our (to some unconventional) view on Gödel's theorem is based on the following. By reflection one also can show formally that $\mathrm{Con}(\mathcal{T}) \to G$. Hence it comes not as a surprise, that G is valid on the basis of the assumed consistency. This has nothing to do with the specialness of the human mind, in which we believe but on different grounds, see the section 'Reflection in spirituality'.

[4] Making it much more reliable.

[5] Like 'This sentence is false.'

[6] 'Is this computation going to halt or run forever?' See Yates [1998]

it should be applied to all aspects of consciousness. Parts that usually are not seen as content, but as a coloring of consciousness, become just as important as the object of meditation. If a leg hurts during meditation, one should be mindful of it. Moreover, one learns not only to see the pain, but also the feelings and reactions in connection to that pain. This fine-grained mindfulness will have an 'intuitive analytic' effect: our mind becomes decomposed into its constituents (input, feeling, cognition, conditioning and awareness). Seeing this, we become less subject to various possible vicious circles in our body-mind system that often push us into greed, hatred or compulsive thinking.

Because mindfulness brings the components of consciousness to the open in a disconnected, bare form, they are devoid of their usual meaning. The total information of ordinary mental states can be reconstructed from mindfulness. That is why it works like coding with the contents of our consciousness as domain.

The reflective rôle of mindfulness on our consciousness is quite similar to that of quoting in ordinary language. As proteins can purify part of our DNA, the insight into the constituents of consciousness can purify our mind. Mindfulness makes visible processes within consciousness, hitherto unseen. After that, mindfulness serves as a protection by not letting the components of consciousness exercise their usual meaning. Finally, the presence of mindfulness reorganizes consciousness, giving it a degree of freedom greater than before. Using mindfulness one may act, even if one does not dare; or, one may abstain from action, even if one is urged. Then wisdom will result: morality not based on duty but on virtue. This is the interaction of consciousness and mindfulness. Therefore, by our definition, one can speak of reflection.

This power of reflection via mindfulness also has another side to it. The splitting of our consciousness into components causes a vanishing of the usual view we hold of ourselves and the world. If these phenomena are not accompanied in a proper way, they may become disturbing. But during the intensive meditation retreats the teacher pays proper attention to this. With the right understanding and reorganization, the meditator obtains a new stable balance, as soon as one knows and has incorporated the phenomena.

Mental disorders related to stress can cause similar dissociations. Although the sufferers appear to function normally, to them the world or worse their person does not seem real. This may be viewed as an incomplete and unsystematic use of mindfulness. Perhaps this explains the enigma of why some of the sufferers become 'weller than well', as was observed in Menninger [1963]. These cured patients might very well have obtained the mental purification that is the objective of vipassana meditation.

Pure Consciousness

In Hofstadter [1979] the notion of 'strange loop' is introduced: 'Something that contains a part that becomes a copy of the total when zoomed out. 'Reflection' in this paper is inspired by that notion, but focuses on a special aspect: zooming out in reflection works via the mechanism of coding. The main thesis of Hofstadter is that 'strange loops' are at the basis of self-consciousness.

I partly agree with this thesis and would like to add that mindfulness serves as the necessary zooming mechanism in the strange loop of self-consciousness. On the other hand, the thesis only explains the 'self' aspect, the consciousness part still remains obscure. I disagree with the title of Dennet [1993]: 'Consciousness explained'. No matter how many levels of cognition and feedback we place on top of sensory input in a model of the mind, it *a priori* seems not able to account for experiences. We always could simulated these processes on an old-fashioned computer consisting of relays, or even play it as a social game with cards. It is not that I object to base our consciousness on outer agents like the card players (we depend on nature in a similar way). It is the claimed emergence of consciousness as a side effect of the card game that seems absurd. See Blackmore [2002] for a good survey of theories about consciousness.

Spiritual reflection introduces us to awareness beyond ordinary consciousness, which is without content, but nevertheless conscious. It is called *pure consciousness*. This phenomenon may be explained by comparing our personality to the images on a celluloid film, in which we are playing the title role of our life. Although everything that is familiar to us is depicted on the film, it is in the dark. We need light to see the film as a movie. It may be the case that this pure consciousness is the missing explanatory link between the purely neurophysiological activity of our brain and the conscious mind that we (at least think to) possess. This pure light is believed to transcends the person. The difference between you and me is in the matter (cf. the celluloid of the film). That what gives us awareness is said to come from a common source: the pure consciousness acting as the necessary 'light'.

To understand where this pure consciousness (our inner light) comes from we may have to look better into nature (through a new kind of physics, see e.g. Chalmers [1996] or Stapp [1996]) or better into ourselves (through insight meditation, see e.g. Goldstein [1983]). Probably we will need to do both.

References

Alberts, B. et al. [1994] *The Cell*. Garland.

Barendregt, H. [1997] The impact of the lambda calculus, *Bulletin of Symbolic Logic*, Volume 3, no 2, 181–215.

Blackmore, S. [2002] *Consciousness, an Introduction.* Oxford University Press.

Chalmers, D. [1996] *The Conscious Mind, Towards a Fundamental Theory*, Oxford University Press.

Dennet, D.C. [1993] *Consciousness Explained*. Penguin Books.

Harrison, J. [1995] Metatheory and Reflection in Theorem Proving: A Survey and Critique, available at `http://www.cl.cam.ac.uk/users/jrh/papers/reflect.dvi.gz`

Goldstein, J. [1983] *The Experience of Insight*, Shambhala, Boston, London.

Hofstadter, D. [1979] *Gödel Escher Bach.* Harvester Press.

Howe, D.J. [1992] Reflecting the semantics of reflected proof, in: *Proof Theory*, Aczel *et al.* (eds.), Cambridge University Press, 229–250.

Menninger, K., M. Mayman and P. Pruyser [1963] *The Vital Balance. The Life Process in Mental Health and Illness.* Viking.

Peitsch, M.C., D.R. Stampf, T.N.C. Wells, J.L. Sussman [1995] The Swiss-3DImage

collection and PDB-Browser on the World-Wide Web. *Trends in Biochemical Sciences* 20, 82-84. See also: `http://www.expasy.org`
Smullyan, R. [1992] *Gödel's Incompleteness Theorems*. Oxford University Press.
Stapp, H. [1996] The Hard Problem: A Quantum Approach. *Journal of Consciousness Studies*, 3 (3), 194–210.
Tarski, A. [1933/1995] *Introduction to Logic*. Dover.
Yates, M. [1998] What computers can't do. *+Plus*, issue 5, available at `http://plus.maths.org/issue5/index.html`.

Acknowledgement. Reprinted with permission of Templeton Foundation Press from the forthcoming *Spiritual Information*, edited by Charles L. Harper, Jr., © 2003. I thank the desk editor Pamela Bond for useful linguistic suggestions.

2. Reflection and the living cell

Lecture by Prof.dr. Peter Bloemers. Text From: Mat Ridley, Genome, The Autobiography of a Species in 23 Chapters, ed. Fourth Estate Ltd, London 1999, pp. 6-9.

The human body contains approximately 100 trillion (10^{14}) *cells*, most of which are less than a tenth of a millimeter across. Inside each cell there is a black blob called a *nucleus*. Inside the nucleus are two complete sets of the human *genome* (except in egg cells and sperm cells, which have one copy each, and red blood cells, which have none). One set of the genome came from the mother and one from the father. In principle, each set includes the same 60,000-80,000[7] *genes* on the same twenty-three *chromosomes*. In practice, there are often small and subtle differences between the paternal and maternal versions of each gene, differences that account for blue eyes or brown, for example. When we breed, we pass on one complete set, but only after swapping bits of the paternal and maternal chromosomes in a procedure known as *recombination*.

Imagine that the genome is a book.

- There are twenty-three chapters, called *chromosomes*.
- Each chapter contains several thousand stories, called *genes*.
- Each story is made up of paragraphs, called *exons*, which are interrupted by advertisements called *introns*.
- Each paragraph is made up of words, called *codons*.

Each word is written in letters called BASES. There are one billion words in the book, which makes it longer than 5,000 volumes the size of this one[8], or as long as 800 Bibles. If I read the genome out to you at the rate of one word per second for eight hours a day, it would take me a century. If I wrote out the human genome, one letter per millimeter, my text would be as long as the River Danube. This is a gigantic document, an immense book, a recipe of extravagant length, and it all fits inside the microscopic nucleus of a tiny cell that fits easily upon the head of a pin.

The idea of the genome as a book is not, strictly speaking, even a metaphor. It is literally true. A book is a piece of digital information, written in linear, one-dimensional and one-directional form and defined by a code that transliterates a small alphabet of signs into a large lexicon of meanings through the order of their groupings. So is a genome. The only complication is that all English books read from left to right, whereas some parts of the genome read from left to right, and some from right to left, though never both at the same time.

(Incidentally, you wi1l not find the tired word 'blueprint' in this book, after this paragraph, for three reasons. First, only architects and engineers use blueprints and even they are giving them up in the computer age, whereas we all use books. Second, blueprints are very bad analogies for genes. Blueprints

[7] The present estimate is 30,000 - 40,000. [H.P.J.B.]

[8] This text was taken from a book of 344 pages.

are two-dimensional maps, not one-dimensional digital codes. Third, blueprints are too literal for genetics, because each part of a blueprint makes an equivalent part of the machine or building; each sentence of a recipe book does not make a different mouthful of cake.)

Whereas English books are written in words of variable length using twenty-six letters, genomes are written entirely in three-letter words, using only four letters: A, C, G and T (which stand for adenine, cytosine, guanine and thymine). And instead of being written on flat pages, they are written on long chains of sugar and phosphate called DNA molecules to which the bases are attached as side rungs. Each chromosome is one pair of (very) long DNA molecules.

The genome is a very clever book, because in the right conditions it can both photocopy itself and read itself. The photocopying is known as *replication*, and the reading as *translation*. Replication works because of an ingenious property of the four bases: A likes to pair with T, and G with C. So a single strand of DNA can copy itself by assembling a complementary strand with Ts opposite all the As, As opposite all the Ts, Cs opposite all the Gs and Gs opposite all the Cs. In fact, the usual state of DNA is the famous *double helix* of the original strand and its complementary pair intertwined.

To make a copy of the complementary strand therefore brings back the original text. So the sequence ACGT becomes TGCA in the copy, which transcribes back to ACGT in the copy of the copy. This enables DNA to replicate indefinitely, yet still contain the same information.

Translation is a little more complicated. First the text of a gene is *transcribed* into a copy by the same base-pairing process, but this time the copy is made not of DNA but of RNA, a very slightly different chemical. RNA, too, can carry a linear code and it uses the same letters as DNA except that it uses U, for uracil, in place of T. This RNA copy, called the *messenger* RNA, is then edited by the excision of all introns and the splicing together of all exons (see above).

The messenger is then befriended by a microscopic machine called a *ribosome*, itself made partly of RNA. The ribosome moves along the messenger, translating each three-letter codon in turn into one letter of a different alphabet, an alphabet of twenty different *amino acids*, each brought by a different version of a molecule called *transfer* RNA. Each amino acid is attached to the last to form a chain in the same order as the codons. When the whole message has been translated, the chain of amino acids folds itself up into a distinctive shape that depends on its sequence. It is now known as a *protein*.

Almost everything in the body, from hair to hormones, is either made of proteins or made by them. Every protein is a translated gene. In particular, the body's chemical reactions are catalysed by proteins known as *enzymes*. Even the processing, photocopying error-correction and assembly of DNA and RNA molecules themselves - the replication and translation - are done with the help of proteins. Proteins are also responsible for switching genes on and off, by physically attaching themselves to *promoter* and *enhancer* sequences near the start of a gene's text. Different genes are switched on in different parts of the body.

When genes are replicated, mistakes are sometimes made. A letter (base) is occasionally missed out or the wrong letter inserted. Whole sentences or paragraphs are sometimes duplicated, omitted or reversed. This is known as *mutation*. Many mutations are neither harmful nor beneficial, for instance if they change one codon to another that has the same amino acid 'meaning': there are sixty-four different codons and only twenty amino acids, so many DNA 'words' share the same meaning. Human beings accumulate about one hundred mutations per generation, which may not seem much given that there are more than a billion codons in the human genome, but in the wrong place even a single one can be fatal.

All rules have exceptions (including this one). Not all human genes are found on the twenty-three principal chromosomes; a few live inside little blobs called mitochondria and have probably done so ever since mitochondria were free-living bacteria. Not all genes are made of DNA: some viruses use RNA instead. Not all genes are recipes for proteins. Some genes are transcribed into RNA but not translated into protein; the RNA goes directly to work instead either as part of a ribosome or as a transfer RNA. Not all reactions are catalysed by proteins; a few are catalysed by RNA instead. Not every protein comes from a single gene; some are put together from several recipes. Not all of the sixty-four three-letter codons specify an amino acid: three signify STOP commands instead. And finally, not all DNA spells out genes. Most of it is a jumble of repetitive or random sequences that is rarely or never transcribed: the so-called junk DNA.

That is all you need to know. The tour of the human genome can begin.

	U		C		A		G	
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys
	UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys
	UUA	Leu	UCA	Ser	UAA	stop	UGA	stop
	UUG	Leu	UCG	Ser	UAG	stop	UGG	Trp
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg
	CUC	Leu	CCC	Pro	CAC	His	CGC	Arg
	CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
	CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser
	AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
	AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
	AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly
	GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly
	GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly
	GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly

A	Ala
C	Cys
D	Asp
E	Glu
G	Gly
F	Phe
H	His
I	Ile
K	Lys
L	Leu
M	Met
N	Asn
P	Pro
Q	Gln
R	Arg
S	Ser
T	Thr
V	Val
W	Trp
Y	Tyr

Figure 4: The 'universal' genetic code and the naming convention for aminoacids. Three codons (UAA, UAG and UGA) code for the end of a protein ('stop').

3. Reflection and Language

Describing and generating languages is an important subject of study.

Formal languages are precisely defined via logical rules. These languages are introduced for special purposes. For example the programming languages describe algorithms, i.e. calculation recipes, used to make computers do all kinds of (hopefully) useful things. Other formal languages can express properties of software, the so-called specification languages, or properties part of some mathematical theory. Finally, some formal languages are used in order to express proves of properties, for example the proof that a certain program does what you would like it to do[9]

Natural languages occur in many places on earth and are used by people to communicate. Part of linguistics uses ideas of formal languages in order to approach better and better the natural languages. The hope cherished by some is to be able to come up with a formal description of a large part, if not the total, of the natural languages. We will discuss mainly formal languages, giving only a hint how this study is useful for natural ones.

A *language* is a collection of *words*. A word is a string of symbols taken from a predefined *alphabet*. A typical questions are

- Does word w belong to language L?
- Are the languages L and L' equal?

Words over an alphabet

3.1. Definition. (i) An *alphabet* Σ is a set of symbols. Often this set is finite.

(ii) Given an alphabet Σ, a *word* over Σ is a finite sequence $w = s_1 \ldots s_n$ of elements $s_i \in \Sigma$. It is allowed that $n = 0$ in which case $w = \epsilon$ the *empty* word.

(iii) We explain the notion of an *abstract syntax* by redefining the collection of words over Σ as follows:

$$\boxed{\texttt{word} := \epsilon \mid \texttt{word}\ s},$$

where $s \in \Sigma$.

(iv) Σ^* is the collection of all words over Σ.

3.2. Example. (i) Let $\Sigma_1 = \{0, 1\}$. Then

$$1101001 \in \Sigma_1^*.$$

(ii) Let $\Sigma_2 = \{a, b\}$. Then

$$\begin{array}{rcl} abba & \in & \Sigma_2^*. \\ abracadabra & \notin & \Sigma_2^*. \end{array}$$

(iii) $abracadabra \in \Sigma_3^*$, with $\Sigma_3 = \{a, b, c, d, r\}$.

[9] If software is informally and formally specified, tested and proven correct, i.e. that it satisfies the specification it obtains five stars. The informal specification and tests serve to convince the reader that the requirements are correctly stated. There is very little five star software.

(iv) $\epsilon \in \Sigma^*$ for all Σ.

(v) Let $\Sigma_4 = \{\texttt{A}, \texttt{C}, \texttt{D}, \texttt{E}, \texttt{F}, \texttt{G}, \texttt{H}, \texttt{I}, \texttt{K}, \texttt{L}, \texttt{M}, \texttt{N}, \texttt{P}, \texttt{Q}, \texttt{R}, \texttt{S}, \texttt{T}, \texttt{V}, \texttt{W}, \texttt{Y}\}$. Then the following is a word in Σ_4^*.

```
MSMLFYTLITAFLIGIQAEPHSESNVPAGHTIPQVHWTKLQHSLDTALRRARSAPAAAIA
ARVAGQTRNITVDPRLFKKRRLRSPRVLFSTQPPREAADTQDLDFEVGGAAPFNRTHRSK
RSSSHPIFHRGEFSVCDSVSVWVGDKTTATDIKGKEVMVLGEVNINNSVFKQYFFETKCR
DPNPVDSGCRGIDSKHWNSYCTTTHTFVKALTMDGKQAAWRFIRIDTACVCVLSRKAVRRA
```

We have encountered it in Fig. 2 of section 1.

(vi) Let $\Sigma_5 = \{\texttt{a}, \texttt{c}, \texttt{g}, \texttt{t}\}$. Then an element of Σ_5^* is given in Fig. 3.

(vii) Let $\Sigma_6 = \{\texttt{a}, \texttt{c}, \texttt{g}, \texttt{u}\}$. Then Σ_6 is "isomorphic to" Σ_5.

Operations on words

3.3. Definition. (i) If $a \in \Sigma$ and $w \in \Sigma^*$, then $a.w$ is defined 'by induction on w'.

$$\begin{array}{rcl} a.\epsilon & = & a \\ a.(us) & = & (a.u)s \end{array}$$

(ii) If $w, v \in \Sigma^*$, then their *concatenation*

$$w{+\!\!+}v$$

in Σ^* is defined by induction on v:

$$\begin{array}{rcl} w{+\!\!+}\epsilon & = & w \\ w{+\!\!+}us & = & (w{+\!\!+}u)s. \end{array}$$

We write $wv \equiv w{+\!\!+}v$ as abbreviation.

(iii) Let $w \in \Sigma^*$. Then $w^\vee$ is w "read backward" and is formally defined by

$$\begin{array}{rcl} \epsilon^\vee & = & \epsilon; \\ (wa)^\vee & = & a(w^\vee). \end{array}$$

For example $(abba)^\vee = abba$, and $(abb)^\vee = bba$.

Languages

3.4. Definition. Let Σ be an alphabet. A *language* over Σ is just a subset $L \subseteq \Sigma^*$ (defined in one way or another).

3.5. Example. Let $\Sigma = \{a, b\}$. Define the following languages over Σ.

(i) $L_1 = \{w \mid w \text{ starts with } a \text{ and ends with } b\}$. Then

$$ab, abbab \in L_1, \text{ but } \epsilon, abba, bab \notin L_1.$$

(ii) $L_2 = \{w \mid abba \text{ is part of } w\}$. Then

$$abba, abbab \in L_2, \text{ but } \epsilon, ab, bab \notin L_2.$$

(iii) $L_3 = \{w \mid aa \text{ is not part of } w\}$. Then

$$abba, abbab \in L_3, \text{ but } \epsilon, aa, babaa \notin L_3.$$

(iv) $$\begin{aligned} L_4 &= \{\epsilon, ab, aabb, aaabbb, \ldots, a^n b^n, \ldots\} \\ &= \{a^n b^n \mid n \geq 0\}. \end{aligned}$$

Then $\epsilon, aaaabbbb \in L_4$ but $aabbb, bbaa \notin L_4$.

(v) $L_5 = \{w \mid w \text{ is a palindrome, i.e. } w = w^\vee\}$. For example $abba \in L_5$, but $abb \notin L_5$.

Operations on languages

3.6. Definition. Let L, L_1, L_2 be languages over Σ. We define

$$\begin{aligned} L_1 L_2 &= \{w_1 w_2 \in \Sigma^* \mid w_1 \in L_1 \ \& \ w_2 \in L_2\}. \\ L_1 \cup L_2 &= \{w \in \Sigma^* \mid w \in L_1 \text{ or } w \in L_2\}. \\ L^* &= \{w_1 w_2 \ldots w_n \mid n \geq 0 \ \& \ w_1, \ldots, w_n \in L\}. \\ L^+ &= \{w_1 w_2 \ldots w_n \mid n > 0 \ \& \ w_1, \ldots, w_n \in L\}. \end{aligned}$$

Some concrete languages

3.7. Definition. (Hofstadter) Let $\Sigma_H = \{\mathsf{M}, \mathsf{I}, \mathsf{U}\}$. Define the language L_H over Σ_H by the following grammar, where $x, y \in \Sigma_H^*$.

axiom	MI
rules	$x\mathsf{I} \Rightarrow x\mathsf{IU}$ $\mathsf{M}x \Rightarrow \mathsf{M}xx$ $x\mathsf{III}y \Rightarrow x\mathsf{U}y$ $x\mathsf{UU}y \Rightarrow xy$

This means that by definition $\mathsf{MI} \in L_H$;
if $x\mathsf{I} \in L_H$, then also $x\mathsf{IU} \in L_H$,
if $\mathsf{M}x \in L_H$, then also $\mathsf{M}xx$,
if $x\mathsf{III}y \in L_H$, then also $x\mathsf{U}y$,
if $x\mathsf{UU}y \in L_H$, then also xy.

3.8. Example. (i) $\mathsf{MI}, \mathsf{MII}, \mathsf{MU}, \mathsf{MUU}, \mathsf{IMMIU}, \ldots \in \Sigma_H^*$

(ii) $\mathsf{MI}, \mathsf{MIU}, \mathsf{MIUIU}, \mathsf{MII}, \mathsf{MIIII}, \ldots \in L_H$.

3.9. Exercise. (i) Show that $\mathsf{MUI} \in L_H$

(ii) Show that $\mathsf{IMUI} \notin L_H$

3.10. Problem (Hofstadter's MU puzzle[10]). $\mathsf{MU} \in L_H$?
How would you solve this?

3.11. Definition. (i) $\Sigma_2 = \{\mathtt{p}, \mathtt{q}, -\}$

[10]See Hofstadter [1979].

(ii) The language L_2 over Σ_2 is defined as follows ($x, y, z \in \Sigma_2^*$).

axioma's	$x\mathtt{pq}x$ if x consists only of $-$s
rule	$x\mathtt{p}y\mathtt{q}z \quad \Rightarrow \quad x\mathtt{p}y{-}\mathtt{q}z{-}$

3.12. EXERCISE. Which words belong to L_2? Motivate your answers.

1. $--\mathtt{p}--\mathtt{p}--\mathtt{q}----$
2. $--\mathtt{p}--\mathtt{q}--\mathtt{q}----$
3. $--\mathtt{p}--\mathtt{q}----$
4. $--\mathtt{p}--\mathtt{q}------$

3.13. EXERCISE. Let $\Sigma_3 = \{a, b, c\}$. Define L_3 by

axiom	ab
rule	$xyb \quad \Rightarrow \quad yybx$

The following should be answered by 'yes' or 'no', plus a complete motivation why this is the right answer.
(i) Do we have $ba \in L_3$?
(ii) Do we have $bb \in L_3$?

Even languages over a single letter alphabet are interesting.

3.14. DEFINITION. Laat $\Sigma_4 = \{a\}$.
(i) Define L_{41} as follows.

axiom	a
rule	$w \quad \Rightarrow \quad waa$

Then $L_{41} = \{a^n \mid n \text{ is an odd number}\}$. Here one has $a^0 = \lambda$ and $a^{n+1} = a^n a$. In other words $a^n = \underbrace{a \dots a}_{n \text{ times}}$.
(ii) $L_{42} = \{a^p \mid p \text{ is a prime number}\}$.
(iii) Define L_{43} as follows.

axiom	a
rule	$w \quad \Rightarrow \quad ww$ $wwwa \quad \Rightarrow \quad w$

How can one decide whether Σ_4 is in L_{41}? The question '$w \in L_{42}$?' is more difficult. The difficulty is partly due to the specification of L_{42}. Language L_{43} has an easy grammar, but a difficult decision problem. For example it requires several steps to show that $aaa \in L_{43}$.
CHALLENGE. Do we have $L_{43} = \{a^n \mid n \geq 1\}$? The first person who sends via email the proof or refutation of this to <henk@cs.kun.nl> will obtain 100 €.

Closing time 1.05.2004.

OPEN PROBLEM. (COLLATZ' CONJECTURE) Define L_{44} as follows.

axiom	a
rule	$w \Rightarrow ww$ $wwwaa \Rightarrow wwa$

Prove or refute Collatz' conjecture

$$L_{44} = \{a^n \mid n \geq 1\}.$$

The first correct solution by email before 1.05.2004 earns 150 €. Is there a relation bewteen L_{43} and L_{44}?

3.15. EXERCISE. (i) Show that $\epsilon \notin L_{44}$.
(ii) Show that $aaa \in L_{44}$.

Regular languages

Some of the languages of Example 3.5 have a convenient notation.

3.16. EXAMPLE. Let $\Sigma = \{a, b\}$. Then
(i) L_1 is denoted by $a(a \cup b)^*b$.
(ii) L_2 is denoted by $(a \cup b)^*abba(a \cup b)^*$.

3.17. DEFINITION. Let Σ be an alphabet.
(i) The *regular expressions* over Σ are defined by the following grammar

$$\mathtt{re} := \emptyset \mid \epsilon \mid \mathtt{s} \mid (\mathtt{re.re}) \mid (\mathtt{re} \cup \mathtt{re}) \mid \mathtt{re}^*.$$

here s is an element of Σ. A more consice version of this grammar is said to be an *abstract syntax*:

$$\mathtt{re} := \emptyset \mid \epsilon \mid \mathtt{s} \mid \mathtt{re.re} \mid \mathtt{re} \cup \mathtt{re} \mid \mathtt{re}^*.$$

(ii) Given a regular expression e we define a language $L(e)$ over Σ as follows.

$$\begin{aligned} L(\emptyset) &= \emptyset; \\ L(\epsilon) &= \{\epsilon\}; \\ L(s) &= \{s\}; \\ L(e_1e_2) &= L(e_1)L(e_2); \\ L(e_1 \cup e_2) &= L(e_1) \cup L(e_2); \\ L(e^*) &= L(e)^* \end{aligned}$$

(iii) A language L over Σ is called *regular* if $L = L(e)$ for some regular expression e.

Note that $L^+ = L.L^*$ so that it one may make use of $^+$ in the formation of regular languages. Without a proof we state the following.

3.18. Proposition. (i) *L_3 of Example 3.5 is regular:*

$$L_3 = L((b \cup ab)^*(a \cup \epsilon)).$$

(ii) *$L_4 = \{a^n b^n \mid n \geq 0\}$ is not regular.* ■

There is a totally different definition of the class of regular languages, namely those that are "accepted by a finite automaton". The definition is not complicated, but beyond the scope of these lectures.

Contextfree languages

There is another way to introduce languages. We start with an intuitive example. Consider the following *production system* (grammar) over the alphabet $\Sigma = \{a, b\}$.

$$S \quad \rightarrow \quad \epsilon \mid aSb$$

This is nothing more or less than the grammar

$$\texttt{exp} \quad := \quad \epsilon \mid a \; \texttt{exp} \; b$$

The S stands for *start*. With this auxiliary symbol we start. Then we follow the arrow. There are two possibilities: ϵ and aSb. Since the first does not contain the auxiliary symbol any longer, we say that we have reached a terminal state and therefore the word ϵ has been produced. The second possibility yields aSb, containing again the 'non-terminal' symbol S. Therefore this production has not yet terminated. Continuing we obtain

$$ab = a\epsilon b \text{ and } aaSbb.$$

And then

$$aabb \text{ and } aaaSbbb.$$

Etcetera. Therefore this grammar generates the language

$$L_5 = \{\epsilon, ab, aabb, aaabbb, a^4b^4, \ldots, a^nb^n, \ldots\},$$

also written as

$$L_5 = \{a^nb^n \mid n \geq 0\}.$$

The productions can be depicted as follows.

$$\begin{array}{l} S \rightarrow \epsilon; \\ S \rightarrow aSb \rightarrow ab; \\ S \rightarrow aSb \rightarrow aaSbb \rightarrow aabb; \\ S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb \rightarrow aaabbb. \end{array}$$

L_5 as defined above is called a *contextfree language* and its grammar a *contextfree grammar*

A variant is

$$\boxed{S \quad \rightarrow \quad ab \mid aSb}$$

generating

$$L_5' = \{a^n b^n \mid n > 0\}.$$

3.19. DEFINITION. A *contextfree grammar* consists of the following.

(i) An alphabet Σ.

(ii) A set V of *auxiliary*[11] *symbols*. Among them S, the *start symbol*.

(iii) A finite collection *production rules* of the form

$$X \rightarrow w$$

where X is an auxiliary symbol and w a word consisting of letters from the alphabet and the auxiliary symbols together; otherwise said $w \in (\Sigma \cup V)^*$, where $\cup$ denotes the union of two sets.

(iv) If there are two production rules with the same auxiliary symbol as its left hand side, for example $X \rightarrow w_1$ and $X \rightarrow w_2$, then we notate this in the grammar as

$$X \rightarrow w_1 \mid w_2.$$

For the auxiliary symbols we use upper case letters like S, A, B. For the elements of the alphabet we use lower case letters like a, b, c etcetera.

3.20. EXAMPLE. (i) L_5, L_5' above are contextfree languages. Indeed, the contextfree grammars are given.

(ii) $L_{41} = \{a^n \mid n \text{ odd}\}$ over $\Sigma = \{a\}$ is context-free. Take $V = \{S\}$ and as production-rules

$$\boxed{S \quad \rightarrow \quad aaS \mid a}$$

(iii) Define L_7 over $\Sigma = \{a, b\}$ using $V = \{S, A, B\}$ and the production-rules

$$\boxed{\begin{array}{lcl} S & \rightarrow & AB \\ A & \rightarrow & Aa \mid \epsilon \\ B & \rightarrow & Bb \mid \epsilon \end{array}}$$

Then $L_7 = L(a^* b^*)$, i.e. all string a's followed by a string b's.

Note that the auxiliary symbols can be determined from the production- rules.

[11] These are also called *non-terminal* symbols.

3.21. Exercise. Given is the grammar

$$\boxed{\begin{array}{lcl} S & \to & aSb \mid A \mid \epsilon \\ A & \to & aAbb \mid abb \end{array}}$$

This time $V = \{S, A\}$ and $\Sigma = \{a, b\}$.
Can one produce abb and aab?
What is the collection of words in the generated language?

3.22. Exercise. Produce the language of 3.21 with axioms and rules as in 3.7.

The name 'Context-free grammars' refers to the fact that the left-hand side of the production rules consist of single auxiliary symbols. (For example the rule $Sa \to Sab$ is not allowed.) One never needs to look at the context in which the auxiliary symbol is standing.

An important restriction on the context-free grammars consists of the *right-linear* grammars.

3.23. Definition. A *right-linear grammar* is a context-free grammar such that in every production rule

$$X \to w$$

one has that w is of one of the following forms

(i) $w = \epsilon$

(ii) $w = vY$ with $v \in \Sigma^*$ and Y an auxiliary symbol.

That is to say, in a right-linear grammar auxiliary symbols on the right of a rule only stand at the end and only as a single occurrence.

3.24. Example. (i) In Example 3.20 only L_{41} is a right-linear grammar.

(ii) Sometimes it is possible to transform a context-free grammar in an equivalent right-linear one. The following right-linear grammar (over $\Sigma = \{a, b\}$) also produces L_7 example 3.20 (iii).

$$\boxed{\begin{array}{lcl} S & \to & aS \mid B \\ B & \to & bB \mid \epsilon \end{array}}$$

Without a proof we state the following.

3.25. Theorem. *Let L be a language over Σ. Then*

$$L \text{ is regular } \Leftrightarrow L \text{ has a right-linear grammar.}$$

Hence every regular language is context-free.

3.26. EXERCISE. (i) Consider the context-free grammar over $\{a, b, c\}$

$$\begin{array}{|lcl|} S & \to & A \mid B \\ A & \to & abS \mid \epsilon \\ B & \to & bcS \mid \epsilon \end{array}$$

Which of the following words belong to the corresponding language L_8?

$$abab, bcabbc, abba.$$

(ii) Show that L_8 is regular by giving the right regular expression.
(iii) Show that L_8 has a right-linear grammar.

3.27. EXERCISE. Let $\Sigma = \{a, b\}$.

(i) Show that $L_{41} = \{a^n \mid n \text{ is odd}\}$, see 3.20, is regular. Do this both by providing a regular expression e such that $L_{41} = L(e)$ and by providing a right-linear grammar for L_{41}.

(ii) Describe the regular language $L(a(ab^*)^*)$ by a context-free grammar.

(iii) Let L_9 consists of words of the form

$$aba \dots aba$$

(i.e. a's b's alternating, starting with an a and ending with one; a single a is also allowed). Show in two ways that L_9 is regular.

3.28. EXERCISE. Let $\Sigma = \{a, b, c\}$. Show that

$$L = \{w \in \Sigma^* \mid w \text{ is a palindrome}\}$$

is context-free.

3.29. EXAMPLE. We now give a grammar for a small part of English. Not all sentences produced by this grammar make sense, though!

$$\begin{array}{|rcl|} S = \langle sentence \rangle & \to & \langle noun-phrase \rangle \langle verb-phrase \rangle. \\ \langle sentence \rangle & \to & \langle noun-phrase \rangle \langle verb-phrase \rangle \langle object-phrase \rangle. \\ \langle noun-phrase \rangle & \to & \langle name \rangle \mid \langle article \rangle \langle noun \rangle \\ \langle name \rangle & \to & John \mid Jill \\ \langle noun \rangle & \to & bicycle \mid mango \\ \langle article \rangle & \to & a \mid the \\ \langle verb-phrase \rangle & \to & \langle verb \rangle \mid \langle adverb \rangle \langle verb \rangle \\ \langle verb \rangle & \to & eats \mid rides \\ \langle adverb \rangle & \to & slowly \mid frequently \\ \langle adjective-list \rangle & \to & \langle adjective \rangle \langle adjective-list \rangle \mid \epsilon \\ \langle adjective \rangle & \to & big \mid juicy \mid yellow \\ \langle object-phrase \rangle & \to & \langle adjective-list \rangle \langle name \rangle \\ \langle object-phrase \rangle & \to & \langle article \rangle \langle adjective-list \rangle \langle noun \rangle \end{array}$$

3.30. Exercise. (i) Show how to produce the sentence

Jill frequently eats a big juicy yellow mango.

(ii) Is the generated language context-free?
(iii) Is this grammar right-linear?
(iv) Produce some sentences of your own.
(v) What is the alphabet Σ?
(vi) What are the auxiliary symbols?

Other classes of languages: the Chomsky hierarchy

3.31. Definition. Let Σ be an alphabet.

(i) A *context-sensitive* language over Σ is introduced like a context-free language by production rules of the form

$$uXv \to uwv,$$

where $u, v, w \in \Sigma^*$ and $w \neq \epsilon$. Here X is an auxiliary symbol. The difference between these languages and the context-free ones is that now the production of

$$X \to w$$

only is allowed within the context

$$u \ldots v.$$

(ii) The *enumerable languages* over Σ are also introduced by similar grammars, but now the production rules are of the form

$$uXv \to uwv,$$

where $w = \epsilon$ is allowed.

(iii) A language L over Σ is called *computable* if and only if both L and $\overline{L}$ are enumerable. Here $\overline{L}$ is the complement of L:

$$\overline{L} = \{w \in \Sigma^* \mid w \notin L\}.$$

A typical context-sensitive language is

$$\{a^n b^n c^n \mid n \geq 0\}.$$

A typical computable language is

$$\{a^p \mid p \text{ is a prime number}\}.$$

A typical enumerable language is L_{44}.

The following families of languages are strictly increasing:

1. The regular languages;

2. The context-free languages;

3. The context-sensitive languages;

4. The computable languages;

5. The enumerable languages.

Let us abbreviate these classes of languages as RL,CFL, CSL, CL, EL, respectively. Then we have the proper inclusions can be depicted in the following diagram.

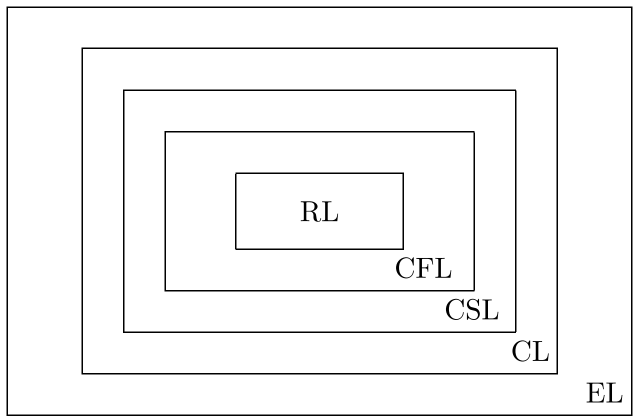


Figure 5: The Chomsky hierarchy

Chomsky discusses the power of these definition mechanisms for the generation of natural languages. He argues that a natural language is too complex to be described by a context-free grammar. Moreover, Chomsky argues that the computable and enumerable languages are too complex to be able to learn by three year old children. The open problem of linguistics is whether a natural language can be described as a context-sensitive language.

Reflection over the classes of languages

There is a uniform way to describe the regular languages. By definition a language L is regular if and only if there is a regular expression e such that $L = L(e)$. This set of regular expressions is itself not a regular language. Reflection over the regular languages pushes us outside this class.

3.32. Definition. (i) A *universal notation system* for the regular languages over an alphabet Σ consists of a language L_u over an alphabet Σ_u and a decoding $d : L_u \to \{L \mid L \text{ is regular}\}$, such that for every regular language L there is at least one code c such that $d(c) = L$.

(ii) Such a universal coding system is called *regular* if the language

$$\{cv \mid v \in d(c)\}$$

over the alphabet $\Sigma \cup \Sigma_u$ is regular.

3.33. Proposition. *(V. Capretta) There is no regular universal notation system for the regular languages.*

We will not give a proof, as it requires some knowledge about the regular languages.

A similar negative result is probably also valid for the context-free and context-sensitive languages. We know that this negative result is the case for the computable languages. But for the enumerable languages there does exists a notation system that itself is enumerable.

3.34. Definition. (i) A *universal notation system* for the enumerable languages over an alphabet Σ consists of a language L_u over an alphabet Σ_u and a decoding $d : L_u \rightarrow \{L \mid L \text{ is enumerable}\}$, such that for every enumerable language L there is at least one code c such that $d(c) = L$.

(ii) Such a universal coding system is called *enumerable* if the language

$$\{cv \mid v \in d(c)\}$$

over the alphabet $\Sigma \cup \Sigma_u$ is enumerable.

3.35. Proposition. *There is an enumerable universal notation system for the enumerable languages.*

Proof. (Sketch) The reason is that the enumerable languages are those languages that are accepted by a Turing machine. Turing machines take as input a string w and start a computation, that can halts or not. Now L is enumerable if and only if there is a Turing machine M_L such that

$$w \in L \;\Leftrightarrow\; M_L(w) \text{ halts.}$$

There is an universal Turing machine M_u, see section 6. This means that for every Turing machine M there is a code c_M such that

$$M(w) = M_u(c_M w).$$

Define $f(c) = \{w \mid M(cw) \text{ halts}\}$. Then given an enumerable language L one has

$$\begin{aligned} w \in L &\Leftrightarrow M_L(w) \text{ halts} \\ &\Leftrightarrow M(c_{M_L} w) \text{ halts} \\ &\Leftrightarrow w \in d(c_{M_L}), \end{aligned}$$

hence

$$L = d(c_L).$$

Therefore $L_u = \{c_M \mid M \text{ a Turing machine}\}$ with decoding d is a universal notation mechanism for the enumerable languages. Moreover, the notation system is itself enumerable:

$$\{cw \mid w \in d(c)\} = \{cw \mid M_u(cw) \text{ halts}\},$$

which is the languages accepted by L_{M_u} and hence enumerable. ■

We end this section by observing that the reflection of the enumerable languages is a different from the one that is present in the natural language like English, see section 1. The first one has as domain the collection of enumerable languages; the second one has as domain the collection of strings within a language.

References

Chomsky, N. [1956]. Three models of the description of language, IRE Transactions on Information Theory, 2, 3, pp. 113-124.

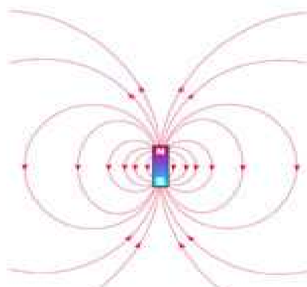
Hofstadter, D. [1979] *Gödel Escher Bach.* Harvester Press.

Sipser, M. [1996] Introduction to the theory of computation, PWS Publishers, Boston.

4. Reflection and Mathematics

The ongoing creation of mathematics, that started 5 or 6 millennia ago and is still continuing at present, may be described as follows. By looking around and abstracting from the nature of objects and the size of shapes *homo sapiens* created the subjects of arithmetic and geometry. Higher mathematics later arose as a tower of theories above these two, in order to solve questions at the basis. It turned out that these more advanced theories often are able to model part of reality and have applications. By virtue of the quantitative, and even more qualitative, expressive force of mathematics, every science needs this discipline. This is the case in order to formulate statements, but also to come to correct conclusions.

Mathematics consists of the study of patterns. In fact, it makes invisible things visible (K. Devlin). Think of magnetic force that is invisible, being visualized by the mathematical laws how moving electric charge and magnetism mutually influence each other.



Maxwell's Equations
$\nabla \cdot \vec{B} = 0$
$\nabla \times \vec{E} + \partial\vec{B}/\partial t = 0$
$\nabla \cdot \vec{D} = \rho$
$\nabla \times \vec{H} - \partial\vec{D}/\partial t = \vec{J}$

Moreover, using this visualization one obtains some kind of control over the phenomenon. These mathematical laws are essential in order to build monitors for televisions and computers, as one needs to know how electron trajectories are being bent in magnetic fields.

The nature of mathematics

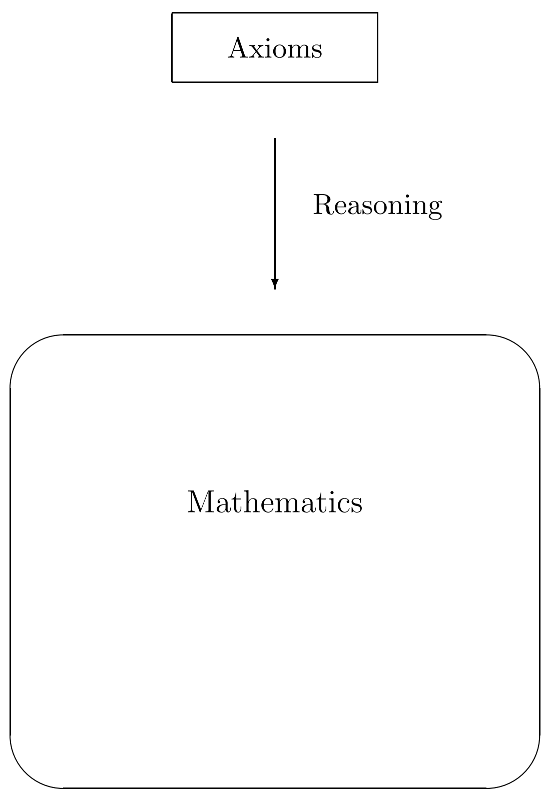
The Greek philosopher Aristotle (384-322 BC) made several fundamental contributions to the foundations of mathematics that are still relevant today. From him we have inherited the idea of the *axiomatic method*[12], not just for mathematics, but for all sciences. A science consists of statements about concepts. Concepts have to be *defined* from simpler concepts. In order to prevent an infinite regression, this process starts from the *primitive concepts*, that do not get a definition. Statements have to be *proved* from statements obtained before. Again one has to start somewhere; this time the primitive statements are called *axioms*. A statement derived from the axioms by pure reason is called a theorem in that axiomatic system. In mathematics one starts from arbitrary primitive notions and axioms, while in science from empirical observations, possibly using (in addition to pure reason) the principle of induction (generalization).

[12] In Aristotle [350 B.C.], Posterior Analytics.

Just a couple of decades after Aristotle and the axiomatic method, Euclid came with his compilation of existing geometry in this form in his *Elements*[13], see Euclid [300 BC] and was very influential as an example of the use of the axiomatic method. Commentators of Euclid stated that the primitive notions are so clear that they did not need definitions; similarly it was said that the axioms are so true that they did not need a proof. This, of course, is somewhat unsatisfactory.

A couple of millennia later Hilbert (1862-1943) changed this view. For him it did not matter what exactly is the essence of the primitive notions such as point and line, as long as they satisfy the axioms: "*The axioms form an implicit definition of the primitive concepts*".

In the light of the axiomatic method the act of creating mathematics can be seen as follows.



The boundary around "Mathematics" is rounded off, in order to indicate that at this moment we have to view it as an open ended collection of theorems. The is because at this stage the notion of "reasoning" is pretty vague.

[13]As was already observed in antiquity the theorems in the Elements were not always proved from the axioms by logic alone. Sometimes his arguments required extra assumptions. The axiomatization of Hilbert [1899] corrected the subtle flaws in Euclid.

Some mathematical phenomena

4.1. EXAMPLE. Consider the sequence

$$1, 4, 9, 16, 25, \ldots$$

What is the next element in this? It is 36 and after that follows 49. We have the sequence of squares. Now we write the difference between two consecutive elements.

$$\begin{array}{ccccccccccccc} 1 & & 4 & & 9 & & 16 & & 25 & & 36 & & \cdots \\ & 3 & & 5 & & 7 & & 9 & & 11 & & \cdots & \end{array}$$

We see that the odd numbers appear. Some questions come to our mind.

- Will this be going on forever?
- Why do we miss the odd number 1 at the beginning?

Let us start with the second question. We want things to become beautiful. We should have started counting with 0 (as do all mathematicians and Montessori children).

$$\begin{array}{ccccccccccccccc} 0 & & 1 & & 4 & & 9 & & 16 & & 25 & & 36 & & \cdots \\ & 1 & & 3 & & 5 & & 7 & & 9 & & 11 & & \cdots & \end{array}$$

Much nicer indeed! The first question is equivalent with saying that the second difference sequence is constantly 2:

$$\begin{array}{ccccccccccccccc} 0 & & 1 & & 4 & & 9 & & 16 & & 25 & & 36 & & \cdots \\ & 1 & & 3 & & 5 & & 7 & & 9 & & 11 & & \cdots & \\ & & 2 & & 2 & & 2 & & 2 & & 2 & & \cdots & & \end{array}$$

More formally we have the following proposition.

PROPOSITION. *Let $a_n = n^2$. Define $b_n = a_{n+1} - a_n$, and $c_n = b_{n+1} - b_n$. Then for all n one has $c_n = 2$.*

PROOF.
$$\begin{aligned} b_n &= a_{n+1} - a_n \\ &= (n+1)^2 - n^2 \\ &= n^2 + 2n + 1 - n^2 \\ &= 2n + 1 \end{aligned}$$
indeed the odd numbers. ■

A more elaborate result is the following.

DEFINITION. Let a_n be a sequence. Define Da as the sequence $b_n = a_{n+1} - a_n$.

Proposition. *Let $a^3_n = n^3$. Then $DDDa^3 = 6$ for all n.*

$$\begin{array}{ccccccccccccc}
0 & & 1 & & 8 & & 27 & & 64 & & 125 & & \cdots \\
& 1 & & 7 & & 19 & & 37 & & 61 & & \cdots & \\
& & 6 & & 12 & & 18 & & 24 & & \cdots & & \\
& & & 6 & & 6 & & 6 & & \cdots & & &
\end{array}$$

Still better is the following. Define $k! = 1.2.3. \ldots .k$

Proposition. *Let $a^k_n = n^k$. Then $D^k a^k = k!$ for all n.*

Some results and open problems

Definition. A natural number p is called *prime (number)* iff (if and only if) $p > 1$ and every divisor of p is either 1 or p itself. For example

$$2, 3, 5, 7, 11, 13$$

are prime numbers, but

$$5, 9, 15 \text{ and } 21$$

are not.

Proposition. (Euclid) There are infinitely many prime numbers. This can be stated without mentioning the concept of infinity:

For every number n there exists a greater prime number p.

In symbols

$$\forall n \exists p\, [p > n \;\&\; p \text{ is prime}].$$

Proof. Given n. Consider $k = n! + 1$, where $n! = 1.2.3. \ldots .n$.
Let p be a prime that divides k.
For this number p we have $p > n$: otherwise $p \le n$;
but then p divides $n!$, so p cannot divide $k = n! + 1$,
contradicting the choice of p. ■

More difficult to prove is the following.

Theorem. (Chebychev) $\forall n \exists p\, [n < p < 2n \;\&\; p \text{ is prime}]$.

Open problems. (i) Is every even number > 2 the sum of two primes? E.g.

one has

$$\begin{array}{rcl} 4 &=& 2+2 \\ 6 &=& 3+3 \\ 8 &=& 3+5 \\ 10 &=& 3+7 &=& 5+5 \\ 12 &=& 5+7 \\ 14 &=& 7+7 &=& 3+11 \\ 16 &=& 3+13 &=& 5+11 \\ 18 &=& 7+11 &=& 5+13 \\ 20 &=& 3+17 &=& 7+13 \\ 22 &=& 3+19 &=& 5+17 &=& 11+11 \end{array}$$

(ii) Are there infinitely many "prime twins"? A prime twin consists of a pair of numbers $p, p+2$ such that p and $p+2$ are both prime. E.g. the first prime twins are

$$(3,5), (5,7), (11,13), (17,19), (29,31), \ldots.$$

Peano Arithmetic

Arithmetic is the theory about the natural numbers. It contains propositions like the ones above. Many such proposition scan be proved from the so-called axioms of Peano. It axiomatizes the properties of the set of natural numbers

$$0, 1, 2, 3, \ldots$$

Following Hilbert, these will be defined via the so-called Peano axioms. The number 0 is taken as primitive. Rather than taking the other numbers as primitive, the successor S, that makes from one number the next one, is taken as primitive.

1. 0 is a natural number.
2. If n is a natural number, also Sn is a natural number.
3. $Sn = Sm \rightarrow n = m$.
4. $Sn \neq 0$, for all natural numbers n.
5. Let P be a property of natural numbers. Suppose that

   $P(0)$
   $P(n) \rightarrow P(S(n))$ for all natural numbers n.

   Then $P(n)$ for all natural numbers n.

Axiom 5 was first formulated by Blaise Pascal and is called the *principle of mathematical induction*. To understand the principle, think of the natural numbers as standing in a long row.

$$0, 1, 2, 3, \ldots$$

If a number has property P, then we mark it. We start with marking the first number, because we know that $P(0)$ holds.

$$0^{\vee}, 1, 2, 3, \ldots$$

But we know also that $P(n) \rightarrow P(n+1)$. This means that if a number is marked, so is its successor. Therefore

$$0^{\vee}, 1^{\vee}, 2, 3, \ldots$$

and

$$0^{\vee}, 1^{\vee}, 2^{\vee}, 3, \ldots$$
$$0^{\vee}, 1^{\vee}, 2^{\vee}, 3^{\vee}, \ldots$$
$$\ldots$$

and we understand that all numbers get marked. It is the principle of dominos standing one next to the other. If the first falls, then it will cause the next to fall, and then the next, etcetera. Eventually all will fall. This is for the natural numbers the way how we get within a finite amount of time some grip over all of them.

Note that there is no notation for the numbers $1, 2, 3, \ldots$. These numbers do not occur in the language of PA (but live in Plato's paradise). Numbers can be represented as linguistic entities called *numerals* as follows.

$$\begin{aligned} \ulcorner 0 \urcorner &= 0; \\ \ulcorner n+1 \urcorner &= S(\ulcorner n \urcorner). \end{aligned}$$

For example $\ulcorner 3 \urcorner = S(S(S(0)))$.

Addition and multiplication

These operations can be specified as follows.

$$\begin{aligned} a + 0 &= a \\ a + S(b) &= S(a+b). \\ a \cdot 0 &= 0 \\ a \cdot S(b) &= (a \cdot b) + a. \end{aligned}$$

Proposition. $\forall a, b, c\ (a+b)+c = a+(b+c)$.
Proof. Given a, b we have to show $\forall c\ P(c)$, where $P(c) := (a+b)+c = a+(b+c)$. We do this by mathematical induction.

Case $c = 0$. Then $P(c)$ states $(a+b)+0 = a+(b+0)$. This holds:

$$\begin{aligned} (a+b)+0 &= a+b, \\ &= a+(b+0). \end{aligned}$$

Induction step. Suppose $P(c)$ holds, i.e. $(a+b)+c = a+(b+c)$. We call this the induction hypothesis. We must show $P(S(c))$ i.e. $(a+b)+S(c) = a+(b+S(c))$. Indeed,

$$\begin{aligned} (a+b)+S(c) &= S((a+b)+c) \\ &= S(a+(b+c)), \qquad \text{by the induction hypothesis,} \\ &= a+S(b+c), \\ &= a+(b+S(c)). \ \blacksquare \end{aligned}$$

Logic

It was again Aristotle who started the quest for logic, i.e. the laws by which scientific reasoning is possible[14]. Aristotle came up with some *syllogisms* (valid reasoning step based on syntactical form) like

$$\frac{\text{No } A \text{ is a } B}{\text{No } B \text{ is a } A}.$$

Aristotle explains this by the particular case

$$\frac{\text{No horse is a man}}{\text{No man is a horse}}.$$

Another of his syllogisms is

$$\frac{\text{No } A \text{ is a } C \quad \text{All } B \text{ are } C}{\text{No } A \text{ is a } B}.$$

Take e.g. men, birds and swans for A, B and C respectively. Aristotle also makes a distinction between such syllogisms and so called *imperfect* syllogisms, that require more steps (nowadays these are called admissible rules). The idea of specifying formal rules sufficient for scientific reasoning was quite daring and remarkable at the time. Nevertheless, from a modern perspective the syllogisms of Aristotle have the following shortcomings. 1. Only unary predicates are used (monadic logic). 2. Only composed statements involving $\rightarrow$ and $\forall$ are considered (so $\&, \vee, \neg$ and $\exists$ are missing). 3. The syllogisms are not sufficient to cover all intuitively correct steps.

In commentators of Aristotle one often finds the following example.

$$\frac{\text{All men are mortal} \quad \text{Socrates is a man}}{\text{Socrates is mortal}}. \tag{1}$$

Such 'syllogisms' are not to be found in Aristotle, but became part of the traditional logical teaching. They have an extra disadvantage, as they seem to imply that they do need to lead from true sentences to true sentences. This is

[14] In Aristotle [350 B.C.], Prior Analysis. One may wonder whether his teacher Plato (427-347 BC) was in favor of this quest (because we already *know* how to reason correctly).

not the case. Syllogism only need to be truth preserving, even if that truth is hypothetical. So a more didactic (and more optimistic) version of (1) is

$$\frac{\text{All sentient beings are happy} \quad \text{Socrates is a sentient being}}{\text{Socrates is happy}}. \tag{2}$$

This example is more didactic, because one of the premises is not true, while the rule is still valid. Aristotle was actually well aware of this hypothetical reasoning.

It was more than 2300 years later that Frege (1848-1925) completed in 1879 the quest for logic and formulated (first-order) predicate logic. He was helped in this by Vieta (1540-1603), who introduced variables to denote arbitrary quantities, Leibniz (1646-1716), who axiomitized equality, Boole (1815-1864), who treated connectives like 'and', 'or', 'implies' and 'not', and Peirce (1839-1914), who studied the *quantifiers* 'for all' and 'there exists'. That Frege's logic was sufficient for the development of mathematics from the axioms was proved in 1922 by Skolem (1887-1963) and independently in 1930 by Gödel (1906-1978). This result is called the completeness theorem for first order logic.

The axiom system for logic has rules that determine the meaning for all the logical connectives: $\neg, \&, \vee, \forall, \exists$, that stand for 'not', 'and', 'or', 'for all' and 'exits' respectively. For example the rules concerning & are

$$\frac{A \quad B}{A\&B} \qquad \frac{A\&B}{A} \quad \frac{A\&B}{B}$$

The first rule has to be read as: if statements A and B are given, then one can deduce $A\&B$. The second rule states that if $A\&B$ is given, then on can deduce A. Other rules are slightly more complex and are beyond these lectures. For an introduction, see van Dalen [1994].

Definition. The language of Peano arithmetic by a context-free abstract grammar. We ned the syntactical categories of *variables, terms* and *formulas.*

$$\begin{array}{rcl} var & := & x \mid var' \\ term & := & var \mid 0 \mid S\ term \mid term + term^{15} \mid term \cdot term \\ form & := & term = term \mid \neg form \mid from \vee form \mid form\&from \mid \\ & & form{\to}form \mid \forall\, var\ form \mid \exists\, var form \end{array}$$

[15] Terms like $x + y$ could be avoided by introducing a predicate $A(x, y, z)$ with the intended meaning

$$A(x, y, z) \leftrightarrow x + y = z.$$

This A then should satisfy

$$\begin{array}{rcl} A_+(x, 0, z) & \leftrightarrow & x = z; \\ A_+(x, y, z) & \leftrightarrow & A_+(x, S(y), S(z)). \end{array}$$

Something similar can be done for multiplication via a predicate $A_\times$. But this is not so convenient. A familiar equation like

$$(x + y) \cdot z = x \cdot z + y \cdot z$$

We have for example

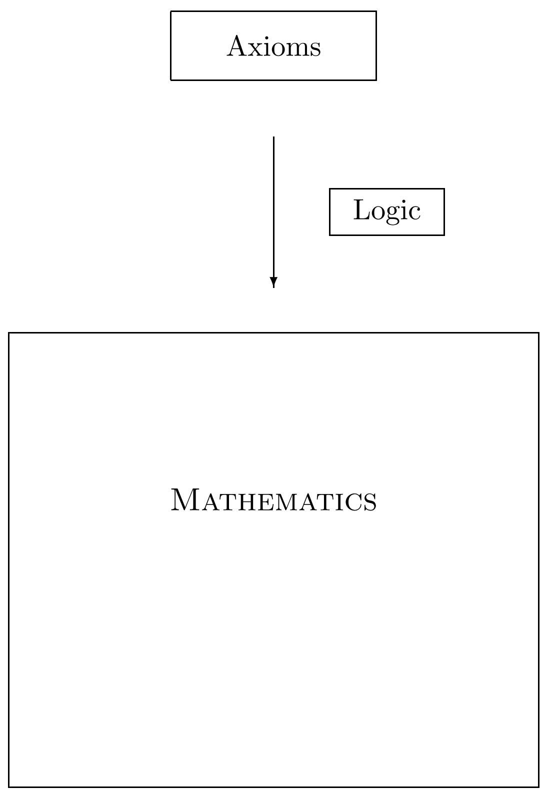
$$\texttt{var} = \{x, x', x'', x''', \ldots\}.$$

One uses $x, y, z, \ldots, x_1, y_1, z_1, \ldots, a, b, c, \ldots$ to denote arbitrary variables.
One uses $s, t, u, \ldots, s_1, t_1, u_1, \ldots$ to denote arbitrary terms.
One uses $P, Q, R, \ldots, A, B, C, \ldots$ to denote arbitrary formulas.

A statement like $A \leftrightarrow B$, that says that A and B are equivalent ("the same") can be written as $(A{\rightarrow}B)$ & $(B{\rightarrow}A)$.

A different view on mathematics: metamathematics

From the moment on that reasoning necessary for doing mathematics was captured by a few logical rules, the collection of mathematical theorems derivable from some axioms became a well-defined total.



Now the boundary around the mathematical theorems provable from the axioms is quite precise. For this reason it is surrounded by a rectangle with sharp edges.

would be difficult to express. Having $+$ and $\cdot$ as primitive operations the defining equations of addition and multiplication need to be taken as axioms. In fact, the functional notation is so convenient, that we assume that for many more functions f we have symbols in the language of PA. This is justified since we can eliminate these symbols via the predicate A_f as explained above for addition.

Now that the collection of mathematical theorems in an axiomatic system is precisely defined, one may ask questions about this totality. For example the following.

- Decidability.
  Is it mechanically decidable (by machine, computer) whether a statement is provable and hence a theorem?

- Completeness.
  Is the collection of provable theorems the same as the true theorems? Or avoiding the notion of 'truth':
  Is for any statement A either provable or refutable[16]?

Leibniz had the optimistic hope to construct a machine that not only would decide all mathematical statements, but even all philosophical ones. The first question he wanted to ask to such a machine was: "Does God exist?" For someone in the early 1700s he had a striking belief in technology. The mathematician Hilbert had as belief that the axiomatic method was so powerful that any statement could be settled, either in the positive or in the negative. Both optimistic views turned out to be falls. Gödel's incompleteness theorem (1931) states that if an axiomatic theory is at least as strong as Peano arithmetic, then if that theory is consistent, there are true statements that cannot be proved. Turing's undecidability theorem (1936) states that even if one does not assume any axioms, it is not mechanically decidable whether a statement is provable or not. For this he introduced the universal computer already encountered in section 3.

Gödel's theorem

The famous theorem of Gödel[1931] will be presented now. It states that if Peano Arithmetic is consistent, then not all true statements can be formally proved. This shows that the axiomatic method has limitations. But it is still very good and in fact the best we have.

The method applies also to other axiomatic systems of arithmetic, provided that they are at least as powerful as that of Peano. The outline of the reasoning will be given first and then we enter some of the details. For a more complete account of the proof see Nagel and Newman [2001] or Gödel's original paper cited below.

1. Arithmetical statements speak about numbers.
2. (Pythagoras) Everything is a number (after coding).
3. Arithmetical statements speak about everything you want (via coding).
4. Arithmetical statements speak about (other) arithmetical statements.
5. Some arithmetical statements speak about themselves (!).

[16] A statement A is *refutable* if its negation $\neg A$ is provable.

6. L: This statement is false.

7. G: This statement is unprovable from the Peano axioms.

8. If PA is consistent (free from contradictions), then G is not provable and hence true!

Conclusion. Interesting axiomatic systems (at least as strong as PA and free from contradictions) are incomplete: there are statements G that are true but not provable.

Now we discuss the details.

1. Arithmetical statements speak about numbers.
This is clear. A notion like "prime" can be expressed formally as

$$P(n) := (n > 1) \,\&\, \forall d[d|n \;\Rightarrow\; (d = 1 \vee d = n)].$$

Here we define the notion "d is a divisor of n", notation $d|n$, by

$$d|n := \exists q\; d \cdot q = n.$$

All formulas in PA speak about numbers.

2. Everything is a number (after coding).
At elementary school you may have invented a trick to code as numbers messages to your friends (and how to decode them). Currently and in past centuries cryptography has become a hot topic, having military implications, see Singh [2000]. A lot of mathematics is involved. But without special needs like secrecy, coding is not very difficult. Pythagoras would have like the fact that a whole symphony may be coded by one (large) number, as happens for example on a music CD[17].

3. Arithmetical statements speak about everything you want (via coding).
One can define now formulas of PA such that e.g. $F(x)$ states that the music coded by x contains a flute passage, or if you prefer a passage of your favorate singer.

These things were novel at the time that Gödel wrote his famous paper. Today in the digital era, these facts are well-known. Now reflection becomes possible.

4. Arithmetical statements speak about (other) arithmetical statements.
Since an arithmetical statements is a member of the formal language `form`, it may be coded as a number $\#A$. Nothing special. Let $\ulcorner A \urcorner$ be the corresponding numeral in PA. That is, if $\#(A) = 3$, then $\ulcorner A \urcorner = S(S(S(0)))$.

[17] If compactness is an issue, then things become harder. Also reliability is an issue: can we reconstruct the music if we loose a bit of the digital information? MP3 is a standard for coding music having both properties of compactness and reliability.

For example one may construct a formula $B(x)$ such that $\texttt{Prov}(\ulcorner A \urcorner)$ states that A is provable in PA.

5. Some arithmetical statements speak about themselves.
This is the essential step in the construction of self-reflection. One can construct a function s_x such that inside PA

$$s_x(\ulcorner A \urcorner, \underline{n}) = \ulcorner A[x := \underline{n}] \urcorner.$$

Here $A[x := t]$ denotes the result of substituting the term t for the variable x in A. Now define $d_x(n) = s_x(n, n)$. Then inside PA one has

$$d_x(\ulcorner A \urcorner) = s_x(\ulcorner A \urcorner, \underline{\#(A)}) = \ulcorner A[x{:=}\underline{\#A}] \urcorner = \ulcorner A[x{:=}\ulcorner A \urcorner] \urcorner.$$

Suppose we want to construct a PA formula "$\texttt{Self}$" that states that it, i.e. "$\texttt{Self}$", is provable. Then we can take

$$\begin{array}{rcl} A(x) & := & \texttt{Prov}(d_x(x)) \\ \texttt{Self} & := & A[x := \ulcorner A \urcorner]. \end{array}$$

Indeed,

$$\begin{array}{rcll} \texttt{Self} & \leftrightarrow & A[x := \ulcorner A \urcorner], & \text{by definition of } \texttt{Self}, \\ & \leftrightarrow & \texttt{Prov}(d_x(\ulcorner A \urcorner)), & \text{by definition of } A(x), \\ & \leftrightarrow & \texttt{Prov}(\ulcorner A[x := \ulcorner A \urcorner] \urcorner), & \text{by the property of } d, \\ & \leftrightarrow & \texttt{Prov}(\ulcorner \texttt{Self} \urcorner), & \text{by definition of } \texttt{Self}. \end{array}$$

6. This statement is false.
Let us call this stament L. It states that L does not hold. Thus L is equivalent with a statement like:

"I am lying now."

If this statement is true, then I am lying indeed, but then (by definition of lying) it should not be true. On the other hand, if it is false, then (by definition of lying again) it is true. This is the famous liar paradox, already known to the Greek[18].

[18] Together with the construction in 5 it shows that there is no "truth predicate" T having as property that

$$T(\ulcorner A \urcorner) \leftrightarrow A$$

holds in PA. Indeed, if such a T would exist, then one arrives at a contradiction by defining

$$L \leftrightarrow \neg T(\ulcorner L \urcorner).$$

This result is due to A. Tarski. It show that decoding is not possible by one truth predicate T. On the other hand, decoding in arithmetic is possible. But for this one needs to classify the set of formulas in to classes Π_n (basically indicating how many quantifier changes $\forall\exists$ occur). For these one has partial truth predicates T_n such that for all $A \in \Pi_n$ one has in PA

$$T_n(\ulcorner A \urcorner) \leftrightarrow A.$$

The upshot is that reflection in the sence of section 1 is possible within PA, not by a single mechanism but by a sequence of mechanisms.

7. By 5. we can construct a formula G such that

$$G \leftrightarrow \neg \texttt{Prov}\ulcorner G \urcorner .$$

8. Suppose that G is provable in PA. Then

$$\texttt{Prov}(\ulcorner G \urcorner)$$

is provable in PA. But by the meaning of G also

$$\neg \texttt{Prov}(\ulcorner G \urcorner)$$

is provable in PA. This makes PA inconsistent. Therefore if PA is consistent, then G cannot be provable. But then G is true, as it stated it own unprovability!

Conlcusion. If PA is consistent then it cannot prove the true statement G. If PA is not consistent, then it can prove everything (including) G, but then PA is not an interesting theory.

The reasoning can be done for every axiomatic theory T at least as strong as PA[19].

THEOREM. Every consistent theory T as least as strong as PA contains a true but unprovable statement G_T.

This result, obtained via reflection, shows the limitation of the axiomatic method. The situation is not terribly bad, however. The axiomatic method is still quite powerful. Some people conclude that the human mind has an essentially non-mechanical basis. The reasoning is: "*We* humans can conclude that G_T is true, but the (mechanical) axiomatic system cannot." See Penrose [1989], [1994]. We disagree, perhaps not with the conclusion,but with this argumentation. The validity of G_T depends on the assumption of the consistency of T. And it is not clear that we can obtain the insight that this is the case. A lucid description of the many theories of mind can be found in Blackmore [2004].

Gödel's theorem states that arithmetic truth and arithmetic provability are not the same. Sharpening Gödel's reasoning Rosser showed that if PA is consistent the Gödel sentence can neither be proved nor refuted in this theory.

Undecidablity

The following theorem of Church makes another metamathematical statement about arithmetic.

THEOREM. (Church) Let T be a consistent extension of Peano Arithmetic. Then there is no computable decision method to determine whether a statement A in the language of arithmetic belongs to T.

[19] For this one assumes that "axiomatic" means that the axioms can be recognized as such. But that is a reasonable assumption.

Similar but for a simpler theory is the following theorem of Turing.

THEOREM. (Turing) There is no computable decision method to determine whether a statement A of Frege's logic is provable in this theory.

This result shows that the ideal of Leibniz to construct a machine that could answer all precisely stated questions cannot be fulfilled.

For details of these theorems see Davis [1965].

For some axiomatic theories there does exist, however, a computable decision method.

THEOREM (Tarski) The theory of elementary geometry has a computable decision method.

The technical proof may be found in Tarski [1951].

Exercises

4.1. Show that $\forall x \forall y\ (x + y) = (y + x)$ by induction 'on y'.

4.2. Define

$$\begin{aligned} x^0 &= 1; \\ x^{n+1} &= x^n.x. \end{aligned}$$

Show by induction on n that for all x

$$(1 + x)^n \geq 1 + nx.$$

4.3. Define an arithmetic predicate Q such that for terms t one has

$$Q[x := \ulcorner t \urcorner] \text{ holds if and only if } t \text{ starts with a } 0.$$

References

Aristotle [350 B.C.]. *Organon*, See also `<classics.mit.edu/Aristotle>`.

Blackmore, S. [2004]. *Consciousness, an Introduction*, Oxford University Press, Oxford.

van Dalen, Dirk [1994]. *Logic and structure*, Universitext, third edition, Springer-Verlag, Berlin.

Davis, M. (ed.) [1965]. *The undecidable. Basic papers on undecidable propositions, unsolvable problems and computable functions*, Raven Press, Hewlett, N.Y.

Euclid [300 BC]. *Elements*, Green Lion Press, Santa Fe, NM. All thirteen books complete in one volume, The Thomas L. Heath translation, Edited by Dana Densmore, 2002.

Gödel, K. [1931]. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, *Monatshefte für Mathematik und Physik* **38**, pp. 173–198. Translated and commented in Gödel [1986]. Another English version based on course notes by Kleene and Rosser is in Gödel [1965].

Gödel, Kurt [1965]. On undecidable propositions of formal mathematical systems, *in:* Martin Davis (ed.), *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, Raven Press, New York, pp. 41–74. From mimeographed notes on lectures given by Gödel in 1934.

Gödel, Kurt [1986]. *Collected works. Vol. I*, The Clarendon Press Oxford University Press, New York. Publications 1929–1936, Edited and with a preface by Solomon Feferman.

Hilbert, David [1899]. Grundlagen der geometrie, *Festschrift zur Feier der Enthullung des Gauss-Weber Denkmals in Göttingen*, Teubner, Leipzig. Later editions published in 1903, 1909, 1911, 1922 and 1930. Translation by Leo Unger of the 10th German Edition: *Foundations of Geometry*; Open Court, LaSalle, 1971.

Nagel, Ernest and James R. Newman [2001]. *Gödel's proof*, revised edition, New York University Press, New York. Edited and with a new foreword by Douglas R. Hofstadter.

Penrose, Roger [1989]. *The emperor's new mind*, The Clarendon Press Oxford University Press, New York. Concerning computers, minds, and the laws of physics, With a foreword by Martin Gardner.

Penrose, Roger [1994]. *Shadows of the mind*, Oxford University Press, Oxford. A search for the missing science of consciousness.

Singh, S. [2000]. *The Code Book*, Anchor Books.

Tarski, A. [1951]. *Decision Method for Elementary Algebra and Geometry*, University of California Press, Berkeley.

5. Reflection and Art

Text by Prof.dr. Anneke Smelik.

To be given during the course.

6. Reflection and Computers

What computers can't do

by Mike Yates[20]

http://plus.maths.org/issue5/turing

Alan Turing is described by Professor P.N. Furbank, overall editor of Turing's Collected Works[1], as "one of the leading figures of twentieth-century science".

Figure 6: Alan Turing.

Sixty years ago his most famous paper was published, introducing the idea of a Universal Computing Machine ten years before the first stored programme digital computer actually ran.

This was only one of a string of varied achievements. It is known now that his work on deciphering the German Enigma code at Bletchley Park during the Second World War made a significant contribution to winning that war, though this remained unknown to his closest friends until after his tragic death from taking potassium cyanide in 1954.

Figure 7: The Enigma machine[21].

[20] Emeritus Professor of the University of Manchester, and an Honorary Professor of the University of Wales at Bangor.

[21] Image source : AGN, University of Hamburg, Copyright 1995, Morton Swimmer.

Turing's wartime work played a significant role in marking out the importance of mechanical computing facilities. Although much of the hack work was done mechanically, an enormous team of human computers was also involved.

Figure 8: Close-up of the coding rotors.

Another feature of his wartime work was its use of probability theory. Some of Turing's work in this area was also highly innovative. It was recognized after the war through the published work of his then assistant (later Professor) Jack Good, without reference to its wartime uses.

Turing's interest in computing continued after the war, when he worked at NPL (National Physical Laboratory) on the development of a stored-programme computer (the ACE or Automatic Computing Engine). In 1948 he moved to Manchester, where the first stored programme digital computer actually ran that year.

Although his connection with that real computer was at best tenuous, he made significant contributions to computing theory, in particular artificial intelligence (the Turing test), computer architecture (the ACE) and software engineering. It is some measure of his contribution that the prestigious Turing Prize in computing science is named after him.

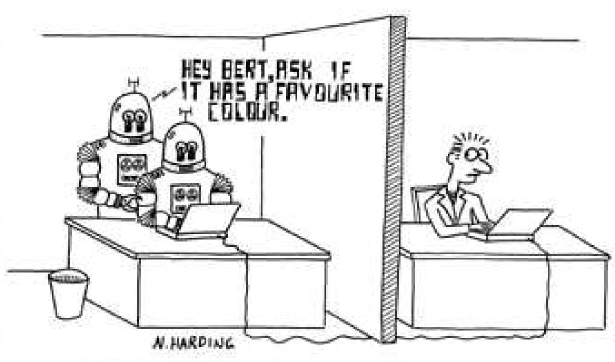


Figure 9: In the Turing test for machine intelligence, an observer has to distinguish between the machine and a human by asking a series of questions through a computer link.

In the Turing test for machine intelligence, an observer has to distinguish between the machine and a human by asking a series of questions through a computer link.

The halting problem

As an example of his thought let's look at a proof that there is no way of telling in general once a computer has embarked on a calculation whether that calculation will terminate in an answer. This problem is known as the "Halting Problem for Turing machines" and was first proved in the 1937 paper[2] in which he introduced his machines.

To lead up to that proof, it is necessary to say a few things about counting and lists or sequences. We say that the elements of a set can be counted if they can be listed in a single sequence.

The set of natural numbers can be listed 0, 1, 2, 3,... and so on ad infinitum - no problem. To list all the integers, positive and negative in a single sequence, you can write 0, 1, -1, 2, -2, 3, -3,... and so on, again no problem.

The fractions take a bit more work. It is usual to do this in 2D, using a table or matrix. Let's just look at the positive ones - it extends to include the negative ones as with the integers.

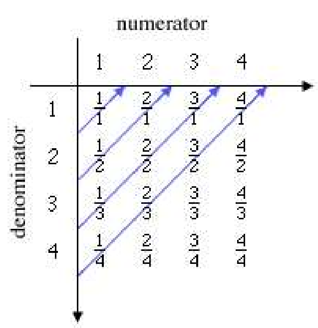


Figure 10: Table of fractions.

The fractions can be counted by tabulating them and then counting them along the diagonals, shown in blue.

There are a lot of repetitions - all the diagonal elements are equal for a start - so this algorithm is a little wasteful. But it does the job. Carry it on for ever and every fraction will be there somewhere in the 2D matrix. To write the matrix out in a single sequence, work up and down the SW to NE diagonals to obtain:

1, 1/2, 2, 1/3, 2/2, 3, 1/4, 2/3, 3/2, 4,...

Next, we come to a very famous theorem, Cantor's Theorem, which says that the real numbers are not countable in this way. The set of real numbers include numbers like π (=3.14159...) which cannot be written as one whole number over another.

Proof of Cantor's Theorem

Let's just show that we cannot count all the binary sequences, in other words, infinite sequences of 0s and 1s.

Suppose we could. We can label each binary sequence B1, B2, B3,... ad infinitum. We will now obtain a contradiction. Let's list the elements of each sequence in a table or matrix as before.

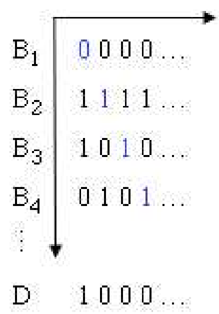


Figure 11: Table of binary sequences. A possible list of binary sequences, the sequence D is constructed by inverting the items on the diagonal, shown in blue.

Now define a binary sequence, D, by choosing a 0 in the first column if B1 has a 1 in that column and 1 if B1 has a 0 in that column. We then choose a 0 in the second column if B2 has a 1 in that column and 1 if it has a 0 and so on. The resulting binary sequence, D, cannot be in the list because if it were it would have to match one of the B sequences, say Bn for some n. But we have just deliberately made sure that the nth column of D differs from Bn. Contradiction.

No matter how we list the binary sequences we can always find a new sequence, D, which is not in the list.

This procedure is called diagonalizing. As you can see, we have given a simple rule for it, so that given a rule for counting out a list of binary numbers then we'd have a rule for computing this diagonal binary number which isn't in the list.

Turing's argument

Finally, let's sketch how Turing's argument (related to an even more famous bit of reasoning by Kurt Gödel in 1931) takes this argument a big stage further.

The proof sketched here is not Turing's original one, but related. Much of Turing's classic paper is taken up with describing his concept of a computing machine and why it is as general as can be. Anything that can be computed according to a finite list of rules, can be computed by one of his machines.

Briefly, a Turing machine can be thought of as a black box, which performs a calculation of some kind on an input number. If the calculation reaches a conclusion, or halts then an output number is returned. Otherwise, the machine theoretically just carries on forever. There are an infinite number of Turing machines, as there are an infinite number of calculations that can be done with a finite list of rules.

One of the consequences of Turing's theory is that there is a Universal Turing machine, in other words one which can simulate all possible Turing machines. This means that we can think of the Turing machines as countable and listed T1, T2,... by a Universal Machine through a sort of alphabetical listing. Turing used this to describe his own version of Gödel's Theorem: that there is no mechanical procedure for telling whether a Turing machine will halt on a given input: the Halting Problem. The unsolvability of the halting problem

Let's represent the result of using the nth Turing machine, Tn on the input i as Tn(i). Suppose that there was a rule or procedure for deciding whether or not Tn(i) halts for all values of n and i.

Machine \ Input	1	2	3	4	5
T_1	5	10	12	?	5
T_2	?	?	4	?	8
T_3	4	?	9	5	?
T_4	?	7	?	4	10
T_5	?	5	?	3	?
D	6	0	10	5	0

Figure 12: A halting rule could be used to make a table of the output Tn(i), using a question mark to represent calculations which never halt. This table is only illustrative, its contents have not been chosen with any particular ordering of Turing machines in mind.

But then by a similar diagonalizing procedure to the one above, we can define a new Turing machine, say D, which will halt for all inputs and return the following output for input i:

0 if Ti(i) does not halt. Ti(i)+1 if Ti(i) does halt.

But this machine D must be one of those machines, in other words it must be Td for some d. However, we just defined it to give a different answer from Td with input d. Contradiction.

The extra sophistication here over the original diagonalizing argument lies in (1) all the listing done is itself computable and (2) any machine Tn may or may not halt in carrying out its computations. None of this enters into Cantor's original diagonal argument. This sort of computable diagonalizing was first used in the pioneering work done by Gödel, Turing and others in the decade before the Second World War, and has remained an important technique. The really hard work lies in formulating the various definitions of computability, but that is another story! What is life?

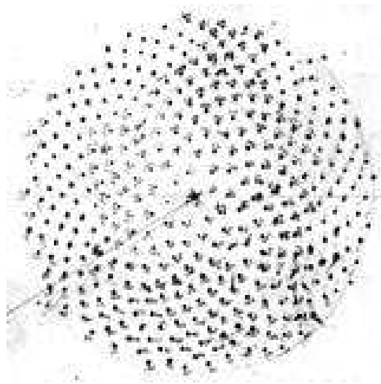


Figure 13: Turing's meticulously hand-drawn sunflower.

In the closing years before his death, Turing was working on something entirely different, something which had been close to his heart since his school days - the origin of biological form - Morphogenesis.

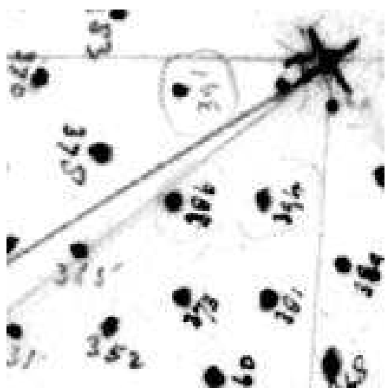


Figure 14: A close-up section.

How could simple cells know how to grow into relatively enormous structured forms? The crucial idea that genetic information could be stored at molecular level had been deduced in Schrdinger's 1943 lecture What is Life?, and Crick and Watson were currently busy in the uncovering of that secret, through the structure of DNA. Given the production of molecules by the genes, Turing was looking for an explanation of how a chemical soup could possibly give rise to a biological pattern.

The first main goal of his theory was an attempt on the classic problem of Phyllotaxis, the arrangement of leaves on a plant. One of the features of this subject which had been known since Kepler's time was the natural occurrence of the Fibonacci series 1, 2, 3, 5, 8, 13, 21,... So it was already established that mathematics had a role to play. (For more about the Fibonacci series see " The life and numbers of Fibonacci" in Issue No 3.)

Turing also proposed that the pattern of markings on animals followed mathematical rules due to chemical signals. This idea had mixed fortunes, though recently biologists' interest has been re-vitalised. Using his theory, researchers in Japan have observed Turing's predicted changes in the patterns on zebra-striped fish.

Further reading

Glance at the web page

`http://www.turing.org.uk/sources/biblio.html`

(The Alan Turing Bibliography, assembled by Andrew Hodges) for further details of the Collected Works.

The definitive work on his life (a compelling read) is:

- Andrew Hodges, Alan Turing: The Enigma, hardback version - Burnett books, 1983, paperback version - Vintage Books, 1992.

  A new angle on Turing can be found in:

- Andrew Hodges, Turing, in the Series The Great Philosophers, Phoenix 1997.

  A guiding force from his school-days was:

- D'Arcy Wentworth Thompson, On growth and Form, Cambridge University Press. 1917 (new edition 1942).

References

[1] Collected Works of A. M. Turing edited by P.N. Furbank, North-Holland, Elsevier Science.

[2] Turing, A. On Computable Numbers, with an application to the Entscheidungsproblem, Proceedings London Mathematical Society (series 2) vol 42, 1936-7, pp. 230-265.

-||-

A very readable history of the creation of the computer is the following.

Davis, M. *Engines of Logic, Mathematicians and the Origin of the Computer*, Norton, 2000.

The classical papers on the theory of computability can be found here.

Davis, M. *The undecidable.* Basic papers on undecidable propositions, unsolvable problems and computable functions, Raven Press, 1965.

7. Reflection and the human mind

Happiness and suffering are both the result of two factors combined: the situation in which one is placed and our consciousness of it. Happiness is not only of personal importance, it is also a necessary factor for ensuring peace in society. Therefore it is important to know the two possible ways for the pursuit of happiness: applied science, which focuses on how situations can be controlled, and spirituality, which focuses on developing the various types of consciousness one can have.

Usually we say that we have a body. Our body may sit in a certain position. Or it is moving around. But is it really ours? When we are ill we prefer not to have this body. When we are healthy it needs food. We are not in full control of our body.

The same may be applied to our mind. Sometimes when we have decided to stay in our room in order to study, the mind gets excited and wants us to go out. If we have the discipline to follow our plan to study our excitement sucks and we can study less. Alternatively, if we give in and go out, then our guilt about not studying may spoil our evening. We see that we are even less in control over our mind.

Says Saint Augustine:

> *If my mind orders my body to do something,*
> *then my body obeys so well,*
> *that one can hardly distinguish between the order and its execution.*
> *If, however, my mind orders my mind to do something,*
> *then it does not listen, even if it is the same mind.*
> *Why this monstrous phenomenon and for what purpose?*

Schopenhauer says something related:

> *We are able to want to do something,*
> *but we cannot* [force ourselves to] *want to want something.*

This section on the mind will describe a way to use reflection in order to gain a higher degree of mental freedom. The method is handed down from classical Buddhism. I do not want to claim that it is the only way to reach this freedom. But the the freedom obtained essentially depends on understanding the functioning of the mind.

Investigating the mind

If we find ourselves in a situation that is not agreeable to us, there are essentially two different ways to attempt to overcome the resulting suffering. On the one hand we can try to change the situation itself. On the other hand we can try to change our consciousness of the situation in such a way that it is no longer experienced as unpleasant. Depending on the circumstances and our possibilities, these attempts will be more or less successful.

It is clear that on the one hand science, technology and democracy have contributed considerably to the way our world can be controlled. On the other hand, we have Buddhism, in particular the 'theravada' school (literary the 'teachings of the elder') that is focussed on the second possibility: eliminating suffering by changing our consciousness. Having completed successfully the course described above implies that one is accomplished in obtaining a different view on the things that happen to us. This state of mind is called *equanimity* and should not be confused with indifference. It may take many years to reach a steady form of equanimity, but the aim is worthwhile. It does not imply that one becomes passive, quite on the contrary.

In order to decrease conditioning of the mind and hence the resulting suffering one needs *(intuitive) insight*. This is not insight through intellectual understanding. The difference is this. When one learns to ride a bike, then one knows that making a turn one needs to bend over. Using mechanics as laid out by the laws of Newton one may calculate the possible angles that are needed to make a smooth bent. But being able to do this, this does not necessarily mean that can ride a bike. The way a child learns to ride is an intuitive one. After a couple of trials he or she intuitively knows how to make a smooth bent.

The insight needed for understanding our mind requires another mental faculty. This is *concentration*. It is interesting, however, that if one wants to be concentrated then one cannot always be in that state. Even if someone would pay you some good money to be concentrated for one hour, you may not succeed.

Nevertheless, concentration can be developed. For this one needs to have some *discipline* and this finally one can make oneself to do this. For example, you all came to class and this requires some discipline that you have been willing to put energy into.

Using discipline, concentration one can develop insight, but not by forcing it. Like in the sports and in music playing one can reach a certain level by cultivating our possibilities. Insight that is obtained will have its positive effect on discipline and concentration, by seeing what are factors that disturb us an finding ways to avoid them. This growing process with its feedback is all part of mental development.

Mindfulness

In the presence of discipline and concentration one is able to develop insight. This is done by applying *mindfulness*. Usually our senses get a lot of input from the world. Mindfulness is the observation of the mental phenomena. When someone calls you a bad name, like 'monster', then you may feel offended and react accordingly. Using mindfulness one first observes the hearing of the word, and then observes the reaction of feeling offended, and then observes the reaction of angriness or perhaps sadness. Mindfulness creates a distance between oneself and the phenomena. One is not sucked away in the stream of consciousness as one usually is.

In order to be able to do this one needs exercising. Sitting meditation is an exercise to increase mindfulness by focusing on breathing (raising and falling of the abdomen). Walking meditation is an exercise in mindfulness on our left and right footsteps. The exercises are done one after the other many times, for say half an hour each.

It is this mindfulness that gives intuitive insights. The *cogito* of Descartes can be seen as an act of mindfulness giving rise to insight. Descartes was wrestling with the question what is the nature of the mind. ‘Do I really exist? There is matter. But is there mind? What can one be sure of?’ Deliberating in similar ways Descartes suddenly observed with mindfulness the act of wrestling thoughts, cognitions. Then he shouted: ‘Cogito ergo sum!’ (‘I deliberate hence I am!’) This ‘Cogito’ should not be seen as the intellectual thinking, but as the mindful observation of the thinking, which act belongs to intuitive insight. By seeing himself from a different level he experienced existence more than he had done hitherto.

The three characteristics

Now we go into the details of the kind of insights one may obtain through mindfulness based on discipline and concentration. If we hear someone say to us ‘monster’, then there is hearing. Usually this hearing is blended with feeling. Mindful observation shows on an intuitive level, that the sound and the emotional contents are different. Rationally this is very easy to understand. To separate the two components on an intuitive level is more difficult. Insight works as a dissecting knife and separates mental phenomena in more primitive forms: the input (through the senses), the feeling (giving value judgments: to obtain, to avoid, to keep), cognizing (distinguishing this from that) and output (reacting through the muscles and inner states).

Our usual view of ourselves as a constant agent that acts in the world will eventually get transformed. We do not have a steady ‘soul’ but consist of a bundle of phenomena. When we realize this on the intuitive level, we see the three properties of life: everything is subject to a constant *change*, is *unsatisfactory* and is *beyond control*. In the Buddhist tradition[22] from which insight meditation comes these are called the three fundamental characteristics: non-permanence (*anicca*), suffering (*dukkha*) and non-self (*anatta*).

Experiencing the three characteristics is a powerful happening. Although the view of ourselves as a bundle of phenomena is theoretically both understandable and even plausible, as actual experience it is something we would like to avoid at all costs. Continuing the investigation of the mind, we see that we usually are covering-up the three characteristics by hiding them with our feelings. As soon as the cover-up wears off, we have to find a new way to hide the characteristics. The way we do this depends on our personality. We can be a *femme fatal* or a wall flower, a Tarzan or an underdog, just to mention a few possibilities, or anything in between or combined. Each personality has a

[22]This is Theravada Buddhism as practised in Myanmar, Thailand and Sri Lanka. Its *vipassana* (insight) meditation is at present widespread in the West. In 2003 there was a program on Dutch TV (*Netwerk* how vipassana was used in prisons. Mindfulness based cognitive therapy, see Segal et al. [2002], is a psychotherapeutic application of the method of mental development through insight.

private way to create feelings to glue the bundle of phenomena together. This gluing together is a symptomatic treatment and makes us slaves of our habits. See Barendregt [1988] for a personal account of the development of mindfulness and the encounter with the three characteristics.

Purification

Mindfulness, that so far has been used as a tool to obtain insight, now can be used as a way to purify the mind. A continuous and concentrated application of mindfulness gives a stable consciousness, one of which one does not become dependent. Here 'concentrated' means that one performs the act of mindfulness as often as possible; 'continuous' means that one does this for prolonged periods.

When one is able to do this, temporary happiness is obtained. But purification still has not taken place. Indeed, one has to *make* mindfulness. This requires energy and concentration. The application of non-interfering mindfulness has indeed a purifying effect. But the effect is not lasting, as mindfulness comes in waves that depend on how we are pushed by our personality. By discipline and concentration the continued concentrated mindfulness should be maintained as long as is possible and in a comfortable way.

This needs to be trained. When this can be done together with calmness, equanimity and bliss one has to surrender. Then the mindfulness may become permanent and automatic. The main bottleneck is to become ready for this. Indeed, letting go is usually coupled with the phenomenon of fear, since humans have the urge of being always in control. Another difficulty to overcome is caused by the tendency to be sidetracked by euphoric experiences. See Barendregt [1996] for a personal account of this process.

Mindfulness and reflection

The essence of mindfulness is attention with detachment. This detachment may be compared to going to a 'meta-level'. Consider the following notions from language semantics. There are objects in reality and names in language.

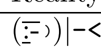
	Reality	Language
1		Maria
2	Maria	'Maria'

Figure 15: Naming

In line 1 of Fig. 15 under 'Reality' we see (the picture of) a (lying) girl; under 'Language' we read her name. In line 2 we see that the name of the girl also occurs in reality; and that the name in quotes is the name of the name. One can state: "Maria is a nice girl" and "The name 'Maria' consists of five letters."

This phenomenon may be compared to the act of mindfulness. The mind is always capturing an object (for example our breathing). This is shown in line 1 of Fig.16, where under 'Object' a process is described that is observed by con-

	Object	Mind
1	breathing	mental event
2	mental event	mindfulness

Figure 16: Mindfulness

sciousness, in this case the observed object is the physical raising and falling of the abdomen. Under 'Mind' the inner awareness of the object is denoted. This awareness of the object in our consciousness can be 'recollected' (this is the literal meaning of *sati*, the ancient Pali[23] word for 'mindfulness'). In that sense the mental event becomes an object for the mind and its awareness is mindfulness, see line 2 of Fig. 16.

Using mindfulness one works, in computer science terms, with the *code* of consciousness rather than with its *executable*. Therefore one is detached without loosing any information. In this way one can react in an equanimous way to phenomena that are 'as if' desire or 'as if' fear. This form of *reflection* gives pure consciousness.

The path of purification using mindfulness has been traversed traditionally in the monastic tradition. A monastery is the 'laboratory for the mind'. This is how a vipassana teacher at a meditation retreat may speak to the meditators.

"Thou dwellers of the great monastery: work with confidence, understanding, effort, concentration and above all mindfulness. At first restrain your senses and stay with their input as much as possible. Make a mental note if your consciousness is pulled elsewhere. This eventually will set you free and your sensory restraint has served its purpose. Be aware of two pitfalls. Too much concentration may give apparent freedom; but you will fall back. Secondly, it is not you who can finish the work. Start with your desire to be unconditioned. At some point you will see that it reaches nowhere. Then let discipline take over and surrender with attention. Do not expect anything and the work will be over soon: bliss of Nibbana[24] becomes permanently accessible. In this life you may use it for the benefit of all living beings."

In the Buddhist tradition it is recognized that the path of purification, can be walked by monks but also by lay people.

Psychotherapy

Partial mindfulness may happen in daily life can reveal a glimpse of the three characteristics. The resulting view of the unsatisfactoriness of our consciousness as loosely bound bundle may create some mental unbalance or hackneyed cover-up as we find in fobias and depression. Mindfulness based stress reduction (MBSR) and mindfulness based cognitive therapy (MBCT) have been developed to treat these. See Kabat-Zinn [1990] and Segal et al. [2002].

[23]The language in which the ancient theory of mind is written.

[24]Pali for Nirvana, the state of pure consciousness.

References

Barendregt, H.P.

[1988] Buddhist Phenomenology, in: *Proceedings of the Conference on Topics and Perspectives of Contemporary Logic and Philosophy of Science*, Cesena, Italy, January 7-10, 1987, (Ed. M. dalla Chiara), Clueb, Bologna, 1988, 37-55. Available at `<www.cs.kun.nl/henk/BP/bp1.html>`.

[1996] Mysticism and Beyond, Buddhist Phenomenology, Part II, *The Eastern Buddhist*, New Series, vol. XXIX (1996), pp. 262 287. Available at `<www.cs.kun.nl/henk/BP/bp2.html>`.

Goldstein, J. & J. Kornfield

[2001] *Seeking the heart of Wisdom: The Path of Insight Meditation*, Shambala Classics.

Kabat-Zinn, J.

[1990] *Full Catastrophy Living: The program of the Stress Reduction Clinic at the University of Massachusetts Medical Center*, New York, Delta, 1990.

Segal, Z.V., J.D. Teasdale and J.M.G. Williams,

[2002] *Mindfulness-Based Cognitive Therapy for Depression: A new approach to preventing relapse.* New York, Guilford Press