Nice examples from Scandinavia

Interdisciplinary teaching combining mathematics with the humanities II: Mathematics and Danish

Workshop 103: Saturday

Contribution to the conference 'Sharing Experiences', Berlin may 2008:
Bjørn Felsager, Haslev Gymnasium & HF, Denmark

This is not intended for publication as a paper, but is just a transcript of what was actually presented in a 'workshop'. It should thus give a fairly detailed impression of the intentions with the presentation even for those, who did not actually attend the workshop.

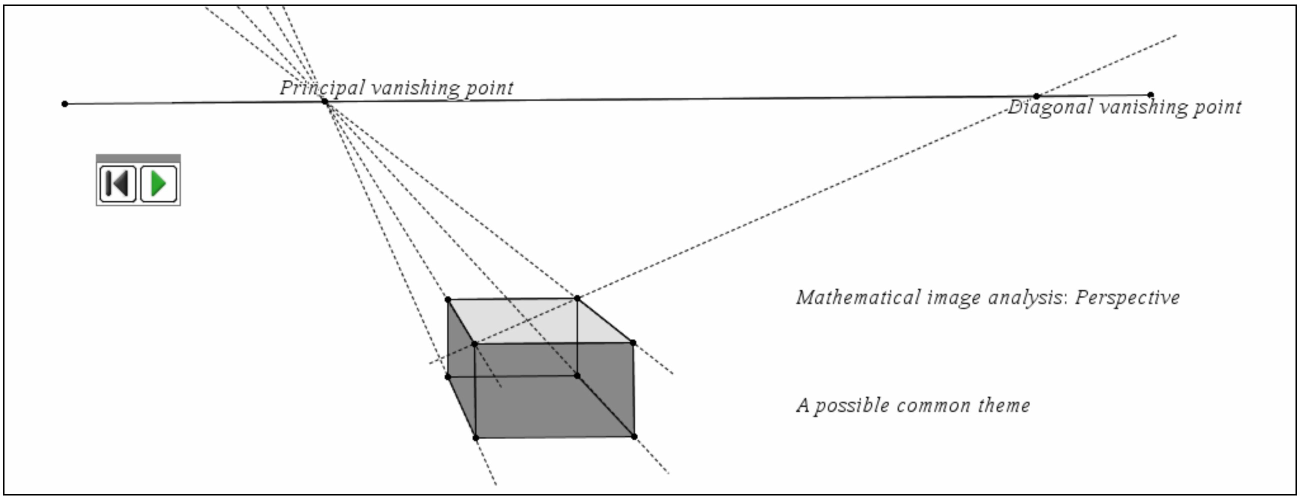
This workshop was the second workshop from Haslev Gymnasium dealing with interdisciplinary teaching combining mathematics with the humanities. The first workshop, presented by Brian Olesen, dealt with mathematics combined with history, and had a particular focus upon cryptology. For details you should consult the materials uploaded with his workshop.

This particular workshop dealt with mathematics combined with Danish. You should substitute your own favorite language for Danish in what follows. It is essentially the primary language, representing the primary subject from the humanities in your own school, that we are dealing with.

The presentation was given as a TI-Nspire presentation, i.e. TI-Inspire was used as an interactive presentation tool. I have uploaded both the original file and the completed file. The **first part** of the presentation consisted of a single screen where I simply outlined the *contents* of the presentation:

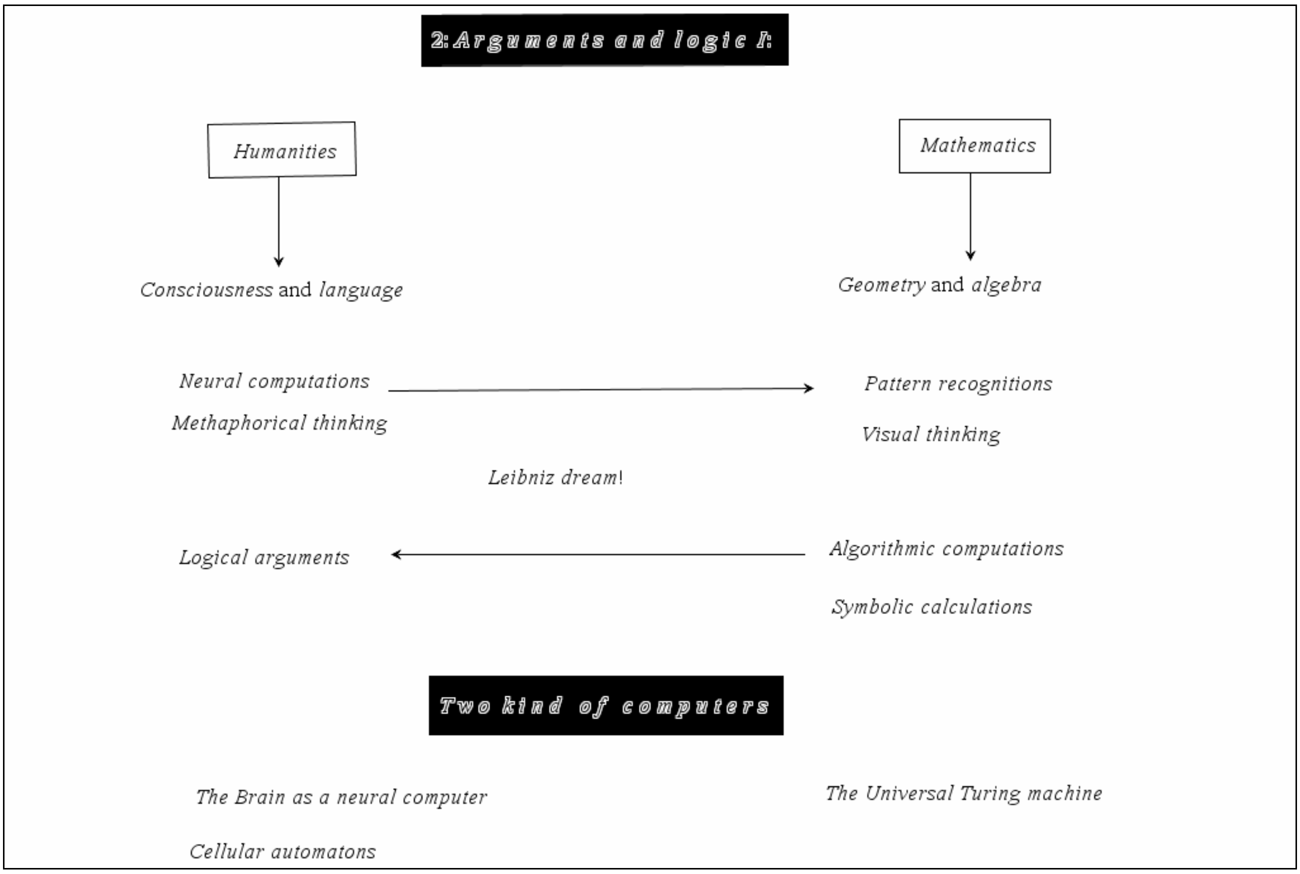
> *Bjørn Felsager – Haslev Gymnasium & HF, Denmark*
>
> **Interdisciplinary teaching combining mathematics with the humanities II: Mathematics and Danish**
>
> (*substitute your own language!*)
>
> 1) Interdisciplinary teaching is a requirement due to the general curriculum reform of the Danish Gymnasium.
>
> Two pilot classes next year – Fields of study: Languages, Social science and psychology.
>
> What's the use of TI–Nspire? Examples:
>
> a) Image analysis
>
> b) Arguments and logic
>
> 2) **Arguments and logic I** – Artificial intelligence: Game of Life
>
> 3) **Arguments and logic II**– Boolean algebras
>
> 4) **Modal logic with TI–Nspire**: Does TI–Nspire actually know when something is true or does it only believe it knows?
>
> A prototypical example!

Here I would only like to point out, that interdisciplinary teaching across different subjects is now mandatory in the Danish Gymnasium, and therefore Mathematics has to develop various themes of mutual interest combining mathematics with science, with social science and with the humanities. In our case we have this year experimented with image analysis from both a humanistic point of view and a mathematical point of view:



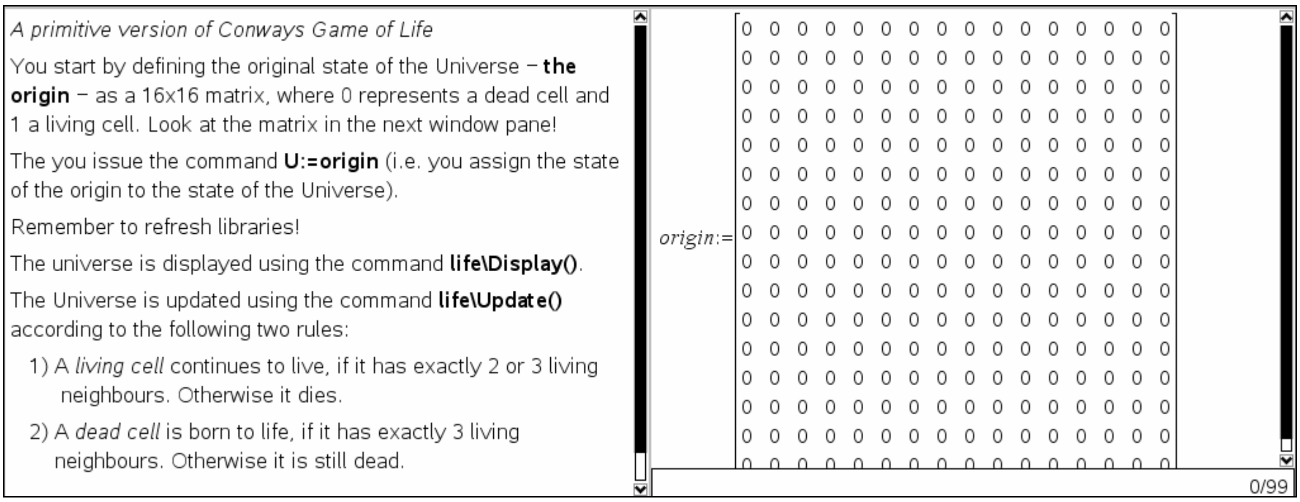
That's the reason for the animation displaying a box rotating in 'true perspective'. I will not enter further into a discussion of image analysis, but just mention that it is a very good subject to combine with the humanities, not only Danish where image analysis is compulsory, but also history and if possible also art (which is not compulsory in the Danish Gymnasium). Also I would like to point out that hopefully a future version of TI-Nspire will incorporate the possibility of importing images, so that you could actually perform the perspective analysis of the image from within TI-Nspire.

So we move on to the **second part** of the presentation, starting with the screen:



This screen is intended to give an overview of some of the themes that we hope to cover in our project: Humanities are very concerned with consciousness and language where as mathematics is very concerned with geometry and algebra. How do we reconcile this? When reflecting upon modes of reasoning it is clear that we have at least two distinct major ways of reasoning: Metaphorical thinking arising from neural computations in the brain and algorithmical thinking that can be associated with symbolic manipulations. In the humanities metaphorical thinking arises very natural, but neural computations are also used extensively in mathematical thinking, where among other things it is associated with pattern recognition and visual thinking. In mathematics algorithmic thinking and logical arguments have been favored, but certainly also the humanities have coped with logical arguments and the art of valid reasoning. It was Leibniz dream, that it should be possible to invent a purely symbolic language, that was able to cope with all kind of mathematical, philosophical, juridical, moral etc. questions and that all questions could be resolved using symbolic calculations, i.e. by systematically transforming the symbolic strings according to well defined rules. This dream seemed to come closer with the advent of the modern computer, the so called universal Turing machine, but today we know that the brain does not function like a Turing computer, thus making it much more difficult to develop artificial intelligence. At the deepest level, the brain is some kind of neural computer performing neural computations. So logical thinking is in some sense not the natural way of thinking, but requires systematic training, where as metaphorical thinking is performed quite automatic.

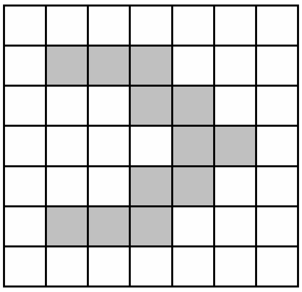
The first thing we want to give the students some impression of is the way a neural computation is performed. The brain is composed of a huge number of neurons, and each neuron is connected to a lot of other neurons, its neighbors. Depending upon the activity in the neighboring neurons, the neuron may fire or stay quiet. We are still looking for good simulations of the activity of such neural networks, but to give the students a first feeling for neural networks, we introduce them to cellular automatons, especially the so called Game of Life:



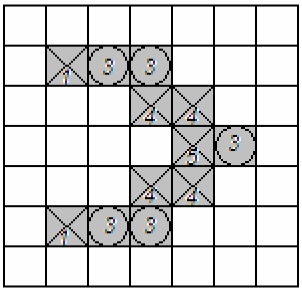
A cellular automaton is a collection of cells arranged in a two-dimensional display known as a matrix. In our case we are using a 16x16 matrix. You can think of each cell as representing a neuron connected to the 8 neighboring neurons. The state of the cell can be either dead or alive. You can think of a living cell as a neuron that fires and a dead cell as a neuron that stays dormant. The activity of the cells is determined by two very simple rules:

1) A living cell continues to live, if it has exactly 2 or 3 living neighbors. Otherwise it dies.
2) A dead cell is born to life, if it has exactly 3 living neighbors. Otherwise it is still dead.

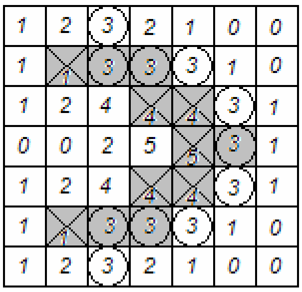
Using these two simple rules you can actually determine the evolution of the cellular automaton by hand. Consider e.g. the following example:



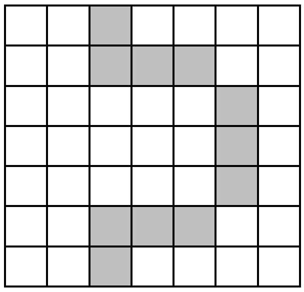
In the first pass we notice which of the living cells that have exactly 2 or 3 living neighbors. They are marked with a ring and they will pass on to the next generation. Other wise they are marked with a cross (in each of the living cell we have noted the number of living neighbors):



In the second pass we notice which of the dead cells that have exactly 3 living neighbors and thus are born to life:



So in the next generation all cells with circles are living, and all cells with crosses die! Thus the next generation looks as follows:



Clearly it will be very cumbersome to run several generations this way. So this is an ideal job for a computer! We will therefore use TI-Nspire to run the simulation[1]. This is explained in the next screen. But first we must edit the matrix **origin** to reflect the above pattern:

$$origin:=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Remember to hit **Enter** so that it is actually stored!

The above pattern is now stored in the matrix **origin**. In the neuron metaphor this represents the input obtained from e.g. the senses.

The matrix origin is then transferred to the matrix **U** (the Universe) that is actually going to evolve dynamically. This is done issuing the command

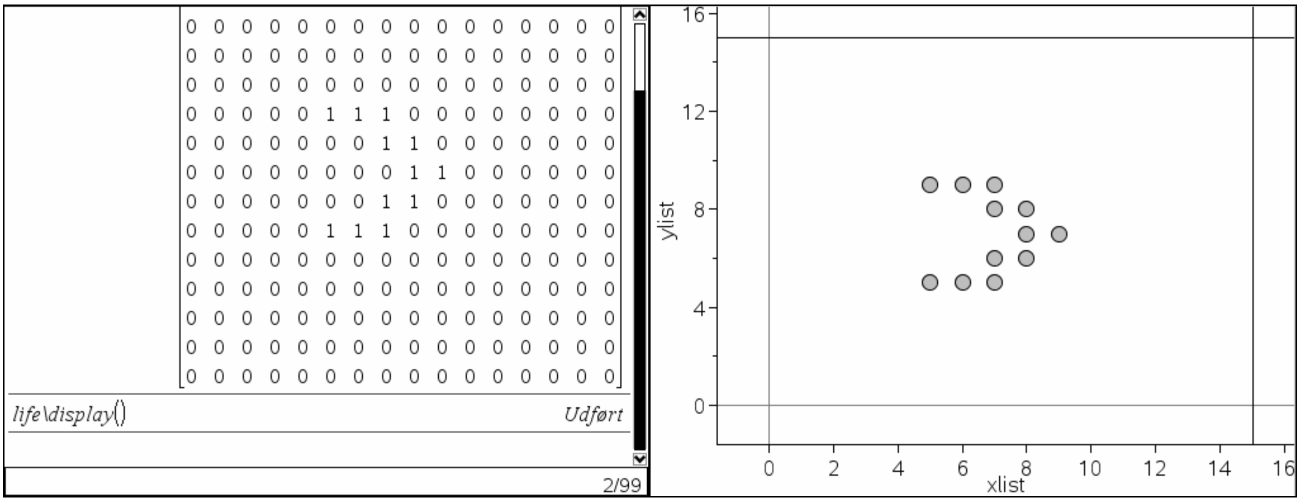
U:=origin.

Next we issue the command

life\Display()

(which is a *private library command* borrowed from the file **life.tns** available from the **myLib** folder on your computer, so you can only know it exists, because you have been told. It is of no use outside this particular simulation):

[1] This requires the installation of a library file in the **MyLib** folder. So if you want to run this simulation your self you need to transfer the file **life.tns** to your own **MyLib** folder!



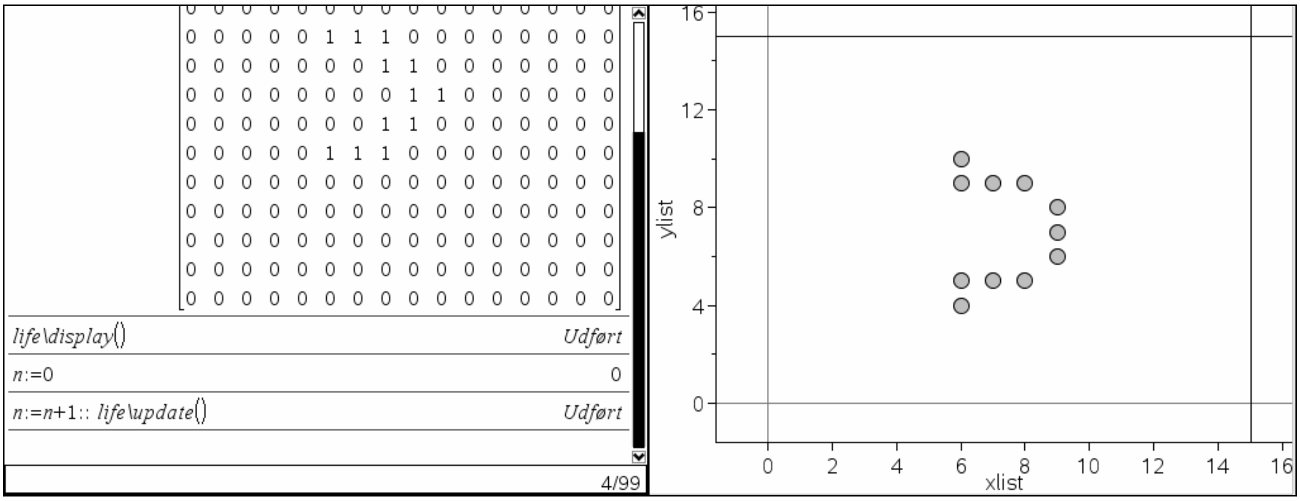
As you can see the initial pattern is displayed nicely in the **Data and Statistics** windows.

To actually run the simulation we will now issue the command

life\Update()

But actually we will combine it with a counter, issued by the command **n:=0**, so we will perform the following double command:

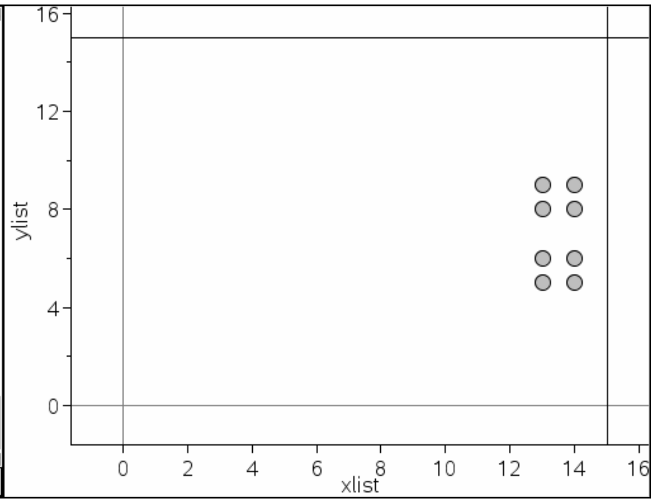
n:=n+1 :: life\Update()



Hopefully you can recognize the pattern we constructed ourselves just a moment ago.

By hitting the **Enter** key repeatedly we now issue the same command again and again and we thus see the pattern unfold before our eyes. For a long time it expands and shrinks but in the end after 138 repetitions it stabilizes in the following static pattern:

```
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n:=n+1:: life\update()        Udført
n                                138
                               99/99
```



In the same way students can play with many other interesting patterns and see them unfold before they settle down in a static or cyclic pattern. Conways **Game of Life** has a very interesting story that you can look up on the internet, using e.g. Wikipedia as a starting point. In the real version the cellular automaton is assumed to have no boundaries, i.e. to fill an unbounded plane. But in the above implementation I have used a cyclic model, i.e. cells at the boundary of the matrix have neighboring cells at the opposite boundary.

Using programming to control the state of the Universe

```
Define LibPriv update()=Prgm
    unew:=u :
    m(x):=mod(x−1,16)+1 :
    For i,1,16 :
     For j,1,16 :
      test:=u[m(i−1),m(j−1)]+u[m(i−1),j]+u[m(i−1),m(j+1)]+u[i,m(j−1)]+u[i,m(j+1)]+u[m(i+1),m(j−1)]+u[m(i+1),j]+
      If u[i,j]=1 Then :
       If test=2 or test=3 Then :
        unew[i,j]:=1 :
       Else :
        unew[i,j]:=0 :
       EndIf :
      Else :
       If test=3 Then :
        unew[i,j]:=1 :
       Else :
        unew[i,j]:=0 :
       EndIf :
      EndIf :
     EndFor :
    EndFor :
    u:=unew :life\display()
    EndPrgm
```

The **key stone** in the program is the enumeration of the number of living neighbours:

```
test:=u[m(i−1),m(j−1)]+u[m(i−1),j]+u[m(i−1),m(j+1)]+u[i,m(j−1)]+u[i,m(j+1)]+u[m(i+1),m(j−1)]+u[m(i+1),j]+u[m(i+1),m(j+1)]
```

The Universe is cyclic!

You shouldn't pay too much attention to the details of this program[2]. But each program has a **key stone** and in this case it is the line:

test:=u[m(i-1),m(j-1)] + u[m(i-1),j] + u[m(i-1),m(j+1)] + u[i,m(j-1)] + u[i,m(j+1)] + u[m(i+1),m(j-1)] + u[m(i+1),j] + u[m(i+1),m(j+1)]

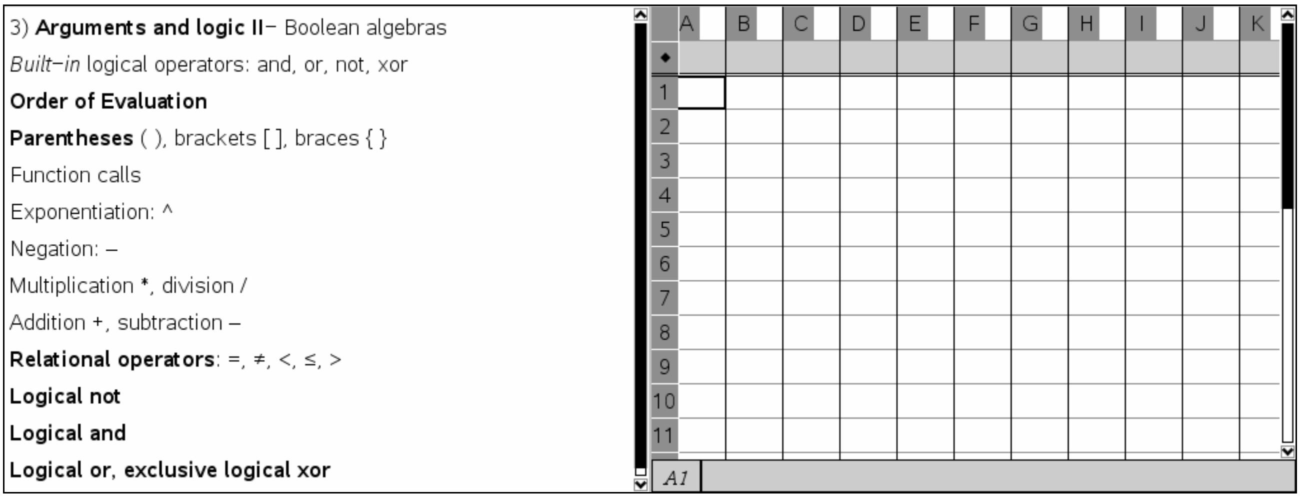
It sums up the contribution from the 8 neighboring cells, where each cell contributes with the number 0 if it is dead and the number 1 if it is alive. It is the current state of the cell together with the number of living neighbors that determine the fate of the cell.

You might also pay attention to the function

m(x):=mod(x-1,16)+1

It has the role of making the cellular automaton cyclic at the boundaries. The counter **m(x)** maps the numbers from 1 to 16 onto themselves, but it transforms the neighbor of 16, i.e. 17, into 0, and it transforms the neighbor of 1, i.e. 0, into the number 16.

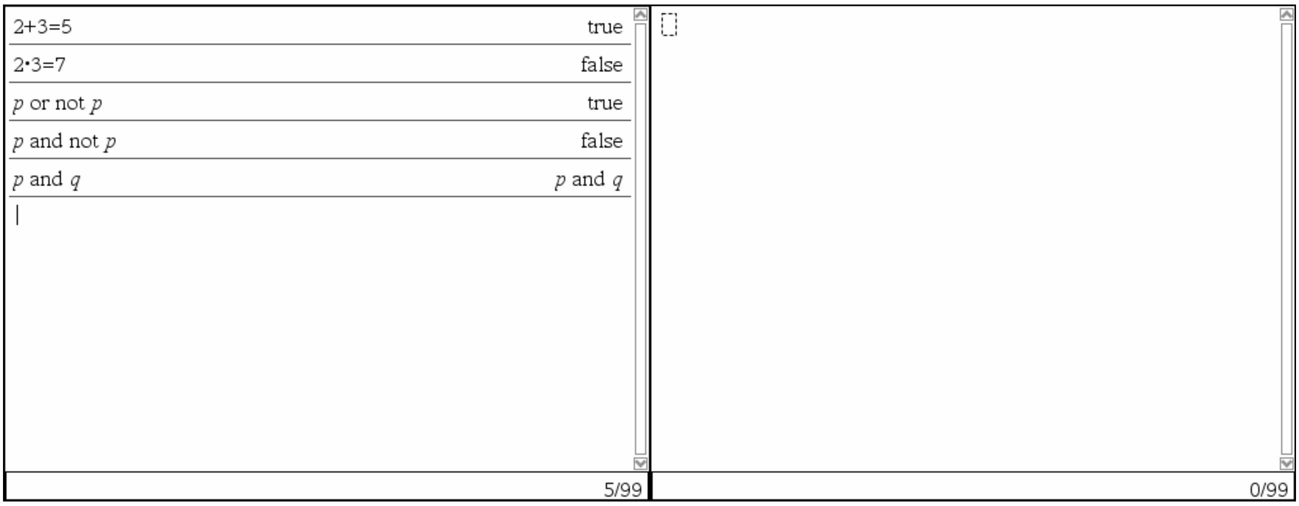
We now turn our attention towards the **third part** of the presentation:



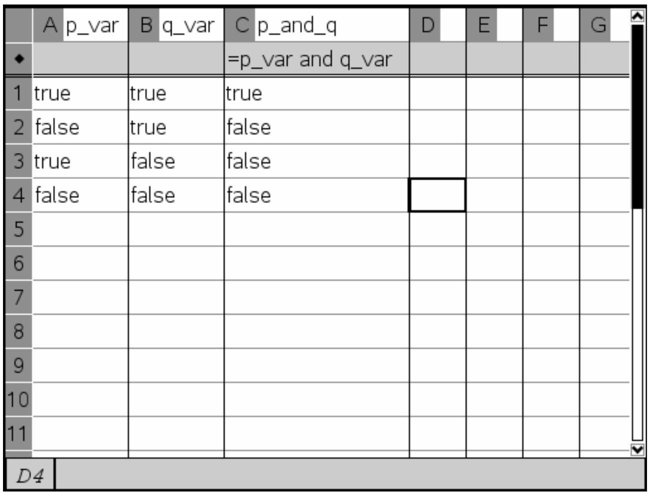
Although logical thinking is not a very natural way of thinking it is still a very important part of rhetoric, i.e. the art of valid reasoning. And part of mathematics contribution is the transformation of the classical Aristotelian logic (based upon syllogisms) into a symbolic language, the propositional calculus (boolean algebras), that fullfills at least a part of Leibniz dream. So any argument is built from primitive proportions ('Berlin is the capital of Germany' or '2+3=5') that are either true or false and can be combined using the logical connectives: not, and, or and xor, that are all supported by TI-Nspire. As long as the primitive propositions are valid mathematical expressions TI-Nspire can be used to determine their truth value:

---

[2] In Denmark programming is not part of the mathematics curriculum!



TI-Nspire also clearly knows the basic rules of Boolean algebra. E.g. it knows that *either a proposition p is true or it is false*, i.e. the statement (*p* or not *p*) is necessarily true independent of the truth value of the proposition *p*. On the other hand *a proposition cannot be both true and false*, i.e. the statement (*p* and not *p*) is necessarily false independent of the truth value of the proposition *p*. Such statements exemplify *tautologies* and *contradictions*. But other statements like (*p* and *q*) has a truth value that really depends upon the truth values of *p* and *q*. You can examine the **truth table** for such a *contingent statement* using the **List and spread sheet application**:



	A p_var	B q_var	C p_and_q	D	E	F	G
◆			=p_var and q_var				
1	true	true	true				
2	false	true	false				
3	true	false	false				
4	false	false	false				
5							
6							
7							
8							
9							
10							
11							

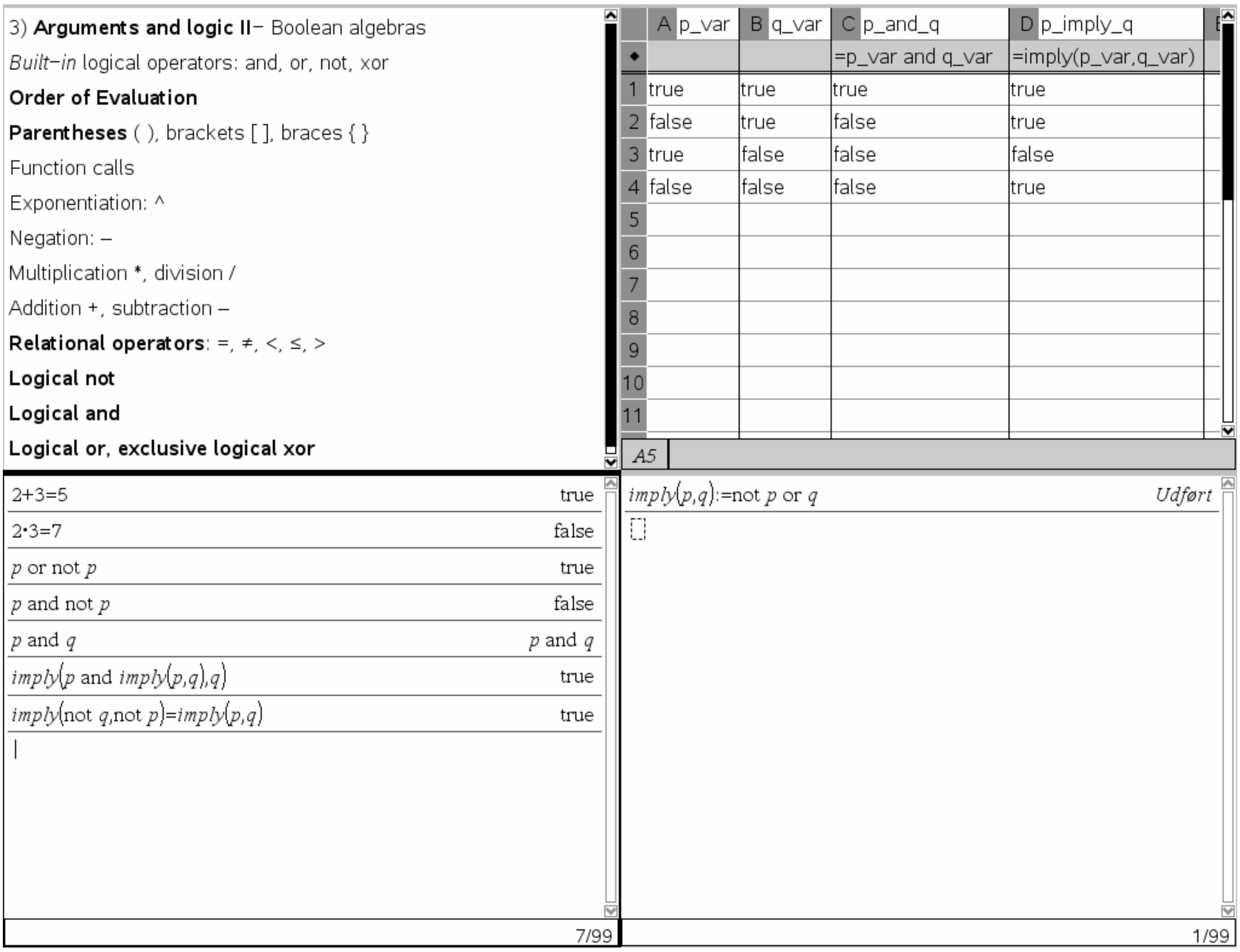
D4

Thus the conjunction of *p* and *q* can only be true if both propositions are true.

TI-Nspire does not support all the logical connectives. Especially it does not include the imply-operator. But then we can just add it as a user defined function:

imply(p,q):=not p or q

As you can see this definition is a little tricky. You can now investigate its truth table as well as some important rules of inference:



The first rule is the rule of *syllogism*: If *p* is true and (*p* implies *q*) is true, then *q* is true.
The other rule is the rule of *contraposition*: To prove that *p* implies *q* you might as well prove the opposite, i.e. that not *q* implies not *p*.

Even though the reduction capacity of the Boolean expressions is impressing, TI-Nspire does not know how to reduce all tautologies. Truth tables are thus indispensable. Also if you want to perform a semantical analysis of a reasoning from a humanistic text, e.g. a part of political speech, it is very useful to break it down into its primitive propositions and the associated logical connectives, and then analyze the truth table associated with the argument.

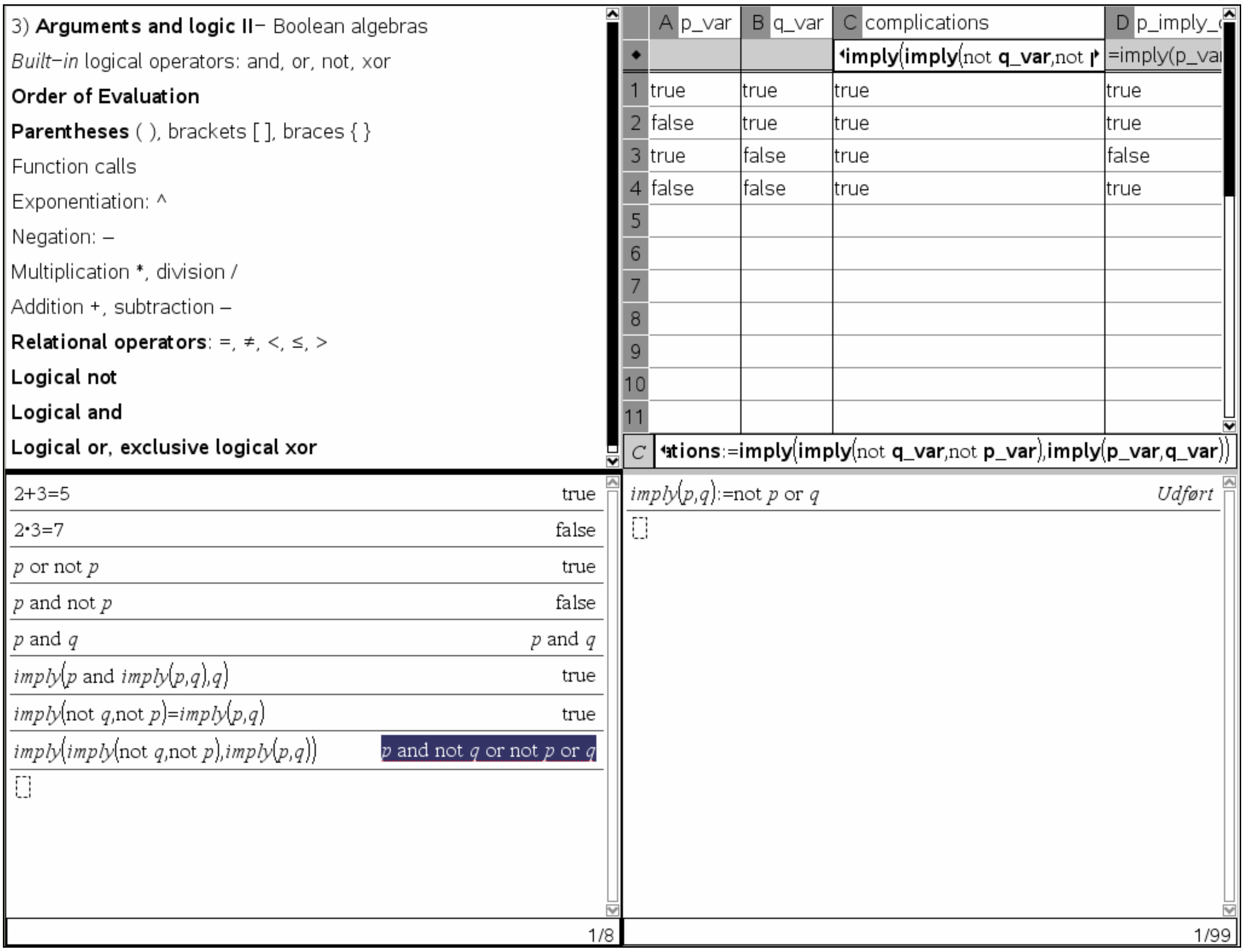
E.g. the following screen shot shows that the logical statement

imply(imply(not *q*, not *p*),imply(*p*,*q*)

is equivalent to the statement

p* and not *q* or not *p* or *q

It can be difficult to recognize this as a tautology, but the truth table reveals this immediately:



Remark: Notice also that it can in fact be difficult to interpret the compound statement

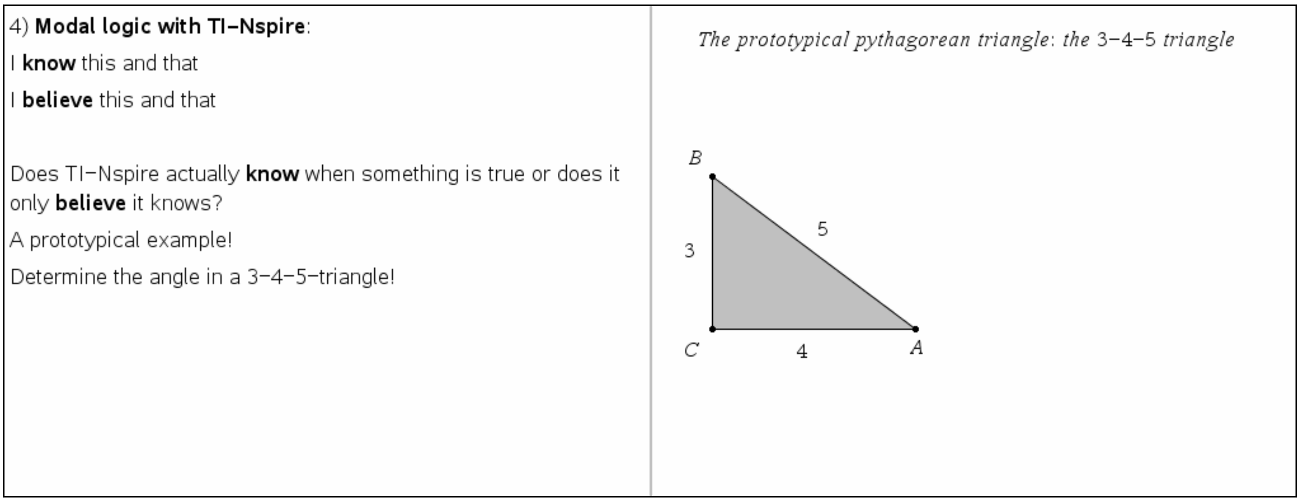
p* and not *q* or not *p* or *q

That is the reason we have enumerated the order of operators in the above screen: First **not**, then **and** and finally **or** (or **xor**). This corresponds to the hidden parentheses:

(*p* and (not *q*)) or not *p* or *q*

But the compound statement (*p* and not *q*) is only true if both *p* is true and *q* is false. If one of these assumptions fails then the remaining part is satisfied. Thus it is a clearly a tautology!

We finally turn to the **fourth part** of the presentation dealing with modal logic and TI-Nspire:

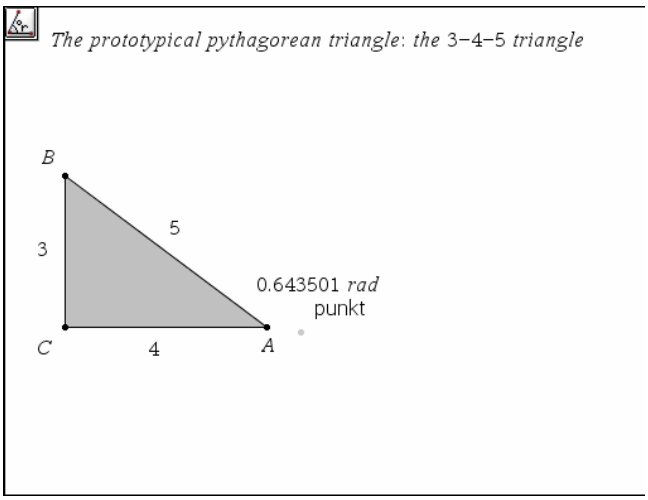


The question is: When TI-Nspire reports that a statement is true (or that it is false), what is the status then of the statement? Does TI-Nspire actually know that it is true (in the sense of classical logic)? Can TI-Nspire actually prove it is true? Or does it just believe the statment is true?

These are deep questions! Modern logic also deals with belief systems: It makes sense to *believe* that something is true ieven if you do not know it is true and you may be a consistent reasoner about your beliefs. But obviously belief is not as strong as actual knowledge. Student tend to think that mathematics and Science are not about believes, but only about safe knowledge – and vice versa for the humanities. But things are not always as straight as we would want them to be!

To make the discussion explicit we will look at a specific example taken from the introductory teaching in mathematics in the Danish Gymnasium. Consider the 3-4-5 triangle *ABC* shown to the right. We will compute angle *A*.

It is very easy to find a numerical answer. We just point out the angle *BAC* and measure it using the built-in angle tool (notice that we work in radians!):



But suppose we are interested in a symbolic answer, rather than a numerical answer. Then different answers may pop up from different students. E.g. a student could decide to compute angle A using the sine-function, i.e.

$$\sin(A) = \frac{\text{opposite leg}}{\text{hypotenuse}} = \frac{3}{5}$$

In that case the student would report the symbolic value as $\sin^{-1}(3/5)$.

But another student may prefer to use the cosine function, i.e.

$$\cos(A) = \frac{\text{adjacent leg}}{\text{hypotenuse}} = \frac{4}{5}$$

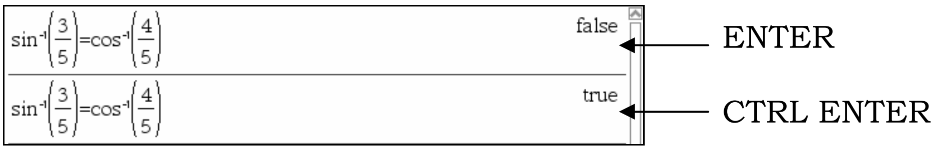
Consequently the student report the symbolic value as $\cos^{-1}(4/5)$.

The question then is: Does TI-Nspire know that these two answers are equivalent, i.e. that the statement $\sin^{-1}(3/5) = \cos^{-1}(4/5)$ is a true statement (like 20+3 = 5)?

Let us see:

So the somewhat surprising conclusion is that TI-Nspire actually believes it to be false! Clearly it cannot know the statement is false, since then it would actually make a grave mistake – but never the less it still believes a true statement to be false!

So what is happening? First let us hit the CTRL ENTER key instead. That forces TI-Nspire to evaluate the expression numerically before it decides whether it is true or false:

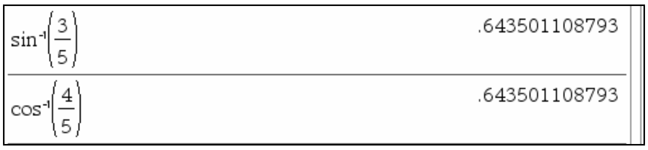


So that is at least a little reassuring: TI-Nspire does at the least know that it is numerically true. But numerical truth is not very convincing. First the slightest round off error may change the numerical truth value[3]. Second a numerical argument does not count as an actual proof (although one may argue from an advanced point of view that numerically coincidence of such expressions involving only integers, does actually force them to be true).

---

[3] E.g. TI-Nspire will report that the statement $\sin^{-1}(5/13) = \cos^{-1}(12/13)$ is numerically false, despite the fatct that it is a true statement related to the 5-12-13 triangle.

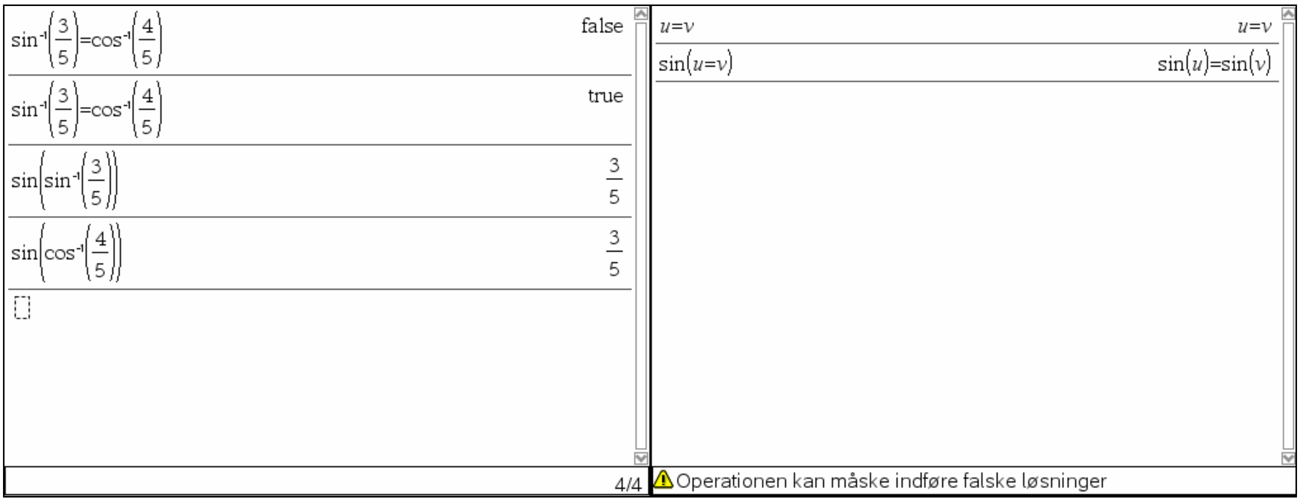
Now if we compute both sides numerically we see that their values are identical (and also consistent with the measurement):



So what is wrong? Is the particular example out of reach for the mathematical abilities of TI-Nspire?

Not at all! Let us first demonstrate symbolically using TI-Nspire's built-in symbolic features that the statement has to be true. So we construct a *CAS-assisted proof*:

Clearly the two numbers $\sin^{-1}(3/5)$ and $\cos^{-1}(4/5)$ are positive numbers in the range $]0;\pi/2[$. Thus they are identical if and only if they have the same sine-value:



Since both sine-values are 3/5, i.e. they are identical, it follows that the above identity $\sin^{-1}(3/5)=\cos^{-1}(4/5)$ has the Boolean value **true**[4].

Next let us check in a similar case, that TI-Nspire actually some times recognizes different expressions for the same angle. In the 3-4-5 triangle you may have established the value of the angle *B* in the form: $B=\cos^{-1}(3/5)$. It follows that the angle *A* can also be represented as $A=\pi/2-\cos^{-1}(3/5)$. Thus the following identity must be true:

$$\sin^{-1}(3/5)=\pi/2-\cos^{-1}(3/5)$$

[4] As shown on the next application, you can in general apply the sine-function to an equation. But in the above case the identity is not an equation but a Boolean expression and thus has the value **true** or **false**. You will therefore get a domain error if you apply the sine-function directly to the identity.

And indeed TI-Nspire report the statement as being true:

$$\sin^{-1}\left(\frac{3}{5}\right)=\frac{\pi}{2}-\cos^{-1}\left(\frac{3}{5}\right) \qquad \text{true}$$

So what's the problem? In one case it works, in another it doesn't. The problem lies in the built-in reduction algorithms. To prove that two different symbolic expressions are equivalent, TI-Nspire reduces both expressions to a standard form and check if the associated standard forms are identical. In the above case e.g. the following auto-reductions are involved:

$$\sin^{-1}\left(\frac{3}{5}\right) \qquad \sin^{-1}\left(\frac{3}{5}\right)$$

$$\frac{\pi}{2}-\cos^{-1}\left(\frac{3}{5}\right) \qquad \sin^{-1}\left(\frac{3}{5}\right)$$

This strategy works very well for rational expressions, and also for simple algebraic expression involving square roots such as

$$\sqrt{5-2\cdot\sqrt{6}}=\sqrt{3}-\sqrt{2} \qquad \text{true}$$

But in general it is too complicated to be able to reduce any expression to a suitable standard form. In fact in general the problem is undecidable, i.e. we will never find an efficient algorithm that can handle arbitrary expressions. So although we can easily fix the above problem we are just pushing the boundary of TI-Nspire's ability a little further: Other expression will very soon pop op that TI-Nspire cannot handle.

This also shows that there are limits to what we can expect from TI-Nspire in terms of symbolic proofs (CAS-assisted proofs). E.g. it is very hard to construct symbolic algorithms that can show directly that the sum of the three angles in a triangle must be pi! Using e.g. the cosine relations you would expect the CAS-system, to be able to recognize the sum of the three angles as the number pi.

$$\cos^{-1}\left(\frac{a^2+b^2-c^2}{2\cdot a\cdot b}\right)+\cos^{-1}\left(\frac{b^2+c^2-a^2}{2\cdot b\cdot c}\right)+\cos^{-1}\left(\frac{c^2+a^2-b^2}{2\cdot c\cdot a}\right)$$

$$-\sin^{-1}\left(\frac{a^2-b^2+c^2}{2\cdot a\cdot c}\right)+\sin^{-1}\left(\frac{a^2-b^2-c^2}{2\cdot b\cdot c}\right)-\sin^{-1}\left(\frac{a^2+b^2-c^2}{2\cdot a\cdot b}\right)+\frac{3\cdot\pi}{2}$$

$$\cos^{-1}\left(\frac{a^2+b^2-c^2}{2\cdot a\cdot b}\right)+\cos^{-1}\left(\frac{b^2+c^2-a^2}{2\cdot b\cdot c}\right)+\cos^{-1}\left(\frac{c^2+a^2-b^2}{2\cdot c\cdot a}\right)|a=7 \text{ and } b=▸$$

$$-\sin^{-1}\left(\frac{17}{22}\right)-\sin^{-1}\left(\frac{89}{154}\right)-\sin^{-1}\left(\frac{1}{14}\right)+\frac{3\cdot\pi}{2}$$

$$\cos^{-1}\left(\frac{a^2+b^2-c^2}{2\cdot a\cdot b}\right)+\cos^{-1}\left(\frac{b^2+c^2-a^2}{2\cdot b\cdot c}\right)+\cos^{-1}\left(\frac{c^2+a^2-b^2}{2\cdot c\cdot a}\right)|a=7 \text{ and } b=▸$$

$$3.14159265359$$

Symbolically TI-Nspire does almost nothing. Numerically it of course computes pi! And it will not help you to apply various constraints, making sure the triangle does exist.

So where does this lead us? First of all it shows us very reassuring that although TI-Nspire is much better than us when it comes to symbolic calculations, we are definitely superior when it comes to reasoning! Fortunately we can recognize many patterns and relations that TI-Nspire does not recognize.

Second we advise our students to adopt the following interpretation of TI-Nspire's abilities:

1. When TI-Nspire reports that an identity is true it actually knows it is true, in the sense that it can prove the identity.
2. But when TI-Nspire reports that an identity is false this does not mean that it is actually false, only that TI-Nspire was not able to prove it is true! It may in fact very well be true unless you have an independent argument for why the identity is false!

In both cases students should be encouraged to back up their findings using independent reasoning such as graphical, numerical or even symbolic arguments.

This concludes the talk!

Concluding remarks:

To summarize we have been investigating various possibilities for interdisciplinary teaching between mathematics and the humanities, in this case Danish, regarding the theme: Arguments and reasoning. We have concentrated on the mathematical aspects and furthermore we have only studied those aspects where TI-Nspire can make a substantial contribution. Three examples were illustrated:

1. TI-Nspire can be used to handle *cellular automatons* (as a metaphor for neural computing).
2. TI-Nspire can handle *Boolean expressions* symbolically and 'numerically' through truth-tables. This can be used in discussions of propositional logic.
3. TI-Nspire can be used to *investigate symbolic proofs* (in the spirit of Leibniz symbolic calculus) thus demonstrating limitations in this kind of symbolic reasoning.

Not all of these examples will be suitable for all students. We expect the first to be generally interesting for students whether their focus is upon the humanities or science. We hope the second will interest at least many of the students whether their focus is upon the humanities or science. But the third part is probably only interesting to students (and teachers) that really like math and science. At the end of the next year we will all be much wiser!

Bjørn Felsager
Haslev Gymnasium & HF
May 2008