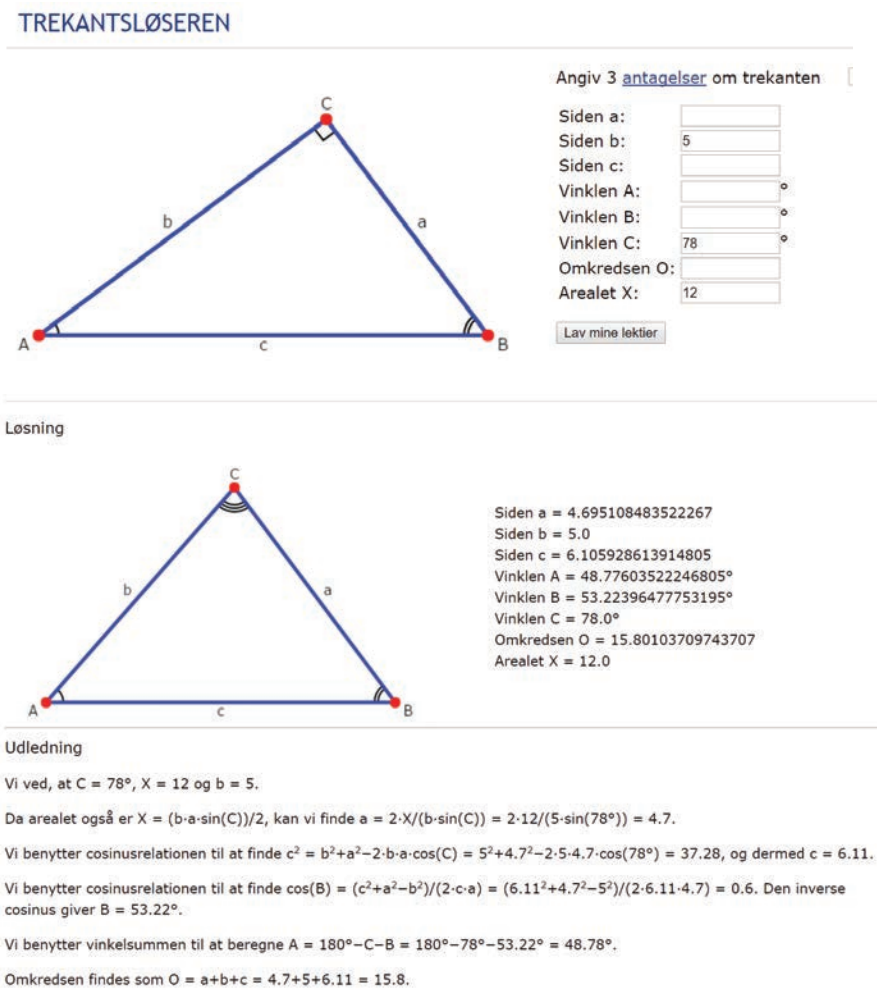
LAV MINE LEKTIER?

Michael Schwartzbach, Institut for Datalogi, Aarhus Universitet, mis@cs.au.dk

Visse typer af matematikopgaver i gymnasieskolen kan løses automatisk. Hvor langt man kan gå og hvilke pædagogiske konsekvenser det har, diskuteres i dette indlæg.

CAS værktøjer til matematik (som Maple, Mathematica og Wolphram Alpha) giver mange genveje til at finde både numeriske og symbolske løsninger til ligninger og avancerede beregninger. Dette udfordrer naturligvis læringen og øger måske spredningen blandt eleverne, da værktøjerne i sig selv ikke er trivielle at benytte korrekt. Er det bedre ikke at tillade dem, eller skal man tværtimod gå meget længere?

I forbindelse med et talentforløb for 1.–semester studerende på datalogiuddannelsen på Aarhus Universitet, cs.au.dk/Challenge, har jeg udvilket to tools, der skal illustrere, at noget af deres tidligere hårde arbejde i gymnasiet faktisk kan automatiseres fuldstændigt. Den datalogiske pointe med dette er dels at vise, at grænsen går længere end de tror og dels at præsentere de teknikker,der muligør dette.



Øverst: Trekantsløserens startside
Nederst: Trekantsløserens svarside.

Det ene tool, *Ligningsløseren*, cs.au.dk/~mis/xxx.html, løser ligninger mellem to vilkårlige polynomnielle udtryk med en enkelt variabel. Det andet tool, *Trekantsløseren*, cs.au.dk/~mis/abc.html, løser alle opgaver om trekanter med 3 antagelser om sider, vinkler, omkreds og areal. Det specielle ved begge tools er, at de ikke bare finder numeriske løsninger (det gør som sagt mange andre allerede), men at de genererer komplette besvarelser med alle udledninger, argumenter, mellemregninger og illustrationer, der (i princippet) kan printes og afleveres direkte til matematiklæreren. Envidere garanterer jeg (lidt i spøg), at besvarelsen vil modtage karakteren 12, hvis den nu blev bedømt særskilt. Begge tools kender til utallige specialtilfælde, der kan simplificere besvarelserne – for at selv den pedantiske matematiklærer vil bevare 12–tallet.

At disse tools findes kan selvfølgelig vurderes på flere måder. De er oplagt interessante for datalogistudenter, men jeg håber, at de også på fornuftig vis kunne indgå i matematikundervisningen i gymnasiet. På den ene side er de uendelige kilder til eksempler og detaljerede forklaringer, hvilket kan bibringe den interesserede elev en dybere forståelse. Mere kontroversielt kan det jo gøre det vanskeligt at stille netop disse typer opgaver, da den mindre interesserede elev jo rent faktisk kan printe en løsning uden at opnå nogen læring. Sammenlignet med andre, mere generelle CAS værktøjer, har de to tools dog den fordel, at eleverne konstant mindes om alle de manuelle skridt, man ville benytte til at finde løsningen – intet bliver fejet ind under gulvtæppet.

Jeg ser situationen lidt analog (men mere specialiseret) til indførelsen af lommeregneren, der også udfordrede visse opgavetyper men muliggjorde andre, så som problemløsningsopgaver, der involverede lidt komplicerede udregninger. I stedet for at eleverne løser opgave a) – f) om andengradsligninger, kan de få opgaver, hvor polynomierne skal genkendes i en kontekst, inden ligningerne skal løses, og tilsvarende med trigonometriske opgaver, hvor trekanten måske ikke gives helt eksplicit i opgaven. Denne tilgangsvinkel vil dog hæve abstraktionsniveauet, hvilket nok også er en pædagogisk overvejelse værd.

Under alle omstændigheder findes sådanne tools nu, og de kan jo ikke forsvinde igen. Der er også oplagt flere slags typeopgaver, der kunne automatiseres på samme niveau, fx funktionsanalyse, differentiering og differentialligninger. Det er i hvert fald et datalogisk princip, at hvis noget kan automatiseres, så er man næsten nødt til at gøre det. Dette brydningsfelt er jo også kendt fra fx spil som fx Skak og Sudoku.

Implementationen af de to tools involverer også noget interessant matematik og algoritmik, der jo også kunne bruges til at uddybe forståelsen af emnerne.

Den vanskeligste del af *Ligningsløseren* er at reducere et polynomielt udtryk til sin simpleste form. Dette gøres ved hjælp af 96 induktive genskrivningsregler, som fx:

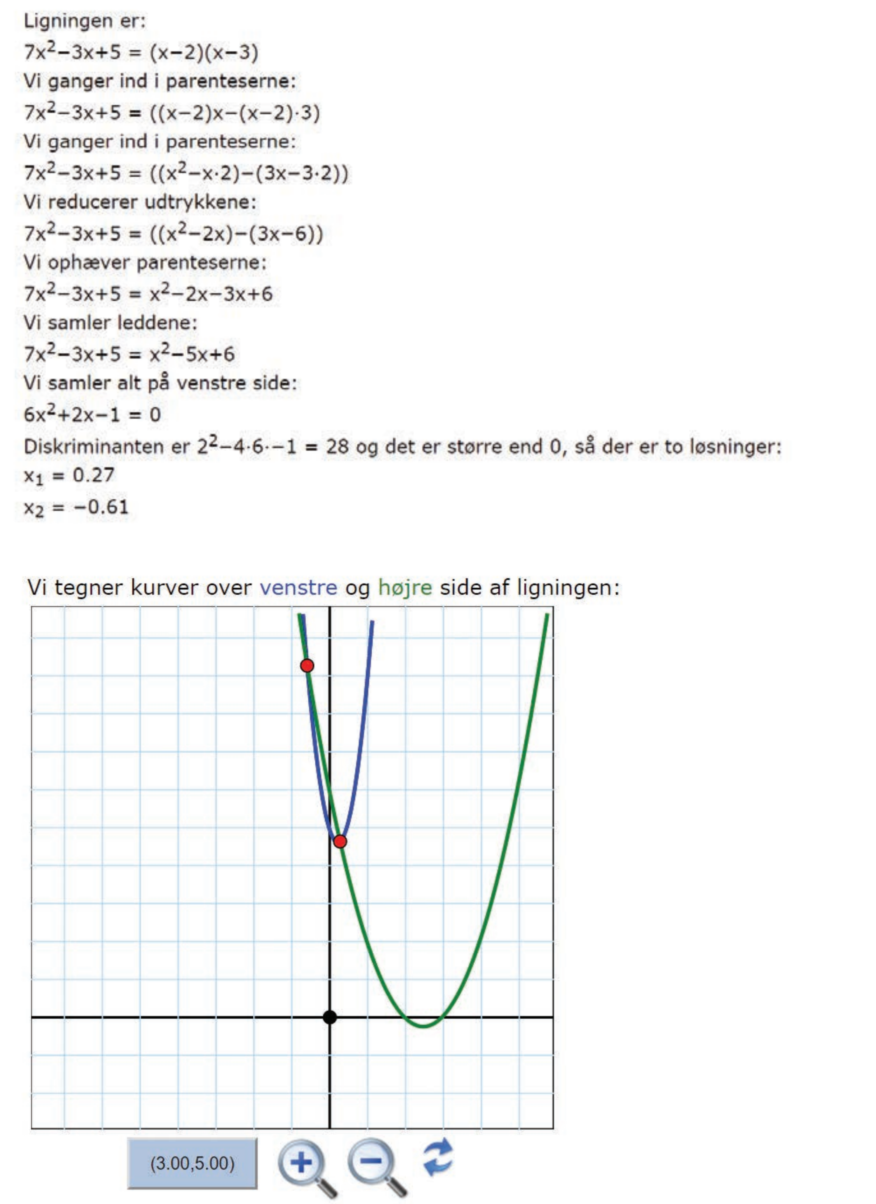
- $E*0 \rightarrow 0$
- $-E_1*-E_2 \rightarrow -E_1*-E_2$
- $E_1 + (E_2 + E_3) \rightarrow (E_1 + E_2) + E_3$

der gentagent forsøges indtil ingen længere er mulige. Ligningsløseren kender også til 8 forskellige former af den klassiske andengradsligning, for hvilke der kan benyttes specielt simple løsningsteknikker.

Ved første øjekast er der med 3 vinkler, 3 sider, omkreds og areal $8 \cdot 7 \cdot 6 = 336$ forskellige opgavetyper om trekanter. Af disse viser de 78 sig dog at være underspecificerede, hvilket vil sige, at de tillader uendeligt mange løsninger (som fx 3 vinkler). De resterende 258, der tillades, kan ved hjælp af maksimal brug af symmetri kollapses til 11 essentielt forskellige trekantsopgaver, der kort kan beskrives som:

- *ABO*, *ABX*, *ABa*, *Aab*, *ABc*, *AXb*, *Abc*, *Oab*, *Xab*, *XAb*, *abc*

hvor *a*, *b*, *c* er sider, *A*, *B*, *C* er vinkler, *O* er omkredsen og *X* er arealet. Disse opgavetyper kan så hver have 0, 1 eller 2 løsninger. Trekantsløseren kender til rigtig mange specialtilfælde for retvinklede, ligebenede og ligesidede trekanter, der kan bruges til at give de simpleste løsninger.



Øverst: Ligningsløserens startside
Nederst: Ligningsløserens svarside.

For den nørdede læser: *Ligningløseren* og *Trekantsløseren* fylder henholdsvis 3161 og 1314 linjer Java–kode, hvilket faktisk er meget lidt. Hvis man er interesseret i flere detaljer, står jeg gerne til i rådighed.

Jeg håber, at dette indlæg kan give anledning til en debat om muligheder og problemer ved denne slags tools.