Computeren som fejlfinder[1]

Af Jesper Møller, ph.d.-studerende ved IT-højskolen i København.

En simpel computerfejl kan sende firmaer konkurs og i værste fald koste menneskeliv. F.eks. blev mindst seks mennesker stegt levende, da de modtog op til 20.000 gange for store strålingsmængder under en rutinebehandling for kræft i USA og Canada for nogle år siden. Mindst to patienter døde af den katastrofale behandling, som skyldtes en softwarefejl.

På IT-højskolen i København forsker man netop i, hvordan *formel verifikation*, dvs. logik og matematik, kan bruges til at undgå fejl i hardwarechips og software, lige fra rumraketter over hospitalsudstyr til hjemmecomputeren.

Mange folks barnetro på, at computere *altid* regner rigtigt, fik et alvorligt knæk i 1994, da en splinterny Intel Pentium Processor dividerede flere tusinde regnestykker forkert. En basal designbommert var skyld i, at processoren var blevet udstyret med fem fejlagtige indgange i en divisionstabel.

Intel måtte gratis erstatte tusindvis af defekte processorer og satte ved samme lejlighed flere hundrede millioner dollars over styr i tabt goodwill.

Fejlen kom dog ikke bag på folk med dybtgående kendskab til computere. Det kan nemlig være en langsommelig fornøjelse at tjekke for fejl i software og hardware, og tit er virksomheder og programmører under stort pres for at sende den nyeste model på markedet så hurtigt som muligt.

Men Pentium processorens regnefejl og andre lignende designfejl kan undgås ved hjælp af *formel verifikation.*

Kort fortalt er formel verifikation, at man opstiller en matematisk model af et påtænkt produkt, f.eks. en microprocessor eller softwaren i en mobiltelefon. Modellen bliver så tjekket imod alle de krav, som det færdige produkt skal kunne overholde for at virke efter hensigten.

Kravene til en regnemaskine som Pentium processoren vil f.eks. være at maskinen skal kunne lægge tal sammen, trække fra, dividere og gange. Hvis Intel havde investeret mere i formel verifikation, så havde de også fundet fejlen i tide. Intel har faktisk lært af lektien med de defekte processorer, og i dag bruger de formel verifikation som en del af deres produktudvikling.

Det geniale ved formel verifikation er, at den udtømmende fejlfinding foretages ganske automatisk. Softwareprogrammer kan nemlig bruges til at tjekke om kravene til et bestemt produkt er overholdt. Med andre ord kan computeren tjekke om en model af et software- eller hardwareprodukt er korrekt i forhold til den givne kravspecifikation.

100% garanti

For bare 5-10 år siden var formel verifikation noget man næsten udelukkende beskæftigede sig med inden for den akademiske verden. Men vores stigende afhængighed af IT i alt lige fra biler, telefoner, mikrobølgeovne til hjemmecomputere har medført, at også industrien er begyndt at interessere sig for fænomenet.

Indtil videre er det mest inden for hardwarebranchen, at formel verifikation er slået igennem, mens softwareindustrien som hovedregel kun bruger de matematiske modeller til "livsvigtige" styringsprogrammer, som f.eks. hospitalsudstyr og biler eller til meget dyre projekter, som f.eks. rumraketter.

I praksis er en matematisk verifikation ofte meget billigere og mere effektiv end de traditionelle test, som foregår efter, at udviklingen af et bestemt program er gået i gang. Fornyelig fandt to amerikanske forskere fra Bell Labs ved hjælp af formel verifikation ti gange så mange fejl i et softwareprogram til en telefonswitch end om-

[1] Publiceret i Orbitalen `www.dr.dk/orbitalen`, september 2000. Redigeret af Mikael Kjærbye, journalist ved IT-højskolen i København.

To katastrofale softwarefejl som kunne have været undgået med formel verifikation:

Fatale strålingsoverdoser
1985–87: Elleve Therac-25 strålebehandlingsmaskiner i USA og Canada indeholder en række softwarefejl, som resulterer i mindst seks alvorlige ulykker, hvor kræftpatienter modtager op til 20,000 gange for store strålingsmængder.
To patienter dør som en direkte følge af de katastrofale overdoser, mens andre bliver svært kvæstede og handicappede. Maskinerne accelererer elektroner til høj-energiske stråler, som kan destruere kræftsvulster med minimal skade på det omkringliggende væv. En mandlig patient fra Texas, som døde tre uger efter at have modtaget en massiv overdosis, beskrev hvordan hans hud "sydede lige som æg på en brandvarm stegepande".

NASA rumraket brænder op i atmosfæren
September 1999: NASAs Climate Orbitor kommer for tæt på Mars' atmosfære og brænder op. Problemet er, at eet hold ingeniører bruger engelske mål, så som pund, fod og tommer, mens et andet hold ingeniører bruger metriske mål, så som kilo, meter og centimeter.
Raketten bliver forvirret af de forskellige mål og kommer for tæt på Mars, før den begynder landingsproceduren og brænder op.

kring ti forskere i den almindelige testafdeling kunne producere.

I princippet giver formel verifikation en 100% garanti for, at programmet overholder de krav, som man har opstillet. Problemet er selvfølgelig, at det kan være svært at opstille de rigtige krav. Derfor kan vi nok aldrig love, at man kan udvikle en 100% fejlfri computer.

Dårlig kravspecifikation årsag til eksplosion af rumraket

Mange softwarefejl skyldes netop dårlig planlægning, hvor udviklerne ikke præcist har specificeret, hvad et bestemt program skal kunne gøre, og hvilke tænkelige situationer softwaren skal kunne fungere under.

F.eks. selv-destruerede det Europæiske Rumagenturs nye raket Ariane 5 efter, det var kommet drastisk ud af kurs bare 40 sekunder efter lift-off i juni 1996. Raketten, der anslås at have kostet 600 millioner dollars at producere, var ellers udstyret med et softwareprogram, som skulle rette raketten op i tilfælde af fejlkurs.

Problemet var bare, at udviklerne havde genbrugt softwaren fra den tidligere raket Ariane 4 og ikke havde forestillet sig, at raketten kunne komme så meget ud af kurs, som den gjorde. Programmet kom således ud for en situation, det ikke var programmeret til at kunne klare. Resultatet var, at raketten destruerede sig selv.

"Fejlfri" alternativer til Microsoft

Generelt sagt er computerindustrien mest interesseret i at rette fejl, som enten kan koste menneskeliv, eller hvor millioner kroner står på spil, som f.eks. inden for rumforskning.

Men mange programmer til vores hjemmecomputere kunne godt blive langt bedre og mere fejlfri, hvis firmaer som Microsoft investerede tilstrækkelig tid og penge i formel verifikation. Tit er de store firmaer imidlertid mere interesserede i at få udviklet den nyeste model med de mest fancy features i stedet for at gøre de eksisterende programmer fejlfrie.

De fleste computerbrugere har måske også accepteret, at softwareprogrammer ofte kommer med småfejl. Men der er praktiske alternativer, som er mere robuste og fejlfri end f.eks. Microsofts produkter.

Inden for forskningsverdenen foretrækker langt de fleste brugere således at benytte Linux produkter fordi de er mere stabile. De er også ganske gratis at downloade fra internettet og så får man den originale programmeringskode med, så man selv kan rette i den, hvis der er fejl.

Derfor er det ikke underligt, at der er sket en eksplosiv vækst af Linux-brugere, f.eks. er der nu flere Linux-brugere end Macintosh-brugere. Men det varer sikkert et stykke tid inden Linux overhaler Microsoft, som for den almindelige brugere har mange flere features og er mere brugervenlig end Linux.