Over de bewijsbaarheid van programmacorrectheid.

1) Omdat "Verdeel en Heers" in eerste instantie het enige principe is, waarmee we iets groots aankunnen, zal een programma naarmate het groter is uit meer componenten opgebouwd moeten worden. Willen we voor het totale programma een zeker betrouwbaarheidsniveau bereiken, dan dient het betrouwbaarheidsniveau van de individuele programmacomponenten hoger te zijn naarmate het aantal componenten groter is.

2) Testen van programma's kan een zeer overtuigende methode zijn om de aanwezigheid van fouten aan te tonen, maar is hopeloos inadequaat om hun afwezigheid aan te tonen. De enige mogelijkheid die ons rest is de correctheid van het programma te bewijzen.

3) Aangezien de (bv. intellectuele) inspanning nodig voor de bewijsvoering critisch afhangt van de structuur van het programma, dient de programmeur zijn programma zo te structureren, dat bewijsvoering nog wel mogelijk is. Ik ga er hierbij van uit dat programma's behalve voor de machine uitvoerbaar voor de mens leesbaar moeten zijn. In het bijzonder dient de programmeur er voor te waken, dat het aantal gevallen, dat onderscheiden moet worden, additief en vooral niet multiplicatief combineert.

4) In dit licht dient het gebruik van de goto-statement ten sterkste afgeraden te worden: de goto-statement is als combinatorische complexiteitsgenerator ontmaskerd.

5) De (bekende) sequentieringsclausules belichamen een aspect van programmeringsdiscipline, waardoor bewijsvoering profiteren kan van de postulaten van C.A.R.Hoare. Van het gebruik van deze postulaten zal een voorbeeld gegeven worden. Aan de hand hiervan zal het begrip "operationele abstractie" als middel ter beknotting van de hoeveelheid bewijswerk worden toegelicht.

6) Aan de hand van een volgend voorbeeld zal het begrip "representationele abstractie" worden toegelicht. Ook dit dient ter beknotting van de hoeveelheid bewijswerk.

7) Betoogd zal worden dat wij om de correctheid van programma's bewijsbaar te houden, programma's moeten leren zien niet als op zichzelf staande objecten, maar als leden van een familie aanverwante programma's: aequivalente programma's kunnen zich hier presenteren als alternatieve verfijningen van eenzelfde abstract programma.

8) Tenslotte zal de hoop uitgesproken (en naar vermogen gerechtvaardigd worden) dat de bewijslast geen taakverzwaring voor de programmeur hoeft in te houden. Integendeel: als hij de bewijsvoering hand in hand met het programma laat groeien lijkt hij een waardevol richtsnoer bij de opbouw van zijn programma gevonden te hebben.

Edsger W.Dijkstra