

## Tot fent el cafè amb el professor Dijkstra

Edsger W. Dijkstra va ser a Barcelona aquest any amb motiu de la Convenció Informàtica Llatina. Per a (ciència), el van entrevistar Albert Llamós i Fernando Orejas. El professor Dijkstra és una de les principals figures mundials al món de la programació. En el número 31, de-

dicat a la informàtica, entre altres treballs, hi inclouem el de Pere Botella que plantejava un panorama general de la programació dels ordinadors, avui. L'entrevista amb Dijkstra i aquest treball poden ser llegits conjuntament.

Fotografies: Anna Boyé

46 (694/Volum 3/novembre 1983

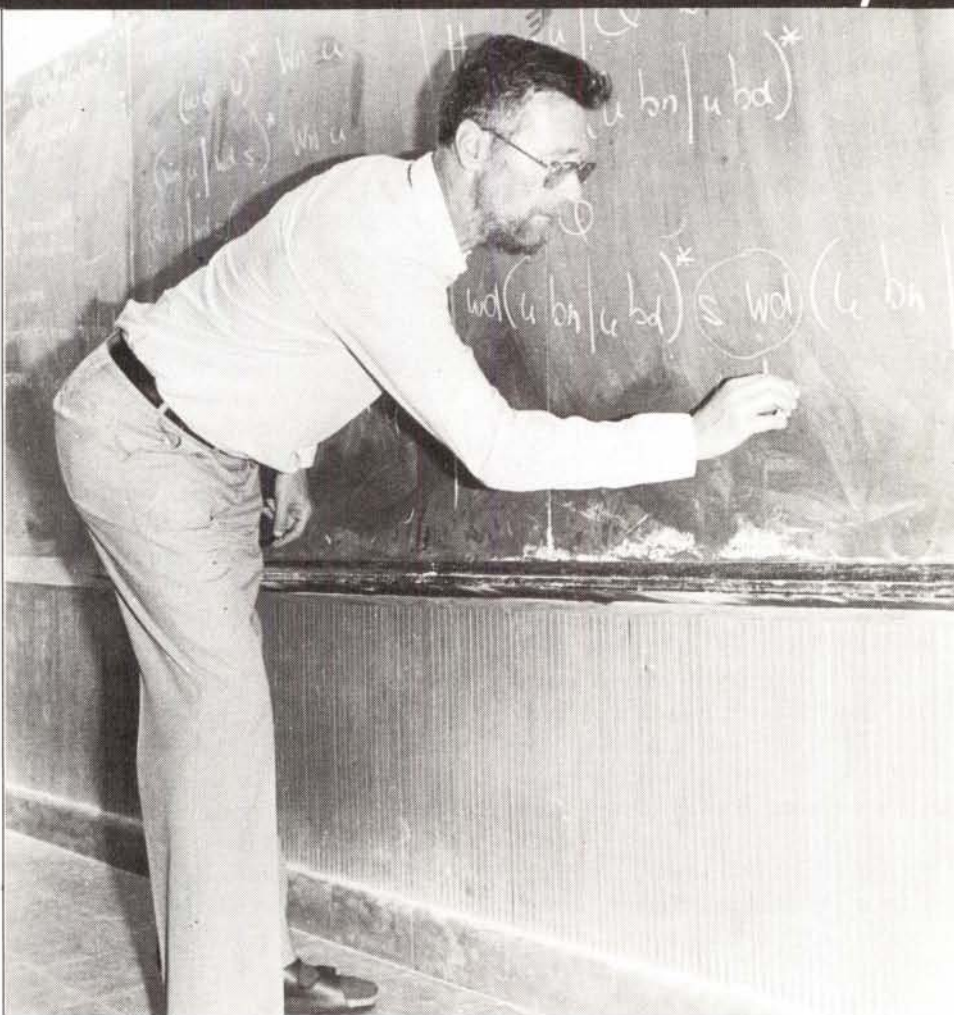
ciència 32)

A l'entorn del 1968 es va produir una presa de consciència col·lectiva sobre la veritable dificultat de construir programes per a computadors. En particular es va constatar la necessitat de canviar els hàbits que havien adquirit els programadors de manera més o menys espontània de vers actituds més sistemàtiques i disciplinades. Aquest canvi d'actitud, originàriament anomenat **programació estructurada**, va indissolublement lligat al nom d'Edsger W. Dijkstra.

Es va introduir en el terreny de la programació molt prematurament. Com ell mateix ho explica, el primer matí de la primavera de 1952, quan era estudiant de física teòrica a la Universitat de Leiden. Pocs anys després va decidir dedicar-se completament a la programació, en una època en la qual aquest terme amb prou feines se sabia quin significat tenia. Tant és així que el 1957, en contraure matrimoni, les autoritats municipals es van negar a acceptar-lo com a indicatiu de la seva professió, i va haver-lo de substituir pel de *físic teòric*.

Posteriorment va formar part del grup que va construir el primer compilador d'Algol 60 i va dissenyar el sistema operatiu THE, que va aportar conceptes que en el seu moment van resultar capgiradors.

Però el professor Dijkstra és conegut sobretot per haver traçat les línies directrius d'allò que havia d'esdevenir l'activitat de programar, que han quedat recollides en multiplicitat d'articles científics i assaigs dels quals són especialment coneguts la diatriba en forma de carta al director "Go to Statement Considered Harmful" (1958) i



la declaració de principis fundacionals "Notes on structured Programming" (1969), que va ser la responsable de la posterior difusió del terme. Tots dos han sigut recollits, juntament amb d'altres, en un llibre de recent edició: *E. W. Dijkstra's Selected Writings*.

A més, ha estat autor de conceptes, notacions, algorismes i exemples que són coneguts i han perdurat en la pràctica actual dels investigadors i els bons professionals de la programació.

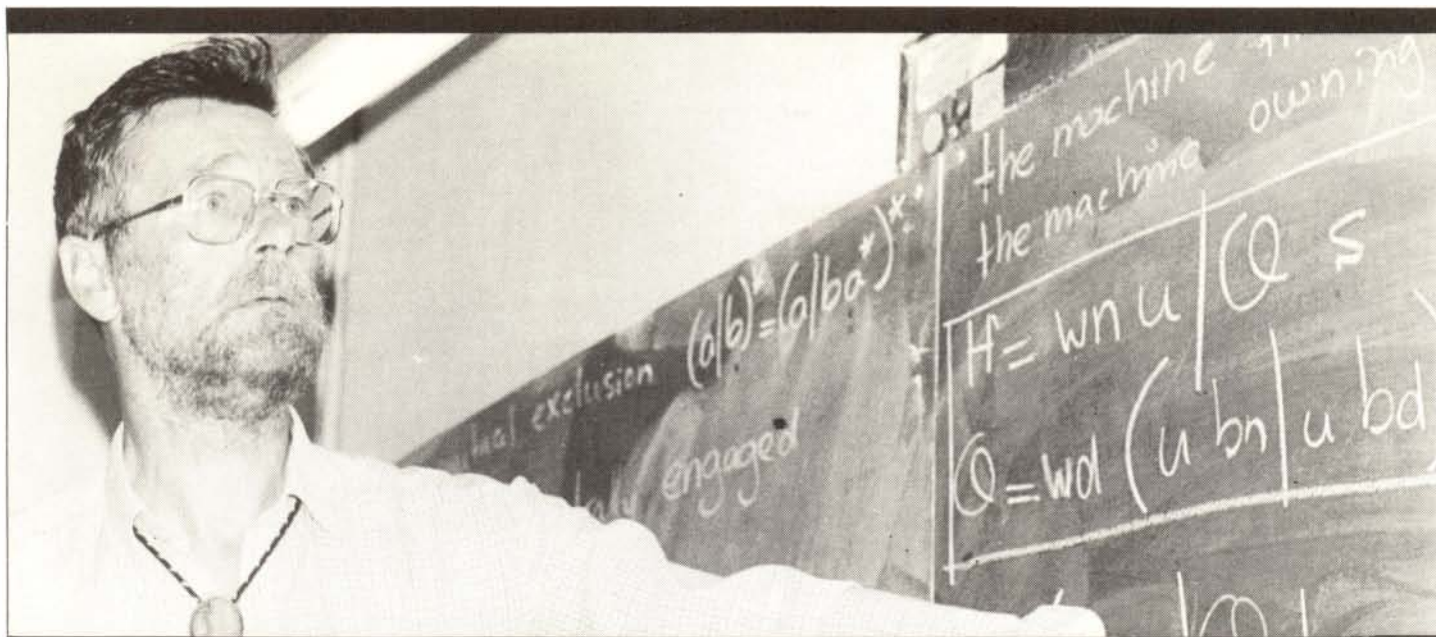
Després d'un bon àpat, el professor

Dijkstra, que s'ha mostrat tot el temps molt amable, es disposa a ser entrevistat.

---

(ciència): —Fa deu o dotze anys, el terme **programació estructurada** havia adquirit un significat molt precís que tenia, àdhuc, una certa connotació combativa. Ara, assimilats totalment o parcialment els seus principis, els nous desenvolupaments semblen haver conferit al terme uns





contorns més difosos.

*Des del vostre punt de vista, què ha ocorregut d'essencial durant aquests anys i en quina mesura s'han assolit els propòsits originaris d'aquell projecte?*

**E.W. Dijkstra:** —Tota la reflexió que s'ha efectuat a l'entorn de la programació d'ençà aleshores procedeix del desig de construir programes tractables intel·lectualment.

De fet la programació hauria d'esdevenir, cada cop més, una activitat matemàtica. Precisament allò que em vaig proposar va ser establir amb cura com formalitzar l'argumentació sobre els programes. Dissortadament, però, no s'ha obtingut ni axiomes ni teoremes. Es tracta d'un camp obert a l'exploració matemàtica.

La base de la formalització que havia de satisfer els meus designis la vaig proposar l'octubre de 1973 a Blanchland, prop de Newcastle, a la reunió del grup de treball W.G.2.3 de la IFIP. Va ser allà on vaig descobrir els transformadors de predicats que treballen cap endarrera. A partir d'aquí s'ha avançat molt en el camp de la semàntica de llenguatges, s'ha treballat molt. El següent pas va ser, per exemple, la tesi de S. Owicki, que ho va estendre al domini de la programació paral·lela.

Allò que resta per fer és fonamentalment prosseguir aquesta conversió de la programació en una activitat matemàtica. De fet la clau de l'èxit en programació ha estat sempre una bona base en aquest terreny. La matemàtica augmenta la nostra capacitat de raonament perquè els objectes que subministra són, en principi, molt còmodes de manipular. Poden ser calcula-

bles, per exemple.

Aconseguir la reducció de la programació a una activitat matemàtica és certament complex, però crec que és possible. Encara més, estic convençut que el meu somni no tardarà a materialitzar-se.

D'altra banda, les matemàtiques són encara molt informals i caldria reduir el raonament matemàtic a un càlcul lògic. És la manera més prometedora d'augmentar la nostra capacitat de raonament.

A més, sempre que les matemàtiques s'han aplicat a una nova ciència, aquesta ha influït sobre les matemàtiques. I crec que el fet d'aplicar-les a la programació les ha d'influir de tal manera que ens subministrin sistemes molt més útils i efectius.

**(ciència):** —*Fa deu anys, el salt qualitatiu que havia representat l'aparició d'aportacions capgiradores en el domini de la validació formal de programes va engendrar efectivament aquesta esperança en molts membres de la comunitat acadèmica. No obstant això, els anys han anat passant sense que hagin aparegut aquells esperats teoremes destinats a simplificar la feina desproporcionadament enutjosa de validar formalment un programa. Per aquest motiu aquelles esperances semblen avui esvaïdes. ¿Podria ser que després de la crisi de la programació haguéssim topat amb la crisi de la validació formal?*

*Abans de contestar res, el nostre entrevistat somriu un moment en silenci. Es fa difícil de saber si està ordenant les idees abans de contestar o bé assaborint l'íntima satisfacció que li produeix poder donar resposta a aquesta qüestió.*

**E.W. Dijkstra:** —El motiu pel qual la verificació de programes no ha esdevingut un procediment d'aplicació pràctica radica en el fet que els sistemes de validació parteixen d'un enfocament equivocat i prenen el text del programa com a punt de partida del procés d'anàlisi. I això és com posar el carro davant del cavall.

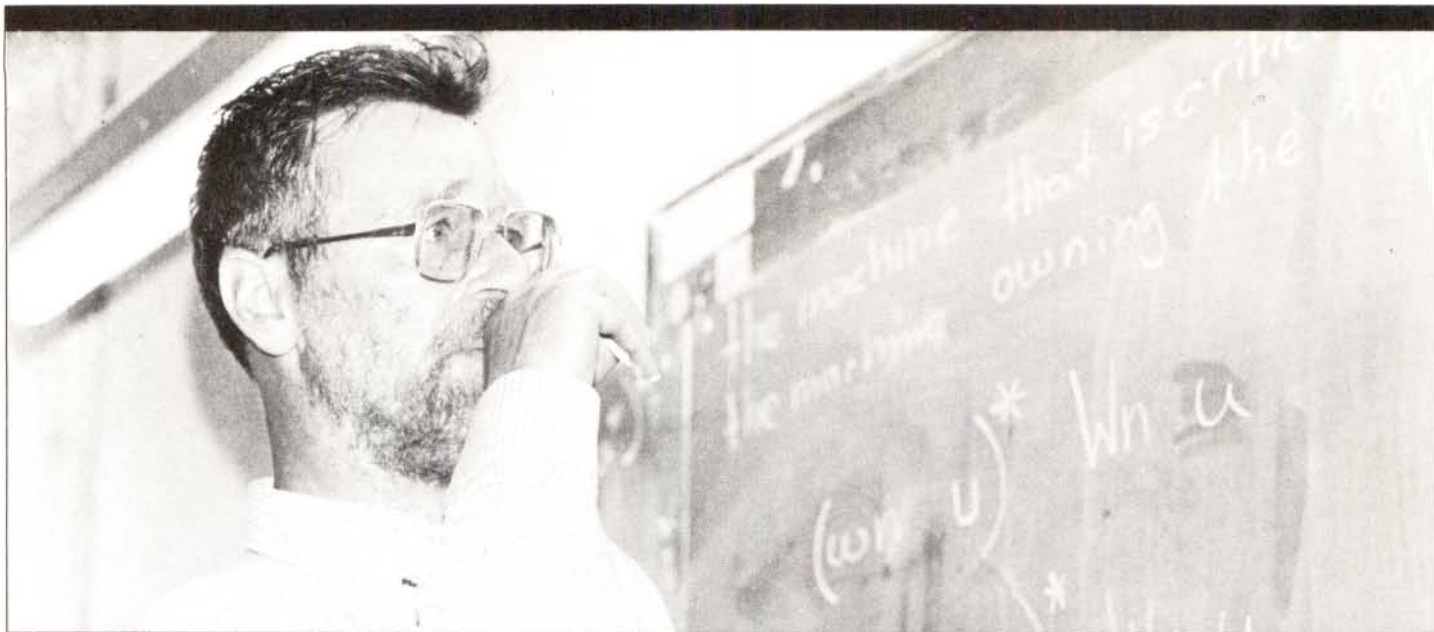
El text d'un programa és tan sols la punta d'un iceberg, que sobresurt com a resultat final d'un llarg procés de reflexió que roman al dessota, invisible. El text d'un programa pot ser un dipòsit molt compacte de treball intel·lectual. En alguns casos no és més que un subproducte. I això és així perquè el seu únic objectiu és la seva execució.

Els mecanismes de prova han d'acompanyar tot aquest procés previ. Comprovar que un programa satisfà una prova esdevé una operació gairebé trivial. El procés invers, enginyar una prova tal que el programa la satisfaci, no ho és en absolut. El maig de 1976 es va completar un verificador de programes escrit en Pascal. La seva execució era terrible: consumia hores a provar propietats que a mà s'obtenien en cinc minuts.

Tant els meus col·laboradors com jo hem posat molta atenció a com presentar el desenvolupament d'un programa. És un procés molt complex que pren de l'ordre de deu vegades la longitud del codi final del programa.

I, dit sigui de passada, en el fet que el text d'un programa doni una visió tan parcial de tot allò que hi ha involucrat, radica la debilitat de totes les anomenades eines d'ajut a la programació. La majoria d'elles parteixen d'una concepció molt estreta de la





programació i giren tan sols entorn del codi del programa.

*(El professor Dijkstra s'expressa en un anglès impecable. Parla lentament, amb una actitud concentrada en les seves pròpies paraules. De tant en tant s'atura amb la vista posada en un punt fix. A vegades, l'interval de silenci és tan llarg que sembla com si ja hagués donat la resposta per acabada i esperés una nova pregunta. Però tot d'una repren la frase que havia deixat penjada i el seu discurs prossegueix el ritme constant del seu anglès meticulós i primmirat).*

**(ciència):** —D'ençà el 1972, moltes promocions formades en hàbits més sistemàtics i rigorosos de programació han completat els seus estudis i s'han incorporat al món professional. Creieu que s'ha notat una diferència apreciable en la qualitat del seu treball?

**E.W. Dijkstra:** —Sí. Absolutament. I això, com tot progrés, no deixa de presentar els seus problemes: els joves que surten de la universitat i s'incorporen a la indústria són més competents que els seus directors. Els conflictes que es deriven d'això palesen que aquest progrés significatiu s'ha produït. En cas contrari, el problema no s'hauria presentat.

**(ciència):** —Si ara us calgués donar un curs d'introducció a la programació, ho faríeu de la mateixa manera que ho vau proposar en el vostre opuscle de 1971 "A Short Introduction into the Art of Programming"?

**E.W. Dijkstra:** —No. Quan vaig donar el curs per al qual aquelles notes servien de suport no hi havia cap alternativa que no fos la intuïció per tal de definir la semàntica d'un llenguatge de la programació de manera operacional. Actualment a la universitat introdueixo els estudiants en el disseny de programes correctes directament a través del càlcul de la precondició més feble, acabament inclòs.

**(ciència):** —I per tal de definir el comportament que ha de satisfer un programa, o algun dels blocs, utilitzeu algun llenguatge d'especificació?

**E.W. Dijkstra:** —No. Simplement el càlcul de predicats. Els llenguatges d'especificació són necessaris, però en un curs introductor el càlcul de predicats és suficient.

**(ciència):** —Considereu que, catorze anys després, Pascal és encara un llenguatge de programació satisfactori?

**E.W. Dijkstra:** —En principi crec que malgrat la seva aparent simplicitat, és encara un llenguatge massa complicat. Quan N. Wirth el va dissenyar, ho va fer perseguint l'objectiu bàsic d'aconseguir un llenguatge que fos fàcil d'ensenyar i que permetés contribuir a la reducció del cost de la programació mitjançant l'estandardització dels hàbits de programació. En aquest sentit ha constituït un èxit rotund. Malgrat l'abundància d'alternatives, l'obstrucció interposada pel món industrial i part de món acadèmic, i no ser abonat

per cap multinacional ni cap organisme oficial, el seu ús s'ha estès a tot el món en deu anys. Tot i que inicialment s'havia dissenyat sota la pressió del repte de vèncer el Fortran en la màquina més ràpida que es coneixia, el CDC 6000, ara es troba fins en les màquines més petites que existeixen. La seva difusió és molt superior a la de PL/I, que és el llenguatge dissenyat i impulsat per IBM.

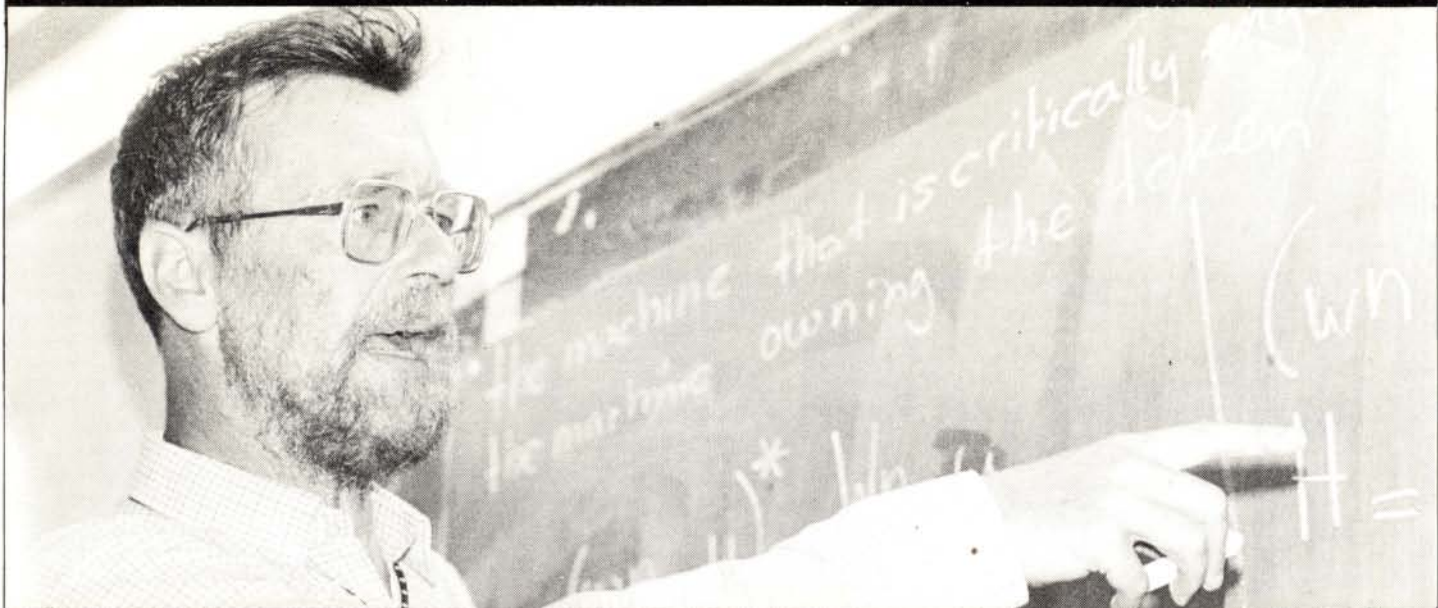
I aquesta proesa ha estat el producte de l'esforç d'un home sol. Això només té una explicació: la qualitat del producte.

**(ciència):** —Recentment, el Departament de Defensa dels Estats Units, que va ser el responsable de la difusió dels llenguatges Fortran i Cobol, ha posat en escena un nou llenguatge destinat a substituir-los, Ada, que conté, es diu, els conceptes moderns de programació. Quina opinió us mereix?

*(El professor, que no ha deixat de fumar en cap moment les seves cigarretes John Player Special, es posa a xuclar encara amb més força. El núvol de fum que es forma al seu entorn no impedeix entreveure, però, una certa lluïssor maligna als ulls enmig del seu rostre habitualment impàvid. I després d'un petit lapse de silenci contesta).*

**E.W. Dijkstra:** —Els requeriments originals d'Ada eren una barreja inextricable d'exigències sobre el llenguatge i sobre la seva implementació. Barrejar ambdós aspectes és un error greu que ve de molt antic i que ja ha conduït a desastres massa estrepitosos.





No fa gaires anys, per exemple, que el manual de Basic contenia el despropòsit de suggerir que òbviament, les variables més usades havien d'aparèixer aviat en el programa, ja que cada cop que apareixien l'interpret les havia de buscar seqüencialment sobre una llista que es construïa en ordre d'aparició.

El fet de tirar endavant el llenguatge representa el menyspreu més absolut envers la comunitat acadèmica. Inicialment es va intentar obtenir la col·laboració d'aquesta, però a la fi això no és gens clar. Tot el treball de N. Wirth, per exemple, ha sigut ignorat. Quan el seu equip va enviar l'informe que havia preparat, se'ls va fer reescriure de nou argüint que era massa negatiu.

Les raons polítiques, i no les tècniques, han sigut les que han determinat la seva versió final.

*(Aquí, el professor s'ha permès la vel·leïtat d'una petita extravagància: s'alça, fa lentament llargues passes al nostre entorn, amb les mans agafades a l'esquena. Inesperadament s'atura en sec. La immobilitat fa el silenci encara més llarg. De cop i volta es gira i continua.)*

El problema és que la complexitat es ven millor. La simplicitat, en canvi, és més difícil d'aconseguir i requereix una llarga educació per tal de poder apreciar-la. És altament reveladora la declaració feta per un general, que probablement no devia conèixer ni tan sols els llenguatges del Departament de Defensa, segons el qual era **obvi que la NATO no estava interesada** en un llenguatge tan artificialment simplificat com el Pascal. No

calia donar-hi cap raó: era obvi.

Originàriament hi va haver quatre propostes per al llenguatge. Tres d'americanes i una d'europea. Aquesta darrera va ser l'elegida. He sentit dir, i és creïble, que aquesta elecció va ser una decisió política conscient. Si no funcionava, els europeus seríem l'objecte del blasme. Com que, a més, Ada és un llenguatge inensenyable, amb molta probabilitat la comunitat acadèmica també seria objecte de blasme. Com a reacció s'obtindria el pragmatisme desitjat.

Les conseqüències de tot plegat seran, doncs, incòmodes, desagradables i no gens desitjables. Però sobretot absolutament injustes.

*(ciència): —Però moltes de les característiques d'Ada serien inconcebibles sense les vostres aportacions.*

**E.W. Dijkstra:** —Ara, a sobre, no me'n doni a mi la culpa.

*(ciència): —Perdó, no era la meva intenció. I pel que fa a la cinquena generació de computadors que preparen els japonesos, què en penseu?*

**E.W. Dijkstra:** —Es una situació menys greu que la d'Ada als Estats Units. Allò que pretenen els japonesos és trobar la pedra filosofal. Inicialment la meva predisposició va ser favorable. Després m'he espantat.

Cal reconèixer que els anys vuitanta han començat tràgicament. El Departament de Defensa dels Estats Units ens ha donat Ada i els japone-

sos la cinquena generació. Ada envienarà la indústria.

*(ciència): —Certament, posat en aquests termes, és patètic. Però ja que s'ha parlat de la competència entre les, diguem-ne, escoles europea i americana, quina és, al vostre criteri, la diferència entre ambdós estils?*

**E.W. Dijkstra:** —No hi ha, de fet, una gran diferència. Només un aspecte és important: el relatiu a la semàntica dels llenguatges, i això no és un problema específic de la programació, sinó de la informàtica en general. Els americans rarament tracten d'entendre un programa com una classe de computacions. El pensen estrictament en termes d'una implantació amb la qual estan familiaritzats. Per aquest motiu diuen encara bajanades com "L'Algol 60 és de ximplés". Tal va ser el cas del manual de Lisp (1962), que vaig ser incapaç de llegir: la seva definició era incompleta i s'intentava tapar els forats indicant com s'implementava. És un recurs abominable que té, a més, conseqüències immediates.

Un llenguatge ha de ser, evidentment, implementat. Però la decisió de com fer-ho depèn substancialment de les possibilitats i limitacions de les màquines disponibles, que poden ser molt dissemblants, i la definició del llenguatge ha de ser prèvia i independent de quina sigui aquesta implementació.

*(ciència): —Actualment els computadors*





*estan estesos pertot arreu i a l'abast de tothom. Sembla doncs que caldria que la gent n'adquirís un cert coneixement tan aviat com fos possible.*

**E.W. Dijkstra:** —Crec que és impossible.

**(ciència):** —Però sembla, si més no, que la programació es podria usar com a eina formativa per als alumnes de batxillerat, en el mateix sentit que s'usa la teoria de

*conjunts. Amb l'avantatge que el computador insta l'estudiant a pensar.*

**E.W. Dijkstra:** —El diccionari d'Oxford defineix la intuïció com a aprehensió immediata de l'intel·lecte sense raonament. Un ensenyament de la programació en aquest nivell, forçadament intuïtiu, pot ajudar ben poc a raonar.

Crec, sobretot, que els nens no estan dotats d'una capacitat d'abstracció suficient per programar. En concret, el mecanisme de raonament més usat

en programació és la inducció. Si la llarga experiència que hi ha en ensenyament de les matemàtiques indica que tal mecanisme no és assimilat pels estudiants en el batxillerat, no veig com se'ls podrà ensenyar a raonar rigorosament sobre programes. Tot al contrari, la proliferació dels microprocessadors produirà efectes deplorables en els hàbits de programació.

**(ciència):** —I pel que fa a Logo?





**E.W. Dijkstra:** —Penso que tot això és una bestiesa.

---

(ciència): —*I respecte a l'ensenyament assistit per computador?*

**E.W. Dijkstra:** —Només conec dos llibres que tractin aquest tema. En qualsevol cas, considero que el computador a l'escola és una moda passatgera produïda per la gran tecnologia i estimulada per les cases comercials.

Crec que les escoles farien una labor molt més profitosa si ensenyessin a presentar ordenadament les idees. És una greu deficiència de molts sistemes educatius no posar cap atenció a saber exposar les pròpies idees per escrit. Es depriment constatar que hi ha centres d'investigació amb especialistes a donar una forma final als textos que han de ser publicats. Aquest tipus de divisió del treball és absolutament insostenible. És espantós comprovar com el món sofreix la distinció entre forma i contingut.

A la Universitat hi ha un club on sovint em reuneixo amb uns deu investigadors en informàtica, tots ells joves acabats de graduar. Una de les nostres activitats principals és llegir-nos meticulosament allò que cada membre del club ha escrit, i expressar les nostres opinions sobre la forma del raonament matemàtic, la manera com s'han introduït les definicions, l'adequació dels identificadors i dels símbols usats, si les comes són al seu lloc, etc.

La conclusió més evident a la qual s'arriba indefectiblement és que la relació entre forma i contingut és inex-

tricable. Jo em sento molt frustrat quan algun deixeble em pregunta sobre si em refereixo a allò que ell ha dit o a la forma com ho ha dit. Hi ha cap diferència?

La majoria de les comunicacions científiques són molt pobrament escrites: són inconsistents, la nomenclatura és inadequada, les formalitzacions són barroques... En el cas d'un poema ningú no dubta que el poeta és responsable de cada mot. En la producció científica hauria de passar el mateix. Per això m'indigna la llibertat que a vegades es prenen els editors d'arreglar els meus textos sense permís, donant per descomptat que el material que els arriba és sempre tan pobrament escrit que poden canviar-lo impunement.

Són aquests aspectes essencials els que hauria de cuidar l'escola, en lloc de deixar-se seduir frívolament per les modes tecnològiques i pretendre ensenyar, a sobre malament, informàtica.

---

**A**poc a poc, la tarda ha anat caient i la conversa, que ha pres de mica en mica un to cada cop més informal, s'esqueixa i es cabdella imprevisiblement per acabar donant un tomb a aspectes de la vida personal. Proporciona una sensació molt agradable el tracte amb una persona que, ultra la seva lucidesa científica, s'ha revelat molt cultivada.

El professor Dijkstra confessa viure amb una esposa, tres fills, dos gossos i un piano de cua. Les edats dels fills són vint-i-quatre, vint-i-dos

i vint, i les dels gossos catorze i deu i mig. El més gran, que ha seguit una inclinació ben diferenciada del pare, però que aquest respecta completament, és afinador de pianos. La filla s'ha especialitzat en idiomes i el fill petit estudia matemàtiques. Cap d'ells no sembla afectat per la fama del pare. Fama que, d'altra banda, ell mateix declara no entendre. Comenta, per exemple, com el 1963, quan només havia escrit tres articles de set pàgines cadascun, l'Association for Computer Machinery li va enviar 100 dòlars per tal que assistís a una convenció a la qual ni tan sols se li demanava que parlés. No s'ha explicat mai com van conèixer la seva existència.

En aquesta ocasió, que va ser la primera vegada que va anar als Estats Units, es va quedar astorat davant l'orientació tan antropomòrfica que tenia la informàtica en aquest país. I atribueix l'èxit de la implementació del compilador d'Algol (1960) al relatiu isolament en el qual s'avia mogut, que li havia permès desenvolupar una aproximació completament diferent al tema.

El seu escriptor preferit és Dorothy L. Sayers, però a part de la literatura també li agrada llegir història. Recentment ha quedat fascinat pel llibre d'Emmanuel Le Roy Ladurie *Montaignon*, que consisteix en la reconstrucció de la vida quotidiana d'aquest poble occità a partir de la informació subministrada pels interrogatoris de la inquisició durant la repressió dels moviments herètics càtars (1294-1324). Ens ho comenta tot aspirant la seva cigarreta, que subjecta amb els dits de la mà esquerra, on llueix un elegant rellotge d'or. De corda, naturalment.