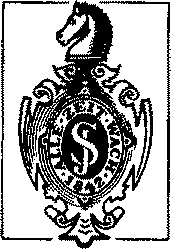
###### Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

92

Robin Milner

A Calculus of Communicating Systems



Springer-Verlag

Berlin Heidelberg NewYork 1980

Editorial Board

W. Brauer P. Brinch Hansen D. Gries C. Moler G. Seegmi.iller

J. Stoer N. Wirth

Author Robin Milner

University of Edinburgh Dept of Computer Science

」ames Clerk Maxwell Building

The King's Buildings Mayfield Road Edinburgh EH9 3JZ Great Britain

AMS Subject Classifications (1979): 68-02

CR Subject Classifications (1974): 4.30, 5.20, 5.22, 5.24

ISBN 3-540-10235·3 Springer-Verlag Berlin Heidelberg New York ISBN 0-387-10235-3 Springer-Verlag New York Heidelberg Berlin

Library of Congress Cataloging in Publication Data. Milner, Robin. A calculus of communicating systems. (Lecture notes in computer science; 92) Bibliography: p. Includes index. 1. Machine theory. 2. Formal languages. I. Title. 11. Series.

OA267.M53. 511.3 80-21068

This work is subject to copyright All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, r郅>r eductionby photocopying machine or similar means, and storage in data banks. Under§54 of the German Copyright Law where copies are made for other than private use, a fee is payable to the publisher, the amount of the fee to be determined by agreement with the publisher.

©by Springer-Verlag Berlin Heidelberg 1980 Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr. 2145/3140-543210

＿

This work was mainly done during my six-rronth appoin tn 巳 址 ， fran 沁 lQUS 七 1 979 to Januarv 1980, at the ca 环 过 er Science dePartmen 七 i n Aarhus Univer s i 七 y , Denmark. Although much of the ground work had been done pre吐 ousl y i 七 was mainly inr esponse 七o th e 扛 encour ag 包在 文止 （七o

咄 e the th eo 巧 nore a ccessibl e 五 d related to 芦 acti ce) , and to their infonred criticism, tha七 廿汜 mater i al rea 中 ed a s arewha七 c:oher en 七 fo nn. I am deeply gra 七 ef ul to them and th eir s 七 udents for all<:M.ing :ne to lecture once a week on wha 七 was, a 七 f ir s 七 ， a loosely connected se 七 of

迈 eas , and for th e 殴 l eaning envirol1!00 砒 扭 which I was able to pu 七

the ideas in order. 工 al so thank Edinburgh Uni vers i 七 y for award.ing :ne five m::inths s abba 七 i cal leave subsequently, which helped :ne to canplete the task in ar e 玉 ,enabl e time.

The calculus presented here grew out of work whi 由 was inspired by Dana Scott's theo 工 y of crnputa 已 on, though i 七 h 在 s since diverged in sare respects. 阰 every stage I have been influenced by Gordon Plotkin; even where I carmo 七 r 七 ace par 七 i.cular ideas to h 扛 n I have

been grea 七 l y illum.inated by our discussions and by his chance remarks, and without them the ou 七 cane would certa.inly be less than i 七 i s . I

沁 ul d also like to thank others with whan I have worked: 函 rge Milne,

with whan I worked out the Laws of Fl<:M Algebra; Ma: 七 th 匋 Hennessy , with whan the notion of abserva 已 on equivalence devel o 店 式 ； and Tony Hoa 迳 ， whose parallel work an different bu 七 s 七 ron gl y related ideas, expressed

.in his "Carmunicating Sequen七i al Processes 飞 has been a s七ran g s 七扛nul us .

Many people have given detailed and helpful criticisms of the manu­ script, and th us 叩 r oved its final fonn. In pc 吐 i cular I thank Michael Go志 n and David MacQueen, who wen七 thr ough it all in detail in a S已记.nar 砒 廿汜 Inf onnati on Sciences Insti 七ute , University of california; this

no 七 onl y e 习 ?C)Sed sane mistakes and obscurities bu 七 gave :ne m::ire confidence in the parts they didn' 七 er 让 i ci se .

Finally, I am very thankful to Dorothy McKie and G.ina Temple for their patience and skill .in the long and .involved task of typing.

1. Introduction

Purpose - Character - Related viork - Evol u 七 i on - Ou 七 l in e .

* 1. 酝 .im:m 七 in 9 on Nondetenni.nistic Machines 9

Tra出 ti onal equivalence of f ini 七e s ta 七e acceptors - Exper.im:mting upon accep 七 or s - Behaviour as a tree - Algebra of RSI's - Unobservable actions.

* 1. S沪chraniza七i on 19

Mu七ual e 习 文江irren七a 七i on - Ca 平沁S 让 i on,r es 七工i ct i on and relabelling - Extending the Algebra of STs - A s 扭 ipl e e:>亏 l e: bin 叩 S 妇 phor es - The ST Expansion Theorem.

* 1. A case s哼 in 叩 chroniza 七 i on and proof techniques 33

A scheduling problem - Building the scheduler as a Petri Ne 七 一

Observation equivalence - Proving the scheduler.

* 1. Case s七udi es in va l u 安 0 工ilnlmi ca 已 on 47

Revi 亏 7 - Passing values - An example: Data Flow - Deri va 已 ons -

An example: Zero searching.

s. 匈n七ax and Seman七i cs of ccs 65

I nr七 educt i on - s沪 七ax - Sel在江止.i cs by der i va 七i ons - Defining behaviour iden 七 H i er s - Sorts and programs - Direct equivalence of behavi ou 工programs - Congruence of behaviour programs - Congruence of behaviour expressions and the Expansion Theorem.

1. Camn.m.ication Trees (Cl's) as a 呻 1 of CCS 84

crs and the dynamic opera七i ons - crs and the s七a已 C opera七i ons -

Cl's def扭ed by recursion - 毕皿c actions and derivations of Cl's -

Strang a:\_ruivalence of Cl's - 政 扣 吐 i ty in the er 呻 1 - Surrm 江 y.

1. Cbservation equivalence and its properties 98

Revi 却 - Observa 七 i on equivalence in CCS 一 心 s ervati on congruence -

Laws of observa七i on congruence - Proof 七o 士miques - Proof of Theorem

7.7 - Further exercises.

VI

1. 运 pr oofs abo u 七 Data Structures 111

Introduction - Registers and ma:rories - Chaining opera 七 i ons -

Pus比沁邓氐 and queues.

1. Transl a七i on .into CCS 126

Discussion - The language P - Sorts and auxiliary def 扛 让 七 i ons -

Trans la 七 i on of P - 沁 女 过 ng procedures to P - Protection of resources.

1. Determina 匀 and Confluence

Discussion - Strong oonfluence - Canpos 让 e guards and the use of confluence - Strong determinacy: Confl uen 七 determina 七 e a 忑 － Proof in r:x::cs: the scheduler again - Observation oonfluence and determinacy.

1. conclusion

What has been achieved? - Is 忑 a pr 呼 anm 扭 g language? -

'Ihe question of fairness -'Ihe no已onof behaviour - Directions for further work.

平这如： Properties of ccngruence and 0'.I凶 valence.

138

158

166

妇ferences 169

工n 七工oduct i on

* 1. Purpose

酝 se notes presen 七 a calculus of concurrent sys 七 ens . 吐 e pres en 迳 已 on is partly infonnal, and a 迦 d at practice; we 畔 ol d the cal cul us 如 ough the medium of exarrples each of 血 ch illustrates firs 七 i ts expressive power, and second the techniques 如 i ch i 七 off ers for verifying properties of a system.

A useful calculus, of c 叩 砒 ing sys 七 年 i.s as of anything else, must have a high level of articulacy in a full sense of the word implying no 七 only ridmess in expression but also flexibility in rranipula 七 i on . I 七 s hould be r-ossible to describe existing systems, to specify and program 芷 田 sys tems , and 七o argue math 已花比 i ca ll y abou七 th 己 n, all withou七 l eavi ng the no七ati onal framework of the calculus.

These are d年回 血 g er 让 er i a , and 迂 may be irrp:)ssible to mee 七 them even for the full range of ccncurren 七 sys tems mich a 迳 the proper ccncem of a ccmputer s ci en已 s七， l e 七 al one for systems in general. Bu七 th e a 七七王耳丈

皿这 七 be made. We believe tha七 our calculus succ宅e 击 a七 l e 王式； to this exu平妇the s 郘 ie notations are used both in defining and in reasoning abou 七 syst 已 芯 ， and - as our exarrples will shCM - i 七 a ppears to be applicable no 七 onl y 切programs (e.g. operating systems or parts of th 己 ） but also to data s 七 ru e­

血 es and, 吐 a certain level of abs 扛 acti on, to haD 如 ar e systems. For

the l a 七ter ha, 汜 ver , we do no 七 cl 忒 .m to reach the detailed level a 七 协 让 ch the correct fl.ll'lctioning of a sys 七 em depends on 七 扛 吐 工 g considerations.

Apart fran articulacy, we 血 a 七 an underlying theory whose basis is a small well - kni 七 coll ectionof ideas and 如i ch jus七迁 i es th e m叩 i pul a已 ons of the calculus.'!his is as 扛 iportan 七 as generality - perhaps even nore

:importan七. Any theory will be superseded sooner or l a七er ; during its life,

understanding i 七 and assessing it are only r-oss:i.ble and worthwhile if i 七

is seen as a logical growth fran rather f 钩 basi c ass 叩 ti ons and concepts. We take this further in the next section, 泣 1erewe introduce our chosen conceptual basis.

One purpose of these notes is 切 provide material for a gradua 七 e course.

With this in mind (indeed, the notes 声 as a graduate course a 七 Aarhus

血 :vers i ty in Autumn 1979) we have tr i ed 切 f ind a good exposi 切 ry se: 扣 笠 ice,

and have anitted sane parts of the theo 巧 - which will appear in technical publications - in favour of case s 七 udi es . 吐 e 工 e are plenty of exercises,

2

and anyone 如 o bases a course on the no 七 es should be able to think of others; one pleasant fea 七 ur e of concur re n 七 sys tems is the wealth and variety of

small bu七 non- tr i vi al exarrples! We oould have included many rrore exarrples in the text, and thereby given grea 七 er evidence for the fairly wide applicability of the calculus; but, since our main a扛n is to pre sen七

it as a calculus, i 七 跌夭叮巳d a gooo. rule tha七 eve 可 exarrpl e program or system should be subjected to sane proof or 七o sane m玉让pul a七i on .

* 1. Character

OUr calculus if founded on two central ideas. The f irs 七 i s obse rv a 七 i on; we aim to 华 s cr ibe a concurrent system fully enough to detemine exactly

沁 at behaviour will be seen or experienced by an external observer. Thus the approach is thoroughly extensional; two systems are indistinguishable if we canno 七 te ll them apart wi th o u 七 pul l in g them apart. We therefore give a£annal defini 已 on of observation e< 不 过 val ence (in Chapter 7) and investigate its properties.

This by no 罕 ans prevents us fran studying the structure of systems. Eve巧r interesting ooncurrent system is bui迁 f ran independent agen七s which camnmicate, and synchronized oamrunication is our second central idea.

俅 regard a ccmm.micatian between two caq为呻 七 agents as an indivisible action of the canposite system, and the heart of our algebra of systems is ooncurren 七 s i已on, a binary opera已on which cx:mposes two inde­ penden七 agents , allowing them to ccmnunicate. I 七 i s as central for us

as sequential ex:叩 ipos i已 on is for s equen七i al progran:ming, and indeed subsuroos the l a 七 ter as a special case. Since for us a program or system description is jus 七 a te 卫 n of the calculus, the s 七 ru ct u 迳 of the program or system

(i ts 扛 1tension) is reflected in the struct哗 of the te 血 . OUr manipulations often consi s七 of 扛 ans f o 五让n g a 七艾:m, y 迳 l ding a tenn with differen七 in t en- sion but iden 七 i ca l behaviour (extension). Such 七 工 ans f o 五 哇 过 ans are familiar in sequential progran:ming, where the extension may jus 七 be a math ema 已 cal function (the "inpu 七 / out pu 七 behavi our") ; for ooncurrent systems however,

迁 seems clear tha 七 f imct i ons are inadequate as extensions.

These two central ideas are really one. For we s up 产 ；e tha 七 the only way to observe a system is to ccmnunicate with i 七 ， which makes the chserver

and system 七 oge th er a larger system. 吐 e other side of this co 江 is tha七 七o place two a 卫耳心 nen ts in carmunica已on (i.e. 七0 CCI"屯>0se them)

3

is jus 七 to le 七 th em observe each other. If observing and carmunica 七 ing are the s 郘 ， 让 fo ll OI 妇 th a 七 ooe canno 七 obs erv e a system wi tho u 七 i ts

par ti 过 pati on . 吐 e analogy with quan 七 um physics may or may no 七 be S\ 耳 汜 亡

豆 ci al , bu 七 the appr oa 中 is unifying and appears na 七 ur al .

We call the calculus 邸 (Cal cul us of Ccmnunica 七 ing Systems).'Ihe 扫江ms of CCS stand for behaviours (extensi a芷）of systems and are subject to equational 1 竺 . This gives us an algebra, and we are in agr eernen 七

with van Emde Boas and Janssen [EBJJ who argue tha 七 Fr.ege ' s principle of compositionality of 晖 aning requires an algebraic fram:: 奴 or k . Bu七 a:s is s立正双ha七 rrore than algebra; for exarrple, deriva已ves andder拉a已ons

of tenn.s play an 扭 portan 七 part 扛 1 describing the dynamics of behaviours.

The variety of systems which can be expressed and dis cussed 年 立 S is illustrated by the exarrpl es 扛1 the text: an agen七 for scheduling task perforroance by several other agents (Chapter 3);'data flcm'

改 :nputa ti ons and a concurrent nun 芦 i cal algorithm (Chap 七 er 4); :merory devices and data structures (Chapter 8); seman 已 c des cr i p 已 on of a parallel prograrrtning language (Chapter 9). In addition, G. Milne [Mln 3] 呻 ll ed andverified a peripheral hardware device - a cardreader - us 江 g an earlier ver廷on of the presen七 i deas .

After these remarks, the character of the calculus is bes 七 dis covered by a quick look through Chapters 1-4, ignoring 运 如i cal details. §0.5 (Ou 七 l ine ) may also help, bu 七 th e next two sections are no 七 es s en 已 al for

a quick appraisal.

* 1. 惩l ated Work

At pre s en 七， th e :rrost fully developed theor:y of concurrency is th a七of Petri and 区 s colleagues. (See for example C.A. Petri, "Introduction

to General Ne 七 'lheo 巧 " [ Pe 七 J, a 记 H. J . Genrich, K. Lautenbach, 瓦 s. Thiagarajan, "An OVervi 窃 of Ne 七 Theo 巧 " [GLT] .) I 七 is 扛屯为 rtan七 七o contr as 七 our calculus with Ne 七 Theo 工 y , in tenns of underlying roncepts.

For t妃t Theo巧 ， a (严 haps the) basic no已on is the roncurren匀

rel a已on OV!江 th e places (ro ndi 七i ons ) an d r七 ans i ti ons (events) of a

system; if two events (say) are in thisr ela 已 on, i 七 in di ca te s th a 七

they are causally independen七 and may occur in either order or si.mul­ taneously.'Ihis relation is conspicuously absen 七 i n our theor:y, a 七

4

l eas 七 as a basic no 七 i on . 如 en we a 兀 贮 o s e two agents i 七 is the s yn chroni一

zation of the 江 mu 七ual a 兀 rm.mica 七i ons which detennines the a兀p 忑 i te ; we tre a七 the ir in dependen 七 acti ons as occurring in 呻 i trar:y order bu 七 no 七simultaneously. 吐 e reason is that we as s 严 of our external ooserver tha七 he can make only one observa已on a七 a 七扛汜； this i.nplies that he

is blind to the possibility tha 七 th e system can support two obs erv a 已 ons

simultaneously, so this possibility is i rrelevan七 to the ex 扫:msi on of

the sys 涵 in our sense.'Ihis assumption is certainly open to (extensive!) debate, bu 七 gi ves ou 工 cal cul us a si.nplicity whi ch 沁 ul d be absen 七 o ther- wise. To answer the na 七 立 al oojection that it is unwieldy to consider all possibl e s这P 平笠坦 (in ter l ea vings) of a se 七 of causally independen七 events , we refer the reader to our case s 七 udi es , for exanple in Chapters 3 and 8, to sati s fy 出 msel f th a 七 our rrethods can avoid this unwieldiness alnost cxxrq;:>letely.

On the other hand, Ne 七 咋 eo r:y provides many strong anal y 七 i c techniques; we mus 七 j us ti fy the proposal of ano 廿 1er theo巩 吐e 纽 phasis in our calculus is upon synthesis and upon extension; alg 吵 !l'.'a appears to be a natural tool for expressing hON systans are bui止，and in shewing tha七 a s ysb 王n m至 七s i 七S specification we are demanding properties of its extension.'Ihe activity

of progranming - nore generally, of sys枉叩synthesis - falls na七ur al l y

into ccs, and 炬 be li eve our 夺 roa ch 七 o be nore articulate in this respect than Ne 七 'Iheo r:y, at l eas 七 on pre s en 七 evi dence . I 七 remains for us to develop analytic techniques 七o ma七ch those of .Ne七 吐 eo r:y, whose a中 i eve ­

men.ts will be a valuable guide.

As a bridge be切11een Net Theo工y and p:i:屯 .taunting languages for roncur­ rency, we should mention the early work of Karp and 阻 li er [如 on parallel program sen 妇 ta . This work bears a rel a 七ion to Ne 七 吐 eo r:y in yielding an analysis of properties.of roncurren 七 sys t ems , such as deadlock and liveness;

让 also cares closer to progranming (in the conventional sense), being a generalisation of the familiar notion of a sequen 七 i al flON chart.

Inr ecen七 p roposals for concurrent prograrrming languages there is a trend towards direct ccn兀n.mica 七i on be 七农夭 m o:np.:,nents or m:x:h让es , and away fr an 女 m 叫 过 cati on through shared registers or variables. Exanples are:

N. Wirth 叹DUI:AA language for 志 ul ar multiprograrrrning", [Wir];

5

P. Brinch Hansen "Distributed Processes; a roncurren七 p rograrrrning concept" , [Bri 2]; C 心 R. Hoare "Ccm:m.micating Sequential Processes", [Hoa 3]. Hoare's "rronitors" [Hoa 2] gave a discipline for the administration of shared resources in concurren七 pr o 乒 如 m扭 g. These papers have helped

切 ards understanding the art of concurrent prograrrrning. OUr calculus differs f:ran all of th 年 1 in two ways: f irs 七 ， 让 is no 七 in 七 迳 accepted sense an 扭 i,per ati ve language - there are no a::m 四 志 ， onl y expressions; second, 让 has evolved as part of a mathema 七 i cal study. In the author's vi€M i 七 i s hard 七o do math 己阻 七i cs with 土耳无 a七i ve languages, though one

may 竺 旦 math 哱 ti cs (or logic) to th 甸 to ge 七 a proof methodology, as in

the well-known "assertion" method or Hoare's axianatic method.

One of the main encumbrances to proof in 扭 严 ati ve languages is the presence of a 印 re- or- less gld:>al :merrory (the assignable variables) .'Ibis was recognized by Hoare in "Ca:mruni ca 七in g Sequen七i al Processes"; although CSP is 扭 ati ve Hoare avoids one aspect of gld:>al rtSIOry which makes cone; 丑 :rent prograrrs hard to analyse, by forbidding any m: 却匹 of a set of concurren 七 p rograms to alter the value of a variable m 平 七 i oned by another 邓 却 匹 . 'lhls signi fi can 七 s 七 ep brings CSP quite close to o 立 calculus, the rrore so because the treat:rrent of carmunication is s:imilar and expressed in s:imilar notation. Indeed, algoritluns can often be translated easily fran

one to the other, and i 七 si re as onable 七 o hope tha 七 a semantics and proof

theory for CSP can be developed fran CCS. Hoare, in his paper and roore recently, gives encouraging evidence for the expressiveness of CSP.

We 江 双 七um to 切 o :rrodels based on non-syndlronized carmunica七i on. One., with strong expressi ve 产 er , is H 窃 吐 七 七 ' s Actor Systems; ar ecen 七reference is [HAL]. Here the ccmnunication discipline is tha 七 each

message sen七 by an actor will, after finite 七扛妇， a立 i ve a七 i ts des七年 a已 on actor ; no structure over wai 七in g messages (e.g. r dering by send一七扛汜）

is imposed.'Ihls, together with the dynamic creation of actors, yields an interes七ing progra:rmrlng method. HCMever, 让 s eems to the author tha 七the fluidity of structure in the 呻 1, and the need to handle the collection of wai tin g 芷 芩 sages , poses difficulties for a tractable extensional theory.

Another non-synchronized m:: 洷 1 , deliberately less expressive, was f irs 七 s tudied by Kahn and reported by h:im and 攸 也 加 至 ［ 凡 QJ. Here the intercarmunication of agents is via unbounded buffers and queues, the

whole being detenninate. Problems have arisen in extending 迂 to non­ determinate systems, bu 七 many non-trivial algorithms find their best expression in this :rredium, and i 七 i s an example of appli ca 已 ve (i.e. no寸n平 ative) prograrrming which yields to extensional treat:nEnt by the semantic 七e chniques of Seo七七. M'.)reover, Wadge [Wad] has recently sh立 切 呻 le calculations can derronstrate the liveness of such systems.

6

A lucid canparative accoun七 of three approa中es - He.吐七七， Kahn /

MacQueen and Milner - is given in [ 叩 ］．

In Chapter 9 of these no 七 es we show how one 勾 咚 of concurrent language - where o::.::irrrnunica 七 i on is via shared variables - may be derived

£ran or expressed in tenns of CCS.'Ihis provides sane evidence that o 立calculus is rich in expression, but we certainly do not cl a 扛 n to be able to derive every language for concu:rrency.

A rather differen 七 s ty l e of pr se en 七 江 g a ro ncurr en 七 sys tan is

exe:nplified by the path expressions of Canpbell and Habennann [CaHJ• Here the active parts of the system are defined separa 七 e l y fran the ronstraints (e.g. the path expressions) which di cta 七 e h 叩 th ey mus 七synchronize. 坟 江 e recent work by Lauer, Shields and others - :ma.it 让 y at N窃 cas七l e - shows tha 七 七hi s model indeed yields to ma.七h ema七i ca l analysis. A very differen七 exampl e of this sepaar已onis the elegan七 work of Maggi ol o- Sche t 七止让 et al [ M曲 J; here the cons 扛 aints are

presen七ed nega七i vel y , by s七a 七扛巧 wha七 con j unct io ns of s七玩 es (of s epara 七e en 七 ag ents ) may 。ccur . This approach h 坴 an advantage for

systems whose 立 贮 nen ts are largely independen 七 (the authors call i 七

"loose coupling"), since then only f 钩 ronstraints need to be expressed.

血 s section has sha-m the surprising variety of possible tr eatn 氐 ts of concurrent systems. I 七 i s nothing like a canprehensive survE,芍 , and the author i s 玉厄氐 tha 七 扛屯ortan七 wor k has no七 been men七i oned, b u七 i 七

will serve to provide sane perspective on the work pre s et 让 .ed here.

* 1. Evol u已 on

'1hls work evolved from an a七七已平 七 七o 江 ea 七 acmrnmi.ca已 on math ema已－ cally. In Milner : "Pn 兀 ess es : a mathemati cal 呻 1 of o::npu七in g agents" [Mil l] a nod.el of interacting agents was constructed, us 年 g Seo 七 七 s·

theory of 改邓血 . 'Ihis was refined and 宇 却 平 ）re algebraic in G. Milne and Milner: "Concurren 七 Processes and their syntax" [MM]. A 七 th i s poin 七 we proposed no progranming language, bu 七 wer e abl e 七 o prove properties of defined concurren 七 behavi ours . For example, Milne in his Ph.D.'Ihesis "A math emati ca l 呻 1 of concurren 七 ca:nputa 已 on" [Mln] proved partial correctness of a piece of har严 e, a card-reader, bui 止f立 m four separa七e ccmpanents as de七过 l ed in its har 面 ar e des cr i p 七i on.

7

Our m:丈 e l at this s七age dr<:M upon Plotkin's and Smyth's Powe江 :danain constructions, [Plo 1, Smy] •. whidl extended Soo 七 七 ' s theory to admi 七non- 金 七enninism. Part of our algebra is s 七 udi ed in depth in [ 阻 l 2].

At this poin 七 th er e were two crucial developnents. Firs 七 - as we had hoped - our behaviour definitions looked considerably like programs, and the r es 钮 lbl ance was increased by merel y 扭p ro vin g no垣已on. The

re s ul 七， es sen 已 all y the language of CCS, is reported in [Mil 31 and

was partly prarpted by discussions with Hoare and Scott. (For ca 丐 让 e 七 e ness , two other papers [ 应 1 4,SJ by the author are included in the 罕 ference

li s七. Ea 中 gi ves a s li gh 已 y di ff er en 七 perspective fran [ 阻 1 3], and di ff eren 七 exampl es . ) The second develop: 仅 平 七 was to realise tha 七 七 he resulting language has many in 七 erpr eta ti ons ; and 廿邱 the POI 妃 r danain

呻 1, and variants of it, may no 七 be the oorr ec 七 ones . A cr i 七 er i on was needed, to reject the wrong in 七 erpr e 七 a 七 i ons . For this purpose, we 七 urn ed to abs 釭 飞 止 i on 釭 扣 i val en ce ; ·two behaviour expressions should have the

same in terp 迳 tati on in the m 幸 l iff in all ca 忒 exts they are indistinguish­ able by abs 叩 吐 i on.

工七 n CM 七u工ns OU 七 th a 七 a def ini 七i on of abserva七i on e 子止val ence (for whi中 admi七tedly the工e are a f 窃 al 七erna已ves) det e 卫面主 s a m文迨1 , up

七 o isarrorphism, and rcoreover yields algebraic laws whi 中 ar e of practical use in arguing abou 七 behahlours . We have strong hope for a se 七 of l 玉 ,;s wh扛::h are in s 哇 sens e ex:项 壮 ete ; in fact the laws given in Chapters 5

and 7 have been shown ccmplete for a s 乓 li fi ed class of fini 七 e ( 七 e rmina 七 ing ) behaviours. In this case, "ccmplete" :rreans tha 七 i f two behaviour expressions are obseJ:Vati an- equival en 七 in all co 砒 exts then they may be proved equal

by the 1 或 s ; this ccmpleteness is shewn in [HM] .

* 1. Ou七l ine

In Chap匡r 1 we 沮 s cus s infonnally the idea of experurenting on, or

observing, a nan-dete卫泊卫担已 c agen 七； this leads 七o the no 已 on of

synchronisation tree (ST) as the behaviour of an agen七 中 apter 2 dis­ cusses nutual experi:rtEnt, or ccmm.mication, between agents, and develops an algebra of STs. In Qi.apter 3 we do a small case-s 七 :udy {a scheduling system) and p正对e saiething abou七 迁 ， an 已 ci pa 七i ng the fonnal defini 过on of abserva 已 on equivalenre and its properties to be deal 七 wi th fully in 玉 pter 7.

8

玉 pter 4 enriches our ccmm.mica 已 ons - up to nCM they have been just syn吐江oni zati ons - to al10v,,1 the passing of values fran one agen七 七o anoth e工， and illustrates the greater express i 妇 p:::,;ver in 切 o nore examples; one is akin to Data FlCM, and the other is a concur:ren 七 a l gor i thm for finding a zero of a continuous function. 吐 e notion of der 拉 ati ve of a behaviour

is in 扛 这 uced, and used in the second example.

In 中 apter 5 we define o 忑 formall y , giving i 七 s dynamics in tenns

of de 过 vati ons (derivative sa:iuences) • 如 s yields our strong congruence relation, under which two p 立 中 ams are congruen 七 i f f they have es sen 已 all y the s畔 der i va七i ons , and we establish several 1玉忘 obeyedby the congruence. In Chapter 6 we pr esen 七 ccmm.micati on trees (CTs, a generali sa 已 on of STs)

as a 呻 1 which obeys these 1 或 s; this 呻 1 is no 七 necess ary for the further deve屯1 兀巴砒 ， bu七 :rtEan七 as an aid to understanding.

Chapter 7 is the core of the theory; observa 已 on equivalence is treated in depth, and fran i 七 we gain our main congruencer el a 吐 on, observation congruence, under which two programs are congru en 七 i f f they canno 七 be distinguished by abse 工 va 七 i on in any context. Having derived SCI! 它 pro perties of the congruence, we use them in Chapter 8 to prove the

correct behaviour of two further systems, both to do with data structures.

工 n Ch 司 ?ter S 9 and 10 we look a 七 sane derived Algebras. One 七akes the fonn of an imperative concurrent prograrrming language, with assi 中 m 印 七statements, "cobegin-coend" statements, and procedures. In effect, we

过 hCM to translate this language directly in 七 0 ccs. 吐 e other is a

restriction of CCS in wh血 de te nninacy is guaranteed, and we indicate

归 p 如 fs abou 七 s uch programs can be s 扭p l er than in 啦 general case.

Finally, in Chapter 11 we try to eval ua 七 e wha 七 h 租 been a 中 i eved , and 皿 ca 扫 directi ons for future :resear 中 ．

CHAP芦 1

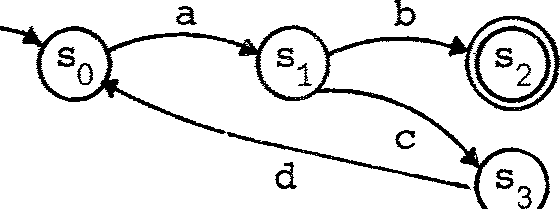
氐 in:ent in g on nondete五ninis七i c machines

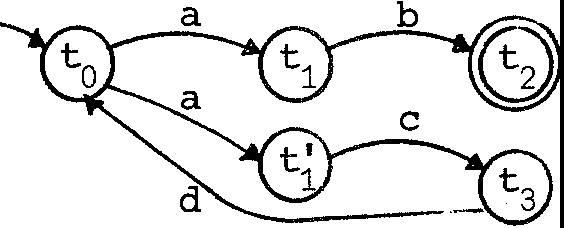
* 1. Tradi 七i onal e<干让val ence of finite state acceptors

Take a pair S,T of nondeterministic accep 七ors over the alphabet

E ={a,b,c,d} :

s





The accepting states of S and T are s2 and 七 2

T

respectively; in 巴王二

detenninistic accep七or s we can always make do, as he年 ， wi th a s in gl e ' 华 ad '

accepting state.

A standard argument th a 七 s and T are equivalen 七 ， 晖 an 让 1g th a 七 the y accep 七 th e saroo language (se 七 of strings), runs as foll a 炟 . Taking s. (resp

l.

七 .) to 正 presen 七 the language accepted starting fran state s. (resp 七 ），we

l. l. l.

ge 七 a se 七 of equations for S, and for T:

s0 = as1

七。= a 七 + at'

s1 =bs 2 +cs 3

2

七 = b 七

s 2 =£

。

;::: ds

七 I = ct

3

七2= E:

七3 = d七0

Here as usual + stands for 皿 on of languages, e: for the language {e:} rontaining only the anpty string, and we can think of the symbol a standing for a function over languages: as= a(s) = {acr; cr Es}

归 by simple substitution we deduce

s = a(be: + eds) • 0 0

By applying the distributive 1 玉 寸 a(s + s') =as+ as' we deduce

s0 = abe: + acds0 1

and we can go further, using a standard rule for solving such equations kn 吵

as Arden's rule, to ge 七

s。= (acd) \*主．

For T i 七 i s even simpler; we ge 七 dir ect l y (wi 七 hou 七 us in g di s r七 ib u 七 i vi 七 y )

七0 ＝忒王： + acd 七0

and the unique solvability of such equations tells us tha 七 S 。 ＝ 七 O , so S and T are equivalen 七 acce pto rs .

Bu七 2竺 th ey equivalen七， i n all useful senses?

* 1. 氐汗汜r 扭 en 七in g UfOn acce p七or s

Think differently abou七 an acce p 七or ove 工 {a , b , c ,d} . I 七 i s a black box, whose behaviour you wan 七 七o in ves 已 gate by asking i 七 to accept symbols one a 七 a t 扛 妇 . So each box has four bu 七 to ns , one for each symbol:

丿 丿

S b d T b d

There are four a七匀吐 c 色吓定工扛ren ts you can do, one for each symbol. Doing an a- exper:i.men七 an S (secretly in s 七ate s0 , bu 七 you don't k 兀 邓 tha 七 ） con-

sists in trying to press the a-bu 七 七 on , with two possible ou 七 cares in general:

1. Failure - the bu 七 to n is locked;

也） Success - the bu七七on is unlocked, and goes do 汛 1 (and secre 已 y a s ta te 扛 ans i ti an occurs).

工 n fact we canno 七 di s 己 nguish be 七 ween S and T , in 七 he ir i ni 已 a l s 七 ate s ,

by any single atanic exper:i.men 七; the a -exper扛汜砒 s ucceeds in each case, and the other three fail.

After a successful a-exper:i.men 七 on each machine, which may yield

丿丿

S b T b d

we may try another a 七 立 C exper irren 七 ， in our aim 七 o see if 也 e machines are equivalent or no 七. Clearly a b-experirren: 七 nail succeeds for S and fails

for T, though the other three exper irren 七 s fail to dis 七 i nguish them. After 扛ying the b-exper扭巳址， th en , can we conclude th a七 s and T are 巴主equivalen七？

No, because S's response to the a-exper 扭 巴 砒 could ha: 归 been di ff e ren 七(for all we kn 切 ） and locked the b-bu 七 to n , while T's response could have been di ff er en七 (for all we kn< 邓 - and i 七 co ul d indeed!) and unlocl 又 ed the

1. bu 七七on . FollCMing this argument further, we may feel forced 七 o a dmi 七 th at 王 f ini te 年 un 七 of experirren 七 co ul d pr ove 七 o us th a 七 S and T are, or are no七， equivalen已

Bu 七 s uppcse

* 1. 工七 is the weather a 七 any manen 七 -which dete:rmines 七 he choice of transition (in case of ambiguity, e.g. T at 七。under an

a-exper:i.rrent) ;

(ii ) 哗 weather has only finitely many states - a 七 l eas 七 as far as choice-resolution is ooncerned;

(iii) We can control the weather .

For s年 mach in e s these ass 吓 ti ons are no 七 so outrageous; for ex 叩 le, one of two pulses may always arrive fi rs 七 wi thin a certain tempera 七 ure range, and outside 血 s range the other may always arrive fir s 七. (A 七 th e boundary of

the range we have the well-kna;m gli 七 ch problem, which we shall ignore here.)

欧 ，by conducting an a-experirrent an S and T under 旦 呈 weath er con­ ditions (always in their start states, -which we have to assume are r eoove 亡able) , we can find tha 七 S ' s b-button is always unlocked, bu 七 tha t T's

b- bu 七 to n is sa 汜 tirres locked, and we can conclude tha 七 廿 1e machines are no 七

缸 val en 七

工 s this sense of equivalence, in which S and T are no 七 equivalen 七 ， a meaningful one? We shall find tha 七 we can make i 七 pre ci se and shall adop 七迂- partly because i 七 yi e l ds a nice theory, partly because i 七 i s a f in e 工(smaller) equivalencer el a已on than the s七平 dar d one (which we can always

recover by introducing the dsi tr 止uti ve l 或 us ed in§1.1), bu七 rrore for the

follCMing reason. Imagine th a七 the b- bu 七七ons on s and T are hidden. 'Ihen in all weathers every successful experiment upon S unlocks saoo visible bu 七 to n:

S (with b hidden) is no 七 deadl ock ab l e

血 l e in s 哱 weather s , and after sare exper扭 ts , all of T's visible bu 七 to ns will be locked:

T (with b 区 dden) is dead.lockable.

除 wi sh to think of a no 江 妇 terministi c choice in such machines as being resolved irreversibly, at a particular m:roent, by inf o 五 花 1ti on flrMing into the system fran an unseen source; if a deadlock can thus arise in one machine

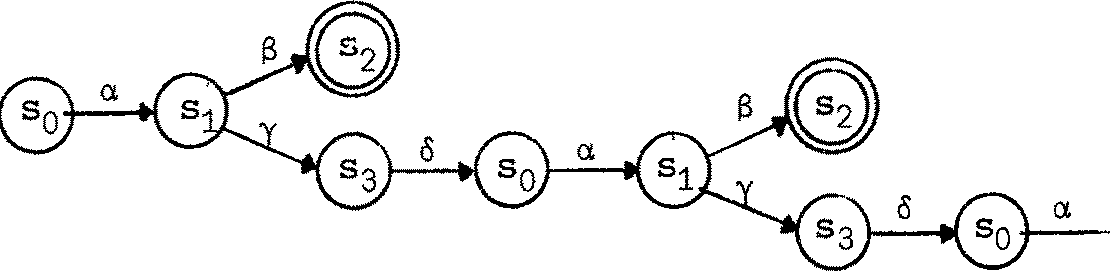
bu 七 not in another, we do no 七 r ega 工 d them as behaviourally equivalent.

* 1. Behaviour as a tree

Because we reject the distributive 1司 a(x + y) = ax + ay , we can no longer take languages (sets of strings) as the behaviours of our machines. We pr ooeed 七o an al te ma七i ve . Fran na,, on we will use NIL instead of e:

to stand for a behaviour which can do nothing (= admits no e} 吓 :er imen 七 ） ； we shall also use Greek l e七t ers for o u工 symbol s - i.e. n玉芷!S of bu七to ns - so you should consider a.,S,y,o as replacerrents for a,b,c,d in our s 乓 le exanple.

Fi rs 七， 七忒(8 the 七工ans i t i on graph for S and unfold i 七 into a 七 工 ee with states as node labels and symbols as arc labels:



Because state names are pre sen 七 we have los 七 no inf onna 已 on; the s ta te r 七 ans ­

让 i on gralil can be recovered fran such a 七五辛. But the exper:irrenter carmo七see the state - he canonly see the transitions.'Ihls leads us to drop the node labels, and take the infinite tree

．二．三＿＿

as the behaviour of S• ----. 0 拿 a, - -

I

Definition A 罕 i s a irerril: 汜 r of a given (fixed) label se 七 A •

We are using a. , f3,y,• • to stand for labels. ('.Ihe use of th e 沁 r d ' l abe l ' i n place of'symbol'will be further rrotivated later.)

I Definition A sort 1.s a subset of A •

We shall usually use L,M,N, •. to stand for sorts. We shall also of 七 en use the word艺立 in place of 'machine ' or ' accep七or ' , so

, S is an acceptor over the al phabe七 i::'

beo 正 S

I

'S is an agent of sort L'.

Definition A Rigi d - = 巳 on Tree I 每 ） of sort L i,; a rooted, unordered, finitely branching tree each of whose arcs is labelled by a

m叩 iber of L •

吐 us the tree in the l as 七 di agr am is an RST of sort {a, B, y, o} • {It is also an 邸 T of any larger sort.)

陷y 'r igid'? Because i 七 i s the behaviour of a rigid agen七 - one which

can make no transition excep 七 th at co rre spon ding 迈 an atanic experirren 七. v祀

s ha 耳 SCXl!l mee 七 o 廿 汜r transitions.

伽y ' syndlronizatian' ? Because we shall later see比 the camn.mication of two agents can be represented in fanning their join 七 tr ee fran their separate trees. Then the join 七 tr ee wi ll 空 在 be rigid, in general, since intercamn.mication between a::mponen 七 agen 七s i s no 七 d:ls ervabl e•

Notice tha 七 f ini te 郎 Ts can be represented as expressions:

is a (SNIL + yNIL)

人旮

is 畔 NIL + ayNIL

and usually there is more than one natural expression:

is aNIL + (BNIL + yNIL) , or (aNIL + BN耳） + yNIL .

乃

Indeed, 十 is both ccmnutative and associative, since we dec lar ed 芘 Ts to be unordered trees - and NIL is easily seen to be a zero for surrmation. To justify these ranarks we 江 def ine the algebra of RSTs.

1. 4 Algebra of RSTs

Ignoring sorts for a no: 年 七 ， we have an el ei 年 七 a 工 y algebra over RSTs, whose oper a已 ons a年：

NIL (nullary operation) NIL is the tree•

+ (binary operation)

A心工 乙沁

s the tree ( i den 七 i fy r oo 七 s )

1

入 ( unary opera七i on , for e ach 入 E A)

6

入 I l is the tree

加 y obey the fo ll 吵 g 1 称 s , as you can easily see:

Associ a七i vi 七y Carmu七a 七i vi ty Nullity

x + (y + z) = (x + y) + z x+y=y+x

x+NIL=x

In fact, these 1 称 s are ccmple 七 e : any true equa 七 i on between RST expressions

can be deduced fran them.

If we consider sorts, andlet 汜T be the set of RSTs of sort L,

L

then NIL is of sort L for any L :

NIL E 邸 T .

L

Further, + takes trees of sort L,M r es pE 汜 旦 ve l y to a

+ E 邸 T X RST RST

L M LuM'

r 七 ee of sort LuM :

and 入 t ake s a tree of sort L 七 o a 扛 ee of sort Lu{ 入 ｝ ：

入 E RST + RST .

L Lu 伈｝

We shall usually fo巧砒 abou七 s orts for the pr esen 七， bu 七 ther e are times

later when they will be essential.

Consider nc::M solving recursive equa七土ons over RSTs. We wish the equ­ ations for our agen七 S of§1.1

s 0 = as 1

s 2 = NIL

S1 = i3S2 + "(S 3

s 3 = 6s0

to def 江e the (扛正扛让七e ) behaviour of S as an RST of sort {a, 13,丫，,s} •

'Ihls set of a平坦已ons has a uni 平1e sol u已 on for the variables s , .. ,s ;

0 3

you can see this by the fact tha 七 th e en t ir e r 七 ee can be deve l op 玉 i top-down

七 o any depth:

S。＝ 入＝｀ ＝ … and so oo.

Warning. No七 ev 可 郔 of recursive equations has a unique sol u 已 on; consider the simple equation

s = s

which is satisfied by any 汜 T (or anything else, for that ma 七 t er ) .

Again, sane sets of equations define no 芯 T a 七 al l . Consider the equa已on s = s+aNIL;

a solution沁uld have to be infinitely branching a七 th e r oo 七. Even if we allailed infinitely branching RSTs, so th a 七

S。=y 分 ... + 00

would be a solution, 让 woul d no 七 be unique since s 。 + 七 w:>ul d also be a solution for any 七 . We defer this problem to Cha p 七 er 5.

．酝 er ci s e 1.1 Can you f:ind a condition on a set of equa 七 i ons

(with RST expressions involving sO , •. ,sn on the r i gh 亡 hand sides)

0 1.

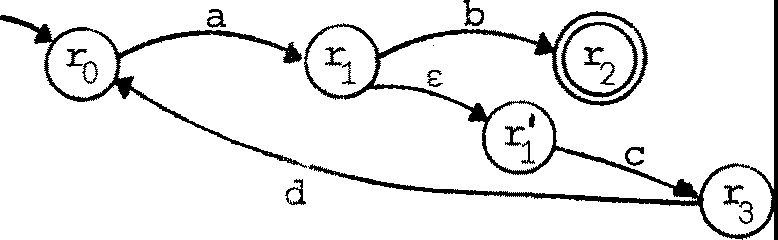
n

s s s

which ensures tha 七 i 七 r:x::,ss es s es a unique sol u 已 on in RSTs?

(H扛止： consider cycles of c: 一 r 七 ans i 七 i ons in transiti.on graphs.)

1.5 Unobservable actions

Under the conventional defi ni 已 on, a nondeterministic accep 七 or may have tr ansi已ons labelled by£, t enull s七r-in g . Consider R , a modi­ fi ca七i on of our s of§1.1 (reverting briefl y 七 o Ranan alphabet) :

R

('!he loop fanned by the a-transition is irr elevan七 to our ca:nparison.)

In the conventional sense, R and S are equival en 七. But wha 七 does the e:-transition mean, in our 叩r e mechanistic in枉江preta ti on? 工七 means tha 七R in state r 1 止 e . af 七 e 工 th e a- bu七ton h 玉; been pressed) 巴致 吐 any 七扛妇

叩 e silently to state r1 , andtha 七 if a b- experimen七 i s never a七temp七ed

让 旦； do so.

Thus, if we a七tenp七 a b- experimen 七 on R, after the successful a­ experimen七， th er e are sane wea th e 工 condi ti ons in which we find the b- button pennanently locked; if on the o廿飞江 hand we a七七己丐亢 a c-exper.iment (after the a-exper.imen 七 ） we shall in 旦 斗 weath er oonditions find the

c-bu 七 to n E!Vl 平 t uall y m1locked {eventually, because although R may 七 或 e a

li 七七l e 七乓 t o decide on its e: 一 r 七 ans i ti on , 让 wi ll do so since no b­

E-=-

experimen七 i s a 七七五pted) .

蓝 er cis e 1. 2 use this as the basis of an argumen 七 tha 七 no 严 of R, S

七 . A ru,or=s basis for the- will be given

later.

Letus return to our Greek al r:h忒汜七， and ask hOii we should wr迂e the equations specifying R's behaviour. 殴 i/e choose the symbol 1: in place of e: (to avoid confusion with the null string), and use 让 ； as a n 窃 una 工 y opera 已 on upon behaviours. The equations detennini.ng the behaviours

r0, •• ,r3 are:

r 。= c,,r

r = Sr + 立＇ r'= Yr3

r = N耳 r3 =or 。

We are assuming th a 七 T 生人 (th e fixed label se 七 ）．

I

Definition A 三 罕 (ST) of sort L is a =ted, 叩 nlered,

fini迳lybranclrlng 七迳 e each of whose arcs is labelled by a m臼让汜r of

Lu{,:} ,

吐 us a 丑 理 ST (an RST) is jus 七 an ST with no arcs labelled , ; i七 i s

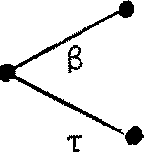
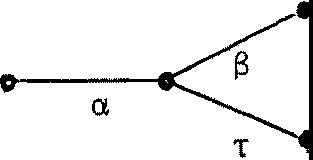
the behaviour of an agen: 七 whi ch can make no s il en 七 七 工 ans i 七 i ons .

Since we are taking the ur 沮 巧 , operation, over STs to be given by

,,A) . 大

袒 can of course deduce the ST-behaviour of R • I 七 i s

y . O .



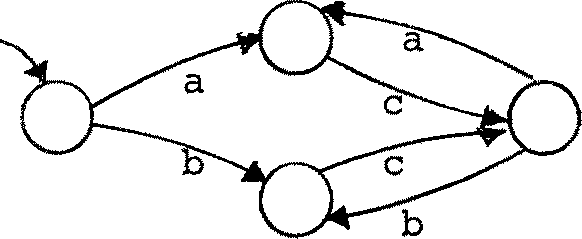
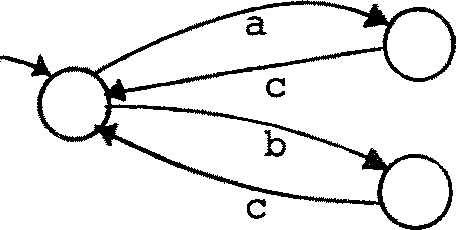
a

8

丫

CL . - -

STs are a s 扭 l e and useful notion of behaviour. They are jus 七 th e unfoldings of behaviour equations, which in turn foll 叩 directl y frcm transition graphs. Of oourse in this way di ff er en 七 r 七 ans i ti on graphs can yield the same ST, frcm which we can be certain tha 七 th ey are indistinguish­ able by exper扭en七．

政 er ci se 1.3 Convince yourself tha 七 th e r 七 ans i ti on graphs

have the same unfolding.

HCMever, di f f eren七 S Ts (o工 tr ans i 已 on grarhs yielding diff eren七 S Ts) may be indistinguishable by exper irren 七 ． 血 s i s true even for RSTsi consider the s 扭 le pair

产\

．

．

a

穸 tch of which admits a single a.-expE 五 :i.men 七 and then nothing else.

But i 七 i s e:ven rrore 七 rue in the case of unobservable actions. 压 ter we shall s七udy an equivalencer el a七i on , obs erva七i on equivalence, over STs, which can (for finite STs) be axiauatized by a fini 七 e se 七 of equa 已 ons added to those given in§1.4 above. To ge 七 a foretaste of the equivalence consider the foll 叩 ing exercise.

Exercise 1. 4 Examine each of the follONing pairs of simple STs and try to decide by info:rmal argumen 七 ， as in 氐 er ci se 1. 2 above, which a 迳 ooserv a 已 on equivalent (i.e. indis七inguish址让e by experiinen七）• You mayr eas on 忒 社 y ronclude tha 七 fo ur pairs are equivalen七， or th a 七 s ix pairs are equivalen七，

bu 七 you should also find tha 七 th e notion of eqi. 让 val ence is not ye 七 pr ec i se . 'Ihe poin 七 of this exercise is tha 七 i 七 i s not 江 i vi al to capture our infonnal arguments by a precise notion.

\b

卢

｀

凶

li

lv

u

w

位

8

，

/

＼

, I '

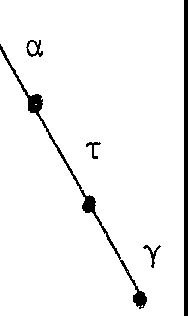
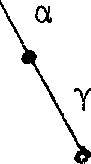
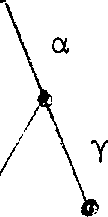
i

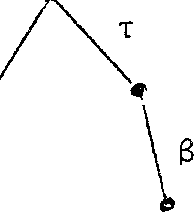
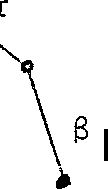
. 1

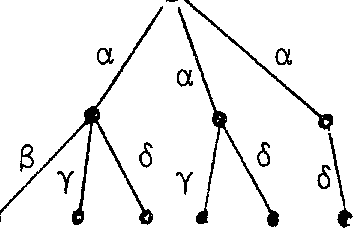
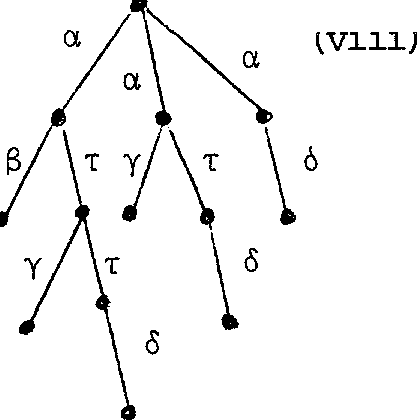
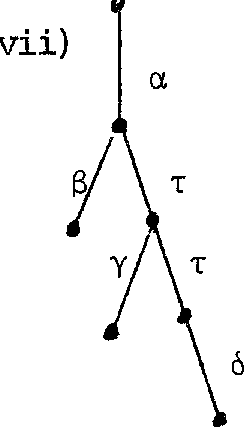
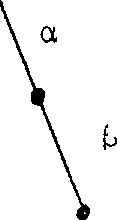
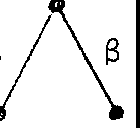
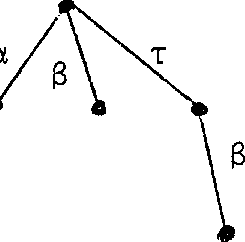
\ ＇

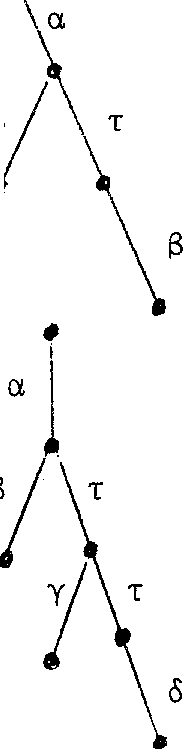
,v,

I





Can you think of sane equational axians of observation equivalence?

II

Chap 七 er 2

S沪chroni zati on

* 1. Mutual exper:imanta 已 on

吐 e success of an a-experiment enables the machine to proceed (to

offer further experiments) ; 迂 al so all芯t eobserver to proceed (七o

at 七己np七 further experiments). This suggests an 中 vious syrrrretry; we

would like to represent the cbserver as a machine, then to represen 七 th e 女 s i te observer/machine as a machine, then to understand h 叩 兰machine behaves for a n 窃 observer .

How should two ma 中 ine s 皿 er act ?

CJ, r;

丿s' & T 三

We mus 七 s ay whi 中 experirrents offered by S may canbine with or (carplerren 七 ） 叩 ch experin:ents of T to yield an interaction. Rather than set up a

label correspondence (e.g. a 七 f;I 0七 n) for each ma 中 年 e canbination, we 扭 tr aduce a 1 让 tl e s 七 ruct u 工 e on our label se 七 A.

We assure a fixed se 七na 玉 ·

of names. We use a, 13, y, • • • to stand for

We assume a set of oo- 平 年 s , 主 s join 七 f ran and in bijection

＾

＾

＾

with i 七；

the bijection is C) :

ct( 拉＿） ct ( E b.)

and we call a. the en- 垣 哇 of

Cl.• Using (-) also for the inverse bijection,

we have

＝ct = ct.

Nc1J1 we as s 哱 A= /:i u /:i

to be＿our se七 of labels.

We shall use

to range over A. We call

入

and

入 改 邓 pl errentazy labels.

'Ihe function () is 叩 a bijection over A We extend i 七 七 o subsets

入

of A;

in particular for any 竺 仁 L,

－L－= {入；

入EL} .

We shal l 攻 tines need the function narne(a} = narre (;;) = a

which we extend to sorts by defining

正 (L ) = { nama (入）； 入EL}.

Now consider the pair of machines

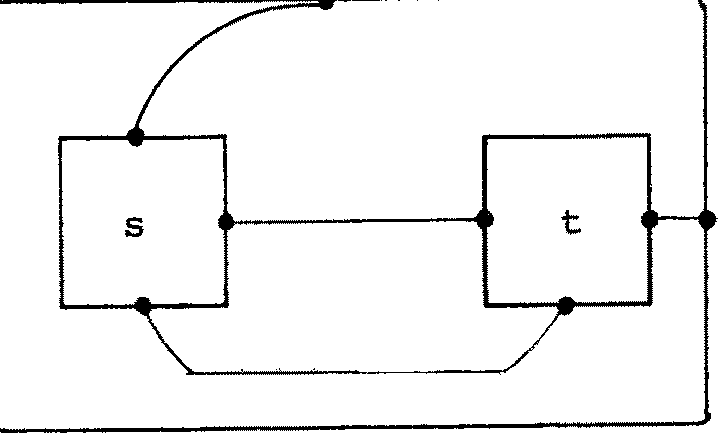
丿

丿

S: 位，8, y} T: {8,y,o}

吐 e na 七 ur al candidate, perllaps, for the cc:mbined niachine S 11 T nay be

pictured thus:

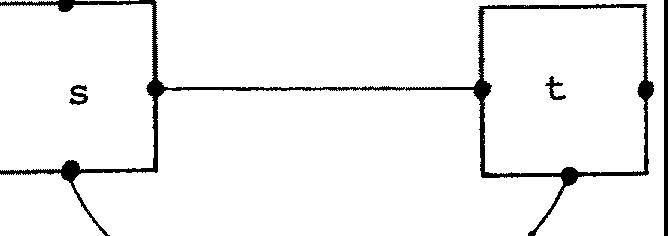
a

父

or:

6

a



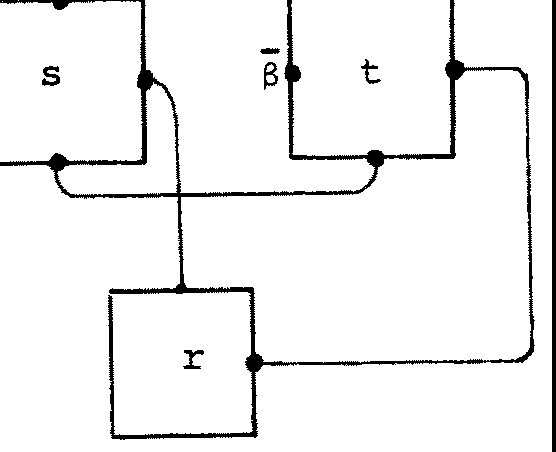
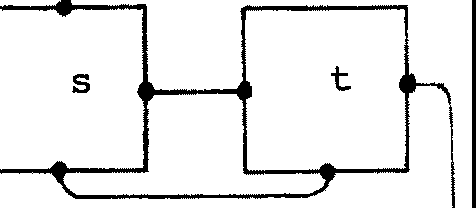
'Ihe in tui 七 io n is tha 七 0 耳 lar 四 1 tary ports, one in each machine, are linked and hidden ( labels 迳 rroved ) , since these links represen七 巴巴王吐observation, while other ports still support external observation.

Bu 七 under this schare there are two disadvantages. Fir s 七 ， cons i der

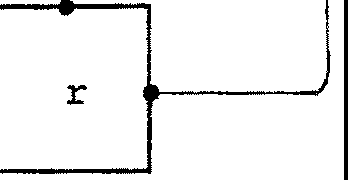
勹

R: {S, 初

We can form R II [SII T) and (Rli S) !I T:

CL 0.

6

each of sort 伍，初 bu 七 cl ear l y di f f eren七 S' s offers of 13--exper.irrents are observed by T in the fir s 七 cas e , but by R in the second case. So II is no 七 as s oc i a 七 i ve .

Second, 让 i s useful 七 o allCM th.a: 七 S's 13 玉于汜r .im:m七-o ff e 洷

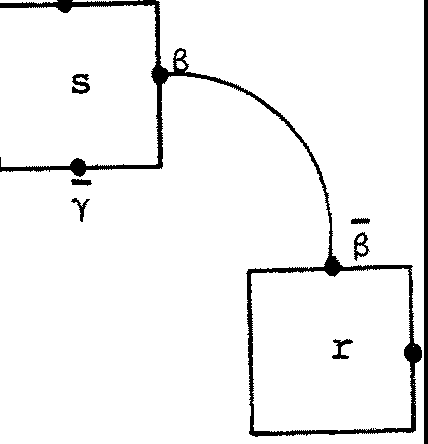
(or 13-capabilities as we shall s 哱 七 imes call th 卸 ）may be d 忑 erved by either R or T (th.a 七 is , each 13- eJ 屯 汜r .irrent on S may be done by either R or T, bu 七 no 七 bo th ) ; this m 咏 e s S into a resource shared by R and T.

吐 e solution is to factor caobinatian into twoseparate operations: one to 兰主 rx,rts , the other to 过生 th 王 We shall use the 沁 r d

叩 si ti on for the first of these operations, and 廿1e seoond we shall call restriction.

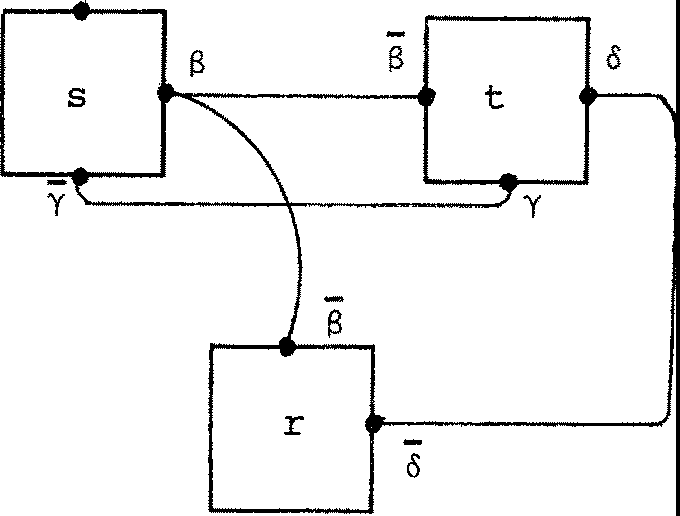
* 1. C叩 s i ti on , restriction andrelabelling-

The canposite Rls of our 切 J machines R and s may be pictured

C(

叩 le for (RIS) IT 袒 ge t

6

Cl

'Iha 七 i s , for each 入， in foJmlin＿g ujV we link e伲 ry 扣rt labell ed 入

in U to every p::>rt l abe ll ed 入 in v.

re-

ht。

yb

fl na e

0 七

y

ecco

g

nacna

d s

i

na

n

,

sa-

s

lT eru

Istc

I(

R

』

los

2

**e**sSiipm

Before def ining 改 兀 贮 :osi 七 i on of behaviours, 1 砒 us look a 七 two 0 七 he 工

i

d

sru.evi

a

i

ssa

o

eruiti

二a

P1，

n

o

.

}

i

a a

g

三

el

ev

i

a

七

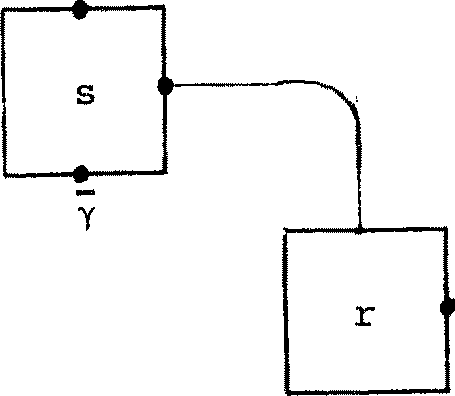
a

operations on pictures.

For each aE6., we define a 贮stf ixed restriction opera七i on \a,

which on pictures jus 七 ne ans "hide the p:lrts labelled a or 于 ， i . e . it drops the labels a and a 仓 丈 m pictures, thus reducing their sort.

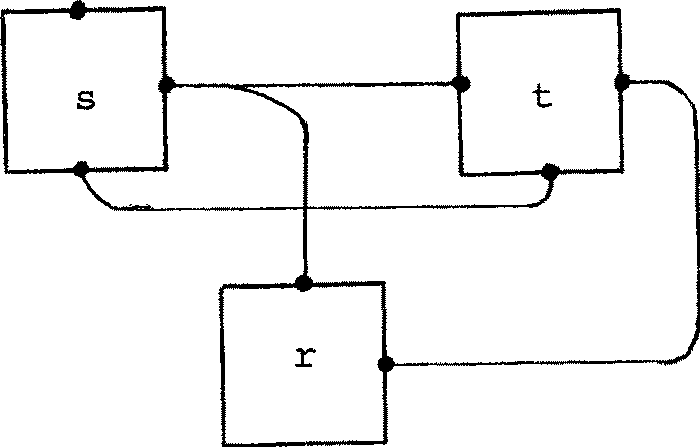
(l

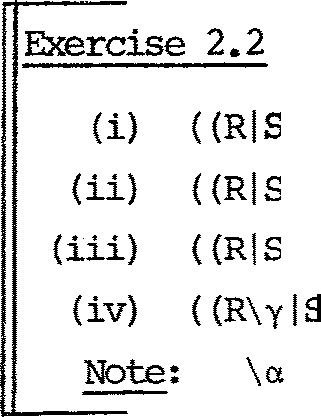
(RjS)\13

(= RII S)

6

((RIS) IT)\B\y\o

Cl

懂 i ch of the f oll 忒 .ing are i den 七 i cal as pictures?

) IT)\8\y\o (v) (RI (SIT)\o)\8\y

) \BIT)\y\o (过） (RI (SIT)\y)\S\o

）＼汁T) \ S\ o (vii) ((RIT)\olS)\S\y

) IT) \8\ o (viii) ((RI T) \o IS\o) \S\ y

binds tighter than I , so th a 七 U! V\a 哇 ans UI (V\a).

Besides its use with canpa过七i on , the r es 七r i ct i on operation by i七s e lf oorresponds to a s 乓 le, rather ooncrete, action:- tha 七 of hiding or 'internalising'certain ports of a machine. Compar e 七 he remarks on hiding the b-bu 七 t ons of two machines, a 七 th e end of§1.2.

Note th a 七 we can define s IIT, where S:L and T:M, by

S II T = (S T) \a ••• \a n where {a , .. o ,a n} = llil!!ES (LnM) •

We shall henceforth abandon the use of upper case le 七 t ers for machines. 'Ihere is a fine di 吐 incti on between the ideas of (i) a machine which may rrove through states bu 七 remains the same machine (a physical no 已 on) and

（ 益 ） a machine-state pair, i.e. a w 可 of specifying a behaviour with a

def.ini七e s 七a 迂 (a ItPre math ema 七 i ca l no 七 i on , exarplified by 七 he nonnal definition of F.inite- s 七 a te Acceptor as consisting of as ta 七 e s e 七 ， a tr ans 止已 on relation, a set of accep七江 g s 七ate s 凸 a s 七a 迂 S 七ate ) o Our lCMer case le七ters co rr es pond 七o the latter idea - indeed, they denote the spec 七fied behaviours (here as S'I's), and i 七 is these which are the dcroain of our algebra; we shall soon see wha 七 r js e 七 c . mean as behavi ours 。

We also have another use for upper case letters; we say that S:L M (where L,M are sorts) is a rel abell 扭 9 franL to M if

(i ) 迂 i s a bijection;

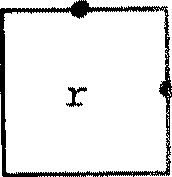
* 1. i 七r espects cx:xapl ernen七S

(i . e 。 S (a ) = S(a) for a, aEL) 。

We define the 区）stfixedr elabell 囥 哼 ati on 岱J, over (pictures of)

machines of sort L, ass 乓 ly 年 placing each l abel 尪L by S( 入）． 'Ihus for r, 七 as above we have

叫t: = 百三



Y

5

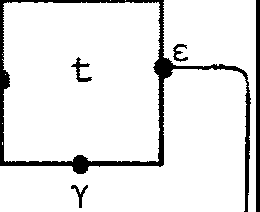
-B

and S: 伐-，y ,o ,o-} + { o, 丫1 £ , -£} , given by

或） = o, S(y) = y, S(o)=t:, 或 ）＝；

is a relabelling; we then have

r< l 七）[SJ =



丿

6

We shall use oonvenien七 忒土江的 i a七i ons in wri 七止巧 r e l abe ll in gs exp li ci 七l y.

吐 US

入/ 气＇．．，．入i 节 or 入卢...入n /Cl凸 Cln

{where et , • o • ,etn are distinct names, and 入 ，…，入n are labels with

distinct names) stands for the relabelling S:L + M given by

1. S(eti) = 入配 if O,EL

l.

1. S( 句 ＝ 入i if Cl 产
2. s(入）＝入 if name ( 山 FO , ••• ,etn

provided that the function so defined 兰 ! a relabelling. So in place of

(r l 七 ）(SJ above, we write

{r l 七）[o/S, e:/.SJ or (r l 七)[oe:/a.sJ.

妇 we see the 1 硒 of the Fl 氓 Algebra (1硒 for the 勾 严 i ti on, Restriction and Relabelling operations) in Theorem 5.5, 炬 shall see that 妇 志 ll ing distributes over 叩 s i ti on, so tha 七 we have

（平 ）［加，e:/ oJ = r[6几，e:/oJ I 七[5几，e:/.S]

(as you can readily check) - even though in strict f 叨 亚 uity o/a, e:/o stands for a differen 七 re l abe ll ing in ea 中 cas e , because r, 七 and r 忤possess dif fer en 七 so rts .

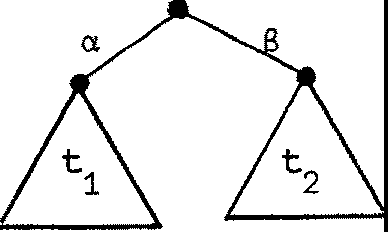
1. 3 ExtenclincJ the Alqebra of s 沪 1chronizati on Trees

We mus 七 nOi/ add our three n 窃 opera 已 ons to the algebra of STs, using intuition abou 七 th e operational meaning of these trees. In future we continue to use 入 七 or ange o 欢江 A, and useµ,v to range over Auh}.

Carq;x:>Sition I :ST L

Consider two STs

xST M +ST LuM

u=

｀

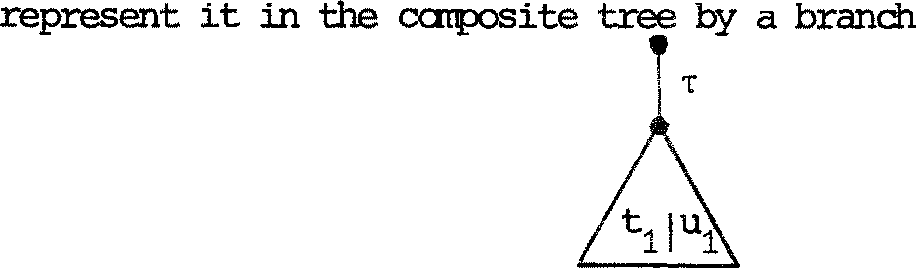
七

For their oc 邓产 i te , four actions are possible. 切 u admits an a-experirren 七

(because 七 doe s ) , so one branch of 七lu will be

Li

This branch represents independen七 actionbyonea:mponeI让，and similar

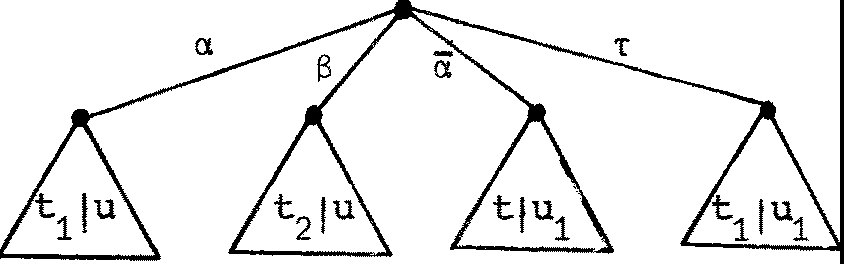
branches exist for a 13-exper乓 n七 on 七 and an a 气 这 户 辽 'im:毗 on u. of these three branches represents interaction be也Teen 七 and u; there is a possible inter act 沁 n, since u's a-offer a:mpl arents 廿s a-offer. Since this action is inter.nal (no 七 observable) we use

T

Pu 七 七 in g all the branches together yields

None

but and



Naiv CC兀I\_POS 土七i on of 七 and u has been defined in tenns of cai:p:,sition of their sons1 clearly this 罕 unts 七 o a re cursi 伲 def ini 七 i on of I

沁 r e precisely, sinre every tree may be written in the fonn

七= Lµ. 七．

1 :5i 细 1 1

， µ. EAU{-r}

l.

(wi th 咋:() i f 七 = N耳），we may define o:::rtpos过七i on as follws:

Def ini 已 on If

七= i壮气

and

u = .u., then

. J J

Iv

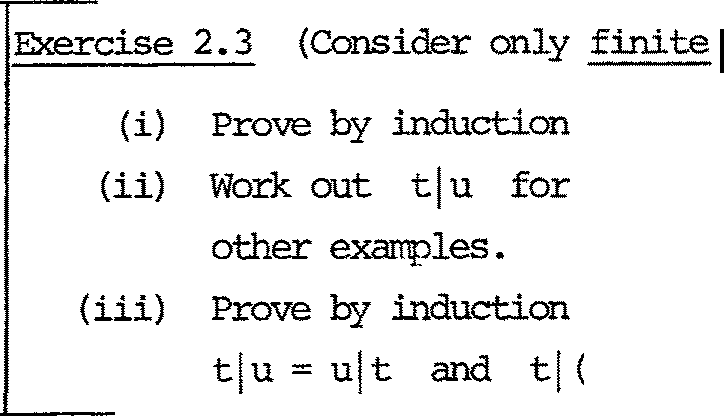
J

中＝狂（七.lu> + 沪．（七Ju . > + L -r <七i luj >

i 1l. J·J

µJ =v.

l. J

STs).

on the depth of 七 tha 七 七INI L =七．

I

=0; and u= 岛

on th e 竺 of the depths of trees tha 七

(u 位）＝（七ju) Iv.

We should criticize two aspects (a 七 l eas t )of our def ini 已 on .

Cons 运 er in g our f irs 七 ex 动平l e of ST CUI平运 i ti on, it can well be argued tha 七 the fo 五 n we gave for 七iu f ai ls 七 o rep re sen 七 th e possible con­ curren 七 acti vi 七 y of 七 and u - for example, we may 七 h 止 寸 <:: th a 七 a

s- exper.iman七 on 七 can be perfo五心d s:imu.l七玉1eousl:y with an ; -exper imen七on u, while (looking a 七 yourr esul 七 f or Exercise 2.3(ii) a 迳o) the ST

for 七 I u rnerely indicates that the two experiments may be per fo nned 兰

either order. Indeed, STs in no way re pre sen 七 七 m 玉 = concurrency.

Two no七 cat 叩 l ete l y conv 江 cin g defences can be given. Firs七， S Ts are simple, and tr ac七abil i 七y in a 兀IOdel has gre a七 a d.van垣 ges ; second, in so far as we wish a'behaviour-object'to tell us 比 111 a system may appear 七o an observer who is only capable of one m汗汜r imen七 at a 七扛芷切

we f in d 江 poss ib l e 七o i gnore 七m坦 ro n currency. You are u巧ed 七o consider this que已s on in greater depth.

The seoond aspect for cri 七 i ci sm is the introduction of -r to represent successful 'mu 七ual observ a 已 ons ' . I f we h 玉 no n 至 d for 抚 1n a.etining I,

we could l eave 让 OU 七 of our theory al 七 cgeth er .

Again, there are two defences, bu 七 thi s tiJre convincing ones. First, consider replacing the third teJ:m in the recursive defini 七 i on of 七lu -

namely the 七enn 汇 T ( 七 i luj ) - by jus 七 I <七. iu.>;

µ.=v. =- 1 J

J

1 J 1

扛 让 ui ti vel y, an intenial action jus 七 vanish es . I 七 七u工ns OU 七 th a 七

is no longer an associative opera七i on , which confl 虹丈8 S 七ror丐l y with our assurrption that the join 七 behavi our of three agents should in no way depend upon the order in which we wire them together before they do any­ thing!

厂

ise 2.4 With this n窃 def 皿 已"' >ork OU七 七I <u 位） and (七luJt v f or 七 = a ! , u = ; I, v = SI to justify the above assertion.

The second defence is that vve must sare土CM express, in the ST ( 七 iu ) \ a when 七 = C\入8, u = !a, the possibility tha七 carmuru＇ .cation between 七 and u can preven七 any S-exper让ten 己

I, and of \a (see below) ,

f。

n。

it

i

江

f

8

『

lam

on

B

人

于｀

=

eht

r e

a＼

dnU七

a e

ht 回 sa

仕

厂

s

益

tUO

e

**s**iikr

'Ihis ST does indeed represen 七 possib l e prevention of a S--experirren 七 ， and 皿 ess we leave STs (and der i ved 函 els ) altogether i 七 i s hard to see hCM such deadlock phenanena can be re pr esented wi th ou 七 石

．

c

.1

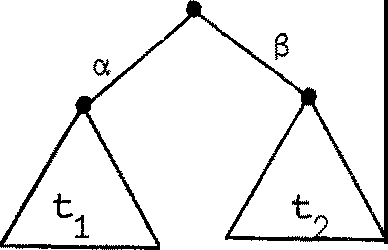
ht

距 s tr i c已 on \a.:ST + ST

L L- 位，心

如 t:,. )

We wish to deny all ct- and ;-experiments, so tha 七 七 \ ct is fo:rrred

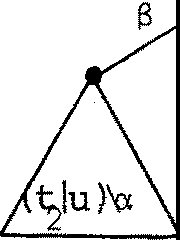
by pruning 或 ay all branches and sub-branches labelled a. or ;•:;: Cons i der 江g

七＝

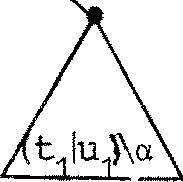
｀

＿＿

u

again, we see th a 七

元

国 u ) \ o. =

还 re fonnally, for 七 ＝ 狂 ．七. we have

* 1 1

I

1

Def皿 已on 七\ a = µ i 上，动µ i ( 七i \a)

An obvious alternative to the res立i c已on operati on 沁 uld be to define

＼入 for ea ch 芷 叫 泣 入 of A by

七 \;>.. =µJ/i ( 七i \;>..) ;

in oth er 沁 r ds , 妇 皿 gh 七 choose to restrict n 蕊 芷s and co-narres independen 已 y, instead of both a 七 once . 'lhi s would, of course, have a correspondingly

di f feren七 ef f ect on pictures. The reason for our choice is in fac 七 to do with the algebra of pictures (FlCM Algebra) tmder I, \a and [SJ: 让 has a parti cula 丑 y sinple algebraic theory [ , Mil 2], which we have not found for the suggested al 迳 ma ti ve.

Rel abe ll 扛 [ SJ-: ST ST

L M

(S:L -+ M a relabelling)

'Ihis operation is as s 乓 le on STs as i 七 i s on pi ct u 年 s ; i 七 jus 七

applies the relabelling S to all labels in the tree. 购 re formally,

for 七 ＝ 狂 ．七 炽 have

i J. i

Def ini ti on 迁 SJ = ?S(µi) (七i [ S J)

工

where we 立 M' adop 七 th e oonvention tha 七 S(T) = T for any relabellings.

An 扭 rt an 七 (th ough no 七 th e only) u 卑 of relabelling is in cases whe re 袒 have several ins 远 ces of a single agen 七 r in a sys 七 em, bu 七each with di ff eren七 l abe ll in g, so tha 七 under canposi已 on they a迳 proper l y linked. We have only to define several'copies'

r. = r[S.J

1 l.

of the generic agen 七 r , and then carpose the r .•

1

One 皿 gh 七 have all c 双 ed rrore general relabellings, using many-one functions over A (so th a 七 di f fe ren 已 y labelled ports care to bear the sarre label) or evenr el a 七 i ons in place of functions (so that oneport could ' spli 七， in 七o two di f fe ren 已 y labelled ports)• Suffice i 七 to say th a 七 th is creates proble:ns in the axi anati za 已 on of Flow Algebra.'!he presen 七 choi ce al la 总 pl en ty of srope.

* 1. A s 砰 l e ex 郘 ipl e : binary semaphores

A binary s 部 畔 ore s, of sor 七 { 11,i}, may be pictured

To gain the semapho年 (Di jks tr a • s P OE 沦 r ati on ) we mus 七 perfo 五 n a 11 一

exper扛 切 识3 炬 l eas e i 七 (th e V opera已on) by a <f,-.玉午er 扛汪mt . Clearly

s = 1T 中 S

expresses the appropriate behaviour (a long thin ST!). Imagine a generic agen 七 p, who改:! cr i 七i cal section we rep re sen 七 by a sequence

<a,$> of a 扛 血 c actibns (experiments upon a resource, say), and whose non-critical section we ignore:

p = 1TCI 的 p .

we wish to place several instances of p

p. = p[S.J = 1ra.B. 怀p . (where S. = a. B-1 祖 ）

1. 1. 1. 1. 1. 1. 1. 1.

in ccmnunication with s, and derive the a::nposite ST. Cons i de 工 jus 七

two copies of p (i = 1,2) and fo血

q = <P1 IP2 !s ) \ 1r\ 中

wh 扛 h may be pi ct 哗 d as shown:

a

1

13 P s p a2

1 13

2

q

We easily derive an equation for the cx:xnposite ST q, using the Expansion 'Iheorem - given in 立 5 - repeatedly. You should read tha 七 sec 已 on with

r eference 七o th e e 习灭江区 io n whi中 foll r:Ms:

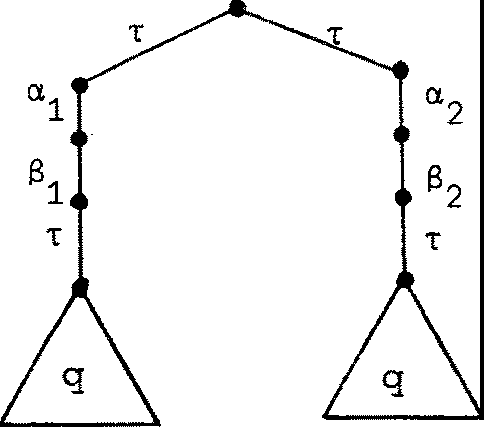
q = {气 早 P 卢 宁 戎P:21 平 ）＼飞

＝ 认{ a.凸觅尼南）＼飞） + -r«P1 la 泸沪乌 中 ）＼汃＄

＝ 丐 生（（中P1 IP2 南 ）＼飞） + Ta.282 ((p1 j 4'P2 I $S) \飞）

= -ru1 B1 1:(( p1 尼 Is 阳 ＼ 中 ） + rn2 $2 -r(( p1 IP2 ls ) \ 飞 ）

＝ 迈 1 S1 -rq + rn 沪泸q •

So q is the ST given recursively by

q=

and 壶 ct l y expresses the fact tha 七 th e cr i 七 i ca l sections of p and

1

p can never 切 erl ap in time, i.e. a sequence like a a B … i s no 七

2 1 2 1

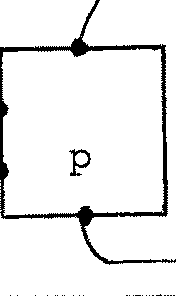
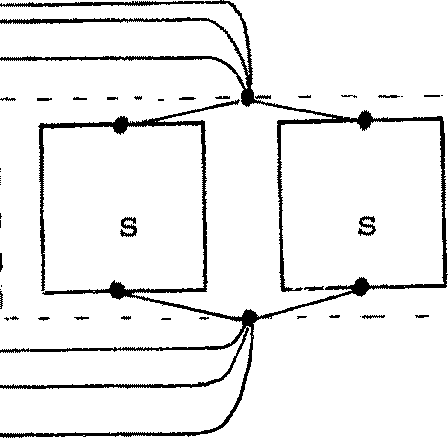
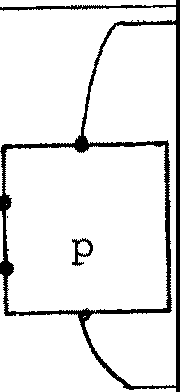
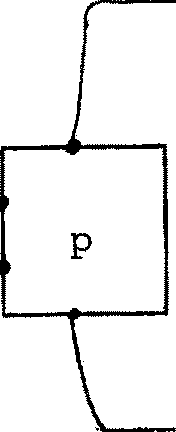
possible.

In fact, an n-bounded sanaphore (n z 1) can be constructed as s = sl s1 ••. 1 s

n 七 irres

this is an exanple of 立项 运 i ti on 泌i中 eff ects no linkage, bu七 wil l yield a multi-way linkage with'user'agents.

吐e 2-l:x>Unded semaphore s2 , with 3 users, can be pictured



s 2

3 3

6

a

2 2

6

a

1 1

a B

Diagram for (p1 飞见!s2 ) \ 飞

(s's border, and its two collector nodes, are fictitious; they are jus 七

2

used here 七 o avoid drawing 12 links in the pictu 工 e ) .

Exercise 2.6 As practice in using the Expansion Theoran, develop the expression q = (p IP IP Is Is ) \ 叭 cp , and draw part of the ST to convince

1 2 3

yourself tha七 a 七 mos 七 two cr i 七i ca l s ec 七i ons can be s irnul七aneous l y ac 七i ve . Can you even derive a se 七 of mu 七 uall y recursive behaviour equations, for which q is the so l u 已 on? I 七' s a b 让 l ength y , bu 七 r,oss ib l e . 'Jhe

deve l oµt 它 址 i s shorter if you take ct1 气 2 气

3 气 ， s1 =s2=s3=s; i.e. deal

with (p lp lp sl ls ) \ 叭 中 年 玩 ead ; then the ST will no七 dis 七inguish the

cri 七 i ca l sec 七 i ons of each copy of p, bu 七 you should be abl e 七 o sh= th a 七 a 七 any r,oin 七 in 七 江 re 七 he excess of a's OV1 匀 S's perfoil!Ed lies in the range [0,2].

* 1. The ST 沁 平 晔 i on Theorem

We consider trees expressed in the fo:r:m

七== Iµ. 七 ．．

l 1

华 i :::n

For a s e 七 {a1 , ... ,a.k} == A of names, we abbreviate \ a1 飞 ... \ak by \A.

Theorem 2.1 ('Ihe 氐 中 主 io n 'Iheorem)

狂七七= (t1 It 2 1 •••• 1 七 ）\A, where ea中 t i . is a sum as above.

m

'!hen 七= I伈（（七 1 ••• I 七.I I .•. I 七 ）\A) ; 1 s J. s m, µ七, a SUI 订正江正 t of 七．，

1 J. m J. J.

name (µ)从｝

＋ 炉 （（七1 I· · • I 气'. I· • · I 阿' I .•. l tm) \A) ; 1 s i < j $ m,

入七．，

J.

a s 中 亚 md of 七. , 3:七' a surrmand of 七．｝

J. J J

Proof 立 七七ed ; it uses propel六;i es of the Fl<::M oper a已 ons i, \a and [SJ, and canbe done by induction on rn.

区

The theorem s ta 七es th a 七 ea 中 br andl of t rorresponds either to an unrestricted action of s晔 七， or to an internal ca:rmunica已 on between

工

七. and 七．

l. J

(Jl. ) <.

* For example consider

( (a 七十 6七I ) I t沁 ＋ 汜 ') I (Sv + y炉））\a\S;

-- ------"- --·

I (

thetheorem gives us

认( (a七十印 ）lu'I (iiv +和））,.1e)

＋飞{ (at + 印） I (沁＋丫记） Jv')\a\S}

+ T( (七回 （函 ＋ 扣 ））\a\S)

+ T((七,I <还+ yu ' ) 位）\a\S)

+ T( ( (a 七十 釴 '> iu ' I 炉）\a\S)

unrestricted ac已ans}

(心- o:::mnunica ti on) (13一a::m囮 ni ca ti on) (y-o:::mnunication)

Exercise 2.7 A l o 七 can be done using ccmpos i 七 i ons of 巳 心 kin ds of e l erren 七 ：

A

YO,

匀 cl e r C = a 衍 C

' 勹 " d- o ( Od + y d) ,',

til wr让e the behav of 勹一。

0---Q

as a restricted a:mpos i 已 on of relabellings of c. (The li 七 七 l e arrows

r epr esen 七 the port at which each oopy of c of fe 立 i ts first e 笃 无 过－ rnen七； the progress of the system can be simul a 七ed by "swinging arrows": try i 七）• Expand the behaviour, to ge七 a r e cursiv巳 def in i已on of an ST whi 中 doesn ' 七 in volve a:mposi 七 io n , res 红 i ct i on or relabelling.

(ii) Design a system (using c only) to behave as the ST

s = a (.(3-rs + -ry-rs) •

Is this equivalent to d?

0 丑呻 3

A cas e s四 in s chroniza已on, and roof te 如 i es

* 1. A schedul 扛巧 probl em

Suppose tha 七 a se 七 {p. ; 1 :;; i sn} of agents all wish 七 o perfonn

J.

a certain task repeatedly, and we 吐 sh to design a scheduler to ensure

th a 七 th ey per fa 五 n i 七 in ro ta 七 i on, starting with p • ('lhis ex五四l e

1

was used in [Mil 5 J.)

沁 r e precisely, the p. are to 巠至 th eir performance of the

J.

task in rotation; 炬 do no 七 impose 出 e restriction tha 七 th eir perform-

ances should exclude each other in tilre (this could be done using a semaphore) bu七 we do impose the r es tr i c七沁 n tha 七 ea ch p. should be

J.

preven七ed fran ini已a七扛巧· th e 七郘 k twice wi th ou 七 carpl e七江 g his f irs 七

initiation. (p. may t:ry this unintentionally, because of bad programning

J.

for example.)

Suppose tha 七 p . requests initiation a 七 l abe l a . , and signals

J. J.

carpletion at 13. {1 s i s n) .'lllen ourscheduler Sch of sort A u B ,

J.

劝 er e A = {a. ; 1 :;; i:;; n} 忒 :d B = {13. ; 1 s i sn} , mus 七 impose two

J. J.

oonstraints on any signal sequence E (A u Bf3 :

1. 网1en all occurrences of 13. {1 s i :;; n) are deleted, it bel 文 双 巳 S

J.

(a a

1 2

••••CnJ.)w •

1. Foreach i , when all occurrences of 釭J ，SJ. (j ;,, i) are deleted,

i 七 beccm 玉

-o

{a. S.) w

二 二·; 二：

:三 J. J.

o七 build

Scheduler Sch:

- -

～、－辛



using also a'start bu七 to n ' ,



S 七ar ter s

In bullding 也e ne七 we have in s tan已｀a七ed C by

J

6

／

1

＋

. 1-

-y

，

y

／

i

y

B，

／

. 1i

6

“}，

／

i1

a

［

c

- l

.1

c

'

,

for 1sisn, 如ere ad已iononsubscripts is Irodule n, so that

,

\ Sch = (s I c1 I· · • I c)n

\ Y1• • · \y n

妇 七 are the behaviours and C ? The starter is there just to

j

s

enable Ci a 七 Y1 and die, so

三

沁s for the cycler, 江 appears tha七 he should cycle endlessly as follows:

(i) Be enabled by predecessor at

（拉） Receive in i ti a已on reques七 a 七

(iii) Receive te rrnina 已 on signal at

扭 ei ther order.

y ;

0＿, ，•

8 and enable successor a 七 0 ,

So we define

c = 云（社C +嘉） l

and this detennines Sch ccrapletely. But does 迂 wor k? In f on 花过l y we can convince ourse l 诧 s th a 七 让 doe s , by arrcw- 却 in gin g. !'Dre fo:i::rnally, there are twopossibil i 已 es:

距thod 1 Show as directly as possible that constraints (i) and (ii)

are 芷t. Forthefirs 七 cons tr a 年 t , this may be expressed as absorbing

氐 e. pe五让七七年 g) a 耳 S . carmuni ca 已 ons , and showing tha 七 七 he re sul 七

J.

is observ a 已 onal l y equivalen 七 to

一心

a. a. •• •Cl.) 1 2

n

Le 七 us make this precise by adopting the convention that if s is any

non-empty label sequence, then sw is the behav 土our given by s 见 从= s(s).

Then wha 七 we wan 七 七o prove, for the f ir s七 consr七 a 止让， i s

1. Sch II (Si w I- •• -I 灯 ）：：：：： （三 .. - 汇

(where ::::: 担 obse rv at i onal equivalence, which we define formally in§3.3).

Using the not a七i on

吓 qi ; i EI} or 门 q .

id 1

for rnul霄 l e ca 屯)()Si已 on , we can re wr让 e (i) as

sch 11 n s.w ::::: 盂．．．；产．

J n

1:::jsn

The required equivalence for the second cons tr ain 七 i s

（江） Schl!(n a,.w J n 仁J ） ～ 盂 f3. )w for each1· ,1::;1:;;n.

护i J j吐 1 1

注thod2 We can specify the beha过our of 匕he ex项 ?let e scheduler by a single par 罕 teri zed behaviour equa 七 i on , in the follCMing way. Cbserve that the scheduler h 坴 七 o keep two pieces of infonnation:

* 1. An integer i (15 巨 n) indicating whos e 七 urn i t 担

to initiate next.

* 1. A s ubs e 七 X of [1,nJ indi ca七in g which agents are

= en 已 y performing the task.

If Spec(i,X) re presen 七 s the required behaviour of the scheduler for

par 罕 ter values i and X, then 炬 can specify the scheduler by

Sp:c(i,X) =Is. Spec(i,X-{j})

jEX J

Spec(i,X) 五 Spec (i + 1 , X u{ 切） + I s.Spec(i,X-{j}

jE:X J

(i EX)

(i X)

'Ihese equations say tha 七 i f p. i s 凸 竺 per fo nning he can :ini 七 i ate , and :in

l.

any case any pJ.(j E X) can signal c 叩 leti on. For thi s 芷 thod we only h 取 e to prove one observation equivalence:

Sch Spec(1,0)

In§3.4 we give part of a p 江 心 f using 洷 thod 1, which may be preferred since i七 dir e ctl y represents the cans七t:'a in ts as specified. M 式 :hod 2 is possible, bu 七 a li 七 七 l e harder.

三

江

s

e

这

s

e sd

ht ee

g

江 n

七

u

sub

',

d

e c

rerayc

h h

七

de tones

n

.m

f s

e i p

d

r I七 eht

e

1

c

yc万

2

e

s

ht esev

二了

g

f i

u o

y

C

na

c

辛g

1

yce

尸

j ht

Exercise 3.2 Build a scheduler which imposes a third cons tr a 扛 止 on a

luf

r

e

1

e

r

.

e 3

ki 孔

. 1 §

m

ht

xE notl

。tced

a

ur S

七

1．

s

.s

g

signal sequence

E (Au B)w :

(iii) When all occurrences of

a.. (1:,;: i:,;: n)

l.

are de l 玩 ed , i 七

be C(平 S ([31 胫... [3n) w •

This constr ain 七 s ays th a 七 the in cyclic order.

pi 皿 芯 七 als o tenninate their 七 as ks

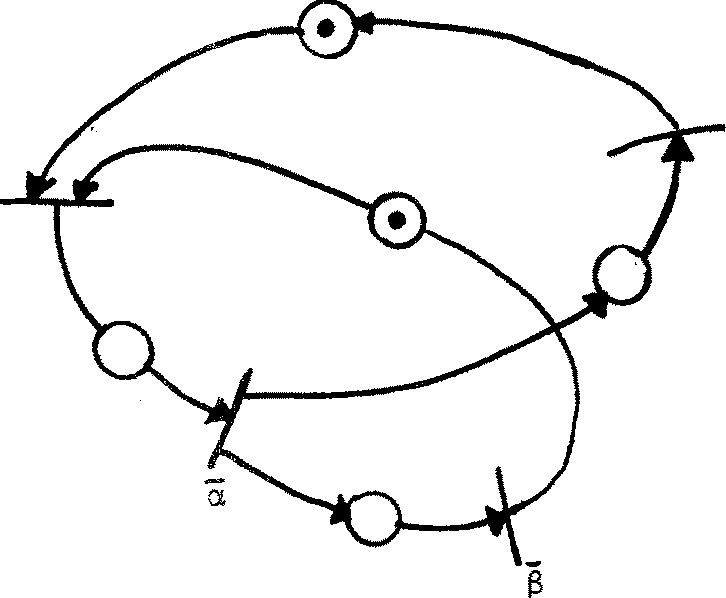
竺 : 'Ihese exercises are playing 七 o sane extent, bu 七 they may hi:Ne S 匀 e significance for buil 血 g asynchronous hardware fran cc 平 nen ts . 加 s remains to be seen.

We shall nCM divert to carpa 迳 our behaviours with Petri Ne 芦 ， informally, using the scheduler as an 壶 平 e. Readers unfamiliar with Net Theo 对 may skip the next section.

* 1. Buildinq the scheduler as a Petri Net

We will use Petri nets in 中i ch the even岱 or transitions are labelled by 芷 叫运 of Au { 叶. In fact, we shall jus 七 0 吐 七 th e

-r labels.

A ne 七 c , for our cycler, is as foll0t-1s, 动 正 江 e circles stand for places and bars for transitions:

6

y

With the in i 七 i al marking as shown, the ne 七 i s clearly live in the usual sense. Bu七 in our in terpre七a 七i on a 入一l abe ll ed even 七 is nerely poten七i al ; 让 n eeds oooper a 七 i on with an event which bears a 改 ::mpl e:aen 七 a 巧 l abe l , or with an observer perfonning a 入 - exper:i.rren: 七 ．

The fl 叩 妇 ati ons J , \ a and [SJ can be satisfactorily defined 0 欢 江 a class of nets (as }bgens Nielsen has shown) in such a way as to yield a Fl Algebra. Here, h 竺 ver , 让 , vill be enough 七 o use only [SJ

。

- the obvious relabelling opera 七 i on - and the derived cpera 已 on II ; if

n 1 and n 2 are ne 迳 of sort Land Mand if {a , ••• ,ak} = n 改 芷 汜 (L n 面 ，

then 1

n1 JI n2 = (n1J n2) \a.1 \ak

may be described as follCMS:

Identify the even 七 l abe ll ed a. (resp a.) in n with the

1 1

event labelled ; . (resp a. .) in n

1 1 2

, for each 1 i, and

then dr叩 the labels a, …，a k

and their carplemants.

［竺： This needs nore careful phrasing if 妇 al 皿 tha 七 nl may 王去扭竖

a 入一even 七 even though 入 E L . Also, in general we llllll 式 ：take care of the

possibility tha 七 n1

- for example - may have two or nore 入 -e ven 七s .

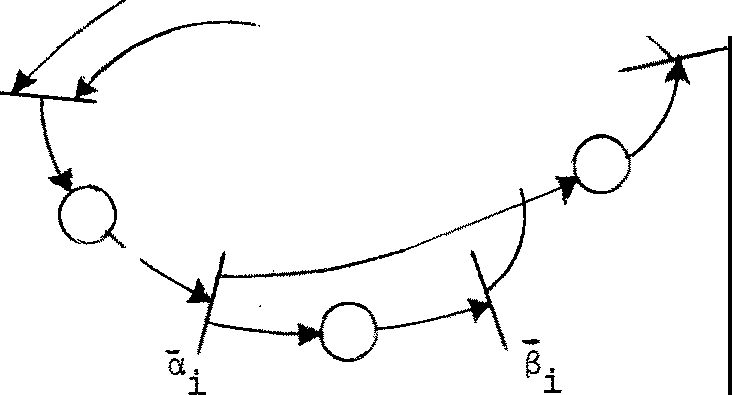
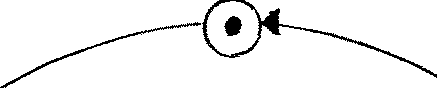
However, if we start with nets n of sort L having exactly one event

l abell ed 入 E L , and confine the use of COlDJ 为 s i 七 i on 七 o pair s n :L, n :M for which L and M are disjoint, then all ne 七 s bui l 七 wi th [SJ and II

1 2

will have exactly one event for each label in their sort].

To illustrate with cyclers, we h 取 e , for ci = c [ o.i 八，气压，yi /y , 乃+1/ oJ:



1

＋

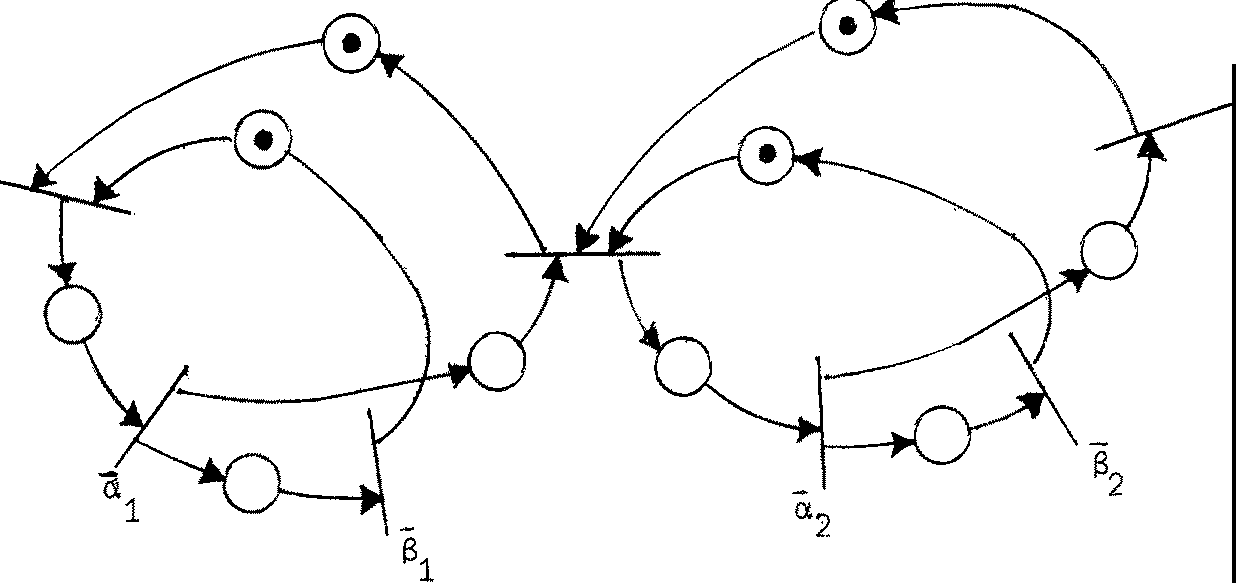
.l

-y

. 1-

Y

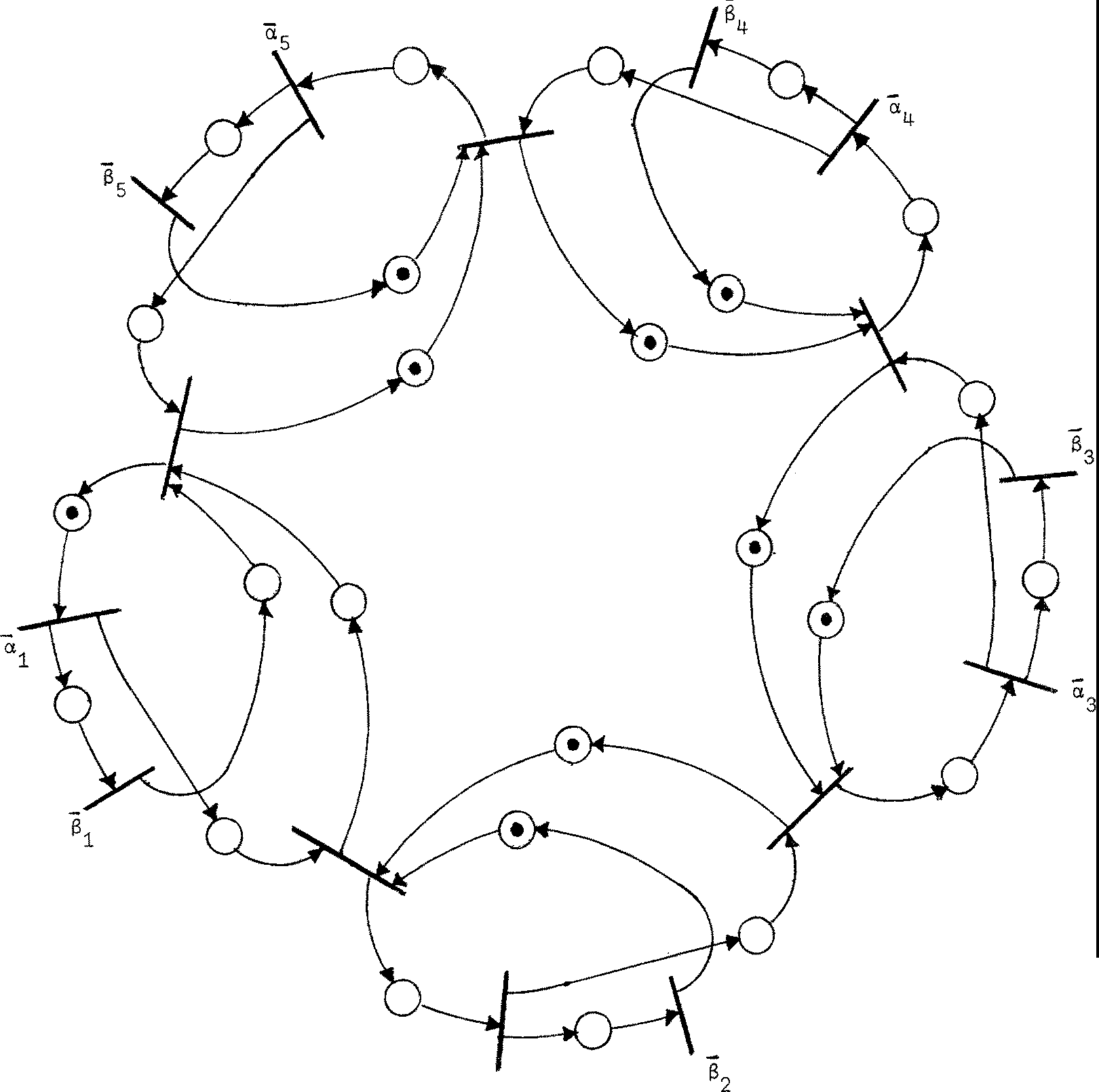
and for c1 lie 2

Y1

3

-Y

Finally we give the diagram for a scheduler of size 5 on 如 i ch you can play the 七oken game:



a 2

吐 e Petri Ne 七 for the scheduler

阮 i ce the sli gh七 che a 七： c has been given a dif fer en七 in i 七i al marking. This would no 七 h ave been needed if we had included a part of the ne 七 f or our s 七art bu 七 to n , and in building the ne 七 we would then find the need for rrore than one even七 l abell ed y - which corresponds to the sha 炬 d port of c in the picture of Sch, §3.l.

There is a growing 血 y of 七 e chni ques for analysis of Per 七 i Nets. For exarrple, the behaviour of 注 江 ked Grafhs is well understood [CoH];

a marked graph is a Petri ne 七 in which each place has indegree and outdegree equal 七 o 1, and our scheduler is in 华 ed a marked graph. Further, much

can be discovered of the behaviour of arbitrary ne 七 s us in g 七 e chni ques fran Linear Algebra due to Kurt Lautenbach (Cl1D, Bonn) 七 o discover Invari an 担(properties which holds for all accessible markings, or 七 oken di s 七 rib u 已 ons) .

Kurt Jensen has pointed OU 七 廿 正 让 \_these techniques are strong enough 七 0 七 e ll us that our scheduler net indeed sa已sfi es 七he two cons江 ain t s specified.

Nevertheless we sha ll 七 ackl e the proof of correctness of the scheduler by our CM1 methods, since we shall see later tha 七 the y apply als o 七 o sys 扫 五 芯which are no 七 so readily represented as Per 七 i Nets (e.g. Sys 扫 珀 记 whose corcmunication structure does not IJ部 巴 in f扭ed) .

* 1. Obse 切 吐 i on e:平止val ence

工 七 i s nCM t ime 七 o be carpl e 七 el y precise abou 七 七 he fonn of equivalence

of agents that \-Te wish to ad 叩 七 . The discussion in Chapter 1 was :i.rrprecise, deliberately so; b 砒 nCM th a 七 we have a case s 七 udy in hand where correctness of an agen 七 has been expressed as equivalence between the agent and its

speci f i ca 七 i on, we have enough m: 丈 i va 已 on to s 七 udy equivalence seriously.

We may f or ge 七 our algebra tempor a过 l y , and 让nagin e s:i.rrply tha 七 we have a se 七 P of agents (or behaviours) together with a family

｛上；µ"'/\ u {T } } of binary re l a七i ons over P. /\ is our label se 七 ， but

妇 can also forge 七 tempor ar il y th a 七 I\ = A u . We shall consi s ten 已 Y use 入

七 o range over A , andµ,v to range over A u { -r} •

p p' means "p admi.七s a 入气 这 p 江 扛汜砒 ， and can transfOllll into p'as a result"

p p• means "p can tr ans f onn 七0 PI UI1obser\ll:式”

We shall wr迂 e P —s 吓' , for s =µ1.…坛 for s哱 p，。... ,p (n:::: O)

\-1 n

。

一

E (AU {计）＊ ， 七o m玄五l th a七

p = p

今历

p 1- 今 p 2 ...•

叩 >p n = p'.

NCM oonsider the 年 SU 止 (s ) of performing a sequence 入 ，．．．，入 of atanic

n

1

exper 扛ren七s on p (n d!-0) .'!heresu七l may be any p'for which

K。k1

T 入1 T 入

kn

．．入．T

p 2 n

; p' (k. l O)

that is, an arbir 七 a 匀 n umber of silent m:Jves may occur before, 玉 江 mg and after the 入 ．

J.

Def 扭 i已 on

for

s

s E A\* , define the relation by: if s

=入1 .•. 入n ， th en

p =s 吁 ' i ff for sane k。••,• ,k :?. 0

k n

T OA1Tk1A2··· 入产

p :p'

We may talk of an s-exi;, 缸 irren七 (s EA\*), and then p 呈 今 p ' rneans

II p admi七s an s --exper :in:en七 and can 七r ans f o:nn to p'as a虑sutl" ; we may also say rrore briefly " p can produce p'under s ".

Note th a七 for th e 巴 中 ty sequence E E A\* , an E--exper :in:en 七 consis t s of

le七tin g the agen 七 pr o ceeds il en已y as i 七 wi s he s , mile observing nothing; for 歪 have

k

p 今 p' if f for s啤 贮 0 p p' Note also the special case p 今 p when k=O.

NON we can s ta 七 e in words wha 七 we shall mean by equivalen 七 agen t s .

p and q are equivalent iff for eve 巧 7 SE A\*

1. For eve 巧1 re s ul 七 p' of an s - exper 扛芷mt on p , there is an equivalen 七 re s ul 七 q ' of a s- experirnen 七 on q.

（过） For eve 巧 re s ul 七 q' of an s --exper 让 它 砒 on q, there is an equivalen 七 re s ul 七 p ' of a s--exper irnen 七 on p.

This appears 七o be a circular defini已on (the formal def in i 七i on will take care of this poin七） bu 七 no 七e f 江 S 七 th a 七 士七 impli es th a七， f or each s,

p admits an s- 巴 <per 扛 ten 七 if f q does.

BU七 i 七 irrpli es much rrore; for example, the two ST's

人

臼

admit exactly the same s -exper i.rren 七 s , b u 七 ne i th er of the two possible resul七s of an a- exper i.rren七 on the f ir s 七 七五涟 i s equi val en 七 七0 七her es ul 七of an a- exper 泗 n 七 on the second.

The notivation for our definition is this: we imagine sw 迂 ch in g p on, perfornrl.ng an experi.rren七， and SW让 ching 让 off again. For q 七 0 be equivalen七， 让 mus七 be pos s ib l e 七o swi 七ch q on, do the same experi.rren七， and sw 让 ch 让 of f in a s 七ate e 平 Ji valent to the state in which p was

sw迂 che d off (and the same, interchanging p and q) .

Our fonnal defini 过 on is in te:a:ns of a decreasing sequence

~01 ::,,1 , • • • , ::,,k I

. • . of (f in er 立 f in er ) e 平 让 val ence relations:

氏 f in i 已 on (Observ a 已 on equivalence) p P 飞 +1 q if£'rt s E A\*

。q is al ways 七m玉习

位） if p

（拉） if q

p'then fors 哱

q'then fors哱

andp' 气q'

andp' 气寸

I

q

I

p

s

S『

q

『

p

＇

，

I

q

I

p

p ::,i q iff V k <'. o. p 气 q (i.e. ""= n /::kI) •

k

．

yb

,

e．

noi

i ,

七

飞~

al

er

U­

ec 扫

nel～K~

a

V 七

·

}

aht

na.no

s g

i

ud

江

～飞y

．

b q

气

hcae

e vo

rp

七 P

a

迨

1

ht 回

e

vo r p

..『

k

q

`il `

a

丑

( no

k

no ～~

3

3 i

p

尸

This equivalencer el a 已 on has many in te re s t 扭 g properties, which we need no 七 examine Ul 让 il Chapter 7 - except one or two.

Fir s 七 ， i t is no 七 neces sar il y 七 m 迨 th a 七 ;::: i s七 e lf sa 已 s fi es the

recurrencer el a 已 on defining 飞~ +1

p I'>! q iff \::,/s E A\*

in terms of ~

次

, that is, the property

（＊）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (i) if | p 呈> p ' | then | 扣' . q >q ' | & p ' 叮 ＇ |
| （拉） if | q 县珂 I | then | 扣' . p 呈>p ' | & P'""q' |

(which is a formal 诧 r s i on of our verbal recursive definition of equivalence given earlier in this section) 参 I 七 i 丢: 七工ue if p and q a 工 e finite STs, bu七 no 七 in general. H 叩 ever , our def in i 七 i on has nicer properties than

one which satisfies (\*).

ForSTs, our binaryr el a已ans - and are obvious;

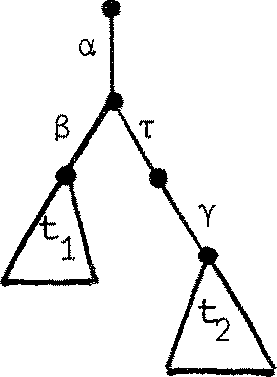
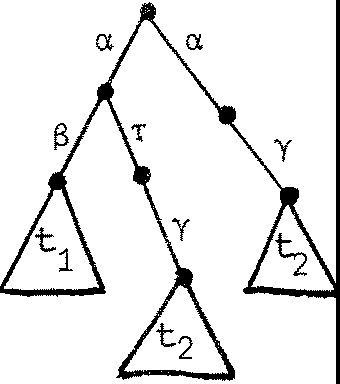
七-七' (resp. 七一仁＞七') iff 七 has a branch 迁' (resp. , 七'). In this

case we shall call t' a 入一s on (resp. , 一s on) of 七 ．

尸i se 3.4 Pro 伲 tha 七 t "' 迂 for STs. inductive proof th a七 七 ""k ·1七）．

(You need a ve 巧 S 乓 le

Le 七 us consider one example of equivalen 七 STs:

u =

七

To check equivalence, i.e. 七 芍 《 u for all k , we mus 七 prov e the inductive step: 七芍<: u :i.nplies 七 气 k+ 1 u. N叩 f or every s "'e:, 七 and u produce

i den已 cal 七攻关名 under s ; under e: , t produces only 七 and u cnly u ,

and t 芍 <: u by induction.

I Def:inition If p 呈吁 ' (cr EA\*) then p' is an s- deriva 七 i ve of p.

(No 七 e tha 七 p is always an 1;:-der i va 七 i ve of itself). We can thus rephrase the definition of in tenns of :

k+1 k

11 I? Rlk+i q iff, forall se:: /\\* ,

p and q have the same s-derivatives

up to Rlk equivalence. 11

巨 3. 5 Pe-examine Exercise 1.4, andverify precisely .m.m pairs are

obser1a七i on equ i va len 七. You should find exactly four pairs.

七

i

七

at

s

g

mni

**.**

8

1

1

0

f

1

+

入n

eht心 dna

g 二

·

入

q

t

,

aht

r

二

e

fo ot

8

r

一

二

郡

a bi a

f

6

尸

One property of agen 七 s i s not respected by ou工 已于让 valen ce . 工七 i s possible for p and q to be equivalen 七 even though p possesses an

＿

`，

e

1

bi

i

. t .

si迳

．

二二

r

七

roJf 坛一1

1．+

入

n

入

a

od

』

e

l

o et

s

七

s s

8 a

S

O hc

infinite sil en七 ccnput a七i on

'T

p 一p 1 -P

2 -今. pk

" Pk+1 一 ．．．

血 ver gen ce ) while q canno 七 di ver ge in this way. The equivalence can

be re f in ed 七o exclude this possibility. See the remarks in§7.3.

* 1. Provin9 the scheduler

工t is C\.11让X 江 SO 宜 to us e 七he direct defini七i on of RI; we shall instead use a few of i 七 s key properties, 动 让 ch a 立 der i ved fonnally in Chapter 7.

We begin by li s 七 in g them, so th a 七 Chap 七 er 7 need not be :read fir s 七 ．

(:::;i 1) 七 '(七 (see Exercise 3.4)

NCM we can see that is n七oa a:ngruence re l a七i on ; th a 七 is , replacing

七 by 七' (when 七～心 ） in u to ge 七 u ' doe s no 七 en s ur e u u'. For ex吽le, NIL,:; ,: NIL, bu七 a. NIL + NIL 中 a.N IL + ,:NIL •

巨i se 3.7 Verify this fact.

So年 gener al 七：：：：七， does not impl y 七 十 旺 t ' + u • But all our other oper a已 ons 空?. preserve 心

I

(a 2 ) 巨 t • =

已汇= : 二了了： Ii I

七 \ Cl.I« 巳 \ a

七[ S J :::: 七' [ SJ

Fortunately, 七 oo , when we apply a guard JJ 七 o equivalen 七 STs 七 ， 七 ， we

ge 七 no 七 onl y µ七""µ 廿 ， bu七 µt :C::; µ七', where ::C::: 担 a s 七rcn ger re l a 七 i on than

::::: which is preserved by 呈 our operations.

作 3)

C::::: is a rongruence re l a七i on, and

七 g 七 ， implies 七 之 七 I •

Beyond these, we need one nore propert y 茄 i ch may look a li 七 七 l e surprising;

we leave its discussicn to Chapter 7.

(::::: 4) 七 十 T 七 ＄ 迂

Apart f 江 m this, the proof bela,; will useonly rather natural properties of our or:er a已 ons , including the E习芙insi on 'Iheorem, all jus 七if i ed by Chapter 5.

悚 tre a 七 only the fi rs 七 constr ain t , mu 汜 l y

Sch II (S w I ... Is l

n

;::; {a •••

-exn) w

(1)

Define the l ef 七 h and si de 七 o be Sch'. t--e shall actua耳y show that Sch'

sa 旦 s fi es the defining equation of (a •••- a )(!J , namely

1 n

Sch',::: -a. ••• -a Sch'.

n

1

(2)

fran which (1) fo ll 硒 ， by general principles whi中 we shall no七 七m涟七 he 迳

(bu 七 see Exercise 7.7).

We may wri 七e Sch'as

Sch'= (s I c'I1 ... I cn'l\y ••• \yn (using general properties of I and \a) , 中 er e

(3)

ci = (ci I 叮）\f3i (4)

represents the

.th

1

cycler with a. penni 七 ted . N 叩 we shall discover

l.

bel<:M that

(5)

I

i

c

1

i＋

-y

, 1-

- 8

i

Y

c **~**

，

,

1

c

so we can use these expressi ons 皿 er中angabl y, by 长 3) , 七o as s i s 七 our expansion of S 匈 ， wh远 run s as fo ll 硒 ：

Sch' g （ y NILjy a. y c'I．．．． Jy -a -y c' ) \y .•• \y

1 1 1 2 1 n n 1 n 1 n

g T( N工L la 1 飞弓飞-;;:2 ;: 3c; I··. Jynan;:1c l\厅.\yn

(the start bu七ton has worked)

g 气 a,2\* ••• rnn ( N 工 日 Ci 同 I • • · · h1 c ) \ y1 · · \ Yn

(leaving c ' 七 o be reenabled)

g 飞 a2,

•••• TanT (NIL I 气Y2 Ci 鸟J ••. 团 \ y1 畸 . \ yn

a, a, ••• a S中 ' as required, by (i:::: 1) and 妇 2) .

n

1 2

Le 七 us no,1 show (5) , for i = 1 say.

ci = (y1a.1($1y2c1 + Y2 131 叩 I 芞W) \ 芞

y1Cl1 (Ty 2C1 + y2 TC1 ) by expansion.

But y2

TC ;C:::

-y c'by (i:::: 1) and 伈 2) , so

2 1

TY2C'+ Y2,Ci§TY2Ci +飞Ci by (i:::; 3)

,C.; T-y c'

2 1

by( 4),

andby subs七让 utin g in the expansion of c1 we ge 七 by {::::: 1), 但 2)

,

c

2

lY

1

-a

1

丫

c～~

11

c

as re 吽 re d .

1

We leave the verification of 七 he second consr 七 ain 七 on 七 he scheduler as an

竺 r ci se 扛 1 Chapter 8. I七 si property than (:::: 4) •

no七 har d , bu 七 uses a s li gh 七 l y rrore general

CTlAPI'ER 4

Cas e s 七u 过 e s in val ue- 立卫 m田让 cat i on

* 1. Revi 窃

So far, we have seen h™ beha 过 ours (STs) may be built using six kinds of opera已on, 七匀 eth er with the all - iroportan七 us e of recursion. 吐 e opera 七 i ons fall into two classes:

* + 1. 历心已c opera已ons (Chapter 1)

Inac七i on Sunmation Ac已 on

N工L

＋

µE A u 忨｝

The dl\_Jnamic oper a 七i ons build nonde七e nrdni s七i c s equ en 七i al behaviours.

* + 1. Sta 已C opera已ans

C中 贮OS让 i on R笠 3r七 i ct io n Relabelling-

\ S

［

(Chapter 2)

(a E f:.)

]

a

—

The static op江a已onses七忒)li sh a fixed l inkage s七ru ct ur e arrong concurrently active beha过ours .

'Ihe 笠 l es gi 戍 m were static canbinations of sequen 七 i al behaviours, yielding sys七ems with!兰竺 l ink age s tru c 七ur e . Bu七 d 泸1am.ical l y-evolvine,

structures can be gained by defining recursive behaviours involving CCillpJSition. The poss ib il i 七 i es ar e 叩 且 t e rich; we give an example, not for i 七s usefulness (whi中 is doub七凸丑） but to i ll us 扛 吐 e the pcr,,er of CCS•

F扛 玩 ， l e 七 us define an op玉石七i on 'vlh.i ch has wide appli ca已 on. 工f

x: L, y: Mand L n M =¢, 豆 th S E L and a E M, the chaining opera 已 on

,,...,\_ is given by

x.-y = {x[o/SJ I y[o/aJ)\o

where o i/: names (L u M) • In pictures:

。百 0a 三

(See§8.3 for a proof tha七r 、i s as soci a 七i ve ; this even holds if LnM-;,,¢.)

妇 consi der in particular p:fo,e,y} and q:fo} given by P = aS丫 (p ""'p) , q = aq

and consider the foll<:JNing der i va七i on:

卢

气p 气 气 --

PP"'P...--.. P"'q

4

罕 p pr 、p p P " p p,..-、p q

.• e 七c ••••.

还 er n a ' s , 2n - 1 西 (and no 叩 r e) can have occurred.

尸

se 4.1 (F= fun). Descrilie the behaviour of p q a bit n=e

precisely - e.g. 四 many 丫 ' s 至 have occurred after n a's?

Exercise 4.2 Build a counter of sort {t,0,1;}

which (i) Can always be incrarented by an ,-exper.ircent;

也 ） Can be decreroonted by a o- exper.ircen 七 if non-zero;

(iii) Can admi 七 a 1;;-exper.ircent only when i 七 is zero.

至 ： in state n, 让 wi ll be sarething like a chain of about n cells. Increroonting mus七 in cr eas e the ce ll - ooun七by one; decrementing mus 七 dec rease the cell -coun 七 by one by

causing one cell to die - i.e. bea:ne NIL. You may need a cbubly linked 中 ain, built by a suitably generalised chaining

operator, and looking like

..夕. 、,:、o :: :o:乡：：Q， :恤，，，＇

趾 our calculus so far has an 扭 portant restriction which makes

it inadequate for prograrrmingi all carmunica七i on is pure synchroni zati<立and no data-values are passed fran one agen七 to another. True, we could in princi ple ' 迳 ad' the contents of the counter of Exercise 4. 2 by seeing hr:M nany decrerrents (<S) are needed before a ?; { te s 七 for zero) is offerred.

'!his would be cuabersane, to say the l eas 七 ， and for the counter as specified it would des扛 oy the coun七 s to red in i 已

So we now proceed to a generalisation of the algebra. In doing so we are f or ced 七o abandon o立 ST in 扫文r平 eta 已on. Wha 七 七akes i 七 S place mus七 wai 七 七i ll Chap七er s 5 and 6; me.王飞mil e the reader m担 七realise tha七 一 f or exar平）le - the equal i 七y S:yI 让心1 be 切 een our rrore general behaviour 郅 pr ess ions is no七 expl ain ed in this chap七er .

* 1. PassinCJ values

灼

Consider the s 叩 l e behaviour

p = a. 8 y p 6

I 巳 s no more than the cycler of Exercise 2. 7, 了

bu 七 i f we think of p玉过＿七i ve labels (a,$) as accepting inpu 七 p ul ses ,

and negative labels (y) as giving outpu 七 pul ses , then p beca:res

a behaviour which "gives an outpu 七 whenever i 七 has received two inputs" (the inputs being demanded in a particular order).

Suppose th a 七 an input a 七 a. consists of rrore than a pulse; i 七 si a value (an in 七 eger , say) • 'Iha 七 i s , a 七 七 anpting an a.-exper.inent on p

consists of offerrinCj a value 七o p at a•. p's behaviour as

p = ax.---

We may then wish to r epr esen 七

where x is a variable (s u 江 o se d to becorre bound to the value received

:in an ci-e 笃 竺 扛 已 ），and - - - is sare behaviour expression dependen 七

upon x, i.e. canta年年g x asa free variable. We say th a 七 th e variable

x is bound by ci , and i ts 竺罕 i s - - - .

('!his is very familiar to anyone 如 kn 叩 s th e 入 - ca l cul us ; th e 出 f f er ence here is that any positive label ci may b 扭 d a variable, whi l e 扛 1 the

入 - cal cul us th er e 担 onl y one b 年 der - th e 严 1 n 兀 ）

We can go further, 江 our aim 七 o 扛 ans fo 五 n p 扭 to a behaviour whose outp u七 val ues depend on i ts 江 pu 七 val ues , and write

p = ax. Sy. - - -

Here S b 扫 ds the variable y • Note th a七 the scope of x is Sy.- - -, while the scope of y is jus 七 - - -. (It would be s 七 upi d to write cix.13x.- - - since then any occurence of x :in - - - would refer to the value bound by

S to x ; the value bound by ci 七o x is inaccessible.)

Su汗>OSe we wan七 th e sum of x and y 七 o be output at y•

吐 at is, in g ene工al for negative la扫ls, a七扫珀:q:itin g a y- experirnen七

on p consists of 担 四 立 巧 a value fran p a 七 了 ． 咋 us negative labels do no 七 bind variables - instead th 匀 quali fy value expressions (which ma.y contain variables)• So we write

p = ax.sy.y(x+y) .p

I 七 i s n<:M proper to 七 alk of an " a v-exper.irnen 七 " rather than an

,,a-exper.irnen 七 " , where v is the value sul::rni 七 t ed by the c:bse 吓 er , and similarly of a "y v-experirnent" where v is the value received by the

中 s erver. So, generalising ther el a已on - of§3.3, we say

入

p 兰 叩 means "p admits a 1v-e,中立irnen七， and can

tr ansfo 血 to p'asr esul七II •

(No 七 e the different sense, acoording to the sign of 入 ． ）

As a general rule then, we can state

必 B - 尘 仁 今 B{v/ x}

where v i s 翌 y value, B is a behaviour expression, and B{v/ x} 哇 ans the r es ul 七 of replacing all unround occurrences of x in B by v.

Ands imil ar 切 (IIDre simply)

沁B B

for the particuli3]'."\_ value v 。 So the follcwing deriva 也 n is possible on p :

一

p = ax.sy.y (x-ty) .p

a3

/34

/3,Y. y (3-ty) .p

, y (3+4) .p

立 P

(See§4. 4 for :rrore abou 七 华 r i va 已 ons . )

妇 we have hardly anything :rrore to add to our language before fin 出 ng tha 七 迂 can be used convenien: 七 l y for programning. As for i ts 江 te r- pretation, we can 江 traduce a general 担 ed fo 血 of ST which we call Ccmnunication Trees (匀），bu七 f or the pr es en 七 we wish to rely on 扛北ui七i ve

understanding.

We shall usually be handing 妥 平 re ss i ons of the fonn

Ia,x.•B. + Is,E .•B'. + I-r-B"

i111 jJJJ k k

where Bi,Bj, are behaviour expressions, the x. are variables,

1

and the E. are value expressions. As for expressions involving

J

carp;,sition (I) and the other opera 已 ons, it will be enough to look

at a simple ex; 五 叩 l e and then give a generalised Expans io n 节 1eor 年 (§2.5).

Consider

B = (必 B1 + 13y.B2) I ;v.B3

We expec 七 as um of 4 七 e 五 芯 ， one involving -r B = ax. (B1 av.B3) + Sy. (B2 lav.B3)

＋社 (( ax . B1 +By.B2) I B3) + 立 (B1 {v/ x} 困）

Note that the "label◄' T does no七 bin d a variable or qualify a value expression. We shall also reserve the right to use other labels in this simple way when they onlyr e 芦 esent synchronization. In fact we

shall all叩 a posi 七i ve label to bind a 七upl e x = x , ••• ,x n of (dis 七in c 七）

variables, and a nega 七 i ve label, to qualify a tuple E = E , ••• ,E of

1 n

value expressions; then for pure synchroniza 七 i on we jus 七 use o 一 七 upl es .

We shall use the 诠皿巠竺 to canprise the prefixes

～ax－,SE and T,

and use g to s七and for a guard. Di jks 亡 a [Di j J in ven 七ed the no 七i on of guard, to stand for sane condi 过 on 七 0 be 哇 七 be for e the execution of

a program part. 工七 i s natural to adapt i七 七0 七he case where 七he con di 已 on is the acceptance of an offerred carrnunica 已 on, as Hoare [Hoa 3] has also done 江 his CSP. We then find th.a 七 th e analogue of Dijkstra's

guarded cam. 叩 ds is provided by s urma 已 on; wer ef er 七 o an expression 飞 ·I\ as a sum of guards, and call each g 贮旯 a 空 空 过 of the expression. We denote the name of g's label by n蕊汜(g ) .

沁中至 i on 吁1eor 匈 (stated and proved as Theoran 5.8).

远 B = (B1 I … I Bm) \A , where each Bi is a sum of guards. 吐en B = 肛g . ((B1J ••• 国 I •.• 1B)\A)·, g.B! a sumnand of B. , name(g) <l: A}

1 m 1 1

＋沪. ((B I… 困｛仓/ xl l ••• I B'. J ••• IB) A); ax.B a sunrnand of

B.,

1

—-\_

J m 1

1 a.E.BJ ! a surrmand of BJ . , i"' ]}

pr:ovided that, 扛1 the firs 七 七enn, no free variabl e 年 鸟< (k 五 ） is bound by g. 因

The meaning of the Theorem is tha 七 all unrestricted actions and all

皿 emal cal 兀lUl'lica 已 o 芯 in B may OC< 亚 ·

Note that our language contains two di 玩 in c 七 kinds of expression - value expressions and behaviour expressions. Consider a 比 B ; E is

the f ir s 七 kind, B the second. We allow the following s止丐让e but import an 七 constru cts in our language:

1. Conditional behaviour expressions.

廷 E 生竺 气 兰竺 邑

where E is boolean-valued. Consider for exa:rrple ax. ( 廷 妇 0 旦 Sx. B 竺 yx. B)

1. Parameterised behaviour definitions. For exa:rrple:

a(y) = 必（兰 妇 y 生竺 函 a (y) 竺 yx a(y))

(ii i ) 血 1 variable declarations. We shall allow cons七rue岱 l ik e

呈 x = 6 至 y = l O 兰 B

and

B where x=6 and

y =10.

'Ihey mean exactly the same - n 年 l y, the s畔 as subs t i t u七in g

6 for x and 10 for y throughout B.

We hope that the language is s 江 屯>le enough to be understood in 七 ui t i vel y, without fonnal syntax. Exact fo 皿 at ion canes later!

* 1. An example - Data Fl<M

We will na,, shoo ha,, to build and verify a s 乓 le system which bears

a strongr el a七i on 七o the Data Fla,, Schema 迳 of Dennis et al [DFL] 。

'lhe task is to build a net which will caupute 2x for arb i tr 扛 y non-negative integer x , given 叩 nents for cauputin g 立 e primitive functions and predicates, and sane standard gating and swi 七 ching catp:)l 芘nts. 'Iha七 i s ,

密 want a net whose 比 出 画 our is observation equivalen 七 to

a= tx-.o2X• a (1)

(We shall of七en use t for input, ;; for outpu 七）• Firs 七， we define sa 妇

standard ca 讥辽年 ts .

1. Un 扛 y function a en 七

For 呻 i trary unary f血 i on f , we define the a 声

才7

1

DO f = 心 (f (x) ) . (DO f )

5

We shall only use s 乓 l e f's ; we are actually trying to build

the behavi ou工

00 bexp

where bexp(x) = 2x , as you can see by carparing (1) and (2).

1. Un叩 pre di ca te 芍 en 七

For 血 i tr ary unary predicate p , we define

扫

ASKp = ix. 旦P 位）主 ；； 1 比 （陋 K p)

竺 。x . (芯K p)

0

1

2 .,\_

Note tha t 廿 汜 val ue X 担 passed unchanged ou 七 of one of the

OU 七 pu 七 p::,rts .

(2)

1. A 9ate

＿

亨

TE = 1x.ox.y. TE

吐 e gate transmits a value unchanged, but m 屯 七

be re-opened at y to 迳 pea 七 ．

邓 G = ix. y. ox.'IRIG y$

（拉） A tr i 哼，er

Like a gate, but must be triggered (or 七 工 i gger sareone

else!) after receipt and before transmission.

1. A source

For ar bi tr 叩 constant value v , a pennanen 七 s our ce of v's is given by

古

＿

DOv = i .av. (DOv)

We use DO , because the ur 豆 巧 , function agen 七 is easily

generali sed 七o n-ary fmi.ction agents, and constants are

jus 七 o- a 工 y functions.

1. A sink

占

SINK = ix.SINK

For discarding unwanted values.

l

1. A swi 七qh\_

SWI'rul c 心（于 望 ·翋虹 + y了 ,,x.swr

### 立）飞去 乃

A generalisation of a trigger;

triggering 丫盒 selects ou七pu 七 port o . •

1 1

'!his is all we need for our example; i 七 i s no 七 a ccnpl e七e (or necess a 立 l y bes七） se 七， and i 七 woul d be in ter es 七in g to design a good se 七 of canponen七S whi ch 如 l d be s 严 adequate for a wide class of data-flow cc兀 puta 已 ans .

We would like to fac七or OU 工 des i gn in 七o a control par七 and a controlled part. Forthe oontrol part, it will be convenien 七 七 o build an agen七 obse rva 七i an- equivalen七 to

X 七扛飞恣

C呻 L = i x! y. • • • •y .'o. C严 L (3)

i.e. forinpu 七 x 迂 wi ll admi 七 X y -exper扛1a1ts followed by a o-exper扛汪m七， and return to its original'state'. we show the ne 七 for realising ca叮'ROL;

让； can be shown by Expansio n 七o sa 七i s fy an equa旦on like (3) with many

in tervening 召s, and this is observa已onequivalent to CCNT.ROL, as we shall see in Chapter 7.

C严 L I

1

'

，;t

i

I

I

午〉,---、-、＼古－＼－－：－－、－｝－勹

I , I I

古

I \ I I

I

＿＿三一一气 ,

、

！：

l ..,'I

，管

I

'.,.'I

...... - - - I I

0

一一令一一一------

飞尸 i ：

l l

I I

I I

## 宁、一- \_)

l I I

古

t

One can check for the right behaviour infonnally, by "ar 立 双 swin gin g" .

Note that the ini 七 i al s 七ate is restored, and tha 七 if ei th er r 七 i gger is

replaced by a gate then 飞 vertaking' can occur, yielding the wrong behaviour.

'Ihe oontrolled part, or body, is to admi 七 a value v a 七 l I , then af七er n y-experiments foll a1妃 d by a 5-exper imen七 i 七 wi ll emi 七千 xv a 七 o and restore itself. Tha 七 i s , we wan 七 to realise

BODY = 订 y . b (y ) 故 ere

b{y) = y.b(勾）＋ 瓦；； y.BODY

1，

了

:

I II

ET

一

一

＿

＿

＿

＿

g

户il l

(4)

-

5-

上l

＿

EY

忙

|

＿

{、』，字， 二

－－

，，

-一

-

－

三

－

、

l -

­

、、

**、**

＇

，

·

，

,

、一

, '

0

\

l

、

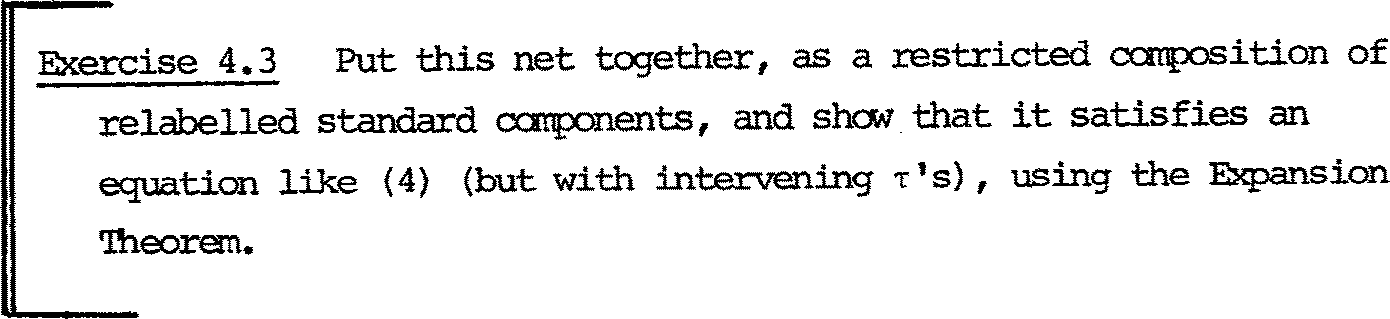
：

II `I1 g \

'

千 、、

I l I I II I\



Having established the behaviour of 虹 IY and CO在氏）L, i 七 i s a s 江叩廷matter to pu七 七h臼n together in su中 a Wey' tha 七 an in pu七 X to the whole system f irs 七 gates a 1 into BCOY, then enters ca 叮 沁 L itself.

'lhe outer pair of gates (p 运 毗 als o in 邸 Y and CCN.l'ROL)

prevent overlapping of successive carputations.

Il i

is to

一0

111

DO bexp

、

、

＿

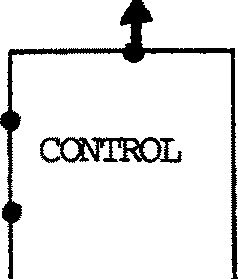
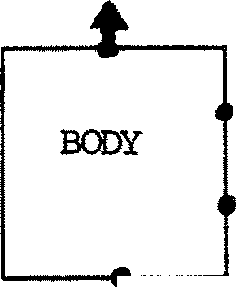
l

1 古古

-

-

:

呵·------

I I I II l III II I l I lil l

`i l l I I I l I II

I

l

千

，

III

一叩- -- --

'

，

夕＿

＿

＿

-

\:1

,

\

．

[

吐 e ex吓 l e s比双s h叩 nets may be bui 止 in m丈 ule s which a年verified separately. 廿 1ere are two remarks:

s

七 ev upa

5 heb.

）

七

d i 2

, j1 `

na 七

d

@ ahtna

）

1

8

ybh , IL

二

s

eeS

g

，

二

saes 主

iL,

L C t

C

七

saevr

dna1 e

ehttni

二

g

mnisa

七 七

aereub

rTht ，

二

A

4

七 00

尸

(i ) 吐 e use of the Expans 的 n Theorem is tedious, bu 七 as we

缅 已 oned earlier i 七 can be irechanised.

也） We have impli c让 l y as s ur芷戏 th a七 if two behaviours are observa已on equival en七， then replacing one by another in any sys权郡 oontext will yield an observa已on equivalent

s ys 饭皿 （吁让s is wha七 j us 七if i ed our tr ea troen七 of BCDY

and cm吓立 - replacing them by their s产cifi ed behaviours) • This ass 叩 ti on is justified for the oontexts we have oonsidered, bu 七 i 七 i s not trivial to prove that this is so.



Exercise 4.5 Construct data flaw nets 七 o carpute the value of y :Eran

in pu七 va l ues x and y , for each of the following prograrrs:

(i ) 竿 p (x) 生 (y: = f(x,y) ; x:= g(x))

(ii) while p(y) do (y:= if q(x,y) then f(x,y) else f(y,x)

x:= g(x))

You wil l 迦 s 七 certainly need sa:re other'standard'agents, and a di ff eren七 W苟r of h 五 1dl in g predicates - since the ccms七工UC七 ＇邸 k 寸does n ' 七 g eneral is e vi艾习 well for non-unary predicates.

* 1. D红 i va 七i ons

In§4.2 we ga 凭 an exarrg;ile of a der i va 已 on of p = ax.sy.y(x+y) .p :

一P a3 - 84 - 一订

f3y . y (3 + y) . p

一～今 y(3+4). p

p •

S 血 l ar 切 ， B = ((ax.B + Sy.B)

2

1

I 的 平 . B3 ) \ 8 has der i va 已 ons

a5

I

B - (B1{5/x} sv.yz.B3)\13 ;

B 工> (B2 仅 / y} I yz.B3) \ 8 立 (B2 仅 / y} I B3{7/z})\13 •

A general derivation takes the fom

叱巧

B) B)

历v 2 B ->...

µnvn B

1 2 n

(which has length n) or may be :inf:inite. We shall often vlr:ite a derivation of length n as

µ1v1µ2v2

B 夕. >.

-rn

µv

. .E 乌 书

n

, or

坰汽·乌骂 µVnn

B)B n

we can abbreviate B - 丑 I by B 圭 B' (n 2".0 )

m n

.and abbreviate B -r .µv.-r i B' 切 B 坚 B' (m, 归 0 ) .

(see also§3.3).

A c叩 l e te derivation is either an 血 血 te derivati 中 ， or a f扫止e der拉a七io n which canno七 be extended (this means Bn = NIL) •

尸

-t

Exercise 4.6 Us 扭 g equa七ions (3) and (4) 年 4. 3, wr 让 e s咋 of the

tians of BCDY, ccmroL and [OOIJY I com<JL)\y\< • deriva已ons are there?

叩 lete

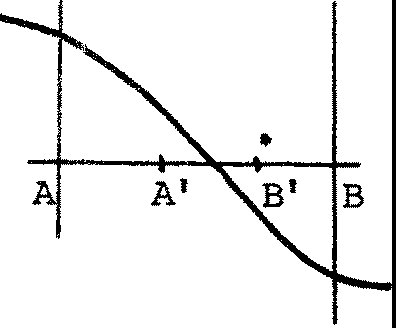
A 立 项 ?l ete finite der 坟 ati onof B represents a possibility tha 七 B

canr ea 中 a poin 七 wher e no furthe 工 acti on is r:ossible; i 七 may dea dl o改 ．

* 1. 5 An exanple - Zero sear 中

We wan 七 to s e 七 two agents p and q to'WOrk together in finding a root for the equa 七 i on f (X) = 0 in the range [A,B] , for a con 七 江 uous function f, 比 ing tha 七 s uch a r oo 七 exi s ts - i.e. f(A) x f (B) :,; o.

工 七 is na 七 ur al to make p and q calculate f (A') and f (B') re spec已vel y, and concurrently, for 切 10 internal p::>ints A'and B'.

f

If p finishes f irs 七 ， and finds tha 七 f (A') differs in sign f:ran f (A) , he can leave a message for q to cane and help him in the n 窃 in te rval [A,A'J, and begin to work wi thin 啦 s interval himself.

If he f 扛坴 f (A') to have the same sign as f (A) , then he should go to help q 扛 l the 扭 terval [A',BJ.

A A•' •B':

•

＋＋

• •

: A" B

He could choose a po.int A" in [A勹B'J or in [B',B]. Kung [Kun, Section 3] made the e l egan 七 s ugges 七 i on tha 七 the points A',B'should not tr isec七 [ A, BJ , bu七 r ath er divide i 七 s o tha 七 th e r a 已 OS AA':AB, B'B:AB and A'B':A'B are equal1 then in the case above A may pick the n 的 po.in 七 A" so th a七 th e n 窃 .interval [A',BJ is subdivided by the 沁 rking points in the s 畔 r a 已 o as [A,B] was subdivided.

'Ihis detennines A',B' as the 还辛 s ecti ons of

．~~．，~~

凡

A,B:

2

A A'B1 B

0 + 0 = 1 ;

、,

0 29,

．切，

-"-'

03凡

O凡

。＝竺.618

2

A 七 any rra:ren 七 th en , th e 工 e a 工 e two p::,ss ib i li 七 i es :

* + 1. p and q are both 沁 r king on golden sections of [A,BJ;

也 ） One of then is working on a golden section pain 七 ， and the other on a pain 七 outs i de the interval {because the other agent h 郘 s hnmk the interval) •

加 泗 utati on stops when the interval has been reduced to less than

sare prede枉立吐ned value'eps'.

AsKung observed, the algorithm can be iroplemmted by giving p

a local variable X (his working point) , q a 1。它让 va工iab l e Y similarly, and represen七in g the interval by a£ 匋 gl 中 al variables which either p or q may inspect and update, using a 立 i ti cal section for the purpose.

吐 us an outline program for P, using canven七i onal and obvious nota 已 on, is:

p = 竿 in te rval eps 生 CRITIC 辽 SECTI CN

辛canpute f (X) ; upda七e globals end

＇

．

s皿 l ar l y for q , and the whole program is

oobeQ-in P 11 q 竺 ·

T. 沁 l dner has given the carplete algorithm [ 皿 J. I am grateful to

1. Salwicki for dr 匈 ing my a 七 ten 已 on 七 o this exa:r 平 l e , whi 中 is a good

one to illustrate differen 七 ooncurren 七 pr ogr 玉 mrlng disciplines.

NCM in a sense p and q are shar 扛 巧 a :r:esource, i.e. the interval, represented by global variables. Hoare and others have made the poin七 tha 七 oode and data associated 泣 th shared resources are be七ter located a七 one site,r athe工 th an di s七rib uted over the sh扛 ing

agents; Hoare proposed 妇 让 ors as a device to achieve this m::x:lularity

[Hoa 2].

Here we propose tor ep 迳 sent the interval as a separate agent,

吐 thout the need for any extra progr 郘 m 江 g oonstruct for the purpose.

The idea is tha 七 p or q sulrnits the 迳 SU 让 of his eval ua 七 i on

to the interval agen七， whi 中 th en h王飞ls him a new evaluation poin七 p, working on X , is represented by

气

巹1妇

＿

p(X) =a (X,f(X)).aX'.p(X')

1

2

a2

-B

and q , working on Y, by

＿

q{Y) = $ 1(Y,f(Y)) .$ Y'.q(Y')

2

Notice tha 七 each submits a .E巳兰 , ar gurc它砒 and function-value,七o the interval.

The interval Int is pa正式记ter ised on A,B,a,b where ini 已all y (and always later) a = f (A) , b = f (B) and ax b:;; o.

By carefully reversing the direction of the interval when necessary,

In 七 ens ure s tha 七 a 七 acy t 扛 芷

p is working either a 七 H A, BJ (left section) or outside the interval;

q

" ＂ " r[A,B] (r i gh七 sect i on) 11 11 ＂ ＂

The interval agen七 has sort {a ,-a , 13 , 13 , p} , and delivers the root

＿ 1 2 1－2 －

finally a 七 p • 工 七 is defined as follows:

In 七(A, B, a , b) =

兰 IA-Bl < eps 兰竺；A. NI L 兰芝

<a1 (X,x) • 旦 X = A'then

廷XX 釭 0

＋

{．\ A'B'B

.) .) ---.

（女 {q}

＋－

为

竺 ； 凡[ A,A' J . In 七(A,A' , a , x )

-

-2

.

.

孚

％凡[ B,A' ] . I n七(B, 配，b , x)

• ( • • -( 一 看

，

竺 砃 I • 皿 (A, B, a , b)

+ 131 (Y,y). 兰 Y = B'then

r

t (q)(p) I

＋

.)

·

—

.

I (p ) 句）

'

旦 y x b s:O

生 82r[B',BJ.工砒 (B' , B, y , b) +

＋

.-)

..)

孛 132r [ B' ,A] . I n七(B' ,A, y , a )

兰 伊 I . In 七 (A, B, a , b)

）竺竺 配，B' = 见[ A, B] ,r [ A, B]

I (p)(q)

• < ••. (. I

1. (p)

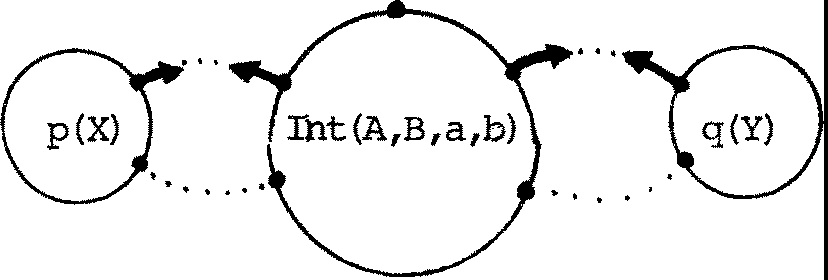
＋.--> •• > •

* 1. (叨

The canpl e 七 e systan is Sys(A,B,a,b,X,Y) =

(p(X) I Int(A,B,a,b) I q(Y)) \a 1 \a 2\13 \132

1

p

('!he arrai/S are marked assuming the case I A - BI eps.)

网 邱 do ,..e want to prove about Sys? s 立 屯 让 y th a 七 every p::,ssible derivation canputes a near-root of f in [A,BJ. (By a near-root z of f, we rrean a Z such tha七[ z - eps,z +eps J contains ar oo 七 ） M:>re precisely, we require

兰 (i ) a= f(A), b = f(B), axb:!:O,

变 (ii ) X = 见 [ A,B] or Y = r[A,BJ,

生 every crnplete deriva 已 on of Sys(A,B,a,b,X,Y) takes the form

Sys(A,B,a,b,X,PZ) 泣L

where Z€ [A,BJ is a near -r oo 七 of f.

工t ' s conveni en 七 七 o pr.ave this by induction on the size of [A,BJ, defined as the l eas 七 n SU 中 th a 七 en 叶 A - B j < eps. For size == O we have

Sys(A,B,a,b,X,Y) - PA

NIL

as the only caaplete derivation, and we are done. For size > o, we can use the Exp五沮 on 'lheo :rem to shaw the following, 故 i ch is enough to cc叩 l e te the proof:

Under conditions (i) and (ii) , every caaplete deriva 已 on of Sys(A,B,a,b,X,Y) extends a derivation

Sys(A' ,B' , a ' , b ' 次 ，切

where the parameters again satisfy (i) and (ii) , and

* + 1. if X = HA,B] and Y = r[A,B] then [配 ，B' J has smaller size;
    2. otheI.Wise either [矿，B' J has smaller size or [配，B'] = [A,B],

X' = 凡 [ A, BJ and Y'= r[A,BJ•

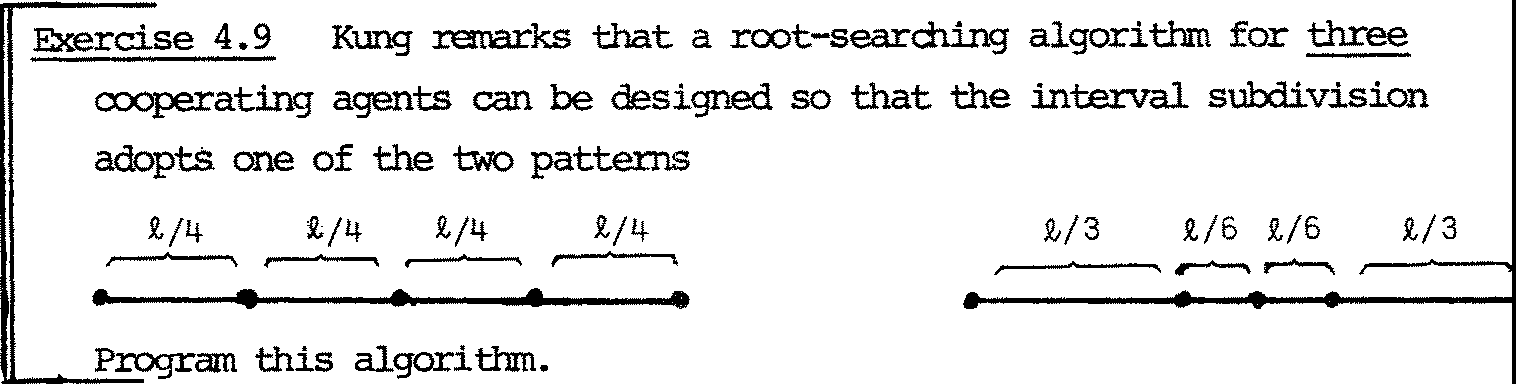
尸 二二：：：::

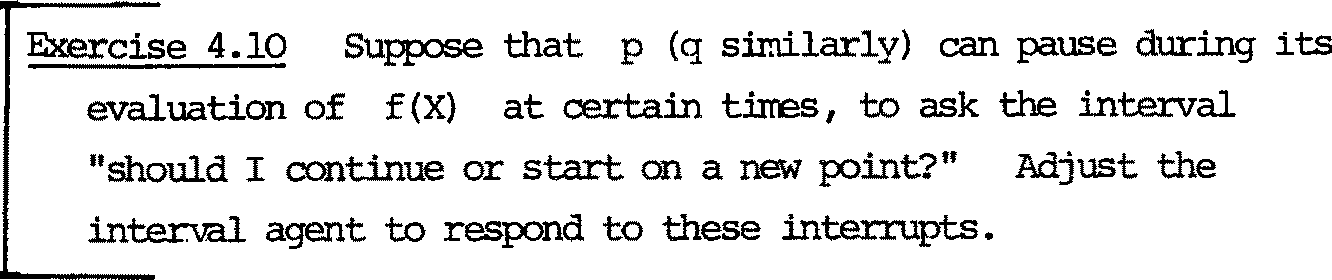
二 ys二二＇二

no 七 a l w 五 y-s after one.

尸二\_:工\_二sn 二th言:. 了，了::二 -

and s 皿 l ar l y a2, S 2; 因 s easy bu 七 no 七 ccmpl e 迳 l y 扛 i vi al .





CHAPI'ER 5

Syntax and Semantics of CCS

* 1. Introduction

We have seen s呻 exampl es of expressions of ccs, representing roth programs and their specifications. We s 五 v tha 七， wi th the introduction

of value-passing, we had to abandon the s 叩 l e interpretation of behaviour expressions as synchronization trees, but in§4.2 we talked of atanic exper乓 nts on behaviour expressions (or on the behaviours for which th 句stand), and this was developed further in§4.4 on der 坟 ations.

We are 正 r eady to present CCS precisely, and to def:ine precisely the atanic ac已ons (and hence the der iva七i ons ) of every CCS progr郡 . On this basis, we proceed in this chapter and in Chapter 7 to develop our

central notion, observation eauivalence of programs. Fram this i 七 i s a short step to a ccngruencer el a 七 i on; 巳 o programs are observation- 正 m 七

iff they are observa已on equivalen七 (i.e. indis七扛飞几让shabl e by ooserva已on) in every oontext. Our proposal is tha七 an obser 炟七ion congruence cl as s 主a behaviour, so tha 七 CCS is indeed an algebra of behaviours, in which eru 土progr 郡 stands for its congruence class.

'1his main developoont is independent of 七 迳 no 已 on of ST. STs may nCM be regarded as a f irs 七 approxirna 已 on (not suffi ci en 已 y abs 扛 act ) to a 邓 1 of CCS without value-passing, and in Chapter 6 we shcM how they may be generalised to crs (cx:mnunicaticm trees) to give a f irs 七 appro ximati on

to a 玉 el of 心 S 至 val ue-passing ; again, the ma.in deve l 匀 兀 1911 七 i s in:­ dependen 七 of CTs, whi 中 are only discussed to aid under 玩 anding. 如 en we eventually def:ine observation equivalence over p 正 平 ams in Chapter 7, it will

look jus 七 l ik e the correspcmding def:ini 已 on in§3.3 over STs, which general­ ises to CTs in an obvious way. Indeed, we expec 七 to find tha 七 七 o programs are equivalen 七 i ff 廿 1e corre s 严 d:ing 函 are so; in 廿 砒 cas e CTs, 出 ough no七 七echnical ly essen七ia l , fit na 七 匹 all y into our picture.

'lhis chapter is devoted to a oong 玉 nee over programs which we call strong conqruence, s:ince i 七 i s s 七五>nger than the observa已cm oong:ruence

S 七 udi ed in Chapter 7. By approaching our proposal in two s ta ges 炬 in 七 r五

duce the properties of behaviour gradually, and with greater insigh 七 than if we tackled observation ccngruence imnediately. In fact we even subdivide the f irs 七 s ta ge in this chapter, approaching strong oong:ruence via an even

stronger relation called direc 七 o 平 让 val en ce .

The CCS l an 晔 ge was in tr 过 uced in the author's "Synthesis of Ccmnunicating Behaviour" [Mil 3J. Ha,rever, the semantic specification by derivations was no 七 gi ven there in detail.

* 1. 匈ntax

Val ue 吟 re s s i ans E

Value expressions are bui 迁 f ran

* + 1. Variables x,y,…
    2. Cons tan 七 syntiol s , and function symbols standing for knCM11 total fmctions over values

using oonventional notation. We also alla,, tuples (E , … ，En ) of

袒 ue expressions.'Ihus each value expression wi th ou 七 var iab l es stands

for a uniquely defined valuei we shall not worry abou 七 the distinction

between such expressions and their values.

We shall also avoid details abou七 th e 七ype s of values and val ue 仑平r es s ­ ions, tho 屯 h we shall have to 芷 mti on scree syntactic ronstr 忒 nts depending on such details (which are standard)•

Labels, sorts andrelabellin9

As in Chapter 2, our labels are A = b. u ti , together with , •

We use o.,$, •• to range over !),, 入 over A, andµ,v,… to range over

Au{,}. A so rt L i s a subse 七 of A; to each behaviour expression B will be assigned a sort L(B). t

A relabelling S : L+M between sorts L and M is as in§2.2.

Ha,;ever, same positive labels o. will be used to bind ( 七uples of) variables, and then will qualify ( 七 uple s of) value expressions; we mus 七 ens ur e

tha 七 S preserves the sign of such labels (i.e. S(o.) E b.) • Moreover, in a Cal 平 l ete treatment we should ha 凭 切 ass ign 七 刃 文 王 to value v 釭 i ables and value expressions, hence al so 七 o labels, and 切 ensure tha 七 r e l abe ll in gs respect the types of labels·. We will avoid these details; 凸 ey need care,

bu 七 woul d only obscure the nore importan 七 as pec ts of se:nantic,s

七 o discuss here.

中i ch we wan 七

t We shall only 晖 et \_ 旦幸 s orts in ex 郘 p l es . 阳 炬 ver , all we n 垒 d 切

assume - for 七 eel 飞 江 cal reasons - si 七 ha 七 A is never exhausted. Infinite

sorts may be of use; see the end of Chap泣 6.

Behaviour i den 七 i fi er s b

粔 pr es uppose a collection of such i den 已 fi ers , each having preassigned

1. an 兰 甡 又 n (b) - the nu:nber of value parameters.

也） a 罕 L(b) .

We assune that the meaning 。f such identifiers is given, often recursively,

by a behaviour expression. For example, in§4.5 we gave meaning to the behaviour i denti 丘 er p by

p(x) = et (x,f(x)).ci x'. p(x')

2

where n(p) = 1, L(p) = {;;1, 沪 ．

Again, a canplete tr eatmen七 woul d specify not jus七 an 旦王主立 bu 七 a

罕 (i . e . l is 七 of par amete 工 臣 吓 沦 s ) for each b.

Behaviour expressions B

Behavi our 呤 re s s i ons are fonned by our six kinds of behaviour operator (§4.1), by par罕 terising beha迈\_our identifiers, andby conditionals.

I廿s convenient to present the fo立洹已 on rules as a 七忒:>le (see belCM), giving for each expression B its sort L(B) and its free variable se 七印 (B) •

We smuld regard the language given by the table as a core language, which we are free to ro也 平d by defining derived behaviour operators (the chaining canbinator r-. of§4.1 for example) and by alternative syntax for cx:mronly occurring patt 缸 :ns.

In wha 七 f oll ows, we shall use

B{E /x , ••• ,E /x }

1 1 n n

七 0 denote 七 he 罕 s ul t of s ubs 七 i t ut 江 g expression E for va工i ab l e

1

x. (ls i sn) a 七 all its free occurrences within B • San: 式 :.imes we shall

l.

abbreviate vectors ( 七 upl es ) of variables and expressions as 又 an d 至 ，

and write a substitu 七 i on as

B { 危及｝ ．

(As usual, such substi 江 七 i ons may require change of bound variables, 切

avoid clashes.)

SYN1'AX TABiiE FOR BEHAVIOUR EXPR 蕊 I 怎

I

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fonn |  |  | B" |  | L(B") | 印 (B") |
| Inaction |  | NIL |  |  | ¢ | ¢ |
| Sumiation |  | B + B' |  |  | L(B) u L(B') | FV(B) u 四 (B' ) |
| Action |  | ax1 , ••• | ,xn | * B | L(B) u {a} | FV(B) 一{ X1 t"• 1Xn } |
|  |  | ciE1 , •••  T.B | ,En | * B | L(B) u {a＿}  L(B) | 式 (B) u UFV(Ei)  印 (B) i |
| 立 s i tio n |  | BIB' |  |  | L(B) u L(B') | FV(B) u FV(B') |
| Restriction |  | B \ ll |  |  | L(B) 一 fo,动 | 印 (B) |
| 距 l abe ll in g |  | B[S] |  |  | S(L(B)) | 印 (B) |
| I den 七if i er  Con 主 ti onal |  | b(E 1, ••• , En(b)）  I if E then B else B'I | | | L(b)  L(B) u L(B') | UFV(El..)\_  i  印 (E) u FV(B) u FV(B') |

I

'!he table shows how B" of sor 七 L(B" ) may be bui l 七 f r an B,B' of sorts L(B) ,L(B'). Parentheses 桽 to be used to make parsing 1.IDambiguous, or to enphasize structure; to avoid excessive use of parentheses we assume the operator precedences

{ Restriction > Action > Carq: 运 i ti on > Surma.lien. Relabelling}

'lhus for example

B l-r: .B'\a + B"[S] 晔主 (Bl ( 石 (B' \ a ) ll + (B"[SJ} •

* 1. Sanantics by deriva七i 牢

We pr oceed 七 o define a binru:yr e la t i on µV

over beha: 过 OU 工 expr es s i ons ,

for ea 中 µ EAU { -r} and value v (of type a 印 ropr i ate toµ) • B 骂 B' may be read "B produces (or can p 志 uce ) B'underµv"; thus if B,B' are in the r el ati on 应 ，a particular atani.c ac 已 on of B - resulting in B'- is indicated.

Referring back to§3.3, we are taking behaviour expressions to be our agents; 扛 劝 扛 ·ds the end of§3.3 we chose STs as agents, and we shall see in the next 啦 pter how to regard CTs as agents.

Note tha 七 i s a particular case of ourr el a 七 ions , since the only value of type appropriate to , is the 0- 七 :upl e .

咋 e r e l a已 ons l: 斗 ar e defined by induction an the structure of behaviour expressions. This neans that all the atanic actions of a ccm- pound expression can be inferred f 叩 the atanic actions of its ill!贮netn(s ) .

Such ar el a已on, th o屯h not indexed as here byµv , probably f irs 七appeared in connection with the J..-calculus. I 七 was called a reduction relation, and the clauses of its definition were called reduction rules. Gordon Plotkin f ir s 七 ma.de re aware of the power and fl ex.i.bi li 切 of such relations in giving meanin g- by-e v吐 ua七i on to prograrrming languages. (In

passing we may note tha 七 the original def ini 七 io n of AI.I. 文 ）iL 68, though strongly verbal, is 扛 1 essence a se 七 of reduction rules.)

InactJ.on

NIL h 在 ; no atanic actions.

Stm:陋已on

F 五 :m B

µV B' in fer B + B µv B'

Fran

1 - 今

µv

B2 -B'2infer B1 2

1 1 2- 1

+ B µv

B '2

'Ihus the atanic ac 七 i ons of a sum are exactly those of its surrnands.

一

We adopt 如 fo ll owing pr es en ta 七 i on of s u 中 inf er en ce rules:

Sun B1哭用

(1)

B1+ B2 这 时

(2} B2 皿 男

扣 B2 坚另

Ac已 on

Ac七 十

(1) ax1, ••• ,xn.B a(v1,···1vn; B{v/x1, ..• ,vn/ 吩

(2) ;:;v1, ... , vn . B 玉 二立 人 B

(3) , . B 乌B

Notes: (i)'.Ihese are no 七 inf er en ce rules, bu 七 axi ans .

1. Act+ (1) holds for all 七 :upl e s (v

—

1

,… ，v n) (of appropriate

七yµ! for a), while Act+ (2) holds jus 七 fo r 七 he 七 :upl e

qualified by a •

1. See§5.5 below for why we consider only values

v1, ••• ,vn

(no七 exp re ss i ons E1, •.• ,E)n

in Ac七 (2) •

C叩 s i t i on

1. B1 -µ-v B'

1

1. B2 骂 鸟

咖咖咖

µV

Can

B1!B2. ,.Bi 1B2

B1.入.V\_B1'

B1 1B2

B1 j B2 鸟 旷 骂

入

V I

B2---,,,B2

B1 1B2

I I

沁 tes : (i) Can+ (1) and (2) express the idea that an action

of B1 or of B2 in the c arpos 平 on B1IB2 yields an action of the c 叩 si te in whi 中 the other canponen 七

is unaffected.

（拉） Can+ (3) expresses tha 七 ccmmmi ca ti an of canponents yields a T-action of the canposite.

Res 扛 i cti on

µv尸

B - 违 ＇ ， 归 {a , 记

B\ a 皿 B' \ a

．眉．

Note: the side condition ensures tha 七 B\ c:. has no av or av

actions.

Relabellin9

腔1

I

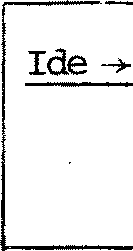
B 皿B'

B[S J罕 矿[ S J

Note: recall our oonvention th a 七 ST = T .

工den 七if i er . Suppose 七 ha t identifier b is defined by 七诠 (possibly recursive) clause

b(x1, ••. ,xn( b ) ){= 旯 四 { 8b) 三 {x1 , · · ·,xn(b)}) We shall discuss suah definilions smrtly. Ourrule is



{v/x1, ... ,vn(b) /xn(b) } ---t,B'

µv

b(v1, .•. ,vn(b µv B'

))

No七e : the rule says, in effect, tha 七 ea ch par 五 汜 ter i zed i den 已 fi er has exactly the s 郘 芘 acti ons as the appropriate instance of ther igh七hand side of i 七s definition.

condi已onal

B )JV,,,B'

B 骂 B'

Can 今 (1) 1 1

(2) 2 2

- — 1- 2 1

if true then B else B -B)JV'

if false then B else B

-µBv '

— 1 2 2

沁 七 e : As with all value expressions wi th ou 七 va 已 ab l es , we assurre th a 七 bool e an- va l ued expressicns eval uate 'au 七 ana 七 i ca ll y ' to their boolean values. See§5.5 belc,; for why we need no 七

consider value-expressions conta 皿 ng variables in these rules.

* 1. Definin9 behaviour identifiers

殴 shall 立 双 ass哱 tha 七 e very behaviour identifier b is defined

by a clause

故 1ere x 1

, ••.

b{x1,…，xn ( b ))

,x n(b) are distinct variables, and'Wh桽 印 ( 压 { x 1 , .•. ,x n(b) }•

The symbol• 飞 s preferred to'='since we are no 七 ye 七 ta lking of the behaviours denoted by behaviour expressions (so'=', inthesense of equality of meaning, would be ou 七 of place) , and also because we will later in this chapter use• = ' 七 o mean i den ti 切 between expressions.

＼ 祀 th us have a collection of clauses defining our b's, andthey may be mutually recursive. Alth 叩 h no 七 act ua ll y e ss en 七 ial , we Sha 耳 江 邓 心 se a slight consr七 a in 七 on the collection,'Which will forbid such def ini 七i ons as

b(x) {:: 涵. NIL + b(x + 1)

｛气{= b 2 十 o..b 3

or

b2 {= b1 匡 b4

in which a behaviour may'call itself recursively wi tho u 七 pas s in g a guard'.

咋 us the foll 叩 ing 扭 penni七ted :

＿

b(x) {= ax.NIL + ,:.b(x +1)

｛气{= b2+ a.b3

and

b2 {=

,,: b1 匡b4

沁 re precisely, we say tha 七 b is tm9U< 江 ded in B if i 七 oc curs in B

wi tho u 七 an enclosing guard. Ther es 七立 C 七ion on our defining cla涟for theb'sisthat there m出3七 be no infinite sequence b. ,b. , ••• such

1.(1) 1.(2)

(j+1)

that, for each j , b.

l.

is unguarded in b1. .(J.) • (I n 七he forbidden

exarrples above there are su 中 seqrences : b,b,b, •... andb1,b2,b1,b2, ... respectively.) Further, for correctness of sorts, we require

L(8b ) 三 L(b)

When the above constraints are met, we shall say that the behavi ou 工identifiers are guardedly well-defined.

* 1. Sorts and programs

OUr fol'.T'lation rules ascr 担 a unique sor 七 L(B) each behaviour expression B ; we wr让 e

B: L(B)

七 o mean'B possesses sort L(B)'. Formany reasons, it is convenient

七o a ll 叩 B to possess all larger sorts as well; s o 炽 decl a 工 e

B : L & L :: M 叩 li es B:M

For example, this all 氓 s us to make sense of an expression like

NIL扭/ a]

since Bia : {正{ B} is a relabelling, and NIL : {a} since NIL : ¢.

An 江 iportan 七 p roperty of atanic actions as defin 玄 江 §5. 3 is the following:

Proposition 5.1 If 一B 叹 B' , and B:L,then

\J E Lu{T} and B': L

Proof By induction on the length of the inference which ensures B - 叹- B', using the ascription of sorts by the fonna 七 i on rules. 因

Although our rules for a 皿 c actions a 印 ly 七 0 arb i tr 叩 behaviour expressions, they fai l 七 o describe fully the meaning of expressions with free var 坛bles. For example, the rul e 罕 尹ves no ac已on for

a(x+ 1). NIL

and 2 平 sa ys nothing for

旦 X.!0 旦兰! ax. NIL 竺 B (女 ）.NIL

Clearly they could no七 de tennine the actions of these expres 廷ons , since actions involve values, no 七 var i abl es , and in the second example even the 迫 竺 of the poss 血 e action depends upon the'value'of x .

他 choose 七 or 笠 ；ard the meaning of a behaviour 呤 res si on B with free var 担 bles 艾 as de tennin 忒 by the meanings of B{ 劝 咬 } for all

val ue- vec 七 or s 芍 ．

Def 扭 i ti on We def 扭 e a program 七 o be a closed behaviour expression,

i.e. one with no free va 工 i abl es .

归 the f ac 七 tha 七 our rules describe the meani 江 s of pl. 屯 J;.dl llS

s a 七i s f actor il y is due 七 o the fo ll oo 江 g:

B

Pr 吓x豆已on 5.2 工f

a program.

is a program and B 鸟 B' , then B' is also

旦 By in duc 已 on on the length of the inference which ensures

B-µvB'.'Ihec中diino tefree variables of each the subst 迂u已on involved in Ac七 (1) , are critical.

, and

* 1. Dir ec 七 equivalence of behaviour programs

因

(In§5.6 and§5.7 we are concemed only with programs).

We :fl叩 tak e up the ques 己 on, posed in§5.1, of 劝 让 ch behaviour

P 立 r ams possess the sarre derivations; this will yield an equivalence relation, which will also be a c 叩 ruence - tha 七 坛 ， any program may be replaced by an equivalent one in any context, wi th ou 七 af f ect ing the behaviour (derivations) of the whole. For example,

B+B' and B'+B

are 出fferen七 p rograms, bu七 we clearly e习立玄土 七h 王n 七0 be 扛让er ch 郘 geab l e in this sense.

A f ir s 七 a pproxirna 七 i on to what vie want may be called direct E!C!l 耳 val ence ;

we deno 七e i 七 by 三 ， and define i七 as fo ll o峦吐

1~ 勹::

B1 三 B 2 B

(B and

1

B 2 are directly equival en 七 ） iff for evE 江 y

(W 扛 nin 中

B1 鸟 B 今 B 产 B •

三 i s no 七 a congruence re la 七 i on.

For example, we may have

B 1 三 B 2'

bu 七 in general

BI B1 申 B I B2

0.. 皿 j B

1

0. .:r:皿 I B2

For example,

NIL l B

；二

1} no 七 乒 ti cal !

I

NIL B2

But the congruencer el a 七 ion we wan 七 wi ll be :implied by 三 ， and so the follCMing laws for 三 wi ll hold for the congruence also.)

In wha七 f oll o,,s 让 i s often convenei n七 七o 1 武 g stand for an

arbitrary guard 奴 ，洷 or T .'Iher esl七 Sg of relabelling a guard is given by s(或） = (Set)x , s 涵 = (Sci) 苍 and ST = T •

哗 name of the label in g is denoted by name(g)•

节 1eorern 5 . 3 (Di re c 七 E 于让val en ces ) • 'Ihe follCM;让灯 dir ec 七 equival ences hold (classified by the leading operator on the lef 七 s id e ) :

(1) B1 + B2 三 B2 +B 1

尸

* 1. B + NIL 三 B

(2) B1 + (B2 + B3 ) 三 (B1 + B2 ) + 鸟

* 1. B + B 三 B

Act 三 ax . B 三 ay . B 守jx} (change of bound variables)

where

～y are distinct variables not in B

Res 三 (1 ) NIL\ 13 三 NIL (2) (B1 + B2) \ 13 三 B1 \13 + B 产

(3) (g.B) \ 13 三{ NIL if 13 = name(g)

g.B\13 oth 笠 双 挂 汜

Rel 三 (1) 泣 L[S] 三 汇 L

(3) (g . B)[ S] 三 Sg . B[S ]

(2) {B1 + B2) [ SJ 三 B1 [SJ +B2[S]

归 in V运 of Sum 三 th e following notations are 1ID.i: 珀 ibi guo us :

I B. :rrearung B +•.• +B (NIL, if n=O)

1 1 n

乓扛n

归五； 江I } nore generally, where I is finite.

工 f each Bi is of fonn gi .Bi , we call such a sun a sum of guards,

and each B. a sum 旧 nd.

1

Can 三 I.et B and C be sums of guards.'!hen B lc 三 I{g. (B'I C) ; g .B'a sumnand of B}

＋ 肛g . (BJC') ; g.C'a smrnand of C}

+ I丘 (B' {动艾} I CI) ; 戎 . B' a sumiand of B

and ;; 吃C' a surrmand of C}

+ I丘 (B' I C屯农｝）; 一av～.B'a surrnand of B

and a 兑 C' a sumiand of C}

I de 三 Le 七 i den ti fi er

b

硕）三月乒反｝

be defined by 硕）{= ; then

Con 三 (1)

(2)

if true then B else B 三 B

1 2 1

—

if false then B else B 三 B

一一

1 2 2

红 To prove each 1 砌 i s a routine application of the definition of the relations -----V---►: • We consider three laws:

位） S um 三 (2) : B +(B + B ) 三 (B +B) + B

- 1av3 123

迳七 B + (B + B )一一B. This can only be due to

一一

一

1 2 3

either rule Sum + (1) , because B

µV

or rule Sun+ (2), because B + B B 叹

and in the l a

七 七 er case, similarly, either B B

2 - 3

B •

2 3 B , µV

or B µV

In each of the three cases, rules Sum 丑 1 ) and Sun +(2) yield

一

(B +B) + B µV

1 2 3 B •

Ther ever se 扭 li cati 立 1 is similar.

过） Res 三 (3) : (a.x.B) \ S 三 { NIL (S =a}

a.～x. (B\S) (S ;a, a.

酌 Ac 七 -+ (1) , the only actions of

a～x.B are of fo血

a~x.B

～

- av 江3{

~v/x~}

～

(for arbitrary v) .

吐us 喊.B) \a has no actions (since Res+ yields none} ;

neither has NIL, which se 七 七 l e s the case J3 = a.

For

(13 五 ），by 些 th e only actions of (五. B) \ 13

are

(ax~. B) \ 13

仅/ x}\ 13

(B\ $) 符床｝

and these are exac 七 l y the ac 七 io ns of

ci芍 B ～ ～

ax～. (B\8).

The proof for guards

一av～

and

T is similar.

1. Can 三 ： B jC 三 I ••• 十 I · · · + I···+I··•.

(We use X to abbreviate ther ight-hai 飞 :i side,)

le 七

B lc 鸟 D .'Ihere areseveral cases.

* 1. B -µv B" , and D = B" le

如

Then B has a surrmand

(by Can + (1)) .

g.B'for vmich g. B' 骂 B"

(by 三 ）．

This action mus 七 be an instance of Ac 七 十

frcm which we can also find tha 七 g. (B' jc ) 且 又 旷 C

{considering the three types of guard). Hence also X 骂 到C = D .

* 1. C 骂 C" , and D = BIC" (by 竺立旦））

咋 e argmen 七 tha 七 X JJV,. D is s imi l a 立

~ \_-c

— 祖 " , C .- C' an d 叹=, , D = B"ic•

(c) B au au

(by can+ (3) ; there is a similar case with a. ,a. exchanged) '!hen by Sum + and Act + , B has a s 中 芷 md ax.B'

and B" = B' { 动 如 ，whil e C has a s urnnand 忒 C' .

Hence, since X has a s ur 晔 诅 飞 (B'{ 动如le ' ) , we have

x 上 B" lc ' = D , as required •

We have now shown by (a) , (b) & (c) th a 七 f or allµ,v andD

Blc 鸟 D =? X 皿 D

and the reverse implication can be argued similarly.

闪

e

ye

.，hT

, I '

— 厂

三

secn

ne oC

1

avd

na

g

f

1-

O

er

onr

ena司

s s .

e eR)

vor

p

, 咖

1

1

．

5

e

厂

s

.1

* 1. Congruence of behaviour p立中ams

sa

1

1

a

ra

一

eR

飞

，

.j

, i11L mu

-i- S

Snu

s

a

g ．

ysa

We 江 M propose to extend or widen our direct equivalencer el a 七 i on 七 o a congruence relation. Apart fran the wish to ge 七 a congruence re l a 七 i on

(so that equivalence is preserved by substi 七 u 已 on of e 平 且 val en 七 progr ams) there is another noti vati on; ' 三 ' r 匀 uir es tha 七 th e r es ul 七 s of ac 已 ons of 妇 va l en 七 pr ogr ams should be i de..'1已 ca l , and i 七 i s reasonable to ask only tha七 th e results should be e子让va l en 七 aga 年 ．

We therefore define the relation 1 -'over programs, 如 i ch we call strong equivalence (we define it analogously to the obser 炟 已 on eqi.: 让 v­ alence of 红 3, bu 七 i 七 is stronger because we do no 七 all ow arbitrary

T-actions to interleave the 中 S 缸 vabl e actions).. we define i 七 in tenns of

a decreasing s匀uence ~a, ~1, • 毯., ~k

,•.

. of equivalence relations:

Definition B~。C

, .'

is always true;

B~ C

k+1

iff for all

µ, V

灶） if

B

（让） if

B 立 B'

c 鸟 C'

then for sane

then for 改

C', C 尘 C'

B·, , B 鸟 B'

and and

B~C iff 欢 ;:;: O. B ~k C

I I

cc

k k

～＇ ～

，

B

(i.e. ～＝ 八 ~k ) •

k

We leave OU七 th e s 扛屯让e proofs tha七 each

~ is an equivalence

k

relation, and tha 七 B ~ C implies

k+1

equivalences is decreasing).

B ~k C

(i.e. the sequence of

[:=cise 5.2

Show tha 七 B 三 C 乓 li es B ~k C

for each k , and

hecne乓lies B~C.

吐 eoerm 5. 4 ~ is a oongruencer el a 七 i on .

沁 re precisely, B ~ B 乓 li es

1 2

B +C~B +C, C+B ~ C+B

1 2 1 2

v

ii .B 1 ~ av .B2 , -r .B 1 ~ -r.B B jc ~B jc , CjB ~CjB

2

1 2 1 2

B \a~B \a , B [SJ~B [SJ

1 2 1 2

and

B {芍戊} ~B {动先 (for all

1 2

a兑B ~ a兑 B

1 2

～v) 扭 li es

Proof

on

We give the p:roof only for caaposi 已 on. that

We prove by induction

B ~ B

k

1 k 2

乓 li e s

B1IC ~k B2IC

For

k = o i 七 i s 七 r i vi al .

!bl assume

B B

1 k+:!. 2

~

Le 七

B jc 鸟 旷 D • 1 1

\ve wan 七 D2

such th a 七

B2 I C 骂 D ~ D

2 k 1

There are three cases:

{a) B 骂 旷 ， and D = B'jC {by 匀 尸 (1 ) ) 1 1 1 1

Then B 立 B' ~ B' • for s B'

咋 2 ,

2 2 k 1

whence B2 \c 立 B; \C by Can 丑 1 )

D (= B'\C)

~

k 1 1

by inductive hypothesis •

(b) C 尘 c•

and D = B IC'

1 1

(by Can+ (2)）

Then

B2 \c 鸟叮C' by Can -->-(2)

Bu七

B ~ B

(since B ~ B) , hence B1 IC'~k B2 Jc'

1 k 2

1 k+1 2

by inductive hypothesis.

~ 一-~

= =

{c) B-入UB'C 入U

1 1 入'~u C '

and

µV 1, D B'IC'

1 1

{by C 叩 式 3) )

书 1en B -B'~ B'

2 2 k 1

for same B',

2

whence B [ C 乌 B' I C' by

2 2

Can +(3)

~ k 气 by 年 duc ti ve hypothesis.

By syn咋 扛 y , of course, if B IC -µnv

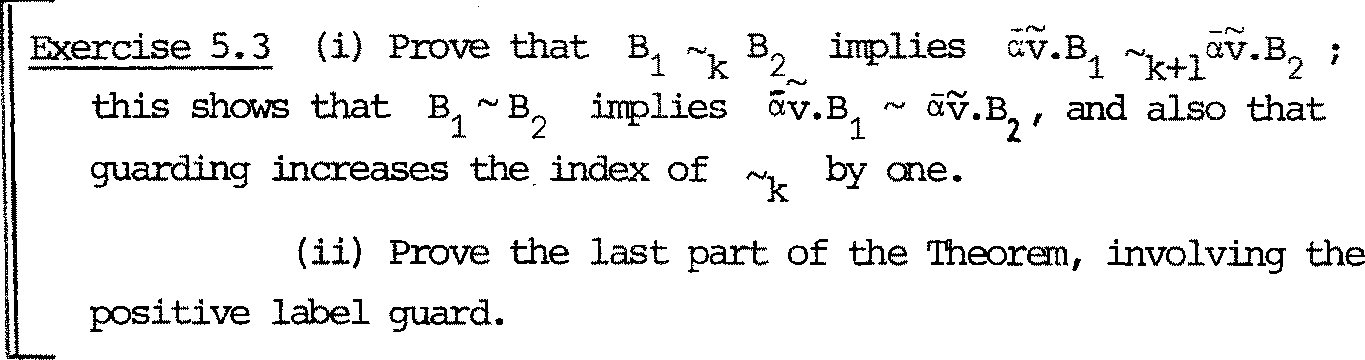
一

µy 2 2

then we find D1 such th a七

B1 IC

D 1 ~k D 2 • 因



We end this section by giving sane useful p攻 es of ~ , in

the fo 血 of equational 1 硒 . Note tha 七 Theorem 5. 3 already gives many of its pro:P=rties, since 三 i s contained in ~ . Since we nm 七 he risk of bewildering the read 缸 wi th a confused mass of properties, let us e.rrphasize s勺笔 s tru c七urE 斗

In Theorem 5. 3, Sum 三 s ta 迳 s tha 七 十 and NIL fonn a ccmnutative

semigroup with absorp 七 i on , and Res 三 ， Rel 三 ， Can one of th e 竺罕 behavi our operations \a, [SJ , I

each describe h 叩

扭ter acts with the

匀 沺 血 c operations +, µv and NIL. In the following theorem g竺～

states

th a 七 I and NIL fo 五n a carrnu七a 七i ve semigroup, while B竺 ~ and B色! ~ s ta 诠

ha,; the static opera 七 i ons 年 ter act with each other. The laws of Theorem

5.5 are only con0:: 江 ned with the s 七 at i c opera 七 i ons ­ the Laws of Fl <:JN 年 [ MM, Mil 2] •

they are essentially

Theorem 5.5 (Stron'f con中uences } The follCMing strong congruences hold:

Can~ (1) B !B ~B IB

1 2 2 1

(3) BjNIL~B

1. B I (B !B) ~ {B IB) jB

1 2 3 1 2 3

Res~ (1) B\a~B

(2) B\a\(3 ~ B邓\ a

(B:L, a n 畔 s (L) )

＿

1. (B 1B)\a~B \a 匡 \ a

1 2 1 2

(B :L ,B :L , a 生 names (L n L))

1 1 2 2 1 2

Rel~ (1) B[I] ~B (I:L-+L is the i den七i 七y relabelling)

(2) B[SJ ~B[S'J (B:L, and S L = S'卜L)

{3) B[S][S'] ~B[S'oSJ

(4) B[S]\8 ~ B\a[SJ (8 = na 哇 (S (a)))

(5) (B IB)[SJ~ B [SJ I B [SJ 1 2 1 2

P玉 底 g i ve the proof of 竺 2) . 工七 i s the har de s 七 - bu 七 al l the proofs are routine inductions.

We prove "i!B1 B克 . B1 I (B) B3) ~k (B11 B) I B3 by induction on k. For k = 0 i t ' s 七r i vi a l .

µv

归 for k+1, le 七 B1 I (B2 jB)3 一 D ; 袒 reqw.re D'such tha 七

(B1 IB2) !B3 旦 D' ~k D •

There are several cases:

JJV

(a) 乌

-----,.. Bi

, and D = Bil (B21 B3 ) 切竺竺 1 ) .

Then (B1i B2 月B3 虫 (Bl IB2) IB3 by C也 (1 ) twice

~ D by induction. k

(b) B2 I B3 骂 C , and D = B1 IC by 竺己 2) .

Subcases

位）乌 这 B' , and C = B'IB by Can+(l) ; i.e. D =BI (B'IB)

2 3 1 2 3

Then B12 1 B2

皿一B1I B by 竺已 2 ) I

1 2 3

so (BIB)jB µV (B jB') !B by can (1)

I

1 2 3

~ k D by induction.

（让）

四

B ,..B'

and C = B I B' by 研 (2) ; similar.

3 入u 3' 沁 2 3

(iii) B2-, B2 , B3-B3 , C = B2'IB3 'andµv =口

so D = B I (B ' 国 ） by Can+(3).

入u 1 2 3

咋 en B1IB 2

---,, 81IB2 by 竺三2 ),

so (B1IB2) I B3 立 (Bl IB2) IB; by 竺已 3) ,

~ k D by induction.

(c ) 乌过 国 ，B2 I B3 过 C , D = B IC andµv = -r by 竺 3) .

Subcases

一

AU

位） B2 B2 , and C = B21B3 by 竺已1 ) ; i.e. D = Bil ( B护牙

咋 en B1 I B2 工 耳IB; by 竺已3 ) '

so (B11B2) IB3-.l (Bi 吵 I B3 by 竺已 1) '

~k D by induction.

一

(ii) B

3

;\.U

B ' 3

, and c = B IB' by 竺已 2) : similar.

2 3

Th 氓 归 have found the required D' 飞 D m each case; Simil ar切空竺 (B 1B) 1 B 皿D , we f 运 D' s uch tha 七

1 2 3

B1 I (B21 B3 ) 骂 D' ~k D .

血 s cc 叩 l etes the m duct 坟 e step, showing

巨 B1l(B2IB3) ~k+1 (B1IB2)IB3

因

ise 5.4 Prove 三 and 三 ）．

need to appeal to Proposi 已 on 5.1.

For the second, you

啼 na,, Sta 七 e and prove a theoran which we need later. I七 depends

cr i 七 i call y on the as s 叩 ti on 廿 苹 al l behaviour identifiers are guardedly well defined (§5.4).

Theorem 5.6 Strong ccngruence'satisfies its defini 已 on' in the follCMing sense:

B ~ C iff for allµ,v

位） if B 应 B' th en for s吐 C' , C 皿 c• and B'~ C',

（让） 迁 c 皿 C' th en fo r 会 S 叩 a B', B 竺 B' and B'~ C'.

竺竺( ) B' ~ C' 乓 li e s B'~kC'for any k ; hence fran (i)

and (ii) we deduce B ~k+1C for all k , by de f in i 已 on , whence B~C •

（＝今） Since B ~ C for all k , 归 have by definition that if

一叹 k+1 \_

.c一叹 气& B'~ C • Bu

f ran our

B B' th en , for each k, 气

kk 七

as s ump已 on th a 七 all bel1aviour i den七if i er s are guardedly well-defined i 七 fo ll Oi 汜 th a 七 { C' ;C 皿 C' } is f ini 迳 (we ani 七 th e details of this 一七）. Hence for sare C',

c 皿 C' and B'~ kC'for infini七e l y many k

and thi s 垣 li es B' ~k C' for 斗 !\_ k , since the relations ~k are decreasing in k , hence B'~ c'.

Thus (i) is proved, and (ii) iss 皿 lar.

因

5.8 Con ruence of Behaviour ressions and the s ion 咋eor匋

Having established definitions and properties of direct equivalence and congruence of l?立习r ams - behavi our 呤 re ssi ons without free variables -

we are ncM in a position to 1i丘 the results to a立过七工a 工y behaviour expressions.

All tha 七 i s needed is to define 三 and ~ ove工 e习立es s i ons as f oll a 炬：

Definition

Le 七 艾 be the free variables occurring in B or B or both.

2

Then

B 三 B iff, forall v, B {芍及｝ 三 B 符／艾｝

2 2

B1 ~ B2 iff, forall 芍 ， Bi { 赣 } ~ B2 碑 ｝

NCMwe clearly want 七 o extend the results of Theorems 5.3, 5. 5 七 o arbitrary expressions; for example, we would l ik e 七o apply Q王旦旦） of Theorem 5. 5 七 o replace

a(x+1). NILINIL by a(x+1).NIL

anywhere in any expression, bu 七 the 1 玉 onl y applies at present to programs, and the expressions shown have a free variable x.

We state wi th 仅 比 pr oof the desired generalisation.

Theoran 5. 7 The relation ~ is a congruence over behaviour expressions.

沁 reover, the r esul 七 s of Theorems 5.3, 5.5 hold o 伲r arbi 七l'.'fil"\.Jexpressions,

with the folla; 心 g adjustnents:

包） In 立 巴 and 坴 三 of Theoran 5.3, replace v (a val ue 七 uple) e 诧 er e by 扂 (a 七 uple of value expressions).

(ii) Add in 立正 the condi 已 on tha 七， in the fi rs 七 (re s p. second) sum

on the r i gh 七 - hand side, no free va 工 i abl e of C (resp. B) is botmd by g.

冈

We naw have enough 七 o prove the Expansion Theor 己 n, which we used in Chapter 4.

'Iheor甸 5. 8 (吐e E>罕晔 i on 吐 eorem) .

邱 B = (B1 1 … ! Bm) \A, where each Bi is a sum of guards. Then

B ~ l{g.((B11··•1B1 •.• !Bm)\A); g.B1

a sum:nand of Bi, name(g)\* A}

＋沪. ((B11 ••• !Bl 酥 }I•• • IBj I•• • IBm) \A) ;

或 Bi a surrrnand of Bi, 盒 Bj a s 叩 畔 of

BJ., i,oj}

provided tha 七 in the f ir s 七 t enn no free variable in 气 (k 八 ） is

bound by g.

Proof. 盎 f irs t show, by induction on m, th a 七

Bi I ••• IBm ~ l{g. (Bi j ••• 冈 ...,Br) ; g.Bi a

surrnand of B . , 1 :,; i s m}

1

＋炉. (B1 I • • • lBj\_{睬 } i •• ,IBjj ••• jBm) ;

a兑 B!

1

a SUITT!畔 of

1. , a~E.B:

l. ］

a

surrrnand of

B . , i, j E { 1 , ••• ,m}, i ;t j }

］

under the p 切 is o of the Theor 亘

Note fi rs 七 廿 诅 t for

m =1 the

second tennis vacuous and the res ul 七 fo ll ows s 扛 屯 让 y by reflexivity

of ~. NON ass 呻 the property for

m 一 1 , withr igh七一han d side C.

Then we have (by congruence)

Bli ••• IBm-11Bm ~ CIBm

and we may appl y 竿 ，gener al is ed as in Theoran 5. 7, since each of and B is a sum of guards - and m:::>reover the side-condition for

c

m

竺 (s ta ted as ( 拉 ） in Theor 甸 5. 7) follc,;. 忘 f rom the pr 叨 i so of the

pre sen 七 the o rem. The property for m then folloi; 炬 byr ou 七 in e , though slightly tedious, manipulations; of course we rely strongly an Can~ (2).

Finally, the theoran follc,; 汜 easil y by repeated use of Res 三 (3) and Sum 三 (3) .

f，

8

因

rp

ehtf

s

s

江

p

ece n

et

s s

. 1l

.ev 细

r

g

8

udht

江

e

eht

ht

f

f。o

s

1 g.

i

vo

atr

p

ed

e

二

e

htht

?

l

w

c lt

c

5

axe

．

5

ee

es

s

. 8

厂

dna

re

In s umn叩 ： we 江 have a p,:亨 f ul set of laws for transfo:rming programs and behaviour expressions while preserving their der 坟 ation pa 七 七 ern . ('Ihese l 或 s are enough 七 o prove the Expansi on '!hear 己 n, 'Iheor em 5.8, for exanple.)

We h 忒 e prepared the way for introducing Cl's, an algebra which satisfies these laws and so may be regar 函 as a 函 el of ccs 如 ch is faithful to i 七 S der 拉 a已 on pa 七terns.

But we should men: 七 i on tha 七 obs erv a 七 ion equivalence c ) (generalised f ran §3 。 3 to admit value-passing) is a wider relation tha 七 our ~, and

sa已 sf i es s七i ll nore equa七i cnal 1 玉 llS .

CHAPTER 6

Ccmnunication Trees (Cl's) as a 呻 1 of C'CS t

6.1 crs and the洹c严atons

压 t us revi.窃 the def让让已on of STs. An ST of sort 还A 担

a rooted, 包卫让el y br an中 in g, unorder ed 七工ee whose arcs are labelled

by rrembers of Lu{叶．

如 other w 屯 of s 屯 rin g this is tha 七 an ST of sort is a f 血 te

L

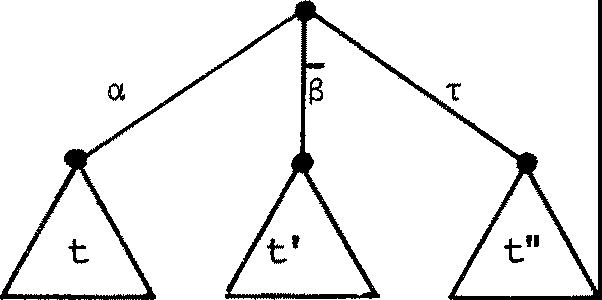
collection (multiset) of pairs of form

<µ, 七＞

(µ €L U 仕｝） where eadl.

is again an ST of sort L.

七

(We allCM this definition to include the possiblity of infinite paths in an ST, though to state this fonnally requires sare 咋 i.th e:na.ti ca l sophistication which we do no 七 wan 七 to be bothered with - the idea of inf ini 七e pa 芦 i s clear enough.)

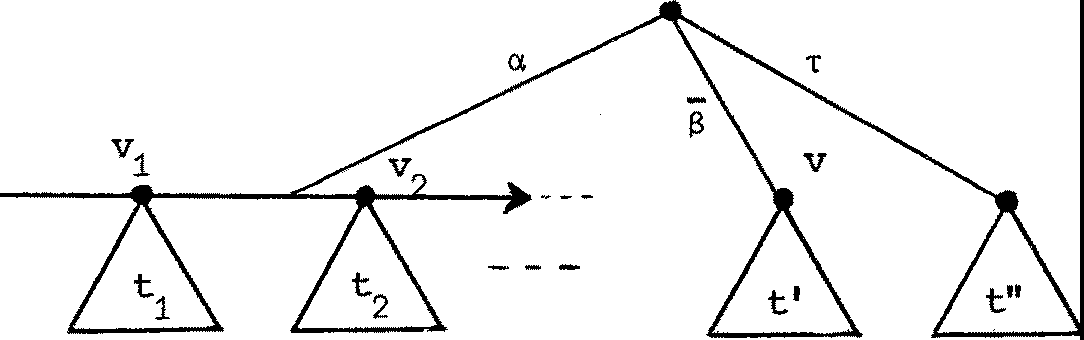
Here is a typical ST:

Now in the language of Chapter 5, positive labels are allowed to bind variables, and negative ones are allc:rwed to qualify values (or value expressions). 吐us, 泣a七 'ha ppens next ' af 七er passing a posi已ve label (= inpu七 guard) depends upon the val ue 包玉甡； less cri已cally, a

value is OU 坏 iut while passing a negative label (= output guard) • Supposing

that

{v。v,1,…} are the values of type appropriate to a, and v is a

value of type appropriate to 百 ， th en a typical CT will look like this:

三。t

v

'!his chap 七 er is no 七 se s en 也 1 to the techni cal 涵 el oµr 四 七 ， and can be

叩 让 ted . Is七 purpose is to ass is七 unders七五1din g by gi vin g 廿汜 nat u工al

generalisaticn of STs to admit value-passing.

indicating (i) tha 七 on passing guard a, 出e inpu七 v . s e l e cts 七

1 1

to'happen nextI 幼

(ii} that V is OU 七 pu 七 on passing S.

We expect this 匀 to be the interpretation of a behaviour program ax.B + $V.B1 + T .B11

泣1er e (i) the programs B{v./x} stand for CTs 七 ·

1 i'

（拉） the programs B'and B" stand for 七' and 七" .

Notice tha 七 the va 工 i ab l e x appears I101 中 er e in the CT; its pu工区汜e

in the p:t 呼: an1 is to sh 叩 hCM B depends upon the value inpu 七 ， and this

dependence is expli ci 七 in the CT; ea ch 七. depends, literally, fran

1

the value v. • (Of course, 沃 can never dr 砌 a 故 ole CT, in general -

1

even to finite depth - because of infinite value danains) • More fonnally, then:

Def ini 七io n A CT of sort L is a finite collection (mu已l

each of fonn

se七） of pairs,

<a,f> (aEL), 劝 1ere f is a family of CTs of sort L indexed by the value se 七 appropr i a 七 e to a.

or 郊 ，<v , t> > ($EL), 故 ere v is a value ap 严 ·opri ate to 13 and t

is a CT of sort L

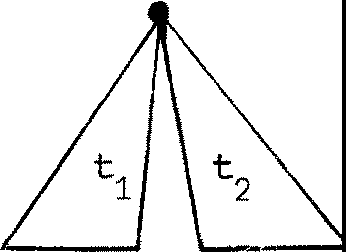
or 吓 ，t> 如 ere 七 i s a CT of sort L.

1. e 七 us denote by CTL the crs of sort L, andby Vx the s e 七 of values appropriate for a. We have, as with STs, an algebra of CTs as follCMS:

NIL (null 扛 ¥ operati on)

NIL is the CT

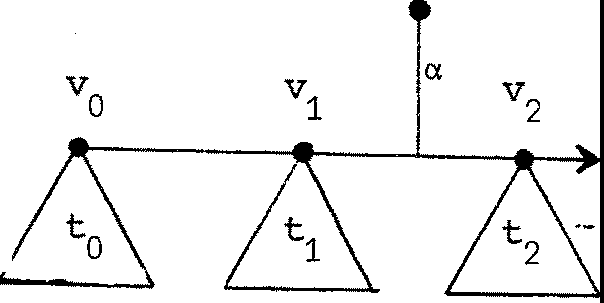
NIL E CT¢.

+ {bin 叩 严 ati on)

&+& is the er

+€er xer +er

L M LuM'

a (a 'rva-叩 " oper a ti on )

TC

eht

\1担

2

七

2 4

v

1

七

1

v

0

V

（

0

七

a

1. takes a se七 of m玉让汜立 of erL indexed by va., whi 中 i s j us 七 a f unc 已 on

f一； va. + erL , and gives a nar归 of erL u伍｝ ; so

* 1. E(V a + erL > + erLu{a.}·

'!his is why 陕 cal l ed a. a V 戈 叮 oper ati on.

Cf.

v

E

v

二

但

-a

＿

，

a

For each v, a.v E CT + CT -·

Cl E V + (CT + CT) 一 .

L Lufo}' Cl L Lufo}

；6

eht s i1

-r (unary operation)

；

'(卢） is the er

't"ECT -+CT.

L L

Clearly there is a very close relation between CCS programs (involving only the dynamic 平 r a ti ons ) and expressions for CTs in this algebra. 血 s is no acciden 已

Corre s ponding 七 o programs NIL, ;;: 立 B, 't". B, B + B'we have CTs NIL, ;;:vt, 让 ， 七 十 七 '. Corre s pondin g 七 b the pro 扛 am ax.B we have a er af; if we wrote the er family f as v 1--+- 七 (v ) then we would

express a.f as

a (v 汗 七(v ) )

Of course there are many CTs 血 中 we carmo 七 wri te 改 如 as expressions, because arbitrary V -indexed families of CTs c anno 七 be wr 让 ten d< 立

Cl

f 血 te l y .

Bu 七 we can, using these nota 七 i ons , begin to define the in 七 erpreta­ 七 i on of CCS in the algebra of Cl's. v 妃 s hal l 欢 ri te the CT whi 中 B s 七 平 ds for as 『 B ] • Then we have

氐 f ini 已on

[NIL] = NIL

[ax.B ] = a (v 吁 B{v/ x}] ) [a.v.B] = a 可 B]

[-r .B] = -r[B]

[B + B'] = [BJ + [ B 勹

1. 2 CTs and the static o:p 江 a 已 ons

we n 叩 sh 叩 tha 七 the static opera 七 i ons I , \a , [ SJ can be defined recursively over CTs. Recall tha 七 a CT is, f o 五 芘 ll l y , a roul 七 i se 七 of

elements 1让e <a,f>, 郊，<v , t> > or 令 ，t> ; 袒 s ha ll ca ll s 吽

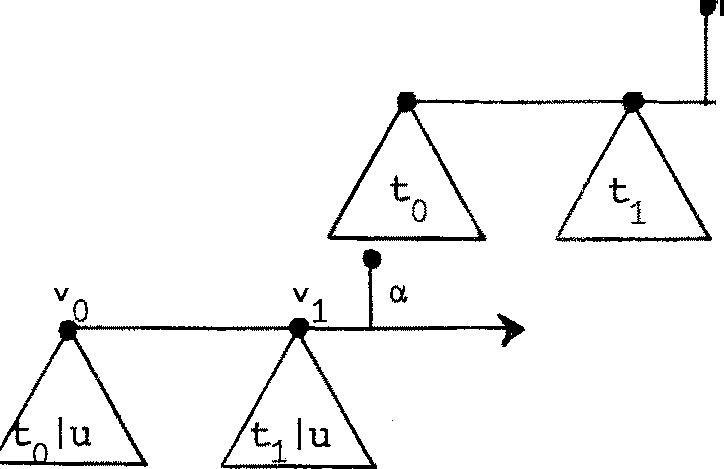
e l 郅 臼 1ts br an 中 es of the CT. We shall content ourselves with a rather infonnal definition of I, \a, [SJ using pictures of branches, rather than defin:ing them fonnally in tenn.s of mul 七 is e ts .

(bin 叩 operati on)

IEcrL x er -+ er

M 切M

迳 七 i:E CTL, 业: erM. 'Ihen 七 ju has the follc:Ming br an中 es :



v。

v 1

For each branch

CL

位） of 七 ， a branch

1. Foreach branch of 七 ， a branch
2. Foreach branch ｀of t, a branch

# ｀ ｀

and s 皿 l ar l y for the bran妇 of u.

（坟） For ea 中 pair of branches

r

:1 a of 七， and

a branch

；竺

上。f 5

and similarly for bran 中 es <o.,<v.,七'>> of 七 and <o.,v. 户 U . > Of U.

J 1 1

('lhus an output branch of u selects a member of 七s' canplerrentary

inpu七 br anch . You should ca 平泊r e this defini七i on with OOlllf儿忑i已 on of

STs in§2.3.)

\a (un叩 严 ation)

\a<:::CTL + CT

正 {a , 心

We could g达吧 the r ecursi妇 defini ti on, bu七 i 七' s enough to say th a七 七\ a is gained by pruning 或 ay all a- and ;; 士 r an 中 es occurring anywhere

in 七

[SJ (unary oreration)

[SJ心 + CT where S: L + M is a relabelling.

L

M'

Again i 廿 s enough to say th a 七 七 [ SJ is gained by repl acing 入 by s 入

everywhere int (入EL ) .

尸

ise 6.1 Give the nacuxsive defintioos of \•, [SJ in the same

style as we defined •I

Now of course, we can continue our definition of the interpretation of behaviour programs, as fo ll c:11 配

Definition [BIB'] = [BJI [B']

[B\a] = [B]\a [B[SJ] = [B][SJ

[if true then B else B'] = [BJ

[if false then B else B'] = [B']

Since our definitions of [] for programs look very trivial, as they should, we must remind ours el v 笚 of the purpose. We are aiming to sh 叩

th a七 when we ar e 沁r kin g wi 廿1 strong equivalence of 区呼 季 (the oon­ gruence relation ~ defined in 氐 7) , and using its properties as

listed in theorems 5.3, s.s (bu 七 3 吐 七七让巧 且旦邑三(4) , the abs o:rp七i on 1 或），

then we are justified in thinkins of the p 立 平 宝 芯 as the CTs th a 七 th e¥

<:l.enote?; CTs are mean 七 pr in ci pal l y to be a helpful Irental picture, or visual aid.

Ther es 七 of th is 中 apter gives the appropriate jus已fi ca七i on . Bu七

f ir s 七 we m出3七 deal with recursively defined CTs.

6 。 3 CTs def ined 切 r ecursi on

举S 哱 as in§5.4 tha 七 our behaviour i den 七 if ies b are defined by

clau至

b(x1,…，xn b( )) 年 号

one for ea中 b. Her e 让 wi ll be ro nvenien七 to suppose th a七 b 0 , b 1 , ••• are the se 七 of identifiers, wi th 王 吐 已 es n , n , ••• , and write B. for

0 1

J.

旯 ， so that the clauses are

i

bi (x1, ... ,xn. ) 年 Bi .

J.

妇 we intend to 址四 tha七 these clauses define, for ea中 i and v巳ctor

～V = V, … ，v of values appropriate for b. , a unique CT as the

1 ni i

interpretation of

* 1. (v～)

．

J.

懂 a 七 ar e these CTs to be? We will call them [b. (v) J. 刊 1en we knCM

v,

J.

th 玩 ， 归 a l so know the meaning of Bi{誘} for ea中 i and this is so because, by our definitions [] so far, ea中 [Bi {, 放 }D can be

rewr扛 ten as a CT expression in 枉江ms of [b. (u) ] for various j and

~ J

u. 如 exai:rpl e will make this clear. Consider the defining clause

应）令 廷 x = o 生竺 百x . NIL 兰竺 ay . b (y ) and call ther igh七h五1dside B。 '!hen

[B{o/x}] = [S-OaNIL] = -SO(NIL) (a CT 郅 pres s i on )

while for any v 于 0

[B{v/x}] = [a.y.b(y) D = a (u t+[b(y){u/y}D) = a (u 叮 b (u) ]) ,

归 'We wish our Cl's b. (句， for each i and v, to be solutions

l.

of the equati ons 叩 er Cl's

[b. {v)J = [B. {v/x}]

v.)

l. l.

(there are very many such equa 已 ons , one for each pair i, Luckily, 定 can P 立 为 e 出 e following:

Pro产 i ti on 6.1 If the beh 动 our identifiers bi are guardedl¥well-defined (see§5.4) then the equations

[b.(v)D = [B.

～位 /～x }]

define a 旦罕 CT [b. (v) ] for each pair (i, v).

Proof Q: 毗 ted .

冈

We can see my this is so, for Iourexampl e 址>0ve , as fo ll 叨 乱

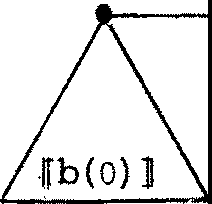
Clearly !b(O)) a 百 O{NTI,) a

。

For any v'Io we have

[b(v)] = a (u 巳 [ b (u) ])

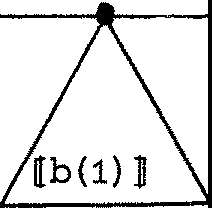
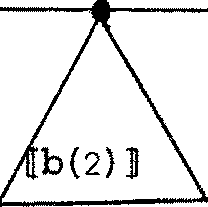
飞 is uniquely defined.

a

2

1

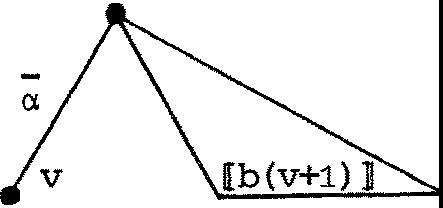
。

so th a七 by us ing 七he t沁 equa已ansr epea七edl y the CT [b(v)] for any v can be developed unarrbiguously to any desired depth.

en the other hand, consider again the for 区 dden ex 吓 le in§5.4 b(x) <= ;ix.NIL + b(x+1).

For any v (a non-negative in 七 ege 工 ） we would have [b(v)] = ;iv(NIL) + 『b (v +1) J

＝

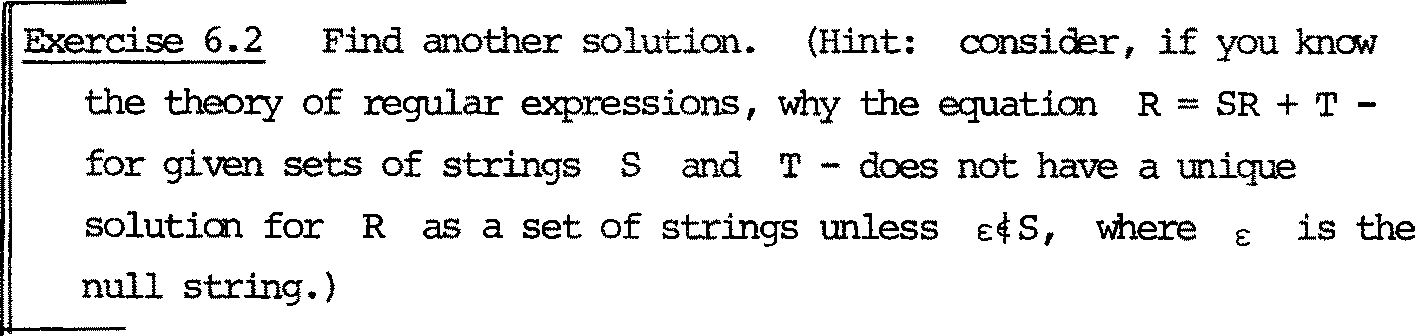
and if vie develop this, we obtain the infinitely branching (forbidden!)

er for b(o):

[b(O)] ＝ A\ ---\_

沁 reover , even if ,;,ie all <:Med 江 f ini te br anching 年 crs this would no 七 迳

a unique sol u 七 i on.



'lb sum up; 欢 a::npl ete our in.terpreta已on of beha吐 our programs as crs by 华 f in.in.g unanbiguously for ea中 b

I Def in.i七i on lf.b 向 ］＝ 旯 誘 }]

Remark There is a :rrore general interpretation than Cl's whi 中 makes sense of unguarded recursions, bu 七 we decided no 七 七 o use i 七 h em 王

* 1. A七cmi.c acti<ns and der i va七i cns of Cl's

工f we wish to 廿让咄 of beha吐our programs as the CTs which they stand for, then - for one thing - we must be abl e 七 o understand the action rel a七i ons -巴匕 ver crs in such a way tha 七 th ey hanronize with the corresponding re l a 已 ons over programs.

。

We therefore start with an in dependen 七 def in 七 i on of the relations

一叹 。侂r CTs. (We could use a differ en 七 芍 叨 比 l fr an — for these

relations, bu 七 i 七 wi ll in fact always be clear whether we a 年 七 alk in g abou 七 a tanic actions of CTs or of programs.)

Def 让让七i on 迳 七 七 be a CT, i.e. a multi se 七 of pairs (as defined 年卢 ）．咖 叩 七 has the a 七吐 c ac 七i ons

(i ) 七4 鸟 f (v) for each membre

七骂::= appropriate for a;

勺 ，仓 of 七 and each v of

* 1. 七一色匕 七' f or each rrenber 郊 ，<V, 七'>> of 七；

(i ii ) 七 t ' f or each member 令 ，七'> of 七．

吐 i s states, for every t, exactly which pai rs 生 ， 七'> are 扭 the

re l a已 on for everyµand v.

尸se 6.3 Lis 七 the a 七 立 C acti 中s of the typical er

§6. 1.

沮 -- 扫

巨ise 6.4 Prove tha 七 七1 十 七2 立 吐 ' i£ f 坐 竺 产 七'

七一坠 七I •

一

2

Exercise 6.4 gives a h 扛 止 of the hanrony 识 :l expect be 七 沃 关 !I1 the act 沁 n re l 平 ons -E4 over CTs and over programs. For if v-ie recall the rules f 红 3, we can rephrase them as fo l 正 s :

。

B + B B'iff either B or B -E4B1

1 2 1 一 2

(the'iff'being jus已fiedby the fact tha七 包 王!::. is the only rule by

which actions of B.. + B can be inferred) •

1 2

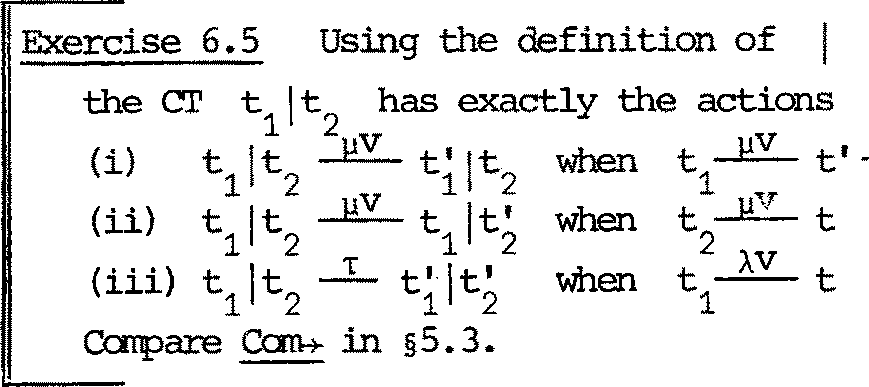
Similarly, the CT cif, which is the rnul 七 i se 七 whose only nernber is

<ci,f>, has only the actions

廷 乌 乌 f (v) , for each v,

which 炽 :i can 叩 are 社 th the fact, fran Act-+(1) in§5.3, tha 七 th e program ax.B has only the actions

ax.B --a-v-:. B{v/x}, for each v.



1,.I

, and

2，

七

入V

1

2

七'.

2

S 哗 l y then the atanic actions of B and its er lfB] are closely related. We state the 迳 lation in a theon 郅 ：

哗 o rem 6.2

* + 1. If B - 还仁屯 I then [B] -巴今仕门；
    2. If l[B] 立 心 ， th en for sare B', B - 巴 B' and [B'] =七'.

竺 Mainl y by induction on the structure of B; but particular care is needed when B = b {v) , and the ass ump 七 i on that the b's a 迳 guardedl y well defined is 江 iportan 七. R1

In other words, the a 七 立 c actions of [B] are exactly [B] -巴 [B' ] where B--].IV.,B'is an at cr吐 c action of B; th i s 哇 ans that in ro正

sider江g 平皿c actions, 让 makes no diffe:rence whether we 廿让咄 of programs or of the Cl's tha 七 th ey stand for.

The next step is to shCM th a 七 thi s holds too in consi der 扭 g s 七 立 改 1g equivalence.

* 1. Strong equivalence of Cl's

We proceed in the same s 七 yl e ; th a 七 i s , we def in e s 七 ron g equivalence

（～） 忒 er Cl's .independently, and then sha,; h<:M it hanronises with s 七 五 :,ng equivalence of programs. Our def.ini 已 on is en 七 i re l y anal ogous 七 0 tha 七 of

~ for programs (§ 立 8 ) ; we use a decreasing sequence ~o, ~1 , ••. ,~ of equivalences:

k'..

Def in i ti on 亡 。u is always true;

七,..., u iff for allµ,v

k+1

|  |  |  |  |
| --- | --- | --- | --- |
| 位） if  也） if | 七一上又呻七' then for sane u',  u u'then for an七＇ | u 且又.,- u ' µv  七 -- 七' | and  and |
| 七 ~ u iff | Vk 2::0.七,...,k u. |  |  |

t'~

ku';

七 I ~ UI•

k

Although we don't need it a 七 pr esen 七 ， we may as wi 己 11 s 七ate the anal og 毡of Theo:rem 5.4.

Theorem 6. 3 ~ is a CXJ11gruence 迳 l a七i on in the algebra of crs. 沁r e

p 军 1s e l y , 七1 ～七 2

:implies

七+ u ~ 七 + u, u + 七 ~ u +七

＿1 ＿

2 1 2

a v (七） ~ av (七）， '(七）~, (七）七 ju ~ 七 l u , u J 七 ~ u l 七

1 2 1 2

七 七 七 七

1 2 1 2

\ a ~ \ a , [ S J ~ [ S J

1 2 1 2

and for f 1 (v) ~ f 2 (v) (for all v) 乓 li es af 1 ~ af 2 •

旦 Anal ogous to'Iheorem 5. 4, and ani 七 te d .

区

陬 at we do need, to caaplete our jus 七 i fi d:a 七 i on of thinking of programs as crs, is the follrMing:

节1eo.rem 6.4 B ~B

1 2

iff [B J ~ [B ] •

1 2

竺 We mus 七 prov e separa: 七 e l y , by induction on k, that

* + 1. B1 ~ k 骂 乓 li es [B1] ~ k[B2];
    2. [B ] ~ [B ] 乓 li es B~B. 1 k 2 1 k2

We do only (1) , l eav 江 g (2) as an exercise.'Ihe case k=O is trivial.

•-• >lhy?

NCM ass 哱 (1 ) a 七 k , and assl.lile B ~ B , and prove [B D ~ [B ] .

1 k+l 2 1 k+l 2 Suppose [B1] 号· 吐en by Theorem 6.2(2)

B

1

So by as sump七i on

B'for s咋 B' , with [B'] = 七 I •

1 1 1 1

B --µV,.B'for sarre B', with B'~ B

2 2 2 1 k 2'

and by Theorem 6.2(1)

[B2 ] 骂 [ B; ] , wi th 七1' = [B'D ~ [B'D by inductive hypothesis.

1 k 2

如 s verifies the fir s七 cl ause in ~'ks +1

def ini 已 on ; the second clause

follONS by syrnret:cy, so th e 江 duct i ve step for (1) is cx:mplete. 冈

厂：：二 二

（．：）；二 ct 二:;;.,:,: 二 二 ee 二

proof is likely to be wrong.

* 1. Equali 七 y in the CT rrodel

Can we have B ~B bu 七 [ B D 牛 [ B ]?'Ihat is, if two programs are

1 2 1 2

strongly equivalen七， are their c:rs perhaps always th e 旦竺竺:?

No, because for exanple T.NIL + T. N工L ~ T.NIL;

bu 七 the two Cl's are TAT respectively.

．

T

d

na

Bu 七 th en perhaps the only difference between the Cl's [B ] and [B ] ,

．

2

when B ~ B is due 七 o the fact tha 七 七 十 七 ＝ 七 is false for Cl's, because

1 2'

殴i allow the presence of i d.en七i ca l branches.

In fact, we f ir s 七 th ough 七 七 ha 七 if we adjusted our def ini 七 i on of crs to be in 七enrs of 旦色竺r ath er than mul七i s e 运 ， then all our results so far would hold, and also we would have

B ~ B iff [B ] = [B) (?)

1 2 2

玫 恁 ver , Brian Mayoh showed thi s 七 o be false, with the followings 扛 耳 让 e

001.mter - ex 郘立e . Suppose x is a Boolean variable, andccnsider the two

programs

B = a,X.C + 必 .c

1 2

B = 这 (if x then C else C ) + 必 (if x then C else C) 2 - 1 2 - 2 1

where C and C do no七 0 卫让 ain x. Clearly we hav 己 onl y the follCMing

1 2

four actions for B :

a.v

B )

1

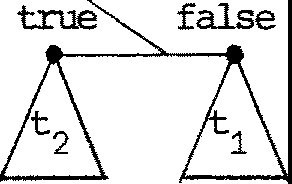
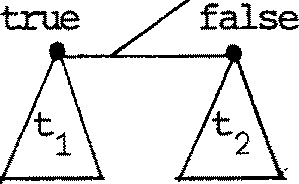
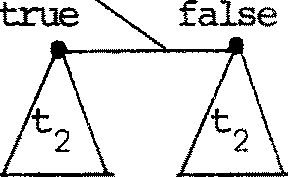
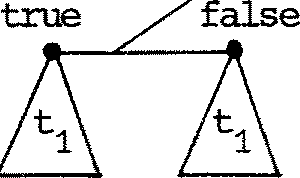
1. , vdtrue,false} andi€{1,2}

J.

and B 芦 ex actl y the s 郘 e four actions. So B ~ B . Bu 七 [ B D and

2 1 2 1

* 1. are di f f eren 七 Cl's:

(). ct a Cl.

.in which ti= [Ci]'id1,2}. So in general [B1 ] 牛 [ B2 ] , though of oourse [B ] ~ [B ] by 吁 氐 rem 6.4.

1 2

We 中 os e to define Cl's as multisets rather than se 七 s of branches,

because 迂 S eE平 ed. tha 七 mul七i se st are a nore concrete .intuitive m::x把 l ;

af 七 er all, 七 o check wheth er 沁 br an che s are id en 七 i cal requires an infinite

年 un 七 Of WO 欢 ! But i 七 is very much a ma 七 ter of 七 坴 te .

Even in the pre sen 七 m::idel , many equali 已 es hold. In fact, if we allCM ourselves to drop the 宝 吐 i c brackets [ D , and 七ake a beha过 our program

七 o denote a CT without this extra formality, then we s 七 ate the follc:Ming:

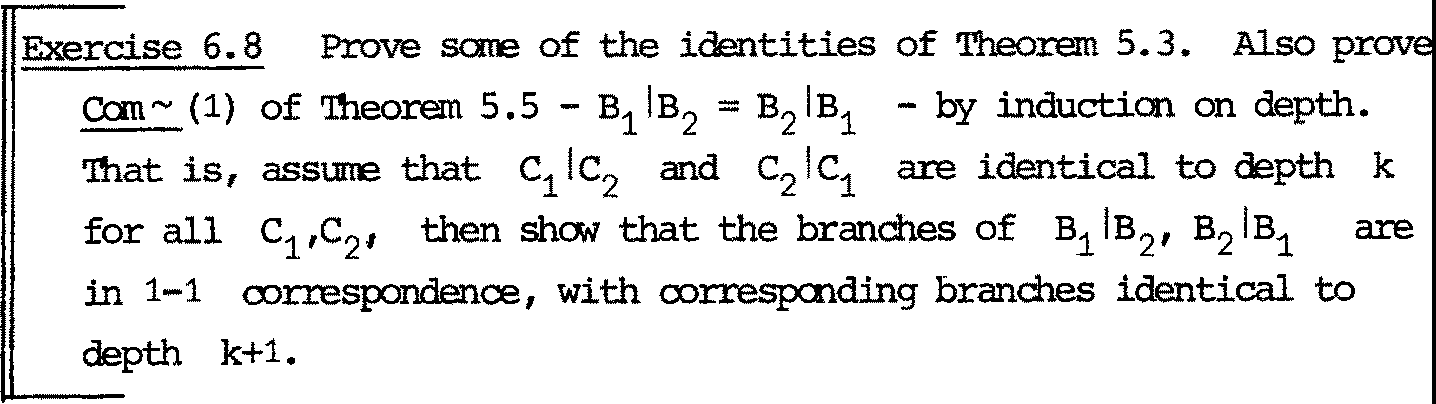
Theorem 6. 5 All the oongruences of Theorems 5. 3, 5. 5 are i den 七 i tie s in the CI'rrodel, excep 七 (4) (abs orp 七 i on ) .

竺 立 七 te d . 工 七 i s a ma 七 ter of proving th a 七 七 he two crs in question - for example (B1IB2) \a and (B1\a) I (B 》 a ) ( 些 竺 (3) in Theorem 5.5), 一

are identical to depth k, for arbitrary k (using inducticn on k).

In f ac七， 七he i den已 七ie s of Theorem 5. 3 can be proved with ou七 any induction.

区



* 1. Suntnary

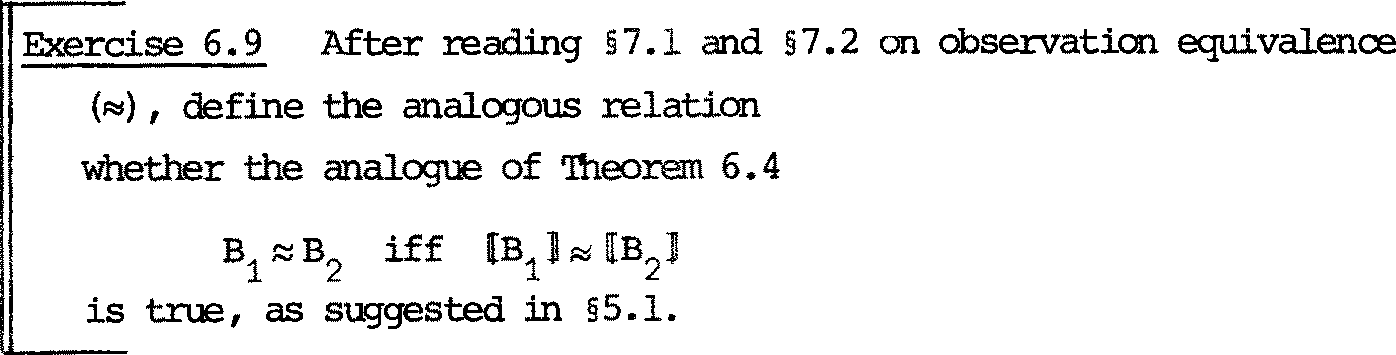
In this chapter we have

1. Constructed crs as an in 七 ui 七 i ve nodel of CCS;
2. Shown th a七， in 00! 正;idering a:七ani c actions and s七ran g ec于让val en re of programs, 归 are justified in ccnsidering these notions as they apply to the denoted Cl's;
3. Shown tha 七 many useful program eqi. 让 val ence laws are actually identities

for C 屿 ．

We have 竺 S 七 udi ed the wi der 罕 l ati an of ab serva 七 i on -equivalen 年 over

programs. Bu七 i 七 t ums OU七 th a:七， f or any eqi.让val enre relation whi 中 i s defined in te 血 s of ....!.: 乌 and/ or ~, 归 can think of this also as as 包 严 val en ce 洷 l a 七 i 中 over Cl's.



::::: over CTs. Then in ves ti ga 迳

Cne further poin七 shoul d be m玄 l七i oned. 'Ihe syntax of CCS is sudJ.

that only a small subclass of Cl's are e:xpressible as programs. In parti 一cular, a er of fo 血 {<a, f 习 can only be e:xpressed by a program 必 B

for whidJ. B, ccnsidered as a function of its free variables x, e:xpresses

the family f schematically. N叩 there are effectively indexed CT-families f which carmo 七 be represented by CCS e:xpressions; ccnsider for ex 吓 le

the family f = {yiI ; iEN}, and l e七 a bind an integer variable, so

that {<a,f>} is the er

a

1 I 2



Y.1 I飞

。

一Y

whose (江 f 扭 i te ) sort is {«,-, ,y ,y , ... }• To express i 七 立1 CCS we may

－－

0 1 2

wish to all<:M labels to be pa 工 五 芷 iti ca ll y 华 penden 七 upon values, and write

ax.y .NIL. In 叩 年 匀 叩 l ex cases y

X X

could also qualify a value expressicn,

or be replaced by a positive paran:etric label binding a variable. Such extensions of CCS may be of real practical value. If we wi sh 七 o ccnsider them, then the theory of CTs 扛 1cre ase s in 扭 irort ance s in ce 让 does no 七carmi 七 US 七 o any particula 工 exp re ss ib l e subclass of CTs.

CHAPTER 7

Cbservation equivalence and its properties

7.1 Revi 窃

In Chapter 6 we s 七 udi ed CTs as a m::x 拒 1 of ccs; this should h 忒 e given ins i gh 七 扛 让 o the laws of strcng ccngruence (~) s七ated in Theol) 已 阻

5.3 and 5.5, since CTs satisfy all these laws excep 七 th e absorption 1 或

B + B 三 B, interpreted as identities. In spite of this sli gh 七 di s cr e pancy,

让 i s still useful to think of programs ' 在 ; ' Cl's.

In 氐 3 we defined a notion of 心 servati on 立 ruJ.val ence (::::::) for STs; in our Da七a Fl Ckl 郅 ampl e (§生 3) we an已 ci pated using i 七 in full C'CS bu 七gave no definition. We s 或 tha 七 i ts purpose was to allc,; 寸 1.IDobservabl e actions (T) to be absorbed into experiments.

Recall also the 竺 ivations of§4.4. We 忒 :ibre vi ated

B ,Ill

B' (m 之 O) by B 今 B'

m n

B, . µv., 今 B' {m,n 2!: o) by B 当 >B'

沁 re generally, we n& abbrevi a 七 e

In() m1

T吨

'"C • 归罕．．．．叫森·

B 归 (k 2!: o, m0 , ••• , 吹 2!: o)

罕 1 ·. .. 中心

by B·r-">B1

which includes the above cases (they corres pond 七 o k = o, k = 1). 工七 al so includes the possibilityµ. = , , so that for example B = 今 B' means

m J. e: m

1. B'for sane m >O, 平 l e B 一 ＝ 汜 ＇ 哇 ans B 4=- B'for sane

m 之 o; but usually we shall h 忒 e µ . E A.

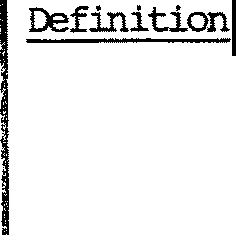
J.

For each s = 入1 V1 ••• • • 入k Vk dA x V) \*, =s is the s-experirrent

relation, and each instance B 今 B' i s called a s-experirren 七. We now def in e 心 serv at i an 购泣 val ence in tenns of s-experirrents.

1. 2 孜 servati 中 equivalence in CCS

Analogous to§3. 3, :::: i s 华 f in ed for programs by a decreasing sequence of equivalences:

B::::。C

.

is always true;

B~~k 丑

C iff for all

蕊(A X V) \*

位 ） if

' '

．

，

I

B

d

＇

＇

I

B

，

(ii) if

s

B ===B

C =s 千 C

then for sane then for sane

B::::!C

＇

c c

～飞 气

I

B

d

nana

，

C B

于『

C B

I

C

iff

\:fk;:,, o. B k C.

Remarks

* 1. There is a question as to whether we need to consi der 卓 s-exper.imen 七 S in this defini 七 扛 m, or if i 七 i s enoU]h to consider only those of length

1 - i.e. we rnigh 七 re pl ace s€(A x V)\* by s E A x V in the def in i 七 i on. 'lhe relation :::: thus obtained is di fferen七， b u七 i 七 七urn s out th a 七 七 he congruence (§7.3) which i 七 in d uces is the s 玉 汜 (assuming only that

CCS includes an equali 七 y pr edi ea 七 e over values), though we shall not

prove i 七 her e. Our presen七 def in i 七i on, using (AxV) \* , has s 匀 芷 wha 七

nicer properties.

* 1. Our def坛i ti on has a the program (crs)

pr立汜迂y which mus七 be pomted OU七． I 七 all o 峦 3

产＝ and

•T•T•

NIL=•

七 o be equivalen 已

k

i

no 3T

itc no 血

u

d

二

IN

y

1

～次

3T

e

ht3

evo T

r

p

4

7

尸

es oNot

i

七0

{-rw

can be defined by b T.b.)

y

咋us, whenever we have proved 压 (e . g. B may be a program and C its specification) we c anno七 deduce 廿 沮 t B has no infinite unseen action, even if Chas none. In one sense we can argue far our def­ inition, s 扛 1ce infinite unseen action is - by our rules - unobserv­ able! But the problem is deeper; 让 i s related to so-called fairness, which we discuss briefl y 江 §11 . 3. In any case, there is a rrorer ef 扫 ed no 七 i an of which respects the presence of infinite unseen acti 中 ， wi th properties close to those 炽 m 平 t i an for the present ane.

『平一皿

errLIN

g

si迳

二

二

．

s y

no lno

S

­

b

七 dna

Y

na

r

noof

七

皇一

ec3T it

* 1. Disregarding the question of whi中 equivalence is correct, if indeed there is a single'correct'one, thefiner equivalence (under a

s li gh七 f urtherr efinemen七）has in ter es 七ing pr o区习士i es . Hennessy and Plotkin [HP 2J haver ecen已y found th a七 i 七 c an be axi a 芷已zed, in

a sense whi 中 we carmot explain here. MU 中! xror e needs to be kn(Mtl

before we can say whi 吐 1 equivalence yields better proof methods

a 七 l eas 七 we can say tha 七 ， if an equivalence can be proved under the refined definition, then i 七 hol ds also under ous 工 .

We now 七 urn to the properties of ;:::.'Ihere are many, bu 七 thr ee are enough to give a feeling for it, and to allow you to read the first case S 七 udy in Chapter 8, if you wish, before proceeding to§7.3.

酝 main thing whi 中 di s tin gui shes :::: fran ~ is the following:

P叩 s i 七i on 7.1 B 女 . B

互 We shON B 节 . B by induction on k. k=O is trivial, sowe assurre for k and prove for k+1:

1. I.et B =s B'. Then also t .B =s 因 ， and 殴 虹 切 B' B' (ea ch

k

~ is an equivalence re l ati a 出 ）

也）压七 t .B 兰兰C' . 'Ihen

either (a) s=e:, and C'is T.B; but then also B

扭 auct i on B"'kT.B

B, and by

or (b) ,.B --4 B =s

C'~ C'

次

C', i.e. B

s

===;,

C'also, and again

'!his canpletes the inductive step, yielding Bl':! k+1-r .B. 冈

This proposition should make you irmediately suspicious of 芍

because 炽 can show th a七 i 七 c 玉皿 be a congruence. In particul扛

胚 C does 罕 扭 l y B + DR>C + D;

e.g. take B as NIL, C as -r .NIL, D as a.NIL -

二

then B by Prop. 7.1, bu 七 B + D 申 2 C + D.

尸二，二。: N二I 二 ":二七

i s 申

1

N 耳 ．

Even so, Theorem 7.3 belCM tells us that "'is near enough a con­ gruence for many ptu平忑 es . First we need 七 o see i 七 s relation with ~

'lheorem 7.2 B~C implies B"'C•

互竺f We show tha 七 B ~ c implies B"'kc by induction on k. A 七 k=O

让 i s trivial; assurre i 七 a 七 k (for all B and C), and pr o 诧 i 七

吐 k+1 . As S UJ 正 : B~C:

s µ').'

位） I.e 七 B =B

n' say B

1 1 :B-+

...

. Bn , 故1er e 改 年 of

theµ.V. 亚 iy be -r, 曲 il e the rema.inder cons 七 i tu te s.'lhen by

µnVn

1 1

Theorem 5.6 used repeatedly, there exi s 七 C , ••• , Cn with

一

一

µ凸 罕 n

C C i -+ ••• c

s

n' 1.e. C =C n

with B. ~C. for all isn.

l. 1

Inparticular Bn ~ Cn, so by induction Bn "" kCn the desired Cn.

, and we have found

四 军 c 兰主 C·then similarly we f 年 d B with B =s B :::; C • 日

n'n n kn

吐 e 叩 rt an ce of this theorem is tha 七 all laws of Theorems 5,3, 5.5 hold also for 气

Theorem 7. 3 Observation equivalence is a rnngruence for all behaviour opera已ons excep七 十. M::>re precisely:

1. B•kC in.,i迨 l . B气忘c , , •• 飞 .c ,

B!Dll:l kCID,

B\a"'kC\a, B[SJ""k'C[SJ

and

~ ~ ~ ~ ~ B{v /x} i>, C{ v/x} for all v

k

～～

implies ax.B$>! kax.C.

1. Hence the s玉 hol ds with the indices k r err硒 d.

旦 坛 七 us j us 七 七咏 e 七he rros 七 in ter es 七江 g case: B气c :implies Blo kclo,

which we prove by induction on k. (This property is 竺 tru e for the

出 fferen 七 obs erva 七 i on equivalence suggested in Ranark (1) above.) Assume

a 七 k , for all B, C, D, and assume B C:

k+1

* 1. 丘 B I D 兰兰,E; then E mus 七 be B'ID', with B 兰¥祖 ', D =呈吵＇ for sare q,r ( 吐 taining canpl 卸 E 砒 a:r:y nanbers which'merge'to fonn 主 in a way which we need no 七 detai l ) .'Ihen for sare C', c 呈 C' and B'f>i k C'by assurnp已 on.

Bu七 then c jo 兰兰 C' I 扩 ， and by the inductive hypothesis B'ID'.::: kc• jD', i.e. E,:; kc• ID'.

也） I.et CI D 兰兰 E, then similarly we find B'ID' such tha 七

印 圭 五 I ID' E. 日

k

咋 e essence of Proposition 7.1 and 书 1eorerns 7. 2, 7. 3 is th a 七 we

can use all our 1 或 s , and cancel T' s 七oo, in proving obse工va七i on

equivalence - provided only tha七 we infer nothing abou七 th e r se subs 也 ut ing c for B under +, when we only k 兀双 眨 c .

Eise

u 让 of

哗 next sec 已 on tells uswha 七 such in f er ences 竺 be made.

* + 1. , - tha 七 B飞氐 叩 ies v.B 气 v. c by mductirn

on k. 配y is induction necessary? (Consider e:-experiments).

As we did for ~, we extend ;:::: to expressions by:

I

Defini 已 on Le 七 be the free variables in B or c or both. 中en B C iffforall v B{v/x}P::C{v/x}.

仅 1en we have

吐eorem 7.4 Proposition 7.1 and 叮 1eorerns 7.2, 7.3 hold also for e:x:pres­ sions.

旦 陀ut ine .

因

Fran nGN on, we dea l 吐 th expressions.

* 1. Observ a 七 i on Congruence

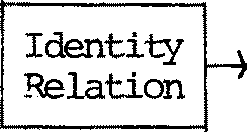
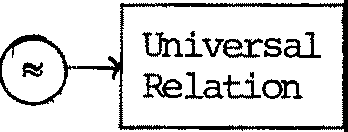
We mus 七 now face the fact tha七 ""' i s no 七 a congruence (see Exercise 7.2). Bu 七 we 沁 ul d like a oongruencer el a 已 on, because we would like to knCM tha 七 i f B and C are e< 于 让 val ent , then in whatever cantext we replace B by C the result of the replacaren 七 wi ll be equivalen 七 七 o

the original - which is only true for an ec; 但 i val ence relation which is a congruence. We have one oongruenoe - strong oongruenoe (~) - bu 七 i 七 i s 竺 s trong; for example a.T.NIL ,fa.NIL.

Can we find a congruence relation which is weaker than ~ (so tha 七

all our 1 称 s , Theor 卸 s 5.3 and 5.5 will hold for i 七 ），and has sane of the properties of :::: 但 o that for example a.T.NIL and a. N 耳 wi ll be congruen 七 ）； Le 七 us dr 或 the order rel a 七 i on (part of the la 七 ti ce of valence rela 七 i ons) arrong our existing equivalencer el a已 ons with s 七ro nger r e l a 七i ons 七0 七he l ef 七， and 攻担 ar e boxesr epre sen七in g cangri 玉 mce s :

Equivalences over beh 叙 i our programs:

产曰忙丑~

We wan 七 七o fill in"?". 工七 m出 共: be s扛 anger than because we do wan 七congru en 七 pr ograms to be observation equivalent. We ge 七 wha 七 we wan七 by the fo ll 叩 ing:

汶 f in i ti on B::::; C (Observation conaruence) iff for every expression con七ext ec J,CCBJ eccJ.

C

I

Theorem 7.5

(1)

C

::::

is a congruencer el a 七 io n ;

1. If e is a congruence and B e c implies B 式 ， then B e c implies

C

B

忒 ．

旦吐出：ted; i七 i s a 卫屯让e七e l y s 七andard, and has no七h 让1g to do wi 七h particular properties of the equivalence :::;. jg

Our'Iheorem says tha C is the weakes 七 congruence s七丘:mger than

七 ..,

(smaller than) :::: •

Caro ll 叩 7. 6

B ~ C implies B .C.. C implies B:::i C •

Proof Imnediate.

区

I 七 i s one thing 七 o define a congruence, another 七 0 knCM i 七 s properties.

We fir s 七 f in d ou 七 more abo u七 th e r e l a 七ion of .,.C section we find scrre 1硒 sat is fi ed by .,.C.•

七。：：：：； in the next

We saw earlier tha 七 s un con 七 exts were cr i 七 i cal for "', because

。

胚 C does no 七 impl y B + D ::::C +队 This leads us 切 explr e a new e< 子 丘 ­

valencer el a七i on .,.＋:

I

Definition B 十 C iff VD. B + D 红 + D • (equivalence in all sun oontexts.)

邸 the er让i cal resu让 i s the follCMing:

Theor 细 7. 7 R+l is a congruence.

竺 See 江 6. This proof i s 竺 standard, but depends strongly on the def ini已on of Ri it is no七 七m正i for the al t em a旦ve in Remark (1) of§7.2,

andtha七 i s 刘1y we chose our def in i 七i on . Theorem 7. 3 is cr i 七i ca l. 区

正 立n this we ge七， fo rt una te l y:

C +

书1eor em 7.8 11:l and "" are th e 三 congruence .

Proof

伈.） B :::::+ c implies B,.C, c by Theorems 7. 5 (2) and 7. 7, since :::::+ 担

s tranger 妇 ：：：：： (take D to be NIL in the definition).

（拉） B ""C C implies B :::::+ 已 S 年 ce [ J+ D is just a special k扭d of

can七ex 七. 113

Now we knCM that we pr ese 玉 "" by s ubs ti 七 uti an except in'+'con­ 七 exts. Wha 七 do we do if we have 胚 C and wish to knCM sanething abou 七B + D and C + D? Luckily, for an importan 七 cl as s of expressions B and

1. we can infer fran 胚 C tha 七 B:::::c C, and 旦竺 inf er tha 七 B + Dr:,;c C + D.

I

Definition B is 罕 if f B-4 B'is impossible for any B'.

Thus a stable behaviour is one which cannot 'move ' 已 es s you observe i 七 ． Stability is important 年 prac 七 i ae ; ane reason why our s ched ul er 江Chapter 3 works, for ex 吽 l e , is tha 七 i 七 wi ll always 迳 ach a s七忒)l e state if i 七 is deprived of external ccmnuni ca 七 i on for long enough. Caup主 宅

the notion of "ri gid" 扫 Chapter l r 袒 may def 江 e a 丑 唾 progr am 七 0 be

one whose derivatives, 江 eluding itself, areall stable.

'lhere are 切 main pror::ositions about s 七 ab il i 七y; f ir s 七 we prove a lemna in a slightly nore general fonn than we need for the pro1)0f 过 七 i ons

bu 七 th e general fa 五 n helps in the proof of'lheorem 7. 7 (skip the lemna if you are only interested in 职 迁 n results, not p 立 为 fs) .

Lemna 7. 9 工f B +c and B 今 B' , then for each k there is a C'such tha 七 c :±=;, C'and B' 芍 C' •

竺 Suppose C'does notexi廷； we find D such tha七 B + D 申 C + D, contrary to ass 叩 已on. Take D to be 入 . NIL, wher e 入 is no 七 in the sort of B or c. 虹 s in ce B B', we have B + D主 B' . But if

C + D 幸 E then either (i) E is C + D, 摔' s in ce C + D NIL, bu七

B' 于 ； or (拉） C尘E , 申k B' by supposition; or (iii) n\_l\_;P. - impossible since D is stable.

Hence B+D 申 C+D, contradicting B :::: 已

因

＋

严 s i ti an 7. 10 工f B ,::;cC then either both are s 七 ab l e or neither is.

Proof Direct fran Lerrma 7. 9 {B' C'not needed) • 区

沁r e 扭陌lOrtan七， for proof methods, is the fo ll O',,y 江 g :

P立和 s i ti 中 7. 11 If B and C

are stable, and B 式 ， th en B;::;CC.

竺 I 七 i s enough to sh& that B + D "'炉+ D for arbitra:ry D, by induction on k. We do the inductive step.

Le七

s

B-+D =E:

- —

位） 工f s=E then either E is B + D, and then C + D = 坠 C + D B+D

, ;::;k

by induction, 竺 D 呈 ::, E, and then C + D 兰 今 E also (B = 兰 E impossible

by stability).

=E

也 ）改扫吐 se either D 兰 E,

and then C + D s also, or B 圭 E,

whence C 至 巨 kE (because B 式 ），whence also C + D 吓

kE.

Thus we have found in each case an F s 上 .C + D = s

贮 E. The converse

k

argumen 七 l S similar, so B + Di::: k+1 C + D. Jg

Now for any guard g 十 T , we can deduce fran Bz C (f or 型 B,C)

t.ha七 g . 胚 g . C ('Iheorem 7.3), andhenoe g. B 竺 g . C since both are stable.

'Ihls 扭 li cati an holds in fact for anyguard, by the following Propos i已on (which is es sen 七i al in the proofs of Chapter 8):

Pr守 si已on 7.12 For any guard g, B C 扭 li es g.B,::;Cg.C.

竺 By the above remarks we need only consider g = T. We p 切 e

口 B + D 勺 k T. C + D for arb i tr 叩 D, by induction on k. Inductive step:

s

邱 T·. B + D = 五 . 'Ihen

包 ） 工 f s=e: the n 主 兰 竺 T E is T .B + D, and then T.C + D 今 T. C + D,,:kE by in duc 七 i on , 竺 D ==;, E, andthen T.C + D 兰 今 E also, 竺 T .B ::b;, E, and then B 幸 =;, E, '炸en ce c 今 F "" kE (since B式 ），when ce also

T.C + D =F E.

k

（ 拉 ） otherwise either D 兰 ::;, E, and then C + Dk E also, or B 辛 ::;, E,

如 en ce C 主 丘 k E (since B 式 ），whence also T.c + D 圭 F i:::: E.

As in Prop. 7.11, this canpletes the proof. k 日

By nOfl these inductive proofs of ""k'appealing to the inductive hypothesis only when e:-E 鸾 泣 irrents are cansidered, are becx:rning familiar; we shall leave th 钮 1 as exercises in future.

* 1. Laws of Observation C01'19ruence

We are going to prove three laws, for 平 ch we have strong evidence tha七 th ey say all th a 七 n eeds to be said abou七 th e s 扛 ange 年 vi s ib l e 1: under ,.C, ; this suggests tha七 th e a pparen已y never-ending s七r e 郘 of 1 玉妇 is dr 砌 江 g 七 o a close! The evidence is tha 七 th es e n 匈 l a1芯 ， t oge 七he r

with those of Theorem 5.3, have been shown to be a::mplete for CCS wi th.a u 七recursion and value-passing. This means that any true s七a 七己心阰 B ,.C,C

（扛1 this restricted language) can be proved fran the 1郘芯； 年 f act the

laws of Theorem 5.3 are quite a lo 七 s impl er witho u 七 val ue- pas s ing , and those of Theorem 5.5 are unnecessary wi th ou 七 r e cur s i on .

One would expect 七 o have to add sare 年 duction pr 江 ciple 让 1 the presence of recursion; what needs to be added for val ue- pass 扭 g is less obvious {bu 七 年 s everal more-or-less na 七 ur al exarrples, 年 el uding

those m Chapter 8, 归 have not needed rrore than we have already) •

Theorem 7.13 (T laws)

* + 1. g. T. B 言 g . B
    2. B + T. B 孚 口 B

(3) g. (B + T .C)+ g.C lg. (B + T.C)

竺 (1) foll 叩s directly fran Prop. 7.1 (T 蟾眨 B) and Prop 7.12. For (2), 炽 mus 七 pro ve for a 立过tr ary D,k

B+T.B+D T.B+D

k

andthis foll 硒 the pattem of Props. 7.ll, 7.12.

For (3) similarly,'v.:e need

g. (B + ,.C)+ g.C + D""g. (B + ,.C)+ D

which follCMS the same pa 如 m , but needs the extra easy fact tha 七 f or

苹 ， i f g .c

s

===;>

E then also g. (B + T .C)

s

=;;,

E • 笘

巨 c担 e 7.3 Canp l e 迳 th e proofs of (2) and (3).

A 职 r e useful fo 血 of (2) is the following:

r

C

Core ll 扛 '1 7. 1 4 B+c.(B+C)

r:; 1:.(B+C).

Proof

7.• - this, by firs七 - l ying (2) to

石 (B妃）； you will

need another 1 或 of +.

区

Q:le may justify the 1 称 s intuitively by thinking of any behaviour B as a collection of action ca声 i li ti es (the branches of its CT), including perhaps sane ,:-actions (the ,:-branches) which are capable of rejecting the other capabi li 七 i es .

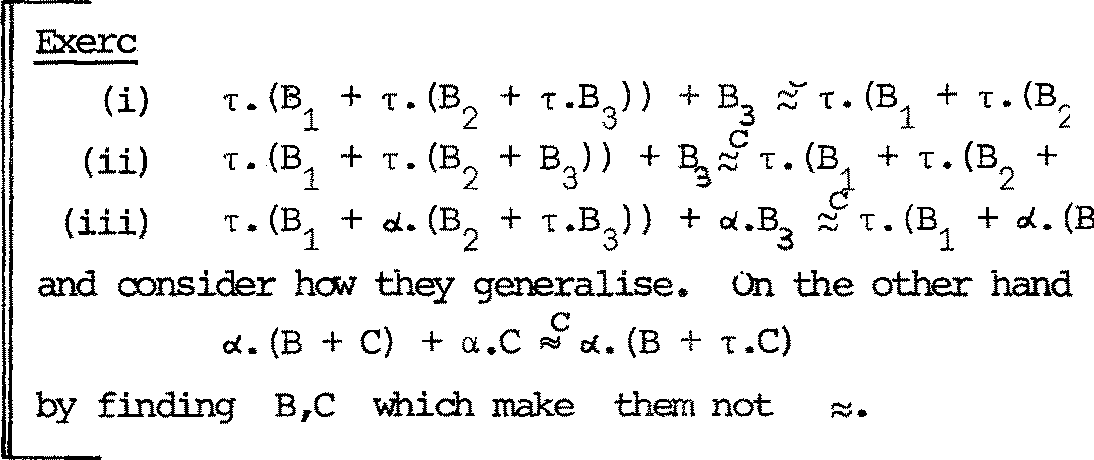
Law (1) may then be explained by saying tha七， under the guard g, the ,:-action of 石 B r e jects 坦 oth er capabilities and therefore has 正 eff ect . For Law (2), the capabil i七i es r epr esen七ed by B are again presen 七 af 七 er the ,:-action of -r.B \_in the context B + -r.B, so -r.B itself has all the p: 邓 红 of B + -r .B. For Law (3) , an observ a 七 i on of

the l e f 七 s i de may reject B by pas s in g 七 he guard g in g.C, but this rejection is already represented in g. (B + ,:.C) . Bu 七 s uch wordy jus 七 i fi ca ti ons badly need support; observa七i on equivalence is 刘沮七gi 伲 s them support here.

Laws (2) and (3) are absorption 1 或 s; they yield many o 廿 1er absorp-

已ons.

.B))



径 7\_.\_5 Prove, directly fran the laws, tha七

+ ,:

B))

3

2

, dis

3

+ ,:.B)

3

prove

* 1. Proof Techni 平芷S

In conducting proofs, we may take the liberty of using "=" in place of "~" or "护 ，ado pting the familiar tr adi七i on tha 七 "= " 晔 ans equali 七y

in the in 枉 :mdE!d interpretation; this helps us to highligh 七 our uses of z , for which care is needed because i 七 i s no 七 a congruence. Wi th 廿让s conven 七 i on , l 砒 us surrroarise the important propel 士 i es .

(i ) '!he 运 s of ~ (Chapter 5) ;

1. B :::;,.B (Proposition 7.1);
2. B == C 扭 li es 胚 C (Corollary 7.6);

（拉） :::; is preserved by all opera 七 i ons ex ce p 七 十 ('Iheo 工 em 7.3);

(v) B江 扭 li es B=C when both stable (Pr opos 平 on 7.11);

位） 陷 C 扭 li es g. 妇 .c (Proposition 7.12);

{vii) The , 1 翋 s (Theorem 7.13).

Since we mentioned tha 七 th e , 1 或 s hav e a 改 项 ?leteness property, why bother with :::; in proofs? The reason is 七 o do with stability. We can often shCM tha: 七 a behaviour B of in ter es 七 ， no 七 s t ab l e i 七 se l f , s a 七 i s fi es

B = T.B\*

for 扫妇 竺兰 B\*; so of course B 哇 \* (b 砒 Bf B\*, by Proposition 7.10!) This expresses tha 七 B stabilises. .Stable behaviours are often easier to handle, and the oonstrained substi 七 uti vi ty of :::; often al la 炟 us to conduct

our proofs mainly in tenn.s of stable behaviours. Chapter 8 should make this poin七 cl e 立 ·

Many proofs can be done with our 1 或 s wi th ou 七 us in g any induction principle, though the laws are established using induction on 气 · 'Ihere is, he1.1汜 ver , a pc王 fu l induction principle - Computation Induction

- due 七o Seo 七七， which we canno七 us e a 七 pr esen 七 since i七 involves a partial order over beha吐OU立沁 We believe tha 七 this principle can be invoked for the finer notion of cbserva 七 i on equivalence all uded 七 0 in

§兀2, 氏邓亟 (2) ; i 七 r emains 七 o be seen hCM iIIpJrtan 七 i ts use will be.

7.6 Proof of Theorem 7.7

Theo:r,匋

7. 7

＋ is a congruence.

Proof Fir s七， we shCM th a 七 B ::::: C 让叩耳 es B + D i,;; C + 防 廿诅七 i 句 we require (B + D} + Ei,;; (C + D} + E for arb i tr 叩 E.

＋＋

Bu七 (B + D) + E B + (D + E) (Theorem 5.3)

""'C + (D + E) (since B ＋C)

"'(C + D) + E.

逆 wer equi 迳

that B C＋＋

implies g.B z g.C ,

B[S干J C，[SJ;

I·'·

* 1. we wan 七 g .B + E"" g .C + E for any E. In each case the proof follows

the pa七tern of proof in Propositions 7.11, 7.12 (these Propos i已ons are

主 in tenns of::::: 气 bu 七 th e proofs are en 七 ir el y in tenns of ::::::) .

+

'Ihe critical case is

+ B ::::: C

扛乒迫 即

之 c 臣 忘

+

urre B::::: C and

prove BJD + E 气 C jD + E, for arbitrary E, by induction on k.

工nduct i ve Step: I.e七 BID+ E 兰主五 ；

位） If s ,,e:, then 兰 主 巴 E 主 E' , and then cjo + E 上 E' al so ,

竺 图D 主 矿， and then cJo心：：：：：k E'for sane F'(since

Bi::1C so B 匠 c In by'Iheorem 7.3) , whence c jD + E 主 沁

（过） If s=e:, then either E'is BjD + E itself, and then

k E' al so .

cln + E 今 c ln + E, 飞 引 D + E by induction, 竺 E ==:k;. E', and then cjo + E 兰 丑 ' al so , 竺 B jo --.!. 叮 3' I D' 幸 五 ' . 'Ihese are n0,1 the three cases:

* + 1. B'is B,and D \_!\_吵'; then CID cjo• and BI D' 江 c jo • by'Iheorem 7.3 so cjo• bp• :::::kE'for sane F', whence

cjo + E 今 妇 kE' as required.

* + 1. D'is D and B ----,. B'; then by I.ernna 7.9 C = 三 I "" k+1B' for s玉 C' (this is the only use of B :::＋:: C -乱延如亟

眨 C is all tha 七 i s needed) , and we also have B' I 妇 C' I D

k+1

fran'Iheorem 7. 3, so since B' I D 今 印 ， c • jo 今 F' "" E'for

k

sane F'. So f in al 凶 c Jo + E 今 C' I D 今 F ' ::::k: E' .

* + 1. B 竺 B' and D 鸟 吵 1; then C 兰 C' 飞 k+i B' f or s啦 C' ,

辛 nee c jo 上 C' I D' ,.. k+iB'!D'by Theorem 7.3, whence

C' !D' 主 F' ""k E'for s玉 F' , whence also CID + E 今 F ' 轩 '. 'Ihus we have found F' in every case so tha 七 CID+ E 兰 今 F'""k E' ; by symretry, we have BID + E jlli! k+1C1D + E which 立 1pl ete s the induction.

囚

* 1. Further exercises

We end this Chapter with same harder exercises, for readers interested in the theoretical developnen七．

尸二:=;;-

二二 resul七， which further

Bi::1 C iff (B C or B :::; t .c or t .B

C

RI C) •

Exercise 7.7 We would l ike 七o kna 寸 th a 七 i f b= a.b and B;::C: a.B

then b :::C:; B; this states that, up to C the re cur s i ve defJ.ni 已 on

：：：：：，

b = a.b has a unique solution. The argumen 七 in §3. 4, proving tl 汜

scheduler correct, used a mild generalisation of thisr esul 七. The follo,;ing exercises lead to a 印 re general theorem (for s 乓 li ci ty , work wi th ou 七 val ue passing) .

* + 1. Prove: if B::::; a.B and CZ a.C then B:::: C.
    2. Deduce: if B :::C: 贮 B and C:::C: a .C 廿 1en B:::C: 己

玫 r e generally, let C[ J be of form

D1 十巨. (D2 十 归·(· • • • ·(Dm +µm·[])···)) for m 1, where a 七 l eas 七 one µ , is no 七 T .

J.

* + 1. Prove: if B式 [ BJ and C::::; C [CJ then B::::: 已

彻） Deduce: 江 B :::C:: C[BJ and C:::C: C[cJ then B:::C: C.

(v) Deduce: if b = C[bJ and B::::C: 贮 BJ then b::::C; B.

" Exercise 7.8 Consider a di f f er en 七 def ini ti on of obs erv a 七 i on equivalence. Fir s 七 ， de f in e a decreasing sequence of pre-orders 0, 乏 1 ' …· ·.. :

B ;5。C is always true ;

B ~<k+1 C iff, forall s ,

s s

if B=>B'then for s 叩 e C', C=>C'and B' C'.

吐uswe take only the fir s七 cl a use of the def in i 七i on of~k+'1I•hen; B 乏 C iff't/k. B ;:;,k C ; B :::: C iff B 乏 C and C:5B.

We 亚 y take ;::; as a candidate for observation equivalence.

1. Prove tha 七 勺k ' ;;;; are preorders, tha 七 ;:::; is an equivalence, and tha 七 B :::: C implies B;::;C.
2. Prove that ;::; is a congruence; in particular, tha 七 B :::: C implies

加. B+D 之 C+D (f irs 七 sh o,; th a 七 ea ch 匀 has this property). 吐 us

;::; and :::: differ, since the la 七ter is no 七 a congruence.

1. Find a simple example in which B ;::; C but B C • Also sh叩 (by

a similar example) tha 七 ;::; does no 七 r es oect deadlock oroperties in the sense of Exer c 运 e 3.6.

吐 is is why we rejected ;::; as our no 七i on of obs erva过on equivalence, in spite of its 改 m 王 t simpler theor:y 。

CHAPTER 8

Sane proofs abou七 da 七a s 七ru e 七ur es

* 1. 工n r七 oduc过on

We have already s 比 双 n same no 七 qui te 七 过 vi al algorithms and systems expressed in CCS. The poin 七 of this chap七e 工 i s twofold. First we wan 七

to shON th a 七 familiar da 七 a structures, as well as algorithms, find na 七 匹 a l expression in CCS; second, we wan 七 七 o i ll us tr a 七 e hON the properties of observation equivalence and congruence all0,1 us 七 o prove tha 七 sy s 七 ems work properly. The data structures here give good proof examples. To what extent they correspond to hardware realisations mus 七 be le£ 七 open, bu 七 i t does no七 a ppear unreasonable tha七 a 七 l eas 七 sane har dwar e s 七ru ct ur es can be

faithfully represen 七 ed in ccs.

* 1. Registers and mar 已 es

The s impl es 七 shar ed resource, which may be the means of in 七 er act i on be切veen oth 如 is e independen 七 agents , is probably a single m 已 刊 Jryr egi s 七 er . Many concurren 七 al gor i thms have been represented in languages which penni 七agents to in七er act onl y 廿 u 文 汜 gh 'shared variables' (usually 'wri 迁 labl e ' as

well as'readable') • We argued in§4.5 tha 七 al gor i thms are no 七 al ways bes 七

expressed this way - many people haver ecen 已 y made this poin 七 ．

But if we do wan 七 a register, readable and writeable by one or rrore agents, its behaviour may be well represented by 斑 (v ) : fo , y} defined by:

REG(v) {= ax.REG(x) + 了v . REG(v)

卓

沁 kin ds of atanic experilren 七 ar e possible:

(write)

REG(v) 早 REG( u)

REG(v) 立 REG(v )

We may also f扭d i七 useful to define

笃 牛＝ 心c. REG(x)

(wri 七 e u) (read v)

(read)

- a register without initial conte七n， whi ch a 七 f ir s 七 a dmits only writing.

工 f we def 年 e relabellings S. = a..y ./ay (1sisn) where the a.,y.

l. l. l.

are all 主 stinct narres, then we can def 年 e a m.arory of sort

{a1,Y1'... ,a.n,yn} by

呻 RYn 立 [ S 1 J I • • • I 立 [ Sn J

l. l.

or, using II to repr esen 七 mul 已 pl e cai 贮 IOSi 七 i on:

86

a2-0

a10

rr

咖 !RY =

n 1 s;i 匀n

立 [ S . J

1.

1 丫2 n

Note th.a七 thi s u妥 of a::r.p忑 i ti on jus七 pl aces the registers side by side; they don't ccmnunicate with ea 吐 other !

迳 t us 江 suppose, rrore realistically, that we want to build a m 五 江 江 y

of size 2k with jus 七 thr ee ports:

位） At a, it receives in sequence the k bi ts 气:-1 ' … ,a。of a

m 卸 r .r address m , Osn<2k ;

1. A七 s i 七 re ce i ves a val ue 七o be wr让ten a七 addre s s m;
2. A七 ？ 让 de li ver s the value stored a 七 addr ess m.

远 us call the 至 0 巧，s tor in g values v = (v。,… ，v2 曰 ， 汽面：fo, 五｝．

We shall adop 七 a convention which is in fact a reality for magnetic core

芷!!Or i es ; destructive reading. 农 ）至 a 芦 value u into address m i n 汽(v ) , the envir o nmen七 wi ll perfo:r::m

气 -1 · ... . 血 。池 . yx...•

andignore the value received a七 y (which is bound to x) ; this value will actually be vm. Thus to 竺 th e 芷 m::>ry a 七 m , the env 扛 :onr汜 砒 f i rs 七

writes an a:t:bi tr 王 -:y value (say O) 七o m , receives and holds v , and

m

writes vm

back a 七 m ; i 七 per fo rms

气- 1 · . . . 气 . so. yx. 气- 1 ..•.－. －.sx.yY. B

where B (the continuing envirornnent behaviour) will use x sa 芷 ilia.v, but ignore y •

In surrmary then, we can express ha.v we wan 七 汽 to behave by saying that for any environment expression B of fonn

气- 1 •••••气.su. yx. B'(1)

tfolla.ving observation equivalence mus七 hol d:

（ 汽面 I B ) \ 归 \ y ( 汽（环{u/ m) ) I B'{vm:凡｝）＼矗 \ y (2) where v(u/m) means (v0, .. ,vm-1,u,vm+l'..,v2 k一1 ) •

'!his rE 芍 山 丘 叩 豆 让 is an example of illa:ill)lete speci fi ca 已 on; we do no七 s peci fy wha 七 hap pens if B supplies too fe 寸 or too many address bi ts, or acts strangely in sa 汜 other way. 工七 i s a na 七ur al in ca 丐,l e 七enes s ,

because we 皿 gh 七 na 七ur al l y ccmpose 芍< with a ' fr on 七 end ' agen 七 whose job

is to receive integer addresses, decode them into b正 sequencesof length k (立叩l aining if the int ege工r e 怎 i ved is outside the range [ 0, 2k一1 ] ) and conduc 七 the correct reading and writing sequences with 汽 . Also, the incanplete specification actuall y 咡 kes the design of 汽 eas y , as we

shall see.

A specification which would be 竺 王 in a:mpl ete 沁 ul d be to demand 芷 江 e l y

that

气 一1 · . . . 气 . s u.yvm ＞ 汽 (v (u/ m) ) ;

汽面

ce rtainly 汽向 mus七 have this de 过 vation for every m = 婓 1 ' , •• ,m 。

and every u , but this 皿 d no七 excl ude unwanted derivations like

汽而 幸 NIL

- deadlock:

归 le 七 us abbreviate {a,f3,y} by L, and def ine 田垃 tr 尘习 sorts

L。= {a O,f3O,Y O'}

L = {a ，队，y }

上 1

1

, asking only tha 七 al l these nar 汜 S

cv., • • , Y1 are distinct andtha 七 cvO.

, f30,YO,a1, f31,Y1 don ' 七 a ppear in 怜 ，

the sort of B • We will also abbrevi 玩 e \a.\f3\y by \L , \a 0\f30\y 0 by \L。,etc., and se 七 S . = a.f3.y.jaf3y, i = 0,1.

1 1 1 1

Fir s 七 we can see tha 七 the Sfe Ci fi ca 七 i on (2) is equivalen 七 to demanding

（ 汽<v> rs。J l B。闷之。( M炒<v硒 ））岱O J l Bo {vnl x } 心。(3)

for any B。of fo 皿 呫lc-1 · . . . • 守o · Sou· Y卒. B0 ; to deduce (2}£ran

1. we note that

（ 汽而 I B) \L = ( 汽面 I B) [S。] \L。 （竺 (1 ) , (2), (4)

＝ （ 汽(v ) [S。J I B[S。J ) \L。（竺 (5) )

＝ （ 汽面[ S 。J I B。) \L。 （竺兰 (3) )

with B1 = B'[S J ;

0 0

the other side of (2) transfoJ:lllS similarly, and (3) can be used to ge 七 ( 2) .

conversely to deduce (3) fran (2) 殴 欢 江 k with R0 = a衍/ a0130y0 , the

inverse relabelling to s。,and use 旦 : (1 ) , (3) 归 ing th a七 R。oS。= I ,

the identity relabelling. Such mani pula 已 ons should becare rou 七 年 e !

We nc,,, cane to the design of 汽 . M 。 (v) , the naoory of size 1 con­

taining v, is given by

M。(v ) = CELL(v) , where CELL(x) i= 勺 YX. CELL(y) (4)

(The a p:Jrt is no七 used. )

卢)

We build ?\+J; (v:w) (v,w each of length 2k; ~v:w~ is their co-nca-ten-

ation) ou 七 of 凶<:: (v ) and 凶<:: (w) by ca:nposing th 甸 wi th N'.DE: LuL0 uL1 ,

whose job is to inspect the f ir s 七 address b让 z which i 七 re ce i ve s and - roughly - tr ansmi 七 th e r es 七 of the ccmnunication to 汽面 or 凶面according as z == o or 1 . Precisely:



虹 E <= az. NOOEz

虹 E. <=

l.

az－.a.－z. －虹 E. + 躯 . 13 x.y.y.yy. 虹 E

(5)

l.

l.

ii

(i=0,1)

and

汽+1(v :w) = ( 汽(v) [ S社 1 汽(w)[ S1 J I 虹 E ) \ L。\ L1 (6)

「

I

-­了

已

\_ -－,－＿

--

**,**

Notice tha七 NODE does no 七 know ha,, many bi 七 s to receive; 让 m比 3七 be

己、｀、它，

I

心

)

，

/

，

i

ready for an address bit 2E a value, and act accordingly.

吐 e diagram on the next page sha,,s M (v) , with arra,,s indicating

3

the initial capabilities of the ex平 nen ts . By swinging arrows about on

让 ，you can convince yourself tha 七 i 七 work s - and th a 七 ， wrong ' s equ en ces

deadlock·e.g. M (~v)

3

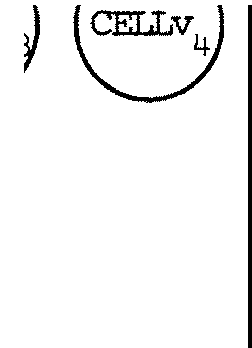
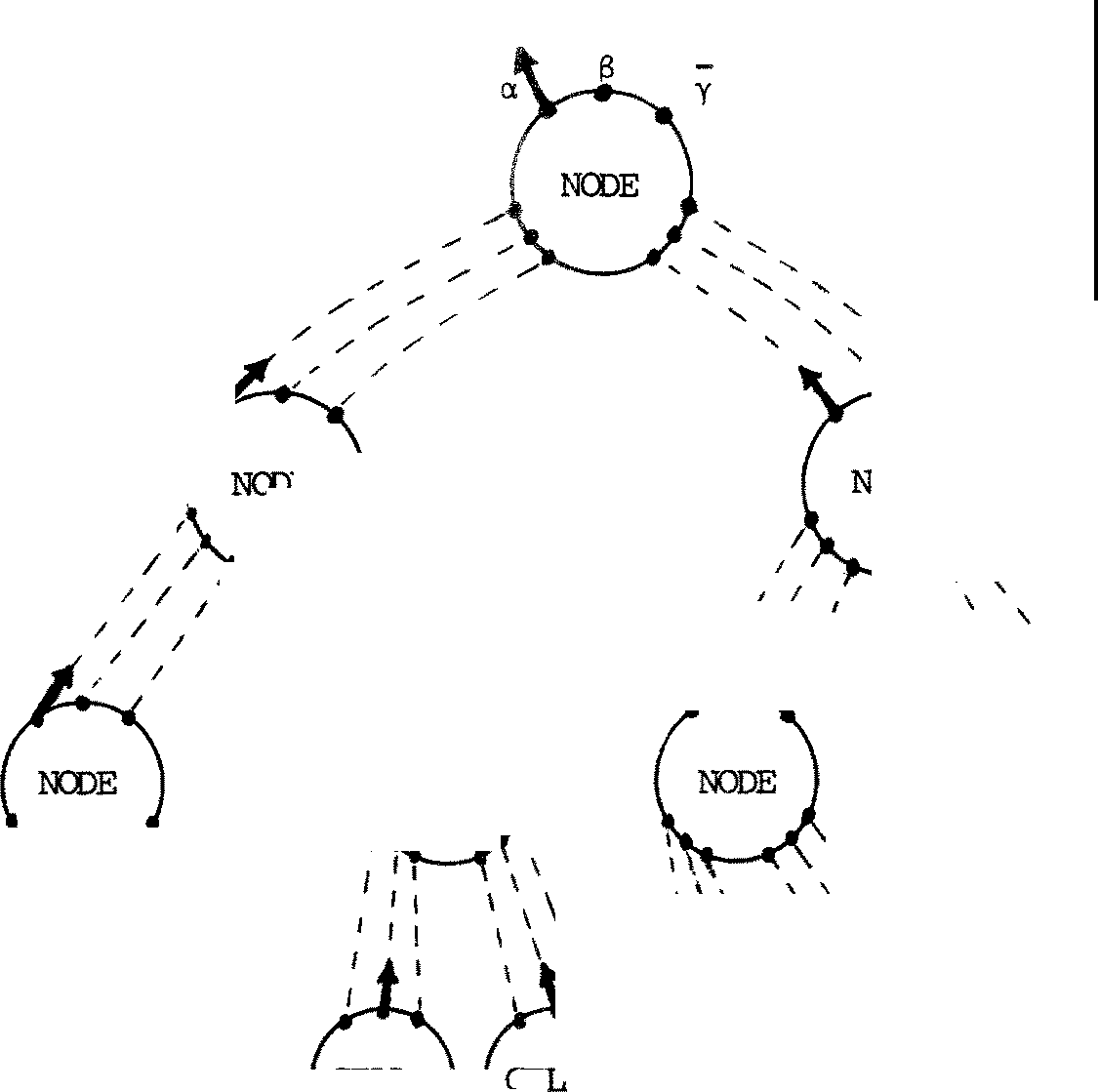
etO .ci1. u > N 耳 ．

（吐 e 迈 ea to use as an ex 郘 iple a rrerrory bui 止 of nodes whi 中 ' use the f irs 七 b 让 to direct 七r af f i c ' cane fran a 七 alk with Nigel Derrett, who told me that this nethod is used in practice.)

Having no,, def in ed 汽 r ather succinctly by (4) 一 (6) , and specified its intended behaviour by (1) and (2), we proceed to prove th a 七 i 七 meets

its specification.





**\** `

\ ＼

｀

`

}

U

＼

`

\＼心)

-

Theo忘 8.1 For any B of fo nn 气- 1" ..••皿。.su.yx . B',

（ 汽<v > 1 a)\L :;:: < 汽(v (u/m) ) I B'{vrrfx} } \L • Proof For k = owe have, sinoe v = (v 。) ,

( M 应 I B) \L = (sy.yv 。 • CELL(y) I su.yx. B') \L

。

= 1:•-r-(CELL(u) I B'{v / x} ) \L (Expansion)

;::: (M 。(v(u/o)) I B'{v。/x}) \L (Proposition 7.1)

as required. Now as s晔 th e theorem for k • Take B of fo 血

气 气- 1 •••••气.su.yx . B'

and consi der 汽岳 ）， where 研 are of length l . We wan 七

（琴声） I B) \L:::: ( 汽+ 1 ((颈 (u咬 > > 1 B'{ <=v 丸）啋/ x} } \L

where m = 吹 1 ' . . . ,mo • By symretry i 七 wi ll be enough to prove this for the case 叹 =o , which is to say we want

(?-\+1(益） I B)\L:;:: ( 汽+ 1 <环(u /m) :w) I B'{vrrfx}) \L 'lhe left-hand side is, by (6) ,

（（ 汽面 [S。J I 心<w> cs J I 虹。E ) \ L \L I B)\L

．干｀

'-...-v-:' 0 1 .

,,\_,

LO L1 L uLu L1 诠

(wr 让 in g sorts below)

＝（ 汽（如 s1 J I < 汽向 [S。J I <立 ）叫B) \L) \L)0

\L 1 (7)

where we have regrouped by repeated use of 三 and by 竺空王），

rerrerrbering th a七 L。归=·Ll 咋 = ¢ .

Nowr ecal l in g 吹 =o , by the Expansi on 'lheor 细

(NODEIB) \L = 1"• ( 虹 E。| 气 1 •••••气.su.yx . B') \L I

:;:: a. 壳 \_1 . ••·.a.0m0 .s 0u.y 0x.(NCDE\B')\L

by Proposition 7.1 and'lheorem 7. 3 • Bu 七 thi s is a B0 of the fonn needed for (3), which we shc:Med equival en 七 to the theorem at k (which we're assuming) ; so recalling Theorem 7. 3 - tha 七 can be s ubs 七迁 ute d except under + - we can r e 江 i te (7) as

< 汽 <w > cs1 J I < 汽（姊(u/m) ) [S。J I Bb{Vrrfx}) \L。)\Ll

where B = (NCD引B' ) \L , so B {vrrfx} = ( 虹 E I B'{Vrrfx}) \L sinoe x is no 七 a free variable in NCOE • Now we can regroup, jus 七 r e versing the

operations by which we go 七 the fonn (7), to ge 七

＝（（ 汽（如(u/m) ) [S 员 1 知w > cs1 J I 叩 ）\L。\ Ll I B'{Vrr/X}) \L

＝（ 汽+ 1 (环(u/m) :w) I BI {vrrfx}) \L as required. 召

Exercise 8.1 Suppose you have available a dea:xler, which accepts an integer (assumed to be in the range [ o, 卢 一 1 ] for s 吐 f ix ed k) and decx:ides i 七 in to its bi 七 跃 习 U 平 ce . 'Ihat is:

D 配 中 E: <= om.气- 1 · . . .

－. 皿。.－,;;• D贮 立 ：{o ,a,,;;} •

'Ihe integer CXllreS in at o , the bits goou 七 a 七 a , and I; signals canpletion.

Design another agen 七 ， cal l ed 巧 还>NT.印叩 ， so that when you ca 平 运 e

军 OOE , FRCm'.END and 汽(v) with appropr i a 迳 re l abell ings and

年 s tr i cti ons you ge七 a system 哑汽面： {a,S,y} satisfying

气面 ～ 皿 (sx .MEM(v (x 加 ）） 十 YVm 笠汽{v ) ) •

(To wr 扛e value u a七 address m, the user perfo五芯 ；in.百u. … ； to

年 ad the Ill 茧 0 巧 , a 七 m and bind the recei v 式 value to y he perfo 五 怡

am.yY. • • • . ) 竺 the desired equivalence.

至 FRCNI'END and D配 ODE mus七 cxx,per a扫 to prod uce 呤 res si ons of the fonn B , so tha 七 you can use'Iheorem 8. l abou 七 卧 位 ） ．

尸 B. 2 can you think of a way 七 0 redesign 汽面 so tha 七 the

outgoing value 年 sn ' 七 have to r 七 avel up the b inary 七正汜？

* 1. a 沮 让让nc, 哼 ati ons

S 屯 杠 汜 e we have agents B1 and B2

e 百 a.CS)

and wish to join them like this:

0--------@

工 七 is na 血 al to define a binary op 红 a 七 i on 一 f or 七 hi s purpose. Defini ti= 压 t B主 ， B,e L2 and 沪 ； 中 en

I

B ' B

、 =

1 2

(B 1 [o/13 J I B 2[o/ci. J) \o where o i nanes(L1 LJL)

Note 出 a七 出＿e ＿definition is specific to 百 and ci.; pe:r:haps we shou 垃

2

write B1 f3 ci. B 2 •

We need to justify our definition by shCMing that the choice of o

doesn 兄 af fe ct 让 . To see this, suppose th a 七 o ' f nar 汜 s (L uL) , o' 邦 。

1 2

Then

(B [o'/sJIB2[ o' / a J) \ 护

= (B尸' / s J !B2 [ o1/ a.J ) \ o' [ o/ li' J by 竺 (1) , (2)

= (B [o'/13J[o/o'JIB [o'/aJ[o/o'J) \o by 三 ( 4) , (5)

= 《

(B [o/BJ IB [ o/ a.J)

1 2

by 距 1~(3) •

Note tha 七 B ..... B may fonn other links, depending on L and L ;

1 2 1 2

this doesn' 七 af fe ct our argurren: 七 ， but we are mainly interested in the

case L 心 = ¢ .

1 2

'Ihe importance of .-. is th a 七 i 七 i s associative; this property is helpful when we need to chain several agents 扛巧eth er . Let us prove associativity. Suppose B :L'B :L , B :L .

08

1 1 2 2 3 3

a@S a@

'Ihen

(B ,,...,B),..B = { (B [o/sJIB [6/aJ)\o[z;;/SJ IB [Z::/aJ) \s

1 2 3 1 2 3

choosing o, z;; 足 nanes {L1 uL2uL) and 沪 ；

3

= ((B 1 [ o/s]1B2 [吵])[z;;/a]\6 1B iz;;/a]\o) \1;;

by 距 l ~(4) and Res~(l) (we are pushing relabellings inwards, pulling restrictions outwards) ;

(B [ 6/ 13][ 1;/S 准 ［ 沪 ］［妇 ] 1B [沁])\归

=

1 2 3

by 竺 (5) and 三 (3) (check its s迈e condi七i on! )

= (B1 [o/aJ IB 2[or/aSJ 1B 3 Cr/ 叮）\z;;\o by 距 1 ~(3) and 距 s ~( 2) ;

= B1....,.(B2 ,...,B i by s 泗 畔 巧 ·

Exactly the s主 can be done for double ch 忒 卫 叩 ； given two agents

0: :0

we wan 七 七 o join them together to give

@\_ 一一3飞

\ 七 B 凸 ，B,2 r., and le 七" •'·归

B1 己B2 = (B1 [ n/ s , e/ oJ I B2 C n/ a , e八J}\ 6\ n where n,e i names(L1uL2) and 哗 ·

be di s tin c 七 'Ihen

工 七 is easy bu 七 tedious to check 七 he as soc i at i vi 切 of 之 We shall use this 妇 ati on in the nex七 secti on .

Both ,...,. an d 乙 gi ve us special cases of'lheorem 5.8, the Expan­ sion'lheorem; we j us 七 s ta te i 七 fo r " , in the sirrple case where

B ,oo•,B : {a 初, ，io e . no labels are presen 七 郅 cep 七 the chaining

1 n

l abel s。

它---言--…飞歹

＿

欧 沺 沮 on 'Iheo rem for'"' If B1,…，Bn :

of guards, then

B 一售 B ,.、.. o o • ..-.B =

1 2 n

位，防，and each is a sum

沁记( B 广了。

＾Bn) ; 必 B a sumnand of B1}

＋汇 v . (B ......乌,..,. … 飞 ） ；玑

Bn' a sumnand of Bn}

+ H 石 (B ,...\_• • • ,...\_旷叱 1 { 誘 },. , •••飞；

1

阮 Bl a sumnand of Bi, ax . Bi 十1 a 过 晔 nd of B赶1 t

All tha 七 廿 让 s says is tha 七 th e only external actions occur at the ends of the chain, and the only internal actions occur be 切 汜 en neighbours. 殴

祖 1 use the corresponding theorem for 8; i 廿s ob吐ous enough, so 炽

do not write i七 d o;讯h

8 。4 PushdCMn.S and 中 e ues

I.e 七 V be a value se 七 ； we use s 切 r ange over Vi< 。

如 社 s houl d be the behaviour PD (s) : {a, 了 } of a pushdown s 七 ore in

血 中 val ues are pushed in at a and 区杠泣 OU七 a 七 y ? Ar eas on 址让e suggestion is

PD(s) <= ax. PD(x:s) +

if s=e: then y$。PD( e:)

(1)

else y (f i rs 七 s } 。 PD(re s 七 s }

Here':'is theprefixing opera七i on OVE江 Vi' , and'$'indicates

句P已迳ss; 殴! tes七 the pushdo;灯1 for a:耳,tm 涟 s by popping and te s七止巧 the value popped 。

'lhus 炽 want to build PD(s) to sa 七拉;fy

PD(e) = ax. PD(x:e) +讳 PD(e)

PD(v:s) = ax. PD(x:v:s) + yv. PD(s)

如 吐 炽 shal l ac 七 ual l y build is PUSH(s) : {CJ.,y} 七o s a 已 s fy PUSH (e) = ax. PUSH (x: e) +祜. NIL

PUSH(v:s) = ax. PUSH(x:v:s) +和. PUSH(s)

the only difference being tha 七 PUSH(e) , when popped, degenerates to NIL .'Ibis is easier to bui 记 ，and i 廿 s also easy to build a sr;ecial fron 七 end , FROOT, so that (2) is sa 七 i s fi ed by

PD(s) = 巧码 已 PUSH(s ) •

(2)

(3)

We build PUSH as a chain of cells, each of which can hold O, 1 or 2 values, terminated by an end cell holding $ • A cell holding

y is

CELL1(y) <= cxx:. CELL2(x,y) + Y:f• CELL 。

竺 1 ('{) : fo－,13,－Yf o} 贮

酝 th e r es 七 of the def ini 七ion is

过

单 (X, '{) : {a, 环 ，c5}

-

立 2(x , y) <=时.CELL1 (x)

单。：如陌心

｀

u

CELL。<= 沉（旦 x= $ 竺 CELL$ 竺 CELL1 (x)) (6)

罕

立立 $

令 ＝ 呕 ．（哑 立

(x ):: CEIL ) + y$. NIL

(7)

we s 切 the successive oonfigura 七 i ons of a typ i cal 竺 i va 七i on , starting f 立 :rn CELL , in the diagram belCM.

1 $

＄

江 j ., (p,sh 6)

切：巨）

-l\_aBS (push 5)

穸

--B B

l T

汇

l yS (砰 5)

an-- B

l

D

T

沁 飞义

l a2 (push 2)

它飞义:

B

l T

#### 它

(8)

＿

'Ihe der i va已on CELL u6.u5.y5.u2

＄

> CELL (2, 6) 今 CELL •

2 — $

NON for any s = (v 1, ••• ,v) n

l e 七 us refine

PUSH (s) = CELL (v ) 今．．：。： CELL (v) ::CELL • (9)

1 1 1 n $

Clearly PUSH (s) is stable; the fourth oonfi gura 已 on in the diagram sh0;vs you th.a七 no r-actions are possible. It is also reasonably clear that every oonfiguration will s七戎社li se , gi ven 七让妇， but th.a七extemal a:mnunication can occur before stability is reached.

* 1. e 七 us see wha 七 we n eed 七 o prove (3), which is ou 工 aim . Fran (9), by the Expansion'Iheorem, we get

PUSH(e:) = CELL = 心 (CELL (x) CELL) + y$.NIL

$ 1 - $

＝ 必 PUSH(x: e:) + y$. NIL

so the fi r s 七 part of (3) is done. (Recall tha 七 we allCM ourselves 七 o write'='whenever we use a oongruence,'~'or':::::c', and th a 七 ' = ' al ways impl i es ' :::::' 。 ) We also ge 七

PUSH(v:s) = CELL (v)::-PUSH(s)

1

必 (CELL (x,v)::::::PUSH(s)) + 丫 v . (CELL:;:: PUSH (s)) •

2

0

we therefore pr cpose 七 o prove

CEIL {u,v):::PUSH(s) :::: PUSH(u:v:s) CEIL。PUSH(s) :::: PUSH(s) •

2

These canrio 七 be oongruences (:::::c) since the left-hand side is un­

(10)

(11)

stable in each case. But':::::'is strengthened to'='by aguard (Pro 产 i ti on 7.12), so for example from (11) we deduce

亿 (CEIL。:::PUSH(s)) = yv.PUSH(s) ;

applying the s畔 七 echnique to (10) we finally reach (3). We have

achieved equality (=) before substituting under 中 ．

To prove (10) and (11) we only need four li t 七 l e lermlas,

grouped 七 ogeth er :

Iamia 8.2

1. 四立 (u, v) 今立

2

1 (w) ;::; CELL (u) 已立立 2 (v ,w)

1. 四 立 2 (u , v) CELL$ :::l CELL (u) 今 哑 立 (v) 已 立

1 1 $

(3) 四 立 0= 笠 立 1 (w) :::: CELL (w) ::;cELL0

1

(4) Cl 立 0

Cl立

$ :::! CELL$ •

Proof All by the 酝 pansi on Theorem1 we need only consider the first in de七ai l.

##### ：它一一切：

We have

CELL2(u,v):;CELL1(w) = 又 (CELL1 (u) CELL2(v,w))

::::J CELL1(u) 已CELL2 (v , w) by'lheorem 7.1 •

For the last, we need the fact tha 七 CE 口 ; 之 :NIL= CE压 ．

$ $

巨三e 8.3 Prove this simple fact. 员

妇 (10) and (11) follCM:

丑平ma 8.3

1. CELL2(u,v)::::;PUSH(s) 2i PUSH(u:v:s)

。

1. CELL PU SH(s) ;,; PUSH(s) •

Proof le 七 s = w 口...,w : to ge 七 (1) , use the def ini 七 i on of PUSH,

and apply Ierrrna 8.2(1} repeatedly, then Ien1na 8.2(2). To get (2), use Ierrrna 8.2(3), (4} similarly. Note th a 七 ;::; is preserved by sinae thel a 七ter is defined wi th ou七 usin g +• 幻

So by wha 七 we did before, we have se 七 七 l ed

吁1eorem 8.4

PUSH(e:) =呕. PUSH(x:e:) + y$. NIL

PUSH(v:s) = 呕 . PUSH(x:v:s) + - PUSH(s) •

员

Exercise 8. 4 Analogous to (3) , we may specify a 丑 竺 by QUEUE(e:) =立. QUEUE (x: e:) + y$. NIL

QUEUE(v:s) = 心 QUEUE(v : s : x) + rv. QUEUE(s) •

(Note th a 七 I : I is being used to J?< 文 ；七 f ix elements to sequences, as well as for prefixing.) Make a very small change to the behaviour of

CELL2(x,y) (5) , and adj us 七 the LJ 五 m 玉 ； to shCM tha 七

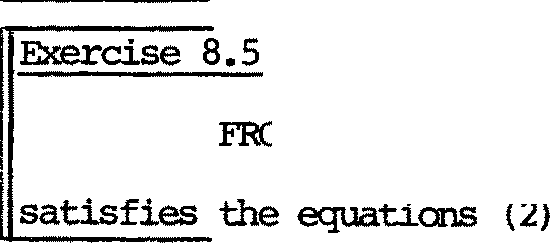
QUEUE (s) = CELL1(v 1) ::;…：：：： CELL1 (vn)。立

(for s = v1,…，v 礼

satisfies the al::ove equations.

{a.压，飞o } so tha 七

PD(s) •



呻 8 PUSH(s)

for

屿 炽 江 e rather careful 扫 our definition (5) of CELL (x,y} ;

2

让 m 耳 push y ck沁 be for e i 七 can pop x• Was this necessary? By

considering diagram (8) and s 血 .lar 远 i vati ons you can p 过 油 l y satisfy yourself tha 七 0 立 2 <x , y ) 竺 巴 be all0t1ed to pop X. 如 a七happens to our proof though? Le 七 us redefine

CELL (x , y ) 牛 = yx. CELL (y) + BY. CELL (x} •

'－

言

2 1 1

We need only make sure tha 七 I.srrrna 8. 2 (1) , (2) s 七 i ll hold. For the fir s 七 ， we have by expansion

0 立 2 (u , v) -..-.,CELL1 (w) =

+

yu. (CELL (v) 乞CELL (w)} -r. (CELL (u) :,CELL (v,w)) (12) 1 1 1 2

：妇勹3

which does no 七 l ook r i gh 七 Bu 七 can the f ir s 七 tenn be absorbed into

the second? By corollary 7.14 - a derived absorp 已 on 1 - 袒 must show

CELL (u}:;CELL (v,w) = yu. (CELL (v) CELL ,(w)) + B (13) 1 2 1 1

f or 改 年 B• Expanding the l ef t - h五过 si de gives

沁三笠

＿

CELL1 (u) CELL (v,w) = 记 (CELL ::'.:;CELL (v,w)) + B

(14)

2

while e:.l 干 玉 血 g part of this gives

##### ：这一一一衮：

0 2 1

CELL 已CELL (v ,w) = -r. (CELL (v)乙CELL (w)) + B • 0 2 1 1 2

虹 pu 七 (14) and {15) 七 oge ther :

CELL (u) :＿:;CELL (V,w)

2

(15)

1 = yu. (-r. (CELL (v) CELL (w)) + B) + B , = B say, 1 1 2 1

= yu. (CELL (v) =CELL (w)) + B by Theorem 7.13 (3)

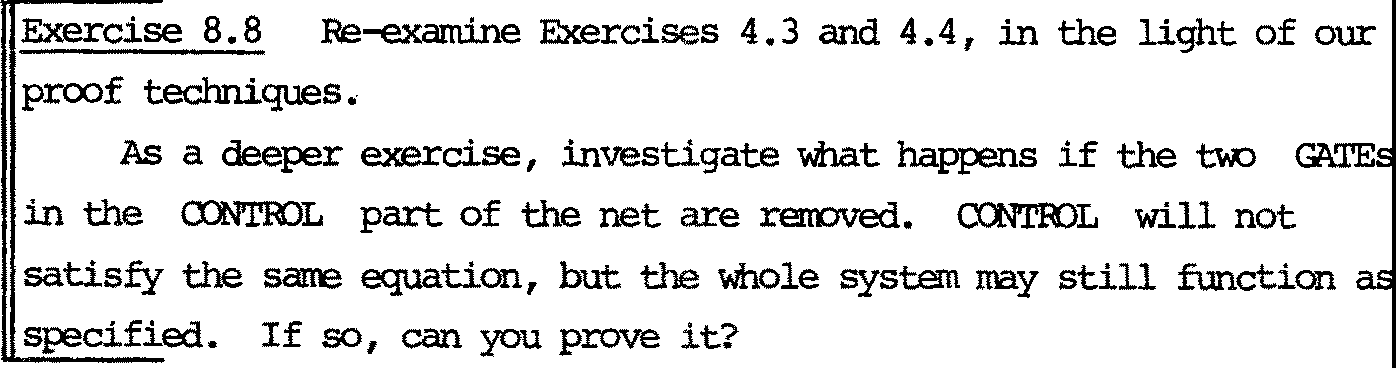
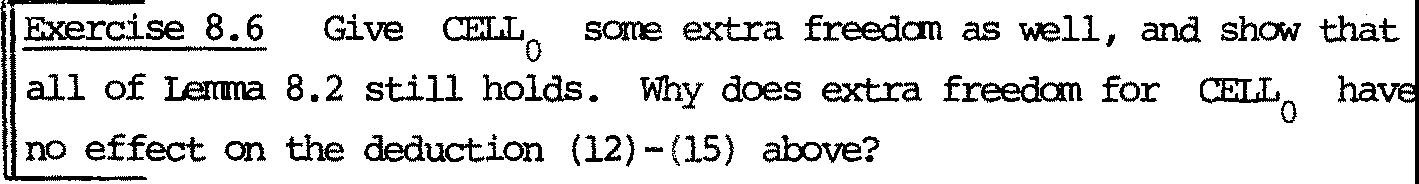
1 1

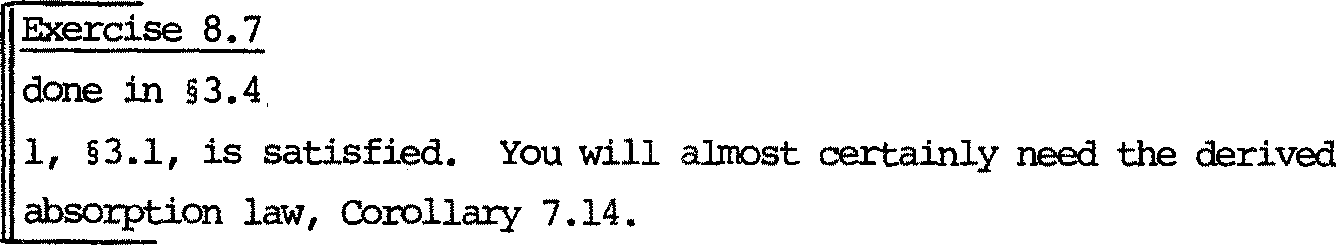
心 .ch is 啦 比 袒 wanted ! We 叩 have (13), and this justifies the step fran (12) to

立比2 (u , v 心 立 辽 1 (w) = i:. (CELL1 (u)=CELL2 (v ,w)) ,

SO 袒 S 七 i ll have I.emna 8.2(1).

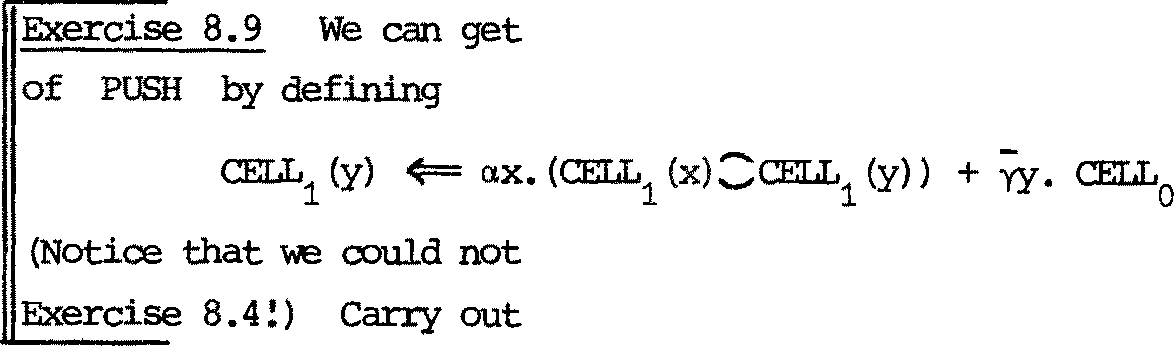
巨 se 8.5 ShCM th.a 七 Ianma 8.2(2) s 七 i ll holds, 七 oo .





呻 l ete the proof of the scheduler, half of 平 ch was

done .in§3.4; i 七 r a:nains to shcM tha 七 the second cons七r-a .int .in Method

过 d of CELL2

cx:mpletely fran the definition

then adapt our sys 七 em 七 o fonn a queue, as in

the proof for this changed system.

CHAPTER 9

Translation in 七 o a 玉

* 1. Discussion

Many concurren 七 al gor i thms can be expressed :in CCS with sare lucidi 切 · On the other hand, the aim in designing a high level concurren 七 l anguage is (in part) 七 o provide and enforce a discipline in 七 he way in which o 卫 屯 立 1en 运ccmnunicate and share their resources, partly t.o pro te c 七 th e prograrmer

fran unwanted deadlocks.'!his often restricts (usefully) the behaviours which may be expressed.

If such a language can be r 七 ans l ated in 七0 ccs, i 七s meaning is thereby detennined; we al so 中 tain a way of reasoning about the language. For example, observation equivalences 虹 ng its programs can be established, and these may yield useful 1 现 1s for program transfonnation.

In this chapter we give a tr ansl a 七 i on for a rather s 扛 叩 l e language. I七 is a s ubse七 of va豆ous languages in use; also Hennessy and Plo廿c:in [HP lJ have specified its s中印七i cs :in de七ai l , in a ve 工y different way.

Our tr ansl a 七 ion is qui 七 e straightforward; the main reason for this is tha 七 th e scop:ing of program variables, which often requires the use of a notion of env扛立m平 t in s巴芷m七i c specif i ca七i ons , is for us represented directly by the restriction operation of CCS. Ha., 汜 ver , when we examine how to translate an enri 吐 lilV:i! 砒 of the language in which procedu 年 s may be def:ined, and each procedure is supposed to admi 七 sever al concurrent activations, we discover a l imita 七 i on of CCS in its pr esen 七 fo 五 n (we can handle a procedure which cannot be concurren 七 l y activated, however) •

'!he translation will be seen to be phrase-by-phrase; each phrase of the language bea: 咋 s a behaviour program which is to 七 al l y :independent of the context of the phrase. (Such translations are sanetimes called macro­ expansions.) We shall wr让 e UCD to mean the 七工ans l a 七i on of phrase C.

For example

0IF E'lHEN C EISE C']

will be oonstructed uniquely fran OED , rrcn and [c ID• 书 让 s 晖 ans tha 七

the construct "IF-'IHEN- EISE 一 " :in the source language can be though 七 of j us 七 as a der i ved 七 em ary behaviour ope 迳 已 en. We can 七 hen think of 七 he en 七 ir e source language as a derived behaviour algebra.

* 1. 'lhe 1五巧立巧e P

Programs of P are bui 止 fr an e} 吓江es s i ons E and camiands C, using assignable program variables x. We suppose a fixed set of functi on 芍叩比1s F, standing for functions f. A oons tan 七 symbol is jus 七 a nul l a 工 y function symbol. We do no 七 spe ci 仑 y the value types of expressions.

The syntax of expressions is just E::= XI F(E, .•• ,E)

(This includes e.g. "+(X,1 ())" which is wr 让 ten 11X+111).

The syn 七 ax of carrnands is

C: := X:=E C;C

IF E 屯 迁 :N C E 岱 E C

邓 I LE E OO C BEGIN X; C END

(Assi gnmen 七 ） (Sequential ca可力廷已on) (Conditicnal)

{I ter a 已 on) (Decl ar a已on )

C PAR C (Parallel en: 平 s i tio n )

INPur x (In pu 七）

OUTPUT E {OUtpu 七）

SK工P (No action)

(Parentheses are used to avoid parsing 吽 iguiti es) .

The main doub七 abo u 七 th e meaning of P i s 七 o do with PAR. For example, can the'concurrent'assignments in the program

X:气；

X:=X+1 PAR X:=X+1

overlap in ti.Ire? 工 f so, the resulting value of X could be 1 or 2; it no七， it must be 2. Our f ir s七 tr an s l a已 on will yield the fo立在江 ； \-le see hCM to ge 七 the l a 七 ter afterwards.

* 1. Sorts and a 血 li ary definitions

Each variable X will be represented by a register (§8.2) of sort

凇心｝．距ca ll in g §8. 2 , we define

立 ：{ex,石 妇 cxx . REG(x}

REG(y): 位，勺仁 辽. REG(x ) + yy.REG(y)

吐 us for X we will have 辽 艾X = LCC[o. 沁 x \ o. yJ;

we will abbrevia七e REG(y) [CL沁X \ o.y J by REGX(y)·

We use Lx = fa. 炉 X } - the OJI 平 erren七 of the sort of 辽 汽 - in defining the sorts of cx:mnands and expressions; we call i 七 th e access sort of x.

Each n-ary function symbol F (denoting :ftmctian f) will be repre­ sented by

厂－丹年令....琴. p (f(x1, ••• ,xn)}. NIL

如 s e sort is {p , •••,pn,p}. So for a ccnstant synbol - e.g. 2 - we have b2 牛 P 2, NIL,

Each expression E with variables X1' …， wi ll be represented by a behaviour program of sort {Yx , … ，丫泾, p}. 'lhus expressions deliver

their resul 七 a t p, and then die; this means that if [E] is the transla­

已 on of E i 七 ha 旦 th e property

•••.pv

ijEl ==> B 乓 l ies B = NIL.

In tr ans la 七 in g caanands we often write, for saoo B,

(UEl I px.B) \p

which we abbreviate to [E]r esul 七 ( px . B) , defining the behaviour operation res ul七 by

B1 re s ul 七 B2 = (B1IB2)\p.

Each ccmnand C wi th 斗竺 var i ab l es x1, ••• , 工 1 be re pres en 七 ed

by a behaviour program of sort L u… UL.. U{ l 声 ， 江 粔 call this program [CD; it uses l ,o for inpu 七 and OU 七 pu 七 and si s i 七s carpl e七i on a 七 ti. I 七 七hen dies,

鸟占

so (C] 兰 今

0 B 扭 l ies B = NIL

Sare auxi l 坛巧, behaviour opera已ons are u翠 ful in defining [CJ;

done= o.NIL

B before B = {B [ f3八］匡 B ) \ f3 (f3 new)

1 2 1 2

par +

B B = {B [cS /cSJ IB [o /oJ I (o .cS .done o .o .done) \o \o 1 2 11 22 12 21 12

(o , o 产 ）

91 o盖f

* 1. 3 , 5,5 and 7.13 to sm< tha 七

before and 竺 are associa ti 定 ， and 竺 is ocmnuta ti 凭 ．

we 心 ha 诧 all we need to define the translations [E] and [C]

inducti 袒 y on the structure of phrases.

区 erc ise 9.2 Prove, by induction on the structure of exp 年 ss i ons and

carma:nd.s I that

* + 1. If E contains variables x1,••• , then [EJhasthesort

坎U…U坎u{句．

1 k

也 ） If the non-local (free) variables of C are x1, ••• , 芍C then

* 1. has the sort 马 u ••• u马｛心 ，o, 江 （泣 te th a 七 X is local
     1. k

(bound) in BEGIN X; C END.)

Many simple equivalences over P can be shCM'l fran the translation.

Here are a f 窃 as exercises.

Exe工ci s e 9.3

1. Prove『SKIP;C] [ C 』
2. Prove [WHILE E 00 CJ~ [IF E'IHEN (C; WHILE E 00 C ) 江 S E SKIP]
3. If X is not a free variable of c, prove

［ 郔 IN X; C ENDJ ~ [CJ

［斑严 X; C; C'END]~ [C; 延 IN X; C'END]

［ 汪 IN X; C PAR C' 式 ] ~ [C PAR ( 邮 IN X; C'END)]

（拉） If X is no 七 扫 E, prove

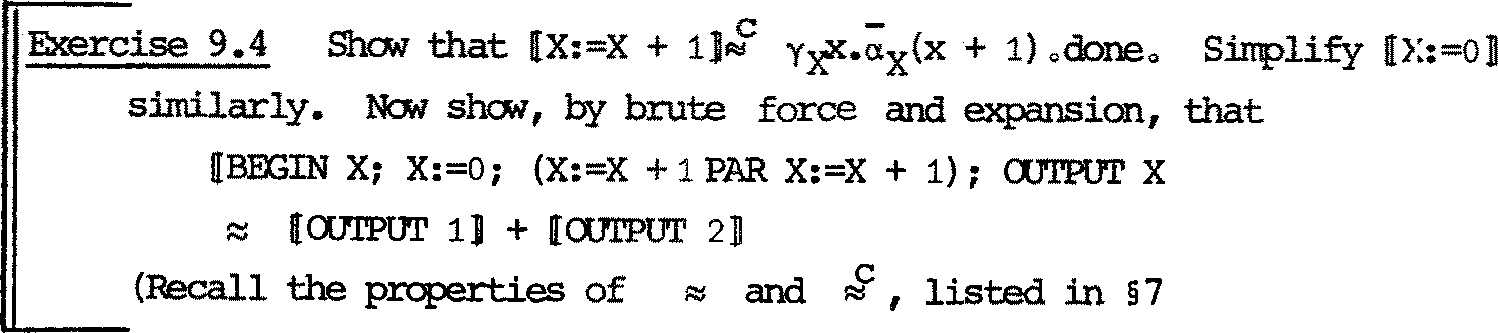
[ B攻江N X; IF E 守正N C 益 E C' END) ~

[IF E 吁邸 （因汜IN X; C END) 益 E (B玫 IN X; C'END)J

and investigate

? [证江tN X; WHILE E 00 C 还ID] ~ [懂IILE E 00 证:GIN X; C ENDJ ?

(v ) 妇 七 can you oonclude fran Exercise 9.1?



END]

.5)

* 1. Trans l a七i on of P For expressions:



＿

[X] = y炉. px . NIL

[F(E1, ... ,E;礼 J = ([E1][p/pJl ••• ![EnB[pJ'p]jbf)\p1... \pn

For carmands:

[X:=E] =『E] r es ul 七 (px . o.} 芒. done ) [C;C'J = [CJ before [C']

[IF E

［网扛立

吁邸 C EI.SE C']=

[EJresul 七 px. (if X then [CJ else [C 勹 I)

E DO C] = w, a 11E 汶 -1 behaviour identifier,

with w 牛 = [EDr esul 七 (px. if x then ([CJ before

w) 生 done)

[BEmN X; C ENDJ =

[C PAR C'] =

(INPUT X D =

[OOI'PUT E]

(ux;xl [C]) \坎

[C] ＿par [C']

tX. 宁 . done

1. result (p x.;x.done)

[SKIP]

= do 迳

沁 王记立S

* 1. 爬 are using \ 奴 七 o abbreviate \ax\Yx'as was done in§8.2.
  2. 吐 e identifier w for th e 邓 I LE carmand mus 七 be dif f eren 七 for every such ccmnand translated. A minor 郅 tensi on to ccs, adding expressions

of the fo 皿

fix b.B

(in which b is a behaviour identifier bound by the prefix "fix") would avoid this inelegance. Such an expression may be understood as

b, where b 牛 = B

where the identifier chosen is distinct fran all others used. ( 吐 e notation can be extended to ma 七 ch the definition of paraneterised behaviour i den 七 if i er s . ) With the "fix" nota 已 on, would write

［ 阳 工 E E 00 C] = fix w. [E] r es ul 七 ( ...) •

* 1. 沁记 年 9 procedures to P

酝 罕 BEGIN X; C END creates a resource X for use by C;

the resource X is represented by a behaviour, accessed through the sort

马·

Procedures (of many different kinds) a 工 e exar 屯 让 es of other resources to create. re 七 us add a ru 戎 s yn 七 怼 cl as s of decl ar a 已 ons D 七 0 OU 工language, with the 呻 r standing that each decla 工 a 已 on D i s 七 0 be accessed through an access sort 母 '!hen we generalise the syntax of blo改 CXIll1旧nds 七o

BEGIN D; C END

and begin the syntactic defini灶on of declara已ons by

D : := VAR X I ...•

The unifo血 扛 ansl a七i on of bl ocl炟 wi ll be

匿 I; IN D; C END] = ([D]I[CD)\1b

and the translation of variable declarations is 江 双

[VAR X] = 立 x (with access sort Ls.J)

Variables are particular in tha 七 the y camrunicate only with their accessors; this is reflected in the fact tha 七 the sort of U:X 议 i s jus 七 坛 Prcx:edures

may, we suppose, contain free variables and call other procedures, so the corresponding behaviours will ha 凭 a sort larger than the caupl emen 七 of the access sort.

Le 七 us define

D ::= VAR X I p 氐 C G (VALUE X, 忘 ULT Y) IS C

G

and add to the syntax of ccmnands

C : := , .•. jCALL G(E, Z)

The prcx:edure declaration indica 七 es tha 七 G is a one-argumen 七 p rocedure ,

taking its argumen 七 (by value) into a loa: 让 vari able X; the body of G

(carmand C) has free variables X and Y and the r esul 七 of the pro-

G

cedure is the value in Y on 立 ipl e 七 i on . The call passes the value of E

as 立 千 年 n 七 ， and assigns the result to variable Z. The access sort of G

i s 七 o be L = {;i ,y } ,

G G G

and we can imnediately write the translation of a

procedure call:



[CALL G (E,Z)] = [E] res ul 七 (px. -acf •Y z.a - z.done)

G Z

We naw have 七 o say tha 七 th e sort of [CJ , when C has free variables

X1, …心c and free procedure identifiers G1,… 舟,' i s

- -

Lx 口 . . uL\ u LG 山 . . uLG 如 ，o, o}.'Ihis will fol归 fr叩 the definition

m

of [DJ for a procedure declaration. (In fac七 sort-checking is a good first guide to correc 七 def ini 已 on, l ik e 七刃:e- checking in good progr 五兀ning languages and dimension-analysis in school mechanics.)

We can give a first approxima 已 on (wrong for a 七 l ea s 七 七 心 re as ons )

to the translation of procedure declarations:

? [ P虹 G(VALUE X, RESULT Y) IS C J = g, where

G

g4= (立 x血 划(a,ef-;. . [ CG] before Y/ •Y (! • NIL) ) \ L沁岭

Notice tha 七 廿让s has sort L uL 一(L奸沪 :S} ) where L is the sort

G C C

of CG·' this will make the sort of the blockr igh七

Are the free variables of C treated properly? Wha 七 out put do

G

炽 expect fran the following ccmnand C。?

BEGIN VAR Z; Z:=3;

B攻江N P虹 G (VAll.JE X, RESULT Y) IS Y:=Z;

还 VAR Z;

CALL G (17, Z); OOTPUT Z

END END

END

'!he answer should be "3", since the body of G should use the outer z. If it used the inner Z the answer would be "no output" since loca 已 ons cannot be used before they are assigned.

尸二声荨二

[CD ：：：：［四 IJI' 3 ]:::: o3.done.

。C

The fi rs 七 mist 咏 e in g above is th a 七 让 i s no 七 much use as a resource, since it dies after one use! Our othe 工 r eso ur ce s (registers) res 七 or e them- selves {with poss 北 l y changed conten 七 ） after use, so we may make g do the

s a晔 ·

Second appr ox扭iati on:

? [ P虹 G (VALUE X, RESULT Y) IS C ] = g, where

G

g 七 （气 尸 为1 空 .;-炉·归 befor e 沪 Ydf •g) ) \ L沁Ly

So the l as 七 thing g does is to restore itself. No 七 i ce tha 七 th e restored

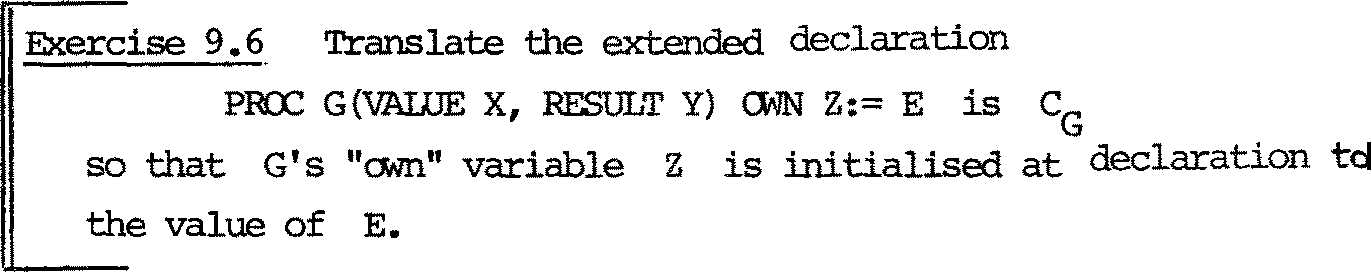
g is of fo皿 (… )\L \L , so its local variables X,Y are —no 七 tho se of

the old g. Bu七 X J

邓 G 七 o have "。双111 variables

you should see hCM to al l<

平 ch are initialized a 七 declar ati on and persist fran call 七 o call.



The second mistak e 扭 g is th a 七 the re is no provision for it to call i七se l f r ecur s i v巳l y. If C con七ams CALL G( 一，一） then i 七 wi ll

G

d 年 四 让 a reply to a v for sane value v, and nothing can 职 涟 七 i 七. \'1ha七

G

coul d 辛 七 i 七? 'Ihe answer is: a fresh resource g for use by C •

G

Taking the clue fran the translation of blocks (which is the way resources

are provided for use), we obtain finally

[PR.ex:: G(VALUE X, 运 ULT Y) IS C ] = g, where

G

g (气 血 引空 今 .(g![CG])\LG before YyY •Ycl •g) ) \ 吩 坅

(with access sort L)

G

i

kc

ehc

,

c }

L -6

tr

s

sa

1

G

cI

fI

9

It is rather hard 欢 K 七o evaluate even s江叩l er eci立 S士ve P progra111S by hand via ccs. Wha 七 'WOUld be the poin七 of eval ua 七扛巧 th己n? Well, 廿汜purpose of our translation is to investigate the power of CCS, and also

g

it

皿

f

ed

s

g，

f。

ed

i1

s

dna h

｀丿

t

｛

hg

¥.m

。

L寸eht

,,j1

h - 。

r

f

Lc

u tr

G OS

L

千g

s

厂

3 eht

es sdl

to indicate tha 七 pr operties of languages such as P (as di s t 江c七 fr an properties of particular P programs) max thereby be established. Bu 七 a check on the validity of the r 七 ans l ati on would be helpful, and oould be provided by a rrechanical CCS simplifier/evaluator. Peter

沁sses has shown !).CM Seo 七七-s r七 a chey sanantic specifi ca已ons expressed

in the lambda-calculus can be checked ou 七 by a lambda simplifier/ evaluator [Mos].

We mus 七 n <:M examine a shortcaning of our 扛 ans l a已on of procedu工e decl ar a已 ons . Since g only restores i tself 生些竺r e turning (y沪

its r es u止，让 fo ll as寸 th a七 al th o ugh there may be con curren七 ca lls of

G within the block of the declaration, for ex 砰 le

CALL G(6,Z) PAR CALL G(7 ,W),

the resu 让 ing executions of C will not be overlapped in 七 江 汜 ； one

G

mus 七 take priority, while the other waits to use the res 七 or ed g. (It

cannot access the inner g provided for recursive calls of G by itself; tha七 i s restricted by \LG.) At first sigh七， we migh七 h ope 七o all<:M for concurren七 acti va 已 ons of G by making g restore itself directly after receiving i 七 s ar gi.m 毗 ：

? [PROC G(VALUE X, RESULT Y) IS C ] = g, where

G

g 今 宁·团（立 刻立 y l(祜.(g![CG])\LG before Yy Y·Ycil·NIL) \ 工'x吁

(Note th a七 we s 七i ll have guarded recursion) . N<:M 七he r es 七or ed g may be acti vated 扭m立 ate l y after the fir s七， and run concurren七l y with it. Bu 七 we canno 七 be sure that the two (or more) g's willr e 七 urn their results 咕 ） 七o the corre c 七 cal l ing sequences - each of which is waiting on

y z.

G

There seems no na七ur al sol u 七i on 七o 七his problem in CCS as i t 兀 双

stands. True, we may generously all<:M s吐 f ix ed number of g's to be created, as separate resources, by the declaration. This could be done by

? [PROC G (VALUE X, 运 ULT Y) IS C ]= gs, where

G

JT

gs{= g.,

1一工 N l.

and for each i

gi 4== CLG,iX 心 知 立 I ( x. (gsI[CGD)\LG before YyY• Y,G i y .gi ) ) \ 坎 屯

with LG = fo G,i'YG,i' 15c L 冬 N} 兀 ·

No已 ce that each gi res七ore s i t se lf 生包竺 carpl et ion; only the N distinct gi can be con curren七l y active.'Ihe calling sequence must also be adjusted:

[CALL G(E,Z)] == [E] re sul 七 ( pX. i:: a .x.y .z.a z.done)

1 :

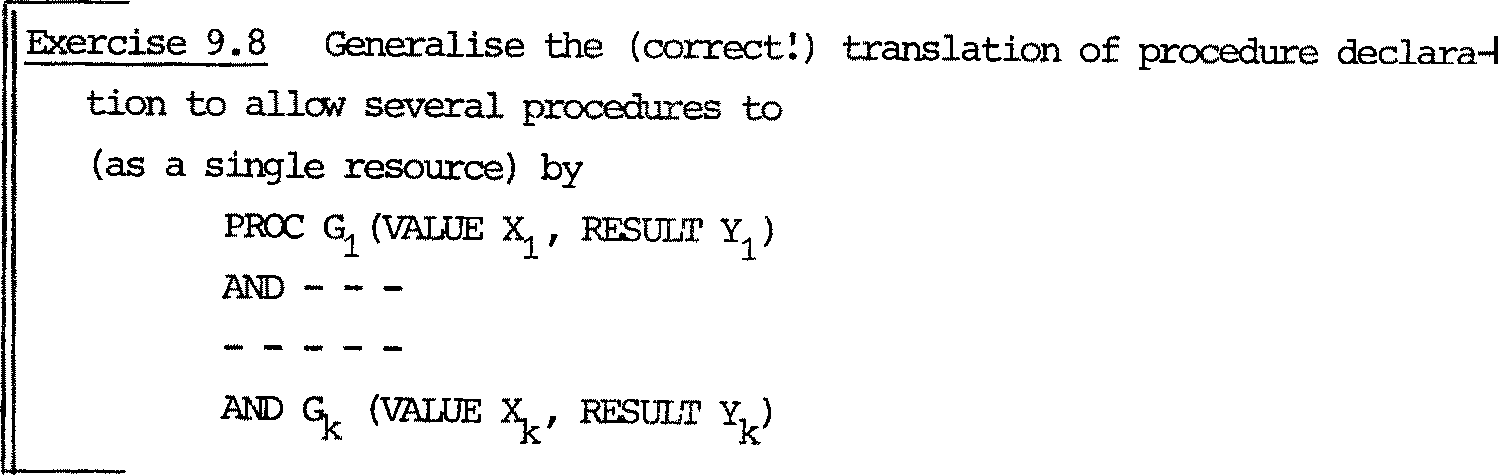
区 N G,1 G,1 Z

'Ihis solution has one a七七工acti on; i 七 may be r eali 玩 i c if we assure a fixed bound N on the number of processors available. Bu 七 we are looking for sol u七i ons a 七 a level of absr七 act i on a七 whi ch :impl erren七a 七i on is no 七 yet considered.

Even so, the 'r i gh 七 ' so l u 已 on is suggested by what impl erren 七 or s often do; tha 七 i s , for each call of G 七 o supply a re 七 urn link along with the argurren 七. Each act i va七i on then kna芯 whi ch l'.'砒 :urn l ink 七o use in r e 七 urn in g its r esu 让 . But in CCS this would mean passing labels (or naires) as values, which we have excluded.

工七 i s no七 七过 vi al 七o gi ve CCS 七hi s abil i 七y , and yet re 七ain the theo工y which we have developed, bu 七 i 七 may be possible (in explora 七 0 巧 di scus- sions wi th 蚐 ens Ni e l sen 炽 have seen sane chances)• The fact that we have no七 rre七 廿让s need un七il nOi/ s ha 最l tha 七 much can be done wi th ou 七

narre-passing, 比 t i ts usefulness is certainly no 七 l imi ted 七 o language translations. We 皿 运 七 l ea ve the ma 七 七 er open.



be 华 cl ared mutually recursively

IS C1

工s

9,5 Protection of resources

We finish this chapter with sane 七 en ta 已 ve remarks abou 七 mut ual exclusion between carrnands in P 中 i ch 沁 uld other:wise run concurren 七 ly . There is no doub 七 tha 七 we can, in CCS,r epr esen 七 sane net.hods for pro- viding mu七ual exclusion, bu 七 to provide methods which are rob us 七 ， :fl exibl e and elegant is a very hard probl 已 of h 扛 扣 - l evel language design which is still no七 f u 耳 y solved though 让: has been s七udi ed for abou七 七en years.

See for example [Hoa 1,21, [Bri lJ. 空 is 皿 prejudiced, and intentionally so, 杠 双 ards the problem; wha 七 i 七 2 翌 do is to provide a means for rigorous­ ly assessing a proposed solution.

If all we wan 七 i s to pr even 七 over l ap 户 玉 l execution of assignment carmands assigning to the same variable, i 七 i s easy to adop 七 th e well­

kncMn semaphore-m-ethod.- 沁- in § 2. 4, define

S即 ：{71,叶牛= 11. 中.s 即

函 X = SEM[飞 炉扣

and redefine

匹 X = (ax 惩 Gx (x ) ) I 函

'Ihe access sort L for resource X be 文 XlES

X

=

L {a,y,ir,q,} X

and the only change in tr ansl a 七 i on is to redefine [X:=E] = ir. [E] r esu l 七 (px . a 尸 . done ) .

Eise 9. ..,..,.,,,k Exercise 9.4 with this n 窃 tr ansl a 已 on, getting

•••• [OOI'PUT 2] instead of .••. i:,:[CXJTPUT l] + 『 OOI'PUT 2] •

An alternative, 七 o all& larger to exclude each other, is to adopt the proposal of Hoare in "T<:Mards a theory of parallel p:r 呼 否ming" (referenced earlier) • 吐 e idea is to all& the progarrrcrer 七 o

declare arbi tr 叩 abstr act resources, by adding a n 钩 decl ar ati on fonn

D::=• ••• 恼这邓立 R

(where R is an arb i tr 叩 i den ti fi er ) and a n 窃 , cc:mnand fonn

C: := • • • • IW 五 H R 00 C

For exarrple, the progranmer may as s oc 远 e a particular R with the outpu 七device, and adopt the di sc 年 l ine that every OOTPUT ccmnand occurs within a 而 四 R • , • " context; he can 出 us protect a sequence of a 冗 Pill' carrnands fran interference. In translation, R is jus 七 a semaphore, so we specify

[RES 叩 沁 E R ]= 亟\ (with access sort 怜 = {nR' 中 R })

and

[WITH R 00 C] = 1TR• [C] before (,PR.done)

Hoare discusses the vir 七 ues and vices of this discipline. In particular, he points out the possiblity of deadly 吽 r ace, or deadlock, as in

血 '1H R DO 叩 吓 R' 00 C) PAR ( 汇 TH R' DO 缸 '1H R DOC')

Bu 七 he observes tha 七 a canpile-time check can preven 七 th i s ; the pI 飞 ,gr an1 must be such tha 七 any nesting of ''WITH R •• 乡 , II a:mnandsI with diS 七 让 1ct R's, mus七 a gree with the decl a工a七i on nes 七止巧" of the R's. For o 立 扛 ans-

lation we mus七 add th a七， in I 飞订：TH R 00 C", C mus七 no 七 con 七a 扛1 ''WI'IH R ••• "

for the 竺竺 R. Also the check mus七 be m立 e s ophi s七i ca ted in presence of procedures, bu 七 can s 七 i ll be done by flow-analysis teclmiques.

归 we can f oi皿 l ys ta 七e deadlo生 fr ee<北兀 for C as follcr., 忘 ：

If [C] =s B is a canpl e七e der i va 已 on (§4.4),

ie B~NIL, then s = r& for sar 巳 r .

(C does not ' di e ' wi th ou 七 s ignall ing canpl e七i on at 句． 如 en the canpile­

time check is satisfied, 江 shoudl

be possible to prove this property of

ccmnands {or a stronger property whi ch 江 li es it) by induction on their structure, though we ha 妇 not done it. Bu 七 f ir s 七 we would have to rarove a s 扭 le source of deadlock - nanely the a 七 tempt to u 延 an unassigned variable.'Ihis canbedone by, for exai 平 le, respecifying

[VAR X] = REG (0) (not LJ 文 ）．

X X

'Ihe proof would be a lot easier withou 七 p rocedures .

CHAPTER 10

Dete nnina 匀 and Confl nce

10.l Discussion

In CCS, non- de tennina 七 e behaviours (in sane sense of determinacy) are the rule rather than the exception. The out e<年 - or even the capabil 让y - of fu七U 虑 observ at i ons may no 七 be predictable, partly because the order of two in 七 er de penden 七 in tem al ccmnunica 七 i ons may affect it, andpartly because of the presence of two or :rrore i den 七 i cal

guards in a sum of guards (e.g. ,.B + ,.B 2

or a.B + a.B) •

Nevertheless, we would probably classify al:rros 七 all our cas e- s 七u 过 es

as de 七e 五诅.nate in 奴工妇 sense; the exo己p已 on is the root-finding algorith!n of Olapter 4, where the r oo 七 f ound depends upon the relative speeds of concurren 七 f i.mct i.on evaluations.

In this chap七er we make precise a no已on of Determinacy, and a related concep 七 Conf l uence , and sh<::M tha 七 a certain easily dlaracterized

subclass of behaviour programs is guaranteed to be determinate.'Ihls class also admi七s a simple proof 七echn i que . I 七 is not a trivial class; for exan:ple, the Scheduling system of Chapter 3 falls within it, and in 且 o. s

we canpl e七e its correctness proof using the special technique.

In this Chapter we shall fo工s impli c让 yr ev巳迂 to pure synchrani za已on; tha 七 i s , no variables or value e 习 ?ress i ons in guards. The results he: 炬probably generalise srcooth l y 七 o full CCS but we have no 七 S 七 udi ed it.

As a fi rs 七 app ro xirna 七 i on , one may think i 七 en o ugh 七 o say tha 七 B is

de tennina七e if, whenever B + B and B + B for s a 汜 入， th en B and B

2 2

are equivalen 七 (e . g . ~ or :.:::) ; of cou 芦 e we would aga 扫 require B and

B2 to be de 均 血 nate . Bu 七 thi s is not enough; for example B 王 NIL may

亟 o hold, implying tha 七 th e cap 咄 吐 li ty of a 入一exper.iroen七 i s not determined - though the OU七文正 i s ! This no已vate s our defini 七i on of ccnfluence. We shall trea 七 no 七 i ons of s r 七 cng confluence and strong detenninacy (so called because they are allied to strong equivalence) in detail fir s 七 - th e y 砬 11

be enough to give us the results we need here - and later we ou 七 l in e a rrore general notion which is allied to observation equivalence.

皿 2 Stron9'confluence

Our notion of strong confluence will no 七 :imply determinacy in the sense of the l as 七 sect i on. We separate i 七 f ran detemdnacy because, by itself, it :implies a property of programs which supports our proof

七 e chnique. Bu 七 de 七 e 立 在 inacy will be needed as well when we shCM that all p:rograrrs wr 迂 ten in a certain derived calculus of ccs are confluent and therefore admi 七 th e technique.

'Ihe follCMing proposition can be read as a def扛让七ion of s t 五X巧

女 mf l uence, excep 七 tha 七 i t ' def ine s ' th e property in tenns of itself:

Proposition 10.1'Ihe behaviour program A is strongly confluent iff

位） Whenever A 归 B and A c th en 兰 迦 汪 µ = v and B~C or there exi 玩 D and E such th.a 七 B D, C 丛 E and D""E •

（拉） Whenever A 丛 B , B is strongly conf l uen 七Proof: 五 m 吽 ate fran the defini 七 i on to folJ.o...r.

因

We may picture oond 让 i on (i) as

一

\)

B

B D

/ }

A

扭 li es eitherµ= v & B~ C or

µ

C

C-'1'E

Such diagrams will be useful in proofs. Note that ifµ= v we have two possibilities; the case B~ C represents in皿 已vely tha七 A 丛 B and A 且 C are essentially the "same action". Our definition of detenninacy 砬11 danand tha 七 thi s 巴巴兰 be the case for µEA, bu七 we

+

do not wan 七 to der.:iand this forµ= T ; A T

+

B and A T C

may arise,

for example, fran two dif f eren七 in 七emal carmunica七i ons .

NON for our fonnal definition. As usual, 识 have 七 o resort to a sequenoe of properties for k 2c o•

芷 f ini ti on A 担 al ways stron9ly 0-oonfluent •

A is stron9ly (k + 1 )- oonfl uen 七 if f

D

¥

B

B

l

位） A 、 implies either ll = \J and B~ C 竺

E

且

C

C

for sore D and E ;

(ii) Aµ+ B implies B strcngly k-conf l uen七

A is StranCJlY CXll'lfluen 七 i ff i 七 i s strongly k-CXll'lfl uen 七 for all k 之 o.

压 t us abbreviate "strongly con王l uent " , "s 七五玉1gly k- ccnfl uen七II by SC, SCk respectively. We f irs 七 want to knrM tha 七 SC is a property of sr 七 ong

equivalence classes, not jus 七 of programs.

P 立 osi ti on 10. 2 工f A is SC and A~A'then A'is SC.

旦 We sh<:M by induction on k tha 七 if A is SC 汝 and A~ A'then A' i s 气 ． 氐 k = 0 there is nothing to pro 伲 . Assume a 七 k , and a 七 k + 1 assume A is SC识+ i and A~A'.

For part (ii) of thedefinition, if A' 比 B' then by Theorem 5. 6

今 B~ B' for scree B ; bu 七 B is 改汶，hence by inductive hypothesis

A µ

so J.S B'.

For part (i), suppose

B' y"'.飞

/ ＼

\_;:.-刁 B

，

A

＼勹

, yielding for sane B,C

c，

A'rv A

'1hen (since A i s 气 +1) eitherµ= V and B~C, so B'~C', or for 改 正 D,E and D',E'

B' D'

V-

飞飞

I

Dl E

I

B

B立今D

µ­

SO

I

I

C

c 二 E

(r'

因

C' E'

Hor.,, 汜 :ver , SC is no 七 pre s erved by ::::

C

＇

．

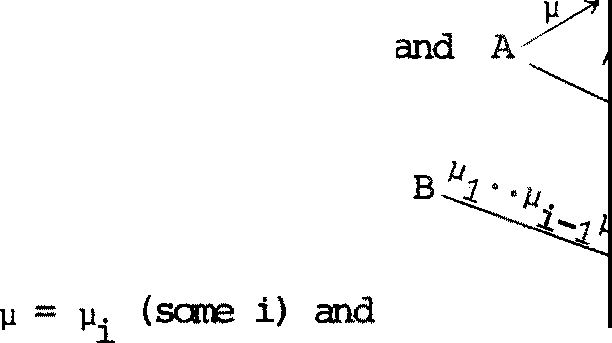
c～~

r。

f or e> 洹 叩 l e

<X. S.NIL + S. a.NIL ::::: <X. S,NIL + S 汀 . a . N 工 L

故 il e the f ir s 七 i s SC, the second is no 七. We 七 忒 企 up this question later.

Forourmain property of SC we f ir s 七 need a lama to do with longer derivaticns.

:r.emre 10. 3

If A is SC

B

, j ..

三

扫..

then

µ.. µ

either

1. 1 n>D

？

C C E

些 By :induction on n • For n=O, C is A and take D,E to be B. A 七 n + 1 -we have

µB

A

µ A 2· 心

广

c

µ 2 •• µ n+1

SO 生 生 主 µ=µ1 and A1~ B , whence B ) D~ C by Theor 细 5 . 6 ,

or (f irs 七 cas e of in duc 已 :ve hypothesis for A)µ=µ. (i:?::2) and

1 1

Ay 气

二

µ B二 二 三

µ1 A C

finding first B ,B'since A is SC , then D'since A1 is SC, then D, 竺 (s econd case of inductive hypothesis)

丑 为· ·.µ n +l

B- 江31 , D

A 5µ2\_°"µl

—三 '

1 A/4... 1 µ

B '

µ 如

1 ，， 、 C ..\_ E

因

Now we can deduce our main property as an :importan 七 s peci al case.

吐eorem 10. 4 (Strong Confluence) • 工f A is SC and A 玉 B then A B.

Proof We show th a七 i f A i s SC and A—-+ B then A ~~kB, by induction on k. Trivial a 七 K 气 ; assume i 七 a 七 k , and a 七 k +:!. assume A is

SC and A B.

(i\_) If B4 B'clearly A 鸟 B' al so .

(ii ) 正 丈

either

s

A= 亨 A' . Then fran Lerrma 10. 3 we have, for 改 兀 它 B'I

s

距今B' or B鸟 B'

f 飞

A' A仁马A"

In th e se 立 d case, since A'is SC (Proposition 10.1), A' A" by

k

inductive hypothesis; bu七 ~ ircplies (Theorem 7.2) so in either

k

case A' kB'as required. 区

The usefulness of the Strong Confluence Theorem is s 扭 ipl y this: a program A may a已 t many ac已 ons , and so may i 七s der i va 已 ve s , bu 七 七o

find a B such tha七 As:::i B we need only follON an £- der i va 已 on (a sequence of ractions) s 七 arting from A, provided we knON A 七 o be SC.

A分 A ．．．．．B．．．一

沁\\\

To follCM all other deriva 七 i ons (as, in effect, the Expansion Theore:n would do when repeatedly appli ed 七 o A, A 口 ...) would of 七 en be heavy work - and is unnece ss 叩 in this case.

In the next section we illustrate this saving an a toy ex 吓 le, which we as s中 to be ro nf l uen 七 (l a 七 er it will be seen 七 o be so an general grounds)• Bu 七 we f irs 七 need to define a class of derived behaviour operations, called cx::mposi 七 e action.

* 1. C<项运 i te 干印 志 ， and the use of confluence

Forµ. EAu"(I (µ

I • • • Iµ> n

迳 a 匀 平 :>s i te 中 邱 (n ;,: 1) whose

J. 1

actions are given as foll a- 态 ， in the style of§5.3 (see Exercise 10.2, end of 扎 o. 4, for richer CXI!lpOSite guards) :

( µ1 1· · · 1 吩. B 乌( µ1 1· · • 1 µ让1 l µi +:1! · · · 心. B

for each i, 1 $i$ n

n= 1 (µ1) . B 乌 B

Fran this i 七 is e 玉 ,,y to deduce the follChling strong equivalences:

ProfOSi 已 on 10.5

(1) (µ) .B~µ.B

1 1

(2) For Il>1, (µ1j ••• jµn).B~ µi' (µ1 1 •· · 1 壮 - 11 让 1 I ••. Iµn) .B 1$l.$Il

* 1. For anypennuta七i on p of {1, ••• ,n}, 归I ••• 心 . B ~(µ的） I• • ·Iµp(n)) .B

Proof 丘 七ted . 囚

For example (aisly).B~a.(sjy).B + s.(a[y).B + y.(ais).B~ (Sjyja).B;

it just rreans that a,S,y canbe done .in any order. Note th a 七 we do no 七

1. 。严 e µ

1

, ••• ,µ n to be di s 七扛lC 七．

In s 咋 pr oof s i 七 is oonveni en七 七o define (µ1j ••• jµn).A to be A, when n=o.

We nCM wan 七 to examine the toy system buil 七 f :ran 七 he cycler of

2

a

官

．

g

g

愉

它

2

夕

、

．、、

. －

巧1

a

s

Exercise 2.7; notice tha 七 c1 is cycling clockwise, while c2 and c3 are cycling anticlockwise. Before going further you rnigh 七 tl:y to guess its beha: 吐 our (as the author did, for five minutes, and got it wrong).

3

We have

c1 七 a.1.s\_.o.c1

飞） '.芍％

c 2 令 a.2.13.y.c2

Y

C3 <= a.3－.y.－o.c 飞 y

3

and

1. is (c1 le 2 Jc)\A. (A ={S,y,o})

3

We as s 哱 s strongly oonfl uen七 悚为, by e习;,ansio n

s a. .s + a. .s + a. .s

~

1 23 2 13 3 12

where s 23 is (f3.o.e le le)\A, s13 is (c 1 1s.y.c le)\A

2 3

1 2 3

and s12 is (e1 !c2 Jy.5.c3)\ A.

By expansion again,

s23 ~ a.2.s 3 + a.3.s 2

｀中er e s 3

is (a•.s.c 和 . e Jc)\A

1 2 3

I

and s2 is (13. o. e1 图1 元5. e3 )\ A,

s13 ~ a. .s + a. .s

1 3 3 1

where s1 is (c1－!S－.y－.e2iY•o.e3)\A, and s ~ a. .S + a. .s •

12 1 2 2 1

妇 we need 七 o consider s。

where s。i s ( a. o. e1 J 和 . c2 国.y . e3 )\ A1

and we find

S。 (o. c1 伈 c2 斤 .J . e3 )\ A

,:

愣

-"" (o.c1 !c2 . e 3)\ A

今 s (t)

't

whence by oonfluence s 。 ;:, s. Also

S ~ a. .s (by 迈中玉 芯ion) ;:, a. .s,

1 1 0 1

s 2 ~ a.2 .s 0 F::J a.2.s

中 il e for S 3

炬 have sanething different:

s3 (a.沪.c2 ic3)\A

~a. .(o.c Iy.c j-y-.o.c)\A by 酝pansi on

3 1 2 3

吐 3. S by the s罕 der i vati on as for S

0

above,

whence by confluence S :::: a. • 趴

3 3

So finally we ge 七

812g {a1 i 吵 . s , s13 图凡） .s ,

and at l as 七

s g 邑 凡Ia3l .s

823g 邑 凡 ）.s

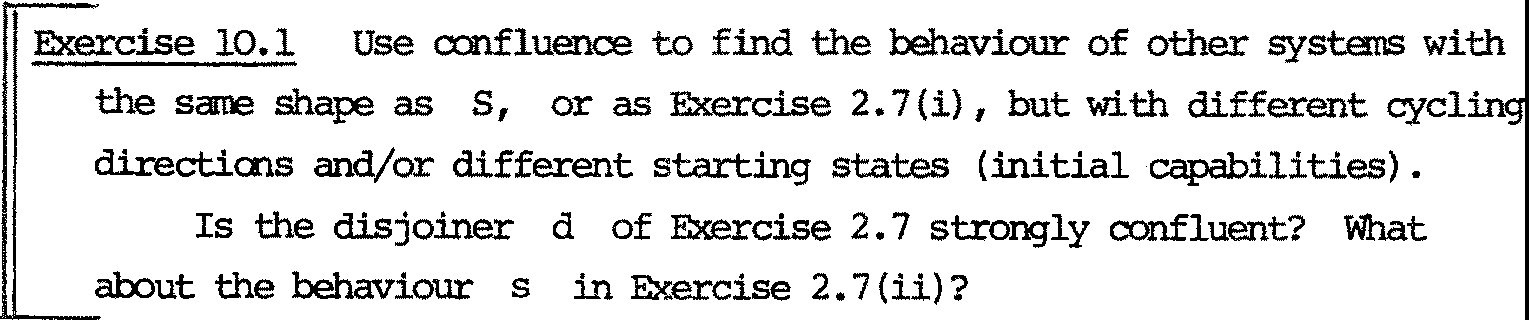
whichs 严 if i es our sys 宅 . It was only a七 (t ) tha 七 we were able to ignore other actions in follor, 血 g a e:-derivation, bu 七 s ud:J. opportuni 七 i es will abound in even slightly bigger systerrs.

Here, we used 立 l!p)Si te actions only to abbr 的 ia te expressions which

we obtained. Later, we will see tha 七 CCI! 贮 ,osi te guarding preser 妇 s oonfluence.

Che final rt=邓 釭 k: in the above cal cul ati 中 we were careful only to as 时 妇strong oonfluence of s, its derivatives, and 呤r ess ions st:I立 19l y equi­ valent to 廿 回n. All this is justified by严 si七i ons 10.1 and 10.2, bu七妇 女 沮 d well have wished to as s 中 巳 cx:,nfl uence of an expres s 扛 :n which is only d:>serva已on eqi.让val en七 to sa 汜 廿让.ng oonfl uen七 As we said earlier, observaticn equivalence does no 七 pres erve strong oonfluence; bu 七 让 does

preserve a weaker fonn as we shall see, and fortunately'Iheora:n 10.4 applies

also to the weaker fo 皿 - so all is well.

* 1. Str on detenni.na中 Conf l uent Detenninate ccs

The na 七 ur a l definition of de 炽 辽m 扛 诅 cy is as fo ll 叩 s :

Definition le 七 入 E A, and 1 过 A be a program. 仅leil A i s s 七ror平Y

入- de 七enninate (入一SD) iff for all k A is strongly 入- k-de tel'.minate

(入- SD ) , where:

k

Every A is 入- SD。:

A i s 入- S D iff

k+i

气：

入 B

位） A

implies B~ c ;

C

也） A B 乓 li es B is 入一SDk .

[ Defmi七1nn A US strongly k- detemrinate I 气 ） iff i 七 is 入一SI:\c for al l 入El\.. A is stroilCj'ly determinate (SD) iff i 七 i s SD for all k

k

入-d e 七e 五吐n acy for partici且ar 入 may have sane use, bu七 we will only

oansider de 七 enn.inacy for 旦县 入．

PrOI; 运 i ti on 10.6

A is SD iff

入 B

位） A

C

乓 li es B~C;

(ii) A- µ

沪

impli es B is SD.

Proof 五media佳 ．

冈

As usual, 炬 have had to make an inductive definition and then prove a 100re usable property. We also have th a 七 SD is a property of strong equivalence classes:

Proposi 七i on 10. 7 If A~A'and A is SD, then so is A'.

互 Anal ogous to Prop::>sitian 10.2 but simpler.

冈

We use the abbreviation SCD ( 气） for "strongly (k- ) oonfl uen 七and strongly (k- )de tenni.na泣 We look for beha: 吐 our opera 已 ans which

preserve sco, and f ir s 七 e l imina 扫 sane which do no 七 ．

Clearly both a.NIL and a.S.NIL are SCD, but a . NIL伈a. NIL ~ a. (a. f3.NIL) + a. (a I f3) .NIL

is no 七 SD, since a.a.NIL + (a 伟 ）.NIL. We shall have to forbid B11B2

except when B1:L1, B2:L2 and L产牙动. Bu七 thi s is not enough; a.NIL and a.f3.NIL are SCD, bu 七

;:;:.NIL!a.B.NIL ~ T.[3.NIL + ;:;:. (a.f3.NIL) + •••••

is not sc,

石

since s.NIL-tB is impossible.'Ihe probl 甸

here is tha 七

the -action of ;.NIL may be observed either by a.a.NIL or externally. In effect (thlnking of pictures) we shall have 七 o pr even 七 th e s ha 工 .ing of ports, i.e. oneport s 屯 port ing two links.

In sumnary, we will forbid B1 I B2, bu 七 al l CM B1I IB2 when B1:L1, B2:L2, L1 n L2=¢; we may call this operati 中 rd- oc 平 运 i ti on (rd= "restricted disjoint").

(Note: we have 叩 S 七 l y avoided the opera 七 i on 11 , and indeed its definition needs sane care. Precisely, it is given by B1j IB2 = (B1IB2)\A where

A= nanes (L(B亡n 丽 ; )）; we can get a different result if we take

A = names (L1 n L2) for arbitrary sorts L1,L2 for which B1:L1 and

B2:L2. Strictly therefore, in each use of 11 we should make explicit the names which are restricted; but in nnst cases these will be implied by the sorts of the argument expressions.)

Also we will forbid B + B (see remark in§10.1) bu七 al 妇 (µ I …I µn ) .B,

1 2

canposite guard.ing, which includes (s 乓 le ) guard.ing as a special case.

We den o 七e by 0CCS the derived calculus whose oper a 七 i ons are:

Inaction(NIL), 釭 ITlfOS 让 e Action, rd-Canposition, Res 红 i ct i on and

距 l abe l l.in9; we nCM show th a 七 eve ry OCCS program is SCD. (Ski p 七 o

且 0 . 5 if you are no 七 扛 1ter es te d .in the proof.)

Pr叩 让ion 10.8 Inacticn, 胆str i ct ion and 距l abe ll in g preserve both the properties 气 and 气 ， for all k.

竺 竺 Clearly NIL is SCI\. le 七 us j us 七 pro ve that if A is 气 I so is A\a; the ranainder are equally simple. For the inductive step on k, suppose A is s 气丑 and

知 Y B\ a µ B

, so A/

C\a. C

Then e i ther 产 '\J and B~ C, whence B\a ~ C\a. also, 竺： for sane D and E, sin 笠 µ 八） 4fo,心，

B - \) 无

l , so also

c.J4E

B\ 矿当D\ o

l

C\o .J!.平 \ a.

(We have of c:吐 se used tha 七 ~ is a cangruenoe, Theorem 5. 4) •

Also if A\已 祖\ a then A 细 ， so B i s 气 ，whence also (by induction)

B\a is 实· 区

For 叩 si te Action we can prov e 叩 r e (which we need in handling recursion later) , namely tha 七 an n-cx 工 1ponen 七 guard raises the level of SC and SD by n:

Pr平 ）sitian 10.9 If A is 气 (re spe ct 坟 el y S片C ) then (µ I … jµn.) .A is

气-tn (respectively SDk如）．

竺 By induction on n, for f 扭过 k. For n =O there is nothing to

prove, since (µj •. -1µn) .A is just A in this case. NCM l e七 A' be

（ 引… I µn+i ) .A, and le 七 US shCM tha 七 A' is 攻飞+n+1 ·

砬 sure

1

了心

f

A

Thenµ,VE{µ , ••• ,µ }. Eitherµ=v, =µ

f

say, and B', C'are both

1 n+1 n+1

(µ I … I µ ) .A up to a pe 皿 uta 七 i on of the guard, whence B'~ C'by

n

1

匠 sit.ion 10.5(3), 竺 严 µn , v=µn+1 say, and then

A',,,J0'B:\_v

C'

<µ 1 1• • - I 壮 - 1) .A

Also, if A' 臣B' then µ =µn+i say, and B1 is (µ 卜.. lµn) ..A which is

1

实 k +n by 血 uctive hypoth es 担 . Hence A' i s 实 • We leave the

廿十

SD part to the reader. 因

Caroll 10.10 If A i s 气 (r es p. s叩 and n 之 1 , then

叮…心 . A i s 气 +1 (resp. SDk+1).

竺 Imnediate , s in ce 气+n 乓 li es 气+i if n 1. 因

Thus far, the operations preserve SC and SD separately. We

can only sha.-1 tha 七 rd -Ca 屯 运 i ti on preserves th 笠 n 七 ogether .

Prop,sition 10.11 If A1 and A2 are 汉气， with A1:L1 ,A2:L2 and L n L =¢, then A II A is SCD •

1 2 1 2 k

竺 Take the inductive step; assume A1,A2 are 实 C\+\_ l and shCM

f 扛 玩 tha t Al II A2 is 改 识十1 . Suppose

A11i¾(:

'!here are essentially four cases:

* 1. B is B1 II , C is Ai II c2 (an

µ

l½\_ action and an

action), and

Ai B1 , yielding

B1II三v

旯ii

4斗c2

C2

生II C2

* 1. B is B II A ,

1 2

1. is C1IIA2

µB1

(two A 1

actions) , and

A 1

C1

Then eitherµ= v and B1 ~ 气 ，whence also Bi II 1½~ C111 , 竺

一

\/

B 1 D 1

J.l 1 ,

C 1 E 1

B1ll A2立 丑 ）11!

l

whence also一

c111 A2上 注 111 A2

也 i ) B is B II B , µ= T ,

1 2

action) , and

A 图

C is C A

1 2

II

'j.

一

(a carmuni ca 已 on and an A

1

1 \)

'--,.

C1

and A2

B2 （ 入 E L1 n L2}

Bu 七 七hen \) ;t 入 , since AiII

Hence

入

A2 \_,,. C is :iropossilile.

\)

一

B 1 D 1

入 ? , whence also

C1--+E1

Bill B 产 D1 11 B2

2

c 1 11 芍二图II B2

(iv) B

c111 c 2, µ= v = T (two ccmnum 幽 cat i ons ) , and

I B

s

i

c

,

1

2 B B

— 入/ "\

B1

s

. 『

A / -, 2

1

A

（入，入 I E L1 介 乌）

* + 1. 入

＼

1

C

C 2

I f 入 ＝ 入I th en 丘 rI also, andsince Al,钓 ar e SDk+l we must have B1 ~c1, 乌~c2, whence also B1II B2~ c1 II c2 •

础 erwi se

一

入＇

一

B 1 D 1 and

入＇

B D

2 2

D

C、

1 1

D

T-

2

B

1

B

2

2

，劝1ence

l

一

入

C 1 E 1

\ 1

c ,E

E

E

T-

2

C

1

C

2 2

Only in the fourth case did we need dete 已 nacy of A1 , A2 •

To 立 p l ete the SC part : if A111 A产 B1 11 B2 then, for i = 1,2

1. is either A. or a µ- or 入气军 i vati ve of A. , hence

1 1 1

II

is 气 and s压， s o SCDk, so by induction B1 B2 is also

SCD •

k

For the SD part i 七 onl y remains to shCM 七 ha 七

B

implies B~C （ 入 E A ) .

C

人了＼

A2

A1

Now either both actions are fr an 气 。r both fran A2 , since

L n L = (our f 扛 S七 use of disjointness} • In the first case

1 2

A 沪 图

1 入

, whence B1 ~ C1

, whence B(i.e. B1 II 气 ）~ C(i.e. C1 !I 沪 ·

C1

s 皿 l ar l y in the second case.

因

工 七 r anains to show tha 七 de f ini 已 on by recursion :in OCCS guarantees tha七 出 e beh a吐 our identifiers a 芘 SCD.

Prop. 10.12 Every behaviour identifier b in OCCS i s 改 for all k.

Proof. By ind uct 沁n on k • By guarded well-definedness, (§5.4），

the definition b 产 恨 may be expanded (by subs ti 七 ut ing 坠 for any

b where necess叩 ） until e画y beh 玑 our identifier is guarded.

Fonnally, we appl y 氐记g ' s lemna to find

bo ~B' b 。

containing no b unguarded. Assuning then that every b is 改气，

。

we deduce tha 七 B' i s SCD fran Props 10.8, 10.11 and Cor 10.10 -

b k+1

the l a 七ter being crucial in raising k to k+ 1 • It follows tha七 b。-

and similarly each other behaviour identifier - is SCDk+1 • 日

酝 er ci se 10.2 We can also allCM guard sequences in canposite guards,

* 1. (a.s) h or even (a.(Slr»lo.'lbese still preserve SCD.

Pr 岱 e the anal 屯 of Prop 10. 9 and Cor 10.10 for canpos 让 e guards defined as foll a知

* + 1. µis acanposite guard

（拉） 工f 91, …, gn are canpos灶七e guards {n 江），so are (g1• • • .gn) and (g1 I•··lgn)•

* 1. Proof in OCCS; the scheduler again

We are in ter es 七 ed in systems definable in OCCS. 吐 e W.f system of 轧 0. 3 is an exanple; each c. there is defined in OCCS, and the

1

system S is also definable in 立 s by

c1II c2II c3

Of course we were able to use the fa 皿 s ~ (e 1 le 2 le)\A since ~

3

preserves sen, and also s ..., a .s + a .s + a .s ; neither of these

1 23 2 13 3 12

are OCCS expressions, bu 七 the faithfulness of ~ to SCD justifies their

use in the proof.

* + 1. e七 us r e 七um to the scheduler problem of§3.1 ; we h 式

I

C 七 位 （引o) .c\_J

郘 d def :m:mg 巨令 c[ 心 如 / a. 阳 J I we ge 七

气~ Yi•气.(飞 Ir 让 1 ) .ci

we also had

Sch 今 (s jc1 j •••• jcn)\r1•••\Yn

and the second part of ou工 s peci f i ca已on dananded

(\*)

Sch II (TI a.w. I JI f3. w) z (气和“

(1)

沪 J j;,e1 J

N<:M - ge 七七让1g 工i d of the s 七art but 七an - we have Sch I':$ 气.(旬沪·望 望 …II en

归 袒 may define, for 2 s; j 红 ，

1 千 气II 寸II s/ I

whence easily

芍～宁.<-r IY扦 1 ). c j

We shall s 切 ，then, that

(\*)

Sch1 ,:,i－a1－.s1.Sch1 (2)

(carpare equation (2) in§3o4, and note the remarks there) where

Sch1<= 1i1• 也因. c1 11 喂 …II 呤

Clearly Sch :::: 壶 l ef t side of equa 七 i on (1) above. 沁 七 i ce that all

1

our defini已ons - in boxes忒:ove - are in OC'CS. Since SCD is a property of ~ equivalence classes, we can use the equivalences (\*) freely.

Sch1 ~ 芞. (( e1 I Y2 } . c1 il c2!1 11 C 以

and

也向） .c1 II c2 II •.• II 咕

3

上 飞 . c1 II r3.c2'J c3JI •.•• II c

＝ 飞. c1 II c2JI c3II ....II c \_1II r1.c (t)

～飞.(c1 II c2 II ....II c'n-1II 飞. c }

while c1 II c2'II .. ·ll c -1II 吁 咕

上 气（ 飞1飞）.c1 II, c2 II• .. II C上 ~ Sch1

Pu 七 七 in g 七 hi s together, using Theorem 10. 4 and known properties of ;..;; , we get S中 1：：：： ll 1• $1, Sch as required.

吐 e crucial part was the long k deriva已on (t) in whi中 the 飞

action could be pers isten 七 l y ignored; without SCD we would have h 改 :l. to

deal with this action by absorption, as we did for the f ir s 七 part of the

s中ed.ule r specification in§3.4. Thus SCD in effect guarantees absorption.

Cne point is worth noting. Fran c'. ~ y .• t •( 五） .c'. we can

J J j+1 J

easily get c'. ""y .c.

J J

J j+1

, and this tr ansfonnatJ.on沁 uld slightly

clarify our proof. Bu 七 we don't kn< 邓 tha 七 SCD is preserved. by "" (in fact we knav i 七 i s 空兰:,in general) • OUr proofs will therefore

be less deli ca 扫 when we have a we 或 er property CXl> whi ch 兰 pres erv ed

by :::: , and which also allCMS a version of'l'heorem 10. 4. We nC"1. tum to

this question.

* 1. Cl:,servation Confluence and Det 竺血坦 匀

I 切 should we arrive a 七 a property OCD, weaker than SCD but supporting our p:roof method {based on'lheorem 10. 4) and preserved by ::::: ? For determinacy, we would probably look a 七

乓 li es B""'C

B

C

/\入

A

as a possibility. Bu 七 the use of - 入

will preven七 p reservati on of

this property by i you will see this if you try diagrams as in

Prop. 10.2. So we 皿 ght try

B

乓 l i es B C

C

入了＼

A

血 s is closer to what we 叽 11 a dop 七 ， b ut notice tha 七 i t already entails

a sort of confluence, for if

B 4-B'then we would have B':::: C also,

whence

B $>! B'(this is because

A 岑B' al so holds).

Sin ce 饶 wan 七 to hanronize with our defini 已 on of "' 炽 do wi sh 七 o

use

= rather than -> ; if we canno 七 se par ate de 七 e 五 ninacy fran confluence

then a definition which covers both seans necessa:ry.

We should also deal

with

=s (s€ A\*) rather than just =入

(入€ A) •

妇 七 s hould confluence say about

r B

A r,s€ A\*

勹·

c

s

工七 should imply sane carmu七a已 vi 七y of ci坛 e rv a已 ons , in so far as

r and

differ; B should admit an abservation which is in sare sense the excess

c

s

of over r , wr让 ten s/r , and

should admi 七

r/s , in such a way

th a七 th e 七心r es ul ts a工e suitably related:

｀卢 叩 li es

c

B=s/=r}D

ll

c垫耘

we shall need to ad jus 七 "i:,," s l i gh 已 y , bu 七 f irs 七 we define r/s. we ge 七 让 by working through r fran lef七 七o right, dele七扛巧· in r

In七ui 七扫汜 l y and in

any symbol whi 中 occurs in (wha 七 r anains of) s .'Ihus r/s is unchanged

s

by a P:江muta 已 on of s , bu 七 de pends upon the order of r .

Definition For r, s e: A\* , r/s, theexcess of r 叩 er s , is given recursively by

e:/s = e:

（入 .r ) / s = 沁 (r / s ) if 入 i s not in s r / (s / 入) otherwise .

Ex郘 ipl es : r r/s 堑

E

c

s y

6

y

ct

6

Y Y

a

6

6

a y y

a a

B

a a

6 6 6

a a a

We lis 七 sane of the properties of "/" wi th ou 七 proof (we write r e兰芒 S

to rnean r is a permuta 七 i on of s):

位） 工f r £ 竺芒! s then r/s = s/r = E: •

* + 1. 工f s £ 竺芒! s' then r/s = r/s',

s/r r,enn s'/r .

* + 1. 工f r and s have no rn巴吐X 江 in C<艾 汀ron then r/s =r , s/r = s .

岱） If r 兰亚 ss' , then r/s E色丑u s' and s/r = E: •

* + - 1. r. (s /r ) 庄 竺 s . (r/s).

= =

* + - 1. r/s s (r/s) /s , r r /s (r /s) • (r / (s/r)) .

12 1 2 12 2 1

'lhere are many others, sane needed in proving the propositions bel0iv, but we will no 七 gi ve those proofs here.

豁 正 define OCD by a sequence {气； k O}:

Definition. A is always OCD 。 .

A is 气 丑 i f f

B

位） A了

c

B 已 D

implies l\k for s玉 D, E ;

c 辛 E

（过）

r

A = 今 B

implies B is OCD

k

A is CO) iff i 七 i s 沈气 for all k;:: o.

Note the use of 气 r ather than ;::; ; 如 s is essential in sh 叩 ing that preserves 心 ．

Thus if A i s 孜 D we have for each k, for exanple:

、

E

B 二 =- D

A / **r** B C

、

A a8/ 8a B

C

扭 li es

implies

c 今

£ B=D

C=€

s

Uk (de 七e rminac:y)

R k

E

今 E

夕 B B = 宁 D

A 扭 li es

c

议K

C =E=: >E

s

A / r rs B

、

C

implies

B=D

C=£

lkl

E

The folla,,ing results hold:

Propos i 七i on 10 .13 If A is a:::D and A 畦 ' th en A'is a:::D. 因

[ Exercise 10.3. Pro,e this by showing tha 七 i f A is <=. and A•21cA'

then A'is a:::D •

k

e:

'Iheorern 10.14 (Confl uence) 工f A is a:::D and A =

B 七 h 匀1 A B.

这 We s 正 让 f or 气 -c by induction an k • For the inductive step,

assurre A is OCD and A =E B •

's

位） If B = B', then clearly A B'also.

s

（拉） If A A'then, because A is OCD,

s

B=B'

l! for sane B', C

k

E

A' = 午 C

Bu七 A' i s a江），so by induction A'~ C , whence A1 ::::: B'as required. 冈

飞 k

罕 os i t i on 10.15 If A is SCD then i 七 i s a 立 囡

Fran this we 扛 阳 dia te l y kn 叩 出 at OCCS, and anything ::::; 七 o a DCCS program, is cm 。 Alth ough these facts do no 七 impl y 让 :i.nmediate l y , we also h 忒 e

P 叩 osi ti on 10.16 The oper a辽ons of DCCS all preserve the prop江切 O工）． 因

沁 remarks should be made. Fi rs 七 ， we do not k:nC1il of any derived calculus of ccs whose programs are all OCD but 正 王 al l SCD. 工七 woul d be ve 巧 , interesting to find one, particularly if it contained systems

平 ch are i n七ui ti ve l y de 七e 五吐 卫 ate in s a 在 s ens e , like earlier case-s 七 udi es

in these notes, bu 七 canno 七 be expressed in OCCS. Fir s 七 of course we would wan 七 七 o extend the presen 七 noti ons , and OC'CS, to al1C1il va l ue- pas s in g 。

Second, the reader may wonder why we introduced SCD at all, since

心 D has the pro i:erty 平 中 we used in proofs 竺 世 pres erv es ; 心

has the advantage tha 七 i 七 i s a property of behaviours ( s cong:r.,.:Ence classes) ,

~

not only of :p五习r ams. 吐 e reason is partly technical; the crucial property of SCD (Cor 10.10), 如 ich provided for recursively defined behaviours in occs, canno七 be es 七忒:ili sh ed for OCD. Also of cou 工 s e the s 扛 an ger no 已 on may yield stronger methods.

In conclusion: we have fotmd a derived calculus of CCS 中 i ch r:ossesses an in teres 七 in g proi:erty, and it i s 贮 ssible tha 七 oth er de 过 v 忒 ca l cu li may be found with useful properties. For confluence and detenninacy, there is a strong connection - s 七 il l to be explored - with notions in Petri's Ne 七'Iheory, particularly the no 已 ons of (absence of) Conflict and Confusion

and the subclass of nets called Marked Gra: 产 [ CoHJ. Other authors have explored confluence in various set 七 ing s . 'Ihe a 豆 gin of the i 圭 a appears

to be the Chur ch 玉os ser theorem for th e 入- cal cul us ; Church-Rosser properties are 沮 scussed by Rosen [Ros]• Hue 七 [Hue ] studied oonditions under which

te mr rewri 七 in g systems are conf l uen 七 ； the principal difference here is that

ourr e江i t ing rela七i ons

µs

-> and =;:, are indexed by labels and sequences.

Keller [Kel] introduces a confluence notion in七o parallel oor平）uta 已 on 干

his rewr迂 ingr el a七io ns are indexed, but his defini 已 on of confluence does no 七 exp l o 让 th e indexing.

吐 e author's impression is tha 七 confl uen ce is a deep notion which (as with mos 七 de ep no 已 ons ) manifests itself very di ff er en 七 l y in di f f er en 七fonnal or math ema七i cal settingso we have no七 i nven te d i 七， bu 七 onl y found

让：s a 飞:! n已ll cl o七hes 。

CHA玉 11

Conclusion

* 1. 1 What has been achieved?

We hope to have sh吵 th a七 our calculus is based on few and simple 迈eas , tha七 让 a ll 吩 s us to describe lSUCcinctly and 七0 maltl土p ula七e a wide variety of 叩 砒 ing agents, tha 七 迂 of fer s rich and various proof

七e chniques, tha七 让 under li es and explains sa飞" con curren七 p rogranming concepts, and th a 七 迂 all ows the precise formula 已 on of ques 已 ons which remain to be answered (e.g. which equivalence relation to errploy) • It also appears 七o have sa:汜 in tr ins i c mathematical in七e re s 七. Thus we

cl 血 t o have achieved, to sorre exten 七 ， th e aims of articulacy and concep 七 ual unity expressed in Chapter 0.

In the next f窃, sec已ons we examine CX:S cr i 七i ca ll y (though briefly) in one or two respects; in doing so s咋 suggesti ons for further work

arise ve 巧 ,clearly. In the final section we propose sane other dir ec 已 ons for the fu 七 ure .

* 1. Is CCS a p哼 arrming l an平沮ge?

工七 is no七 uni vers a ll y agreed wha七 quali fi ca七i ons jus 七 i fy 七 he 七 i tl e "prograrmd.ng language". I.e七 US 七巧, to examine CCS cr i 七i cal l y with respect to s呻 poss ib le qualifications.

First, we have no 七 s ai d how to impl erren 七 让 on a canputer (with one or many processors). Implerrentation of concurrent programs raises a hos 七of di f fi cul 七 quest ions. 'lb s 七 a 迂 wi th , such a program is often (a 七 l eas 七in our case) non-de 七 enninate ; should i ts ' impl errenta 已 on ' be able to

foll 叩 any possible execution, by having the 区 邓 江 to toss a coin fram

tine to 竺 or by using a machine whose parts nm a 七 unpredi ctab l e relative speeds? Or is i 七 nore correct to 七 a 止 of , not a single impleman­ talion, but a 至 of impl 妇 ta ti ons for each program, each implerrentation being detenninate?

Again, would one allCM an impl arenta ti on 平 ch is, if no 七 sequen 七 i a l , conducted under sane centralised control?'!his would be rather unsatisfactory, since the calculus is designed to express he七er ar chy 租rong concurren已y

active cc 平 nen ts . Bu 七 S 让 兀 汜 扫 ；can express systans whi 中 generate unboundedly many s uch 叩 一 l1E!I1ts , 迁 i s natural to e 习 定 ct an :implemen­ tation to 五 加 血 ster (not necessarily in a hierar 中 i c ma:nn 红 ） the allocation of a fixed nUll匹 of processors in executing the a::xrp::ments.

如 扭 p l errenta tion prabl 甸 arises, even with CCS pi 一 :呼 季 wi th a fixed nur妇 of concurrent o 卫 贮 0 11eI1ts , and even if there are enough p.roaessors to go round. In the general case where the cauponents are

呻 i tr ar i l y linked and where ea 中 one may have at each n:anen 七 an arbitrary se 七 of o::mnunica 七 i on capabilities, our primitive notion of syndlronised camrunication does no七 a drni七 dir ect re ali s a已 on by ha 仄 为花江 e (at l eas 七 by curren 七 te chniques) as fa 工 as the author knows. Jen:y Schwa 立 [Sch ] has exposed the difficulty andproposed a sol u 已 on, 故 i ch can indeed 坎

s 扭 p le in special cases but is no 七 so in general. So CCS does not (ye 七 ） have the property that its primitives have pr 江 吐 七 i ve re al is a 已 ons . We claim rather to have found a camrunication primitive which all 竺 othe r disciplines of cx:mnu 过 cati on (e.g. by shared variables, or by bounded or unbounded buffers) to be defined, 玉 平 中 can be handled rnath ema 七 i cal l y . 卫 1ere is no a priori reason that anysuch pr 扛 吐 七 i ve should also be

S乓le to realise. Bu七 we mayo:兀pa迳 the pr扛吐已 ves of th e 入- ca l cul us

(functional abstraction and a汗）lication), or of carbinatory logic (the

匀 lbina to rs an d 立 tbina ti on ) ; ten years ago these may have 妇 平 th ough 七

to require very indirectr ealis a已on, ev曰1 via sofb 花江e , bu七 th ey are nCM being realised directly by hardware.

Le 七 us look a 七 ano ther quali fi ca 已 on usually expected of a practical progranming language. It should no七 onl y have a pc邓 江 f ul and no七 too

red unden 七 s e 七 of cans七ru cts , bu 七 s houl d also enoou豆ge disciplined and lucid progranrn:i.ng.'lhis can mean th a 七 i ts cans 七 ru cts a 迳 ooncep 七 ual l y rather non-pr扭让ti ve ; cons 迈er the sophisticated array manipulations of皿 L 68, or - closer to ooncurrency - the non.it.ors of Hoare. On the other hand a calculus, as distinct fran a progranming language, should oontain only a small se七 of ooncep 七ua ll y pr 扛吐七i ve constructs (i 七 wi ll

be hard to theorize abou七 让 o therwis e) , and s houl d 五五巴江 l a立尹坊ilrpartial with respect to design decis ions 如远 aim a七 I gcodI program­ ming.'Ihen the calculus can serve as a basis for defining practical languages, or for building practical hardware 吐 f i gur ati ons . Of course one canno七 dis t 扛巧U拉出 sh arp l y be切 een the a.ins of car瓦e ptual pars.inony and practical utility, bu七 i 七 i s fairly certain tha七 a language for

writing good large pray 止 血 s wi ll itself be too large to smve as a theoretical tool, and its design may well be rrotivated by current 罕 ementa ti on techniques; when these change i 七 can grow obsolete.

胆七un 已ng to th e 入- ca l cul us as a pri.rre ex五环让e , 士七 i s ncm widely accepted as a iredi.um which can be used to define and discuss sequential

algori 廿 皿 s, and richer languages for them. Althouqh ccs is not as small and

s扭 le, 让: is int.ended as a s七ep 七劝花江d，s such a iredi.um for a 文1curren七 systems. We also hope to have shown tha 七 at l e 坴 七 sa 飞 ccncurren 七systems can be expressed lucidly in CCS; pe 吐 aps this is because i 七 is no 七 ye 七 small enough!

11. 3 '!he 严 七 io n of fairness

In tems of CCS we may state a property, 血 ch is arguably a property of real syste:rs and should therefore be reflected in a rrodel: if an agen 七 pers i s ten 已 y of :15 红 s an experimen 七 ， and if an observer persisten已y a七ta:np七s i t , then i七 wi ll f!'J'en 七ual l y succeed. A m:x':lel

血 ch reflects this property is sorretimas cal l ed 些 兰 . Is CCS fair?

Consider the program

B = rwlA•NIL, 中 er e rw may be defined by b r .b, '!he only actions of B are

B rwI NIL and B B ,

So B has no £- der i va ti ve 如 .dl does not offer a 入哱 r iment ; 廿lis

may plausibly be taken to 晖 an tha七 B pers坛七en 已 y offers the rug王江扛飞汀七．

归 if we consider only the derivations of B, the in f ini te 知 i va-

w

tion B 土 s uggests tha 七 th e experimen 七 i s no 七 bo und 七 o succeed even if

a七tEmpted by an observer; hence we may 中 oose to infer that ccs is not fair.

On the other hand if we consider observation equivalence, we can easily deduce

B l>i 入. NIL

and we argued in Chapter 1 tha 七 if an agen 七 of f ers an exper:imen 七 and

has no al ternati 凭 action - as here 入 . NIL has no al terna 七 i ve to its

offer of an 入 imen七 - then an observi江 ' s a七七王pt a七 th e exp 立 扛正江比

is bound to succeed. 工 七 七 her efo 年 s ean.s tha 七 the ins ens i 已 vi 七 y of l':i to infinite unobservable action makes CCS fair, 砒 leas七 for this one example.'!his is slightly strengthened by noticing tha 七 the agents

B = 入 . NIL + Tw ,

1

B = 入 . NIL + -r( 入 . NIL + Tw

2

)

, •••

which do 竺 per s is ten tl y offer a 入戈 笃泣 血 平 七， ar e 还主: equivalen七 七o B

(though all equivalen 七 to each other).

Indeed, 怅 may tentatively font邑ise "Bp 江s i s ten 已y of f ers 入“

for 扛 b i trar.y B as follCMS:

Definition 主竺旦 if f B 辛 丑 ＇ 乓 l i es 3 B" . B' 兰咋 " .

吐 en it is easy to p 五 为 e tha 七

旺 C 乓 li es 江 (B mus 七 入 牛 ==} C must 入 ）

却 in g tha 七， under 廿让s definition, observation equivalence resp:汒运the persistence or non-persistence of offers.

邸 this is very fa工 f ran a denons tr a已on th a七 CCS is fair; for exarrple, there are alternatives to the above definition, anda much

立 e detailed investi gation s笠歪 necessary to decide which is correct. Even i f 炬 coul d conclude tha 七 CCS is fair, with the present notion of observation equivalence, the fact remains tha 七 o ther equivalences (see the remarks in 江 2) which res 忱 t epresence of infinite unobservable action - and are th ere for e 旦生虫：in vi 窃 of the above discussion - may

have other factors in their favour. We mus 七 l eave the question open.

Other authors have focussed nore directly on the fairness issue. 芘 ueli [Pnu 1, 2], for exarrple, sh竺 h<:M "eventually" (closely allied 七 o fairness, as seen fran the first paragraph of this section) can be represented in a 坛平r al logic. I七 woul d be in ter es 七扛飞r to OJr 心ine

SU中 a tre a江en 七 wi th our algebraic m已thods .

11. 4 The notion of behaviour

'Ibis work has been conce:rned thro 屯 OU 七 沁 th expressing behaviour.

殴 have tried no 七 to prejudge wha 七 a behaviour is, bu 七 ra ther regard i 七

as a congruence by considering which expressions can be distinguished by

observation. A七 f irs 七 we h匀汜d this appro中a w巩让d lead us to cne obviously

best oongruence relation, anden 已 已 e us to say tha 七 - within our chosen 吵 of expression - we have defined behaviour.'Ihis has not transpired; the discussion in§7.2 shows th a 七 th e 迳 i s s 七i ll latitude for choice in the definition of observation equivalence, and s玉 (th ough no 七 al l ) of the choioes induce di fferen 七 oongruenoes .

However, we have p立 对i ded a se七tin g in which the l a七迁 ude for choice is no 七 embarr ass ing l y gre a 七 ， and in which the consequences of each choice

can be examined. It is no 七 扛 叩 robabl e that a bes 七 choi ce will thus arerge. Furthenrore, although the calculus itself cannot claim to be canonical since alternatives exi玩 for the basic opera七i ons and 廿1eir der i vati onal 芷 aning, the s畔 approach to behaviour can be taken for many alternatives.

Our 哇 th ods should be contr as 七ed with wha 七 has often been done in providing a denota 七i an al s anan七i cs for prograrmrl.ng languages, follCMing the work of Seo 七 t and S r 七 a chey [SSJ .'!he rethod - a very fruitful one- is to define outrigh 七 one or several seman 已 C d:!nains, built fran s江屯）le danains by such st 玉1dard neans as Cartesian product, function space construction and (for nondeterminism) a 四 erd anain construction [Plo 1, 岛 J 1 then the semantic interpretation of phrases in these danains is specified by induction on phrase structure. The approach has given .irmense ins i gh七， and ye 七 i 七 was found tha 七 th e ma七ch between denotational and opera七io na l meaning was sat1:丈扛芷扫 扛屯王五 ect ;

this misma七中 was fi rs 七 exp:,sed by Plotkin for a 七刃文艾i 入一cal cul us [Plo 2]. We found a misrna 七 ch again for the rrodel of concur.rent processes presented in [妞 J• There is no re ason 七 o expect, a prior 斗 th a 七 an exp li ci 七 l y presented denota ti onal 呻 1 will ma七中 the opera七i an a l neaning; the

la 七七er should serve as a criterion for the correct deno七a已 onal rrodel, no 七 vi ce versa {see also§0.4). Of cou 运 e , 士 七 would be s a 七 i s fy ing to find an expl ic 扛 presenta 七 i on of a rrodel which does neet the criterion; this may entail extending our 年 pertoire of danains and 虹血 ronstru c­ tions, as found in [ HP lJ where so-called nondetenninistic danains and a

tensor produc 七 i s used.

We can surn:narise our approach, then, as an a 七 tempt to calculate with behaviours withou七 knowing wha 七 th ey are exp li ci 七 l y ; the calculations are justified by opera 七 i an al m 巳 min g, and may help tc:Mards a be 七 ter imderstanding - even an expli c让 fonnula已on 一 of a danain of behaviours.

11.5 Directicms for further 沁 r k

(i) In Chapter 9 we explained a s 乓 le 坦 gh 一 l eve l language in terms of

ccs. 工t will be interes七年 g to see how far such languages can be so explained, and how CCS may help in their design. For exarrple, 年 tha 七 chapter we exposed an apparen七 de fi ci en cy of the calculus, 如 ch could be rerrov if we all 叩 ed. labels to be passed as values

in ccmnunication. Wha 七 e ff e ct would such an extension have on our th eo巧r? And is the extens 沁 n really necess 扛 y , or can we find a way of simulating label-passing with CCS as i 七 s tan ds ? (An analogy is tha七 the 入- cal cul us does not take the no杜ons of 芷叩立y and

assignrren七 as primitive, but can simulate th已正）

(ii} Although hardware devices can be described abstractly as in§8.2, 让 is no七 cl ear 比为1 to extend the calculus to deal with detailed 七扛泊卫g consi dera已ons, or to bring 迁 in 切 ha:口江n y with existing description mathods which deal with timing. We have sore grounds for hope here; for exanple, Luca Cardelli [Car] has recently oon-

structed an algebra of analog processes (whose ccmnunica 已 on signals are time functions} and has sha,n1 i 七 妇 be a FlCM Algebra [Mil 2]

tha 七 i s , 让 s a 七 i s fi es the laws presented in'Iheorem 5. 5. HCMever, FlCM Algebra deals only with our static operations (Cc平 s i ti an , Restriction, Relabelling) and i 七 i s the dynamic operatioI 芷 (Action, S 中 ma 七 i on } which are rrore cx::mni 七 ted to the idea of discreteness and synchronisation in ccmnunication. I am no 七 ccmpeten 七 to judge

wheth er 让 i s desirable, fra:n the engineering pain 七 of vi 窃 ， to build

hardwar e 立 江l[X)nen 七s whi 中 r eal i ze these dynamic op:江ati ons . An al ternati 诧 may be to 切 to find a continuous version of CCS, bu 七hCM to do it is unclear.

(iii) In Chapters 9 and 10 we wer e 扣 l e to find two interesting derived calculi. In particular OCCS, detenninate CCS, has certain s 乓 le properties which facilitate proof. (Since Chapter 10 was written, Michael Sanderson has with l i 七七l e difficulty extended OCCS to a 旦 o value-passing.) 工 七 is 扛 屯 为 rtan 七 to is ol a 七 e other subclasses of behaviour, characterised by int ui七i ve l y s 扛匹让e properties, and to

find for any su 中 subclass a derived calculus which can express only i ts 正 叫 妇 . Of particular interes 七 ， for exanple, 沁 uld be a

calculus of dea 和 心 free beha: 过 ours . Again, 让 woul d be illuminating 七 0 f ind 如 七 certain kncMn nod.els correspcnd to derived calculi; possible cases are Kahn;MacQueen networks of processes [ 叩 J, and

the DataF l c 双 呻 1 of 氐 血 s e 七 al [DFL].

（坟） As far as proof n:ethods for CCS are cancemed, 殴 appear only to have made a beginning. On the theoretical side, we should look for ccrrrplete axianatizatians for subcalculi, where these are

possilile; the results in [昭J and [HP 2] go s en巳 way 扛双ards this.

Ch th e 印 re practical side, a:npleteness (whi 中 may not be p:>SSible for the full calculus anyway) is less jnportant than a repertoire 区 王 ful and manageable techniques. In our examples we have found a fw useful tedmiques; in particular we found i 七 useful to 沁 rk not just with congruence (::;:;c) but with equivalence (l':I) also, and this 扭炬立 ately suggests tha七 o ther predicates of beha吐 our may be used with advantage. Further, we often wish to shCJil tha 七 an agen 七

孚 ts an ina:nplete specification, i.e. one wh 扛 h does not detennine a unique behaviour; this was illustrated by the examples of Chapters 3 and 8. In these examples the in a:nple 七 e s pec i fi ca 已 on could be expressed within the terms of CCS, and we 沁 ul d like to discover

how far this is possible in general, and whe 廿 汜 工 一 泣 1en possible -

让 is na 七 ur al .

M 如 re particularly, concerning proof techniques, the question of recursive defini 已ons and induction principles needs furthe工 S七udy . For our definition of observation equivalence and oongruence we are able to identify a class of recursive defini 七 i cns which p:>Ssess

unique sol u 七 io ns (up to ::;:;。 r

C ) ; see Exercise 7.7 • We believe

this class can be oonsiderably widened. 工七 was this uniqueness 泣让中 al l 竺 d us to do certain p立沦fs , e.g. the s中eduler proof in Chapter 3, withou七 a ppe汜 l ing to 玉 w induction principle. Bu七

as we ranarked a 七 th e end of§7.5, we believe tha 七 th e Ccraputa 七 ion Induction principle of Seo 七 七 wi ll apply in the presence of a finer version of observation equivalence.'Ihe strength of this principle is that 让 沁 r ks wi thou 七 as s uming uni 平 正 sol uti ons of recursive definitions; 迂 allCMS us to deduce properties of 旦七 sol uti ons with respect to a partial ordering of behavio u 运 . Bu七 i 七 :rer庄让适

to be s een 比 田 扭 ipOrtan 七 th e principle will be in practice; rroreover,

since the finer observa七i on equivalence a穿 ars to be 画 air (in the sense of §11.3) there is a delicate and diffi cul 七 prob l em in re l a 七 江 g P 如 f th eory 七 o the conceptual correctness of the m:x 迨 1.

we are no 七 di s cour aged by the em 江 gence of this problem. en the contrary, we believe i 七 to be intrinsic to concurren 七 0 元 U 已 ng , no 七平 e l y a defect of our approach, and are rather pleased to see i 七

釭 芦 ge in a sharp fo 五 n.

(vi) Finally, and fundamentally, h 氓 ver successful we may 运 in

沁 立 :in g within ccs, its pr 运 ti ve constructs deserve re-examination.

Are th 窃 the small es 七 pos s ib l e se 七? Are other constructs needed

to express ar i 中 er class of behaviours? Ha,, can we relate Petri

Ne七 'Iheory to the ideas of observa七i on and syndu 文>nized ccmnunica 七i on? By repeatedlyr e 七 um.ing to such basic questions we may hope to ge 七closer to an underlying theory for distributed ccnputation.

APPEND工X

Properties of CO!l<]Yl王叩 .ce and equivalence

Direct equivalence Strong oongruence

～~C～~

心Serva七i on equivalence

岱 erva已on congruence

C

••• §5.6

••• §5.7

••• §7.2

••• §7.3

B 三 C implies B ~ C 扭 li es

B s,:i C 乓 li es B;:;C ••• Ex 5.2, Cor7.6

Observation congruence "r:,ic.. is also 华 noted by equality

"=", though many laws {as their narres indicate) hold for strong oongruence "~" or even direct equivalence " 三 fl .

Except where indicated, the 1 玉 妇 are those of'Iheore: 芯

5.3 and 5.5 generalised by'Iheoi:; 甸 5. 7 .

Sl:m芘i七i on

三 (1)

(2)

(3)

B +B =B +B

1 2 2 1

B + (B + B) = {B + B)+ B

1 2 3 1 2 3

B + N 工 L = B B+B=B

Action

竺 Cl兑 B = a.y . B{茄｝

where y is a vector of 沮 stin ct variables

no 七 in B •

Coo!J?Osition

竺 I.e 七 B and C be sums of guards.

'Ihen

BIC = }:{g. 但' I C) ; g.B'a sumnand of B}

+ l{g. (B!C'}; g.C'a s,mmand of C}

+ I伈 (B' { 踌 } IC'); 战. B' a surrmand of

Band a. E" .C'a SU!lll印 d of C}

+ I伈 (B' I C' {～E/一x }，) ; aE.B a surrmand of Band a 记 c• a surrmand of C}

provided th a 七 in the f ir s 七 (second) Sllll 兀 曰 nd

no free variable of C(B) is bound by g.

立 (1)

(2)

(3)

B 1B = B IB 1 2 2 1

B (BIB)= (BIB)IB

1 2 3 1 2 3

Bl 皿 = B

Res 七过 cti an

Res =＿

(1)

(2)

(3)

皿 \ S = N 耳

(B + B)\(3 = B \S + B \(3

1 2 1 2

(g.B)\S = {NJl., 迁 s=n哱 (g )

g. (B\ S) otherwise

三 (1)

B\a = B

(B:L, at names (L))

(2)

(3)

B\a\6 = B\6\a (BIB)\a= B \alB \a

1 2 1 2

(B :L ,B :L , 叶 names

1 1 2 2

＿

(L nL))

1 2

腔 l abe l lin CJ

竺旦 (1 )

(2)

(3)

竺 兰 (1)

(2)

(3)

(4)

皿 [ SJ = NIL

(B + B) [SJ = B [SJ + B [SJ

1 2 1 2

(g.B)[SJ = S(g). (B[SJ)

B[IJ = B (I:L + L the identity mapping) B[SJ = B[S勹 (B: L and S「 L = S' L) B[SJ[S'J = B[S'.,SJ

B[S]\B = B\a[SJ (B = naI18 (S (a)))

(B 1B)[SJ= B [SJIB [SJ

1 2 1 2

Identifier

竺 I.e 七 b (x ) 月i then

碑）＝却耽｝

C.Onditional

1. On 三 (1 )

(2)

true then B else B = B

1 2 1

if false then B else B = B

一

1 2 2

Unobservable action T

`

(1)

(2)

(3)

(4)

g.-r.B = g.B

B + ,.B = ,.B

g. (B + ,.C)+ g.C = g. (B + ,.C} B + T. (B + C) = T. (B + C)

* • • Theorem 7.13

••• Car. 7.14

b

t

Observa七i on 已下让 :valen ce

* 1. B 釭 . B … Prop . 7.1
  2. is preserved by all operations excep 七 十 .• .'lheorem 7. 3
  3. B:::: C 扭贮li es B = C when B,C stable ••• Prop. 7.11
  4. B:::: C 扭p li e s g.B = g.C ••• Prop. 7.12

Expansion

运 B = (B1 I ••• IBm)\A, where each

B. is a sum of guards.'lhen

l.

B = 肛g . ((B1 I ••• IB1 I ••• !Bm)\A);

气'. a surrroand of Bi, narre (g ) 志 A}

+ }:h.((B1 I …!Bi ' { 耽 } l ••• l B.' I …J Bm) \ A) ;

忐. B.' a s叩 却 :l of B., 芷. B.'a sumnand

l. l. J

ofB.,i:j=j}

J

provided tha 七 in the f irs 七 tenn no free va 工 i ab le

in (k =l= i) is bound by g.

* • • Theorem 5. 8

References

(I n 七h 去天 r ef er ences , I.NCSn stands for le e七U工e No七es in Computer Science, Vol n, Springer Verlag.)

阳r il J P. Brinch Hansen, 严 at in g Systems Principles, Prentice Hall, 1973.

[Bri2 J P. Brinch Hansen, "Distributed processes; a concurren 七 pr ogr amning concept", carm. ACM 21, 11, 1978.

[CaH] R. Cc 呵 :,bell and A. Habennann, "The specification of process synchronization by Path 氐平迳ssions" , I1祀S 16, 1974.

[Car] L. cardelli, "Analog Processes", To appear in Proc 9th MFCS,

Poland, 过80.

[CoH] F. Cctrm:mer and A. Holt, "Marked directed graphs", JCSS 5, 1971. [DFLJ J. Dennis, J. Fosseen and J. Lindennan, "Data flCM schanas",

INCS 5, 1974.

[Dij J E. Dijkstra, "Guarded cat 即血 s , nonde七e 江 ninacy and formal derivation of programs", carm. ACM 18, 8, 1975.

四 J P. van 旺 :le Boas and T. Janssen, "The 乓 iact of Frege's principle of canpositionality for the sanantics of programning and natural languages", 距 port 79-07, Dept. of Math ema 七 ic s , University of

Ams 七 er dam, 1979.

[GLTJ H.J. Genrich, K. Lautenbach and P.s. Thiagarajan, "An 叩 erv i ew

of Net Theoryl,l

Proc. Advanced Course on General Ne 七 Theory of

Processes and Sys坛延 ， 七o appear in I.NCS, 1980.

[HAL] C. Hewi七七， G. At tar di 田飞I H. Liebermann, "Specifying and proving propert 迨 of guardians for distriliuted systaus", I.NCS 70, 1979.

[HM] M. Hennessy and R. Milner, "On observing nondetennini 郔 and concurrency", to be presented a 七 8th ICALP a 七 沁毗 er dam, and appear in 丑心 ，1980.

[HoalJ C.A.R. Hoare, "TOWards a theory of parallel programning", in Opera 七 ing Systems Techniques, 及 cademic Press, 1972.

[Hoa.2] C.A.R. Hoare, "M:>I让七or s : an OJ?E江 a 七年 g sys tem s七ru e 七立 in g concept", Ccmn. ACM 17, 10, 1974.

[Hoa.3] C. 凡 R. Hoare, "Carm.micating Sequential Processes", carm. ACM 21, 8, 1978.

瞳 l J M. Hennessy and G. Plotkin, "Full abstraction for a s 扭 iple parallel programming language", Proc 8th 娅 它 ，Czechos l ovak 扭 ， INCS 74, 1979.

[HP2J M, Hennessy and G. Plotkin, "A te:rm model for CCS", 七 o appear in Proc 9th 妇 ，Poland, 1980.

[Hue] G. Hue 七， •Conf l u 五比r ed uc七i on s : ab s 江 act properties and applications to te:rm-r ei 心 i t in g systems", Report No. 250, IRIA Laboria, Paris 19·77.

[Kel J R. Keller, "A fundanental theor 甸 of asynchronous parallel

~ 兀 iputa 已 on' 勹 Par all e l Processing, ed. T.Y. Feng, Springer, 1975.

［四 J G. Kahn and D. MacQueen, "Coroutines and networks of parallel processes", Proc. 工FIP Congress, North Holland, 过 77.

IKnnJ H.T. Kung, "Synchronized and asynchronous algorithms" in Algorithms and Canplexity, ed J.F. Traub , Academic Press, 1976.

［出11] R. Milner, "Processes; a matherna 七 i cal model of canputing agents" , Pr oc 江平c Col:);叩 um ' 73, ed. 沁 se and Shepherdson, North Holland, 1973.

[Mil2J R. Milner, "Fl°'11 平 aphs and flow algebras" , J. ACM 26, 4, 1979.

[Mil3J R. Milner, "Syn 归 si s of crn 血 皿 eatin g behaviour", Proc 7th

妞 屯 S, Poland, INCS 64, 1978.

[Mil4J R. Milner, "Algebras for 立叮m血 ca 七in g sys 七ems" , Report CSR一25- 78 , Canputer Science Dep七 ， Edinburgh Univers 让y, 1978.

[MilSJ R. Milner, "An al gebr 吐c theory of synchroni za已on", INCS 67, 1979.

[Mln] G. Mi血 ， "A math ema七i ca l model of concurren七 carq;兀七a 已 on" ,

Ph.D. Thesis, Canputer Science Dept, University of 囡 inburgh, 1978.

［战 J G. Milne and R. Milner, "Concurr en 七 p rocesses and their syn七ax " , J. ACM, 26, 2, 1979.

［沁s J P• Mosses, "SIS, Seman七i c Impl enenta已on System", DA工Mr Report MD-33, Aarhus Uni 凭 r si 七 y , 1979.

［坟J D. Mad;注汜n, "吵 l s for distributed canpu 七 in g" , Report No. 351, IRIA-Laboria, Paris, 1979.

四 J T. Miildner, "On synchronizing tools for parallel programs", Report 357, Ins 七 . of canputer Science, Poli sh 沁 adany of Science, Warsaw, 1979.

［螂] A Maggio l o 一Sche 七 七 年 i , H. Wedde and J. WinkCMSki, "M:xlelling a Solution for a control probl 卸 in distributed systems by restrictions, 平 S 70, 1979.

[Pe七J C.A. Petri, "Introducti on 七o General Ne七 Theory '勹 Proc . Advanced Course on General Net Theory of Processes and Systems, 七 0 appear in INC.S, 1980.

[Plol'J G. Plotkin, "A po 车 dana in construction", SIAM J. Camp 5, 1976.

[Plo2J G. Plotkin, "LCF considered as a pr 哼 arrroing l angua ge 飞 峦 5 , 3, 1977.

卧 hl J A. 即 ue li , "The 坛 虾 poral logic of programs", 19th Annual Symp. on Founda 已 ons of Conputer Science, Providence R.I. , 1977.

际 u2 J A, Pnueli, "'The 七己丐心r a l s eman已 cs of con curren七 p rograms" ,

丑 心 70 , 1979.

［沁S] B, 沁 sen, "Tree manipulation systems and Church-Rosser T 氐 r ems" ,

J. 虹 20, 1, 1973.

[Sch] J. Schwarz, "Di s tr ib u 七 ed synchronization of processor camnmica 七 i on" ,

Internal Report, Dep 七 of Artificial Intelligence, Uni ver s i 七 y of Ed土nbu工gh , 1978.

[Smy] M. Smyth , "P domains", J笠S 16, 1978.

[SSJ D. Seo七七 and C. S 扛 ach ey , "Tow.江 ds a math ema七i ca l seman七i cs for ca:nputer l anguages 飞 Proc . S 泗 . on Computers and Au四 ta ,

隘 cr owave Res. Ins 七 . Symposia Se 已 es , Vol 21, Polytechnic Ins 七of Brooklyn, 1972.

[Wad] w. Wadge, "An extensional treatment of data fl o付 dead l ocl守，

辽祀S 70, 1979.

[Wir] N. Wirth, " 心 UIA: A language for modular multiprograrrrning", Report 18, ETH Zurich, 1976.