UNIVERSITY OF SUSSEX

s g ven n ter s of a reduct on re at on between ***configurations*** u t sets of λ_{cv} c osed express ons or progra s [illegible]nfortunate y t s operat ona se ant cs s not co pos t ona n t at t e be av our of a λ_{cv} express on or ndeed con gurat on s not deter ned by t at of ts const tuents

Here we g ve a co pos t ona operat[illegible]ona se ant cs n ter s of a abe ed trans t on syste for ***μ***CML progra s [illegible] s not on y descr bes t e eva uat on steps of progra s as n [illegible] but a so t e r co un cat on potent a s n ter s of t e r ab ty to nput and output va ues a ong co un cat on c anne s

e t en proceed to de onstrate t e usefu ness of t s co pos t ona oper at ona se ant cs by us ng t to de ne a vers on of ***weak observational equiv-alence*** [illegible] su tab e for ***μ***CML e prove t at odu o t e usua prob e s assoc ated w t t e c o ce operator of CC our c osen equ va ence s preserved by a ***μ***CML contexts and t erefore ay be used as t e bas s for reason ng about CML progra s In t s paper we do not nvest gate n deta t e resu t ng t eory but con ne ourse ves to po nt ng out so e of ts sa ent features for exa p e standard dent t es one wou d expect of a ca by va ue λ ca cu us are g ven and we a so s ow t at certa n a gebra c aws co on to process a gebras [illegible] o d

e now exp a n n ore deta t e contents of t e re a nder of t e paper

IN EC[illegible]ION 2 we descr be t e anguage ***μ***CML a subset of CML It s a typed anguage w t base types for c anne na es boo eans and ntegers and type constructors for pa rs funct ons and de ayed co putat ons t ese ast are ca ed Event types It as t e standard constructs and constants assoc ated w t t e base types and w t pa rs and funct ons In add t on t as a se ect on of t e CML constructs and constants for an pu at ng de ayed co putat ons `spawn` gener

fst	$A \times B \to A$	transmit$_A$	chan $\times A \to$ unit event
snd	$A \times B \to B$	receive$_A$	chan $\to A$ event
add	int $\times$ int $\to$ int	choose	A event $\times A$ event $\to A$ event
mul	int $\times$ int $\to$ int	spawn	(unit $\to$ unit) $\to$ unit
leq	int $\times$ int $\to$ bool	wrap	A event $\times (A \to B) \to B$ event
sync	A event $\to A$	never	unit $\to A$ event
always	$A \to A$ event		

FIG E

nce **A**v ed ate y eva uates to t e constant v we ave

$$\overline{\mathbf{A}v \xrightarrow{\tau} v}$$

e c o ce construct choose e s a c o ce between ***delayed computations*** as choose as t e type A event A event → A event o nterpret t we ntroduce a new c o ce constructor ge ge_2 w ere ge and ge_2 are guarded express ons of t e sa e type en choose e proceeds by eva uat ng e unt t can produce a va ue w c ust be of t e for [ge], [ge_2] and t e eva uat on cont nues by construct ng t e ***delayed computation*** [ge ge_2] s s represented by t e ru e

$$\frac{e \xrightarrow{\sqrt{[ge\ ],[ge_2]}} e'}{\text{choose}\, e \xrightarrow{\tau} e'\ [ge\ \ ge_2]}$$

e notat on ntroduced n s unfortunate as t s used n 4 to represent t e ***internal choice*** between processes w ereas ere t represents ***external choice:*** we ave t e fo ow ng aux ary ru es w c are t e sa e as CC su at on

$$\frac{ge \xrightarrow{\alpha} e}{ge\ \ ge_2 \xrightarrow{\alpha} e} \qquad \frac{ge_2 \xrightarrow{\alpha} e}{ge\ \ ge_2 \xrightarrow{\alpha} e}$$

s ends our nfor a desc a 4 t 2 su aon

For any purposes strong b s u at on s too ne an equ va ence as t s sens t ve to t e nu ber of reduct ons perfor ed by express ons s eans t w not even va date e e entary propert es of β reduct on suc as ***Id*** = w ere ***Id*** denotes t e dent ty funct on (fn x ⇒ x) e requ re t e ooser ***weak bisimulation*** w c a ows τ act ons to be gnored

s n turn requ res so e ore notat on Let $\overset{\varepsilon}{\Longrightarrow}$ be t e re ex ve trans t ve c osure of $\xrightarrow{\tau}$ and et $\overset{l}{\Longrightarrow}$ be $\overset{\varepsilon}{\Longrightarrow}\xrightarrow{l}$ e any sequence of s ent act on fo owed by an ***l*** act on Note t at we are ***not*** a ow ng s ent act ons after t e ***l*** act on Let $\overset{\hat{l}}{\Longrightarrow}$ be $\overset{\varepsilon}{\Longrightarrow}$ f $l = \tau$ and $\overset{l}{\Longrightarrow}$ ot erw se en $\mathcal{R}$ s a ***first-order weak simulation*** ff t s structure preserv ng and t e fo ow ng d agra can be co p eted

e

$$\begin{tikzcd} e \arrow[r, phantom, "=^h"] \arrow[d, "l"'] & e_2 & & e \arrow[r, phantom, "=^h"] \arrow[d, "l"'] & e_2 \arrow[d, Rightarrow, "l_2"'] \\ e' & & & e' \arrow[r, phantom, "\approx^h"] & e_2' \end{tikzcd}$$

as ... w ere $l \ \approx^{hl} l_2$

P OPO I ION $=^h$ ***is an equivalence.***

P OOF ar to t e proof of Propos t on □

s atte pt fa s owever s nce t on y oo s at t e rst ove of a process and not at t e rst oves of any processes n ts trans t ons us t e above μCML counter exa p e for $\approx^h$ be ng a congruence a so app es to $=^h$ s fa ure was rst noted by o sen 2 for CHOC

o sen s so ut on to t s prob e s to requ re t at τ oves can a ways be atc ed by at east one τ ove w c produces s de n t on of an ***irreflexive simulation*** as a structure preserv ng re at on w ere t e fo ow ng d agra can be co p eted

$$\begin{tikzcd} e \arrow[r, phantom, "\mathcal{R}"] \arrow[d, "l"'] & e_2 & & e \arrow[r, phantom, "\mathcal{R}"] \arrow[d, "l"'] & e_2 \arrow[d, Rightarrow, "l_2"'] \\ e' & & & e' \arrow[r, phantom, "\mathcal{R}"] & e_2' \end{tikzcd}$$

as ... w ere $l \ \mathcal{R}^l \ l_2$

Let $\approx^i$ be t e argest rre ex ve b s u at on

P OPO I ION $\approx^i$ ***is a congruence.***

P OOF e proof t at $\approx^i$ s an equ va ence s s ar to t e proof of Propos t on e proof t at t s a congruence s s ar to t e proof of eore 4 n t e next sect on □

However t s re at on s rat er too strong for any purposes for exa p e add(,2) $\not\approx^i$ add(,add(,)) s nce t e r s can perfor ore τ oves t an t e s s s s ar to t e prob e n CHOC w ere $a.\tau.P \not\approx^i a.P$

In order to nd an appropr ate de n t on of b s u at on for μCML we observe t at μCML on y a ows □ to be used on ***guarded expressions*** and not on arb trary express ons e can t us gnore t e n t a τ oves of a express ons ***except*** for guarded express ons For t s reason we ave to prov de ***two*** equ va ences one on ter s w ere we are not nterested n n t a τ oves and one on ter s w ere we are

A pa r of c osed type ndexed re at ons $\mathcal{R} = (\mathcal{R}^n, \mathcal{R}^s)$ for a ***hereditary simulation*** we ca $\mathcal{R}^n$ an ***insensitive simulation*** and $\mathcal{R}^s$ a ***sensitive simulation*** ff $\mathcal{R}^s$ s structure preserv ng and we can co p ete t e fo ow ng d agra s

$$\begin{tikzcd} e \arrow[r, phantom, "\mathcal{R}^n"] \arrow[d, "l"'] & e_2 & & e \arrow[r, phantom, "\mathcal{R}^n"] \arrow[d, "l"'] & e_2 \arrow[d, Rightarrow, "l_2"'] \\ e' & & & e' \arrow[r, phantom, "\mathcal{R}^{n\circ}"] & e_2' \end{tikzcd}$$

as ... w ere $l \ \mathcal{R}^{sl} \ l_2$

and

$$\begin{tikzcd} e \arrow[r, phantom, "\mathcal{R}^s"] \arrow[d, "l"'] & e_2 & & e \arrow[r, phantom, "\mathcal{R}^s"] \arrow[d, "l"'] & e_2 \arrow[d, Rightarrow, "l_2"'] \\ e' & & & e' \arrow[r, phantom, "\mathcal{R}^{n\circ}"] & e_2' \end{tikzcd}$$

as ... w ere $l \ \mathcal{R}^{sl} \ l_2$

L2

[illegible]

clusions:

$$\begin{tikzcd}
\square \arrow[rrr, hook] \arrow[d, hook] & & & \approx \arrow[d, hook] \\
\square^{h} \arrow[r, hook] & \approx^{i} \arrow[r, hook] & \approx^{s} \arrow[r, hook] \arrow[d, hook] & \approx^{n} \arrow[d, hook] \\
& & =^{h} \arrow[r, hook] & \approx^{h}
\end{tikzcd}$$

P OOF For eac nc us on s ow t at t e rst b s u at on sat s es t e cond t on requ red to be t e second for of b s u at on o s ow t at t e nc us ons are str ct we use t e fo ow ng exa p es

$$\begin{aligned}
(\mathsf{fn}\,\boldsymbol{x} \,\square\, \mathsf{add}(\ ,2)) &\ \square^{h}\square\ \ (\mathsf{fn}\,\boldsymbol{x} \,\square\, \mathsf{add}(2,\ )) \\
&\approx\ \square\ \ \mathsf{let}\,\boldsymbol{x} =\ \ \mathsf{in}\,\boldsymbol{x} \\
\mathsf{choose}(\mathsf{receive}\,\boldsymbol{k}, \mathsf{tau}(\mathsf{receive}\,\boldsymbol{k})) &\ \approx^{i}\square^{h}\ \mathsf{tau}(\mathsf{receive}\,\boldsymbol{k}) \\
\mathsf{add}(\ ,2) &\ \approx^{s}\boxminus^{i}\ \mathsf{add}(\ ,\mathsf{add}(\ ,\ )) \\
&\ \approx^{n}\boxminus^{s}\ \mathsf{let}\,\boldsymbol{x} =\ \ \mathsf{in}\,\boldsymbol{x} \\
\mathsf{never}() &\ \approx^{h}\boxminus^{n}\ \mathsf{tau}(\mathsf{never}()) \\
&\ \approx^{h}\boxminus^{h}\ \mathsf{let}\,\boldsymbol{x} =\ \ \mathsf{in}\,\boldsymbol{x}
\end{aligned}$$

w ere

$$\mathsf{tau} = \mathsf{fn}\,\boldsymbol{x} \,\square\, \mathsf{wrap}(\mathsf{always}\,\boldsymbol{x}, \mathsf{sync})$$

Note t at t s sett es an open quest on 2 of o sen s as to w et er $\approx^{i}$ s t e and s nce $\mathcal{R}$

refinement $\widehat{\mathcal{R}}$ be de ned

$$\widehat{\mathcal{R}}^{\boldsymbol{n}} = \{(\boldsymbol{D_n}[\boldsymbol{e}], \boldsymbol{D_n}$$

P OPO I ION 4 *If $\mathcal{R}$ is an equivalence then $\mathcal{R}^{\bullet\square}$ is symmetric.*

P OOF A var ant of t e proof n

It suf ces to s ow t at f $e\,\mathcal{R}^{\bullet s}\,f$ t en $f\,\mathcal{R}^{\bullet s\square}\,e$ and t at f $e\,\mathcal{R}^{\bullet n}\,f$ t en $f\,\mathcal{R}^{\bullet n\square}\,e$ w c we s ow by nduct on on e If $e\,\mathcal{R}^{\bullet s}\,f$ t en e t er

- $e = D[\![e]\!]\,\widehat{\mathcal{R}^{\bullet}}^{s}\,D[\![f]\!]\,\mathcal{R}^{s\circ}\,f$ and $e_i\,\mathcal{R}^{\bullet s}\,f_i$ so by nduct on $f_i\,\mathcal{R}^{\bullet s\square}\,e_i$ so $f\,\widehat{\mathcal{R}}^{s}$ $D[\![f]\!]D\,\widehat{\mathcal{R}}^{s\square}\,[\![e]\!] = e$ or
- $e = \mathsf{fix}(x = \mathsf{fn}\,y \square e^{\square})\,\widehat{\mathcal{R}^{\bullet}}^{s}\,\mathsf{fix}(x = \mathsf{fn}\,y \square f^{\square})\,\mathcal{R}^{s\circ}\,f$ and $e^{\square}\,\mathcal{R}^{\bullet n}\,f^{\square}$ so by nduct on $f^{\square}\,\mathcal{R}^{\bullet n\square}\,e^{\square}$ so $f\,\widehat{\mathcal{R}}^{s}\,\mathsf{fix}(x = \mathsf{fn}\,y \square f^{\square})\,\mathcal{R}^{\bullet s\square}\,\mathsf{fix}(x = \mathsf{fn}\,y \square e^{\square}) = e$

e proof for $\mathcal{R}^{n}$ s s ar □

e can use t s resu t to s ow t at $\approx^{\bullet\square}$ s a b s u at on

P OPO I ION 4 *When restricted to closed expressions of μCML^{+}, $\approx^{\bullet\square}$ is a hereditary bisimulation.*

P OOF By Propos t on 4 4 $\approx^{\bullet}$ s a ered tary s u at on and so $\approx^{\bullet\square}$ s a ered tary s u at on By Propos t on 4 $\approx^{\bullet}$ s sy etr c and so $\approx^{\bullet}$ s a ered tary b s u at on □

s g ves us t e resu t we set out to prove

HEO EM 4 *$\approx^{s}$ is a congruence, and $\approx^{n}$ is an uneventful congruence.*

P OOF Fro Propos t on 4 $\approx^{\bullet}$ s a ered tary b s u at on so $\approx^{\bullet} \square \approx^{\circ}$ and by Propos t on 4 2 $\approx^{\circ} \square \approx^{\bullet}$ so $\approx^{\bullet}$ and $\approx^{\circ}$ are t e sa e re at on nce $\widehat{\approx^{\bullet}} \square \approx^{\bullet}$ we ave t e des red resu t by Propos t on 4 □

5 Properties of Weak Bisimulation

In t s sect on we s ow so e resu ts about progra equ va ence up to ered tary wea b s u at on o e of t ese equ va ences are easy to s ow but so e are tr c er and requ re propert es about t e trans t on syste s generated by μCML^{+} A t oug uc re a ns to be done on e aborat ng t e a gebra c t eory of μCML progra s we ope t at t e resu ts n t s sect on nd cate t at t s equ va ence can for t e bas s of a usefu t eory w c genera ses t ose assoc ated w t process a gebras and funct ona progra ng

e ave g ven an operat ona se ant cs to μCML by extend ng t w t new constructs ost of w c correspond to constructs found n standard process a gebras ese nc ude a c o ce operator □ a para e operator □ and su tab e vers ons of nput and output pre x ng e pre xes n μCML^{cv} ave a

s g t y unusua syntax t e r equ va ents n CC are g ven as

CCS prefix	*μCML^{cv} equivalent*
$k\ x.P$	$k \square \mathsf{fn}\,x \square P$
$k\ v.P$	$k\ v \square \mathsf{fn}\,x \square P$
$\tau.P$	$\mathbf{A}() \square \mathsf{fn}\,x \square P$

e now exa ne t e extent to w c □ and □ act e c o ce and para e opera tors fro a process a gebras

e can nd b s u at ons for t e fo ow ng and ence t ey are sens t ve b s ar

$$\Lambda \square e \square e$$
$$(e \square e_2) \square e \square e \square (e_2 \square e)$$
$$(e \square e_2) \square e \square (e_2 \square e) \square e$$

us □ sat s es any of t e standard aws assoc ated w t a para e operator n a process a gebra However t s not n genera sy etr c because of ts nteract on w t t e product on of va ues

$$v \square e \square e$$

For exa p e

$$\square\Lambda \square \Lambda \quad \Lambda \square \square$$

s eans t at we can v ew t e para e co pos t on of processes as be ng of t e for

$$(\|_i e_i) \square f$$

w ere t e order of t e e_i s un portant Note t at t ***is*** portant w c s t e r g t ost express on n a para e co pos t on s nce t s t e a n t read of co putat on and so can return a va ue w c none of t e ot er express ons can

e c o ce operator of μCML^{+} a so sat s es t e expected aws fro process a gebras t ose of a co utat ve ono d a t oug t can on y be app ed to guarded express ons

$$\Lambda \square ge \square ge$$
$$(ge \square ge_2) \square ge \square ge \square (ge_2 \square ge)$$
$$ge \square ge_2 \square ge_2 \square ge$$

s eans t at we can v ew t e su of guarded express ons as be ng of t e for

$$\bigoplus_i ge_i$$

w ere t e order of t e $\boldsymbol{ge_i}$ s un portant

In fact guarded express ons can be v ewed n a anner qu te s ar to t e ***sum forms*** used n t e deve op ent of t e a gebra c t eory of CC 2 e can nd b s u at ons for t e fo ow ng and ence t ey are sens t ve b s ar

$$(\boldsymbol{ge} \ \Box \boldsymbol{ge}_2) \ \Box \ \boldsymbol{v} \ \Box \ \ (\boldsymbol{ge} \ \ \Box \ \boldsymbol{v}) \ \Box \ (\boldsymbol{ge}_2 \ \Box \ \boldsymbol{v})$$

$$\boldsymbol{ge} \ \Box \ \mathsf{fn}\, \boldsymbol{x} \ \Box \ \boldsymbol{x} \approx^s \boldsymbol{ge}$$

$$\mathbf{A}\boldsymbol{v} \approx^s \mathbf{A}() \ \Box \ \mathsf{fn}\, \boldsymbol{x} \ \Box \ \boldsymbol{v}$$

Fro t s we can s ow by structura nduct on on t at a guarded express ons are of a g ven for

$$\boldsymbol{ge} \approx^s \bigoplus_{\boldsymbol{i}} \boldsymbol{ge_i}\, \mathsf{ge}$$

λ_{cv} express ons Instead of u t sets we use ***configurations*** of μCML^{cv} expres s ons g ven by t e gra ar

$$C \,\square\, Conf \;=\; e \mid C \square C \mid \Lambda$$

Note t at con gurat ons are restr cted for s of μCML^{+} express ons s w fac tate t e co par son between t e two se ant cs s nce t can be carr ed out for con gurat ons rat er t an μCML express ons

e se ant cs of s expressed as a reduct on re at on $= \square$ between con gurat ons and reduct ons ave four ndependent sources e rst nvo ves a sequent a reduct on w t n an nd v dua μCML express on and t s n turn s de ned us ng anot er reduct on re at on $\boxminus\!\rightarrow$ t e second s t e spawn ng of new ***computation threads*** w c resu ts n an ncrease n t e nu ber of co ponents of t e con gurat on t e t rd s co un cat on between two express ons and t e ast s requ red to and e t e always construct e need notat on for eac of t ese and we cons der t e n turn

e operat ona ru es for sequent a reduct on are de ned ***in context*** n t e sty e of r g t and Fe e sen and t e contexts t at per t reduct on are g ven by t e fo ow ng gra ar

$$E \;=\; [\cdot] \mid E\,e \mid v\,E \mid c\,E \mid (E,e) \mid (v,E) \mid \mathsf{let}\,x = E\,\mathsf{in}\,e \mid \mathsf{if}\,E\,\mathsf{then}\,e\,\mathsf{else}\,e$$

e re at on $\boxminus\!\rightarrow$ s de ned to be t e east re at on sat sfy ng t e fo ow ng ru es

$$\begin{array}{rcll}
E[c\,v] & \boxminus\!\rightarrow & E[\delta(c\,v)] \qquad (c \,\square\, \{\mathsf{spawn}, \mathsf{sync}\}) & \underline{\text{const}} \\
E[(\mathsf{fix}(x = \mathsf{fn}\,y \,\square\, e))\,v] & \boxminus\!\rightarrow & E[e[\mathsf{fix}(x = \mathsf{fn}\,y \,\square\, e)/x][v/y]] & \underline{\text{beta}} \\
E[\mathsf{let}\,x = v\,\mathsf{in}\,e] & \boxminus\!\rightarrow & E[e[v/x]] & \underline{\text{ et}} \\
E[(v,w)] & \boxminus\!\rightarrow & E[\square v, w \square] & \underline{\text{pa r}}
\end{array}$$

Here eac ru e corresponds to a bas c co putat on step n a sequent a ca by va ue anguage e s ou d po nt out t at t e ast ru e does not appear n t s p c t n eppy s state ent t e syntact c c ass of t e ter $(v\ , v_2)$ s e t er ***Exp*** or ***Val*** t s a b gu ty s reso ved n favour of ***Val*** e ave ade t e gra ar una b guous and ave added an exp c t reduct on ru e for reso v ng a b gu ty

Note t at t e de n t on of $\boxminus\!\rightarrow$ s not co pos t ona t e reduct ons of an express on are not de ned n ter s of t e reduct ons of ts sub express ons e fo ow ng Le a w be usefu n ater proofs and s ows t at we can recover co pos t ona ty

LEMMA ***If*** $e \boxminus\!\rightarrow e^{\square}$ ***then*** $E[e] \boxminus\!\rightarrow E[e^{\square}]$.

P OOF By exa nat on of t e proof of t e trans t on $e \boxminus\!\rightarrow e^{\square}$ □

o capture reduct ons w c nvo ve co un cat on t s necessary to de ne a t on 2 as t e $\boldsymbol{\mu}$CML^{+} se ant cs and we now co pare t e In order to do t s we extract a abe ed trans t on syste fro t e $\boldsymbol{\mu}$CML^{cv} se ant cs by de n ng

$C \xrightarrow[\surd]{\tau} C'$ ff $C \Longrightarrow C'$

$C \xrightarrow{v} C'$ ff $C = C'' \, v$ and $C' = C'' \, \Lambda$ up to assoc at v ty and Λ eft un t

$C \xrightarrow{k\,v} C'$ ff $C \, k \Longrightarrow C' \, v$

$C \xrightarrow{k\,x} C'$ ff $C \, k \, x \Longrightarrow C' \, ()$

e w t en s ow t at t s abe ed trans t on syste s wea y b s ar to t e $\boldsymbol{\mu}$CML^{+} ts

HEO EM 2 ***The μCML^{cv} semantics of a configuration is weakly bisimilar to its μCML^{+} semantics.***

e re a nder of t s sect on s devoted to prov ng t s resu t A t oug t e sty e of presentat on of t ese two se ant cs are very d fferent t e resu t ng re at ons are very s ar and t ere are essent a y on y two sources for t e d fferences e rst s t at certa n reduct ons n $\boldsymbol{\mu}$CML^{cv} w en ode ed n t e $\boldsymbol{\mu}$CML^{+} se ant cs requ re n add t on so e ouse eep ng reduct ons A typ ca exa p e s t e reduct on

$$(\mathsf{fn}\, x \, e)v \longmapsto e[v/x].$$

In $\boldsymbol{\mu}$CML^{+} t s requ res two reduct ons

$$(\mathsf{fn}\, x \, e)v \xrightarrow{\tau} \mathsf{let}\, x = v\, \mathsf{in}\, e \xrightarrow{\tau} e[v/x]$$

s prob e s and ed by dent fy ng t e set of ouse eep ng reduct ons suc as t e second reduct on above w t n t e $\boldsymbol{\mu}$CML^{+} se ant cs ese turn out to be very s p e and we can wor w t ouse eep ng nor a for s n w c no furt er ouse eep ng reduct ons can be ade

e second d vergence between t e se ant cs concerns t e treat ent of spawn express ons n $\boldsymbol{\mu}$CML^{+} ay spawn new processes w c g ve r se to e equ va ence ≡ s a strong rst order b s u at on w c respects ouse eep ng t at s a re at on $\mathcal{R}$ w ere we can co p ete t e d agra

$$\begin{tikzcd}
e \arrow[d, "\tau_H"'] & \mathcal{R} & e_2 & \text{as} & e \arrow[d, "\tau_H"'] & \mathcal{R} & e_2 \arrow[d, "\tau_H"'] \\
e' & & & & e' & \mathcal{R} & e_2'
\end{tikzcd}$$

and s ar y for $\mathcal{R}^{-}$

P OPO I ION ***≡ is a strong first-order bisimulation which respects house-keeping.***

P OOF ee t e Append x □

e can a so s ow a very strong correspondence between reduct ons of μCML^{cv} con gurat ons and t e r t dy nor a for s

P OPO I ION ***If $C \xrightarrow{\tau_H} {}' e$ and e is tidy, then the following diagrams can be completed:***

$$\begin{tikzcd}
C \arrow[r, "\tau_H'"] \arrow[d, "l"'] & e & \textit{as} & C \arrow[r, "\tau_H'"] \arrow[d, "l"'] & e \arrow[d, "l"] \\
C' & & & C' \arrow[r, "\tau_H'^{\circ}"] & \equiv e'
\end{tikzcd}$$

and:

$$\begin{tikzcd}
C \arrow[r, "\tau_H'"] & e & & C \arrow[r, dash] & {}
\end{tikzcd}$$

c ude c anne generat on t w be necessary to adopt t e ***context bisimulation equivalence*** or g na y deve oped n [illegible] In s ort a t oug se ant c t eor es are be ng deve oped ndependent y for t ese anguages any of t e tec n ques deve oped w nd ore genera app cat on

Appendix

[illegible] s sect on s devoted to t e proof of Propos t on [illegible] and Propos t on [illegible] But rst we need so e aux ary resu ts [illegible] e fo ow ng t ree Propos t ons state

4 M Hennessy *Algebraic Theory of Processes* MI Press

C A Hoare *Communicating Sequential Processes* Prent ce Ha

oren Ho stro PFL A funct ona anguage for para e progra ng In *Proc. Declarative Programming Workshop* pages 4

Doug as Howe Equa ty n azy co putat on syste s In *Proc. LICS 89* pages 2

Doug as Howe Prov ng congruence of s u at on order ngs n funct ona anguages npub s ed anuscr pt 2

A an Jeffrey A fu y abstract se ant cs for a concurrent funct ona anguage w t onad c