UNIVERSITY OF SUSSEX

COMPUTER SCIENCE

Towards a Behavioural Theory of Access and Mobility Control in Distributed Systems

M. Hennessy
M. Merro
J. Rathke

Report 01/2002 October 2002

Computer Science
School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH

ISSN 1350–3170

Towards a Behavioural Theory of Access and Mobility Control in Distributed Systems

/ NN * / / RRO an J TH

ABSTRACT. We define a typed bisimulation equivalence for the language PI, a distributed version of the **π**-calculus in which processes may migrate between dynamically created locations. It takes into account resource access policies, which can be implemented in PI using a novel form of dynamic capability types. The equivalence, based on typed actions between configurations, is justified by showing that it is *fully-abstract* with respect to a natural distributed version of a contextual equivalence.

In the second part of the paper we study the effect of controlling the migration of processes. This affects the ability to perform observations at specific locations, as the observer may be denied access. We show how the typed actions can be modified to take this into account, and generalise the *full-abstraction* result to this more delicate scenario.

1 Introduction

b av our o pro ss s n a str but s st p n s on t r sour s t av b n a o at or ov r t s r sour s or a pro ss s now o t s r sour s a var ov r t r or an a quat b av oura t or o str but s st s ust b bas not on on t n r nt ab t s o pro ss s to nt ra t w t ot r pro ss s but ust a so ta nto a ount t na r sour nv ron nt n w t ar op rat n In our approa u nts w ta t or

$$| \; M \approx N,$$

w r **N** an **M** ar s st s an r pr s nts t r o put n nv ron nt Intu t v t s ans t at **M** an **N** r t sa b av our

- t o put n nv ron nt a var na a r t n bot t ov ra r sour s ava ab to **M** an **N** an t vo v n now t at us rs a a u u at o t s r sour s

s s v op n t r s o t an ua PI a v rs on o t **π** a u us n w pro ss s a rat b tw n o at ons w n turn an b na a r at As xp a n n r sour a ss po s n PI a b p nt us n a *p t* bas t p s st t us

na top o t pap r s t t o *r on* on t b av our o s st s In t rat on o pro ss s s un onstra n r vant r u t on ru s

k⟦goto l.P

ta s ar v n n t on p w r w a so onstrat t pow r o t s an s

r a n r o t pap r s vot to xt n n t r su t abov to t s an ua pow r o ont xts w an us t s apa b t **moves** to ontro a ss to s t s turns out to b v r o p x o s p att rs w a r ss t as w r t on or o t s apab t a ow s **move**$_*$ w t $*$ b n a w ar t us t nv ron nt as t s apab t or a o at on **k** t n o at ons av rat on r ts to **k** w ar on a ow t r t nv ron nt as rat on r ts to **k** as b or or **k** s n **T** A ount r xa p s v n n t on

It turns out t at w ust b ar u about t o at on at w n or at on s arn In or at on about **k** arn at **l** an not b us w t out t apab t to ov to **k** How v r t s n or at on ust b r ta n b aus t at ov apab t a subs qu nt b obta n s a s to a or o p at or o nv ron nt w r or s

- o at ons at w t st n pro ss s a b p a **T**
- *o* ava ab n or at on on apab t s at o at ons
- s ar *o* ava ab n or at on

ta s ar v n n t on w a so onta ns a n ra sat on o t t p a t ons o abov to t s or o p at nv ron nts

M, N	*st s*
l⟦P⟧	Lo at ro ss
M | N	Co pos t on
new **n** **M**	a op n
0	r nat on

P, Q	*ro ss s*
u V P	utput
u **X** **P**	Input
goto **v.W**	rat on
if **u** **v** then **P** else **Q**	at n
newc **n** A **P**	C ann a r at on
newreg **n** G **P**	st r a r at on
newloc **k** K with **C** in **P**	Lo at on a r at on
P | Q	Co pos t on
P	p at on
stop	r nat on
U, V, W	*s*
α₁, . . . , αₙ , n >	tup s
α, α′	*G n r s nt rs*
u	I rs
u₁, . . . , uₙ @**u, n ≥**	Lo at I rs

I GUR 1 Syntax of PI

---

onstru t if **u** **v** then **P** else **Q** a or o r urs on **P** an t r or s o na r at on

- newc **a** A **P** t r at on o a n w *o* *nn* o t p A a **a**
- newreg **n** rc A

s nt n t t r a **P** runn n at o at on **l** s a b o b n w t t para op rator **|** an na s a b s ar b tw n t r a s us n t onstru t **new e** w r s on o A rc A or K

ro ss s s s st s an n t p s a onta n o urr n s o var ab s an t s a b boun n t onstru t **u X P x** app ars n t patt rn **X** t n a o urr n s o **x** n an **P** ar boun s a s to t not ons o r an boun var ab s aptur avo n subst tut on o rs or var ab s **P{v/x}** an **α** qu va n s ar a stan ar apart ro subst tut ons nto t p s w s not qu t s nta t t ta s o subst tut on nto t p s a b oun n n t on sa t at a s st or pro ss t r s *os* t onta ns no r o urr n s o var ab s

an ua a so onta ns b n n onstru ts or na s **newc n** A **P** **newreg n** G **P** an **newloc k** K with **C** in **P** n pro ss s o w a so av t not ons o r an boun na s n t r s an as usua t n t on o **α** qu va n s t r s w on r b t r us o boun na s

UCTION S NTIC s s v n n t r s o a b nar r at on b tw n *os* s st s

$$\mathbf{M} \rightarrow \mathbf{N}$$

an s a n ra sat on o t at v n n or PI It s a *on* *t t* uu

Bas p s	B	**int | bool | unit | □ | ...**
Lo a C ann t p s	A	r□ □ | w□ □ | rw□ , □
		prov **<**
Capab t p s		**u** A
Lo at on p s	K	loc 1, . . . , n , n ≥
st r a p s	G	rc□A□
a u p s	C	B | A | G | A @u | A @K
rans ss on p s		C1, . . . , Cn , n ≥

Types

3 Typing

In t s s t on w out n t t p s us to ontro r sour s an t a o pan n t p n s st start n po nt s s ar to t t p n s st o o but t r ar t r a or r n s

- us a n w at or o t p s *r st r n t p s* to xp t ana t r sour na s w an b s ar b tw n r nt o at ons
- t p s xpr ss ons ar a ow to onta n var ab s t r b v n r s to w at w a *n* t p s t onstra nts t p a on a nt b av our s t r n na a b nstant at on o t s var ab s
- not on o t p nv ron nt s an t o not xp t onta n asso at ons b tw n na s an o at on t p s

3.1 The Types

o t on o t p s s an xt ns on o t os us n o to w t

OC CH NN T P , ran ov r b A an a b r str t to r a
on apab t r

(U -CTOP)

n u t on b tt n

$$\mathsf{loc}\ \mathbf{u}_1\ \ A_1, \ldots, \mathbf{u_n}\ \ A_{\mathbf{n}}\ \{\!\!\{ ^{\mathsf{v}}/_{\mathsf{x}} \}\!\!\}$$

$$\mathsf{loc}\ \mathbf{u}_1\{\!\!\{ ^{\mathsf{v}}/_{\mathsf{x}} \}\!\!\}\ \ A_1\{\!\!\{ ^{\mathsf{v}}/_{\mathsf{x}} \}\!\!\}\ \ \ldots\ \ \mathsf{loc}\ \mathbf{u_n}\{\!\!\{ ^{\mathsf{v}}/_{\mathsf{x}} \}\!\!\}\ \ A_{\mathbf{n}}\{\!\!\{ ^{\mathsf{v}}/_{\mathsf{x}} \}\!\!\}$$

r turn a r ss t s t nt r s a pr an r turns t answ r at t pr r a r ss

$$\mathbf{s}[\![\ldots \mid \text{quest} \; \mathbf{x}, \mathbf{y}@\mathbf{z} \;\; \text{goto}\, \mathbf{z}.\mathbf{y} \; \mathit{spr} \; \mathbf{x} \;\; \text{ping} \; \mathbf{X} \; \mathbf{p} \ldots \;\; \text{kill} \; \mathbf{X} \; \mathbf{k} \ldots]\!]$$

H r t nt r s boun to **x** w t a r ss ons sts o two parts a ann boun to **y** at so *n no n* s t boun to **z**

A t p a nt r s n at **c** ta s t or

$$\mathbf{c}[\![\; \text{newc}\, \mathbf{r} \;\; \text{rw}\, \mathbf{bool} \;\; \text{goto}\, \mathbf{s}.\text{quest}\, \mathbf{v}, \mathbf{r}@\mathbf{c} \; \text{stop} \mid \mathbf{r} \;\; \mathbf{z} \ldots]\!]$$

H r a n w r turn ann **r** s n rat an a pro ss s s nt to t s rv **s** w t t nt r to b t st **v** an t r turn a r ss **r**@**c** anw ba at t nt t r su t s awa t on t o a ann **r**

t p o t s rv at t port **quest** not **p** abov ta s t or r **q** w r **q** s a tup t p rst o pon nt s **int** w t s on s a t p or a r ot ann at so *n no n* o at on t a t t at t o at on o t nt s un nown or arb trar a ows t s rv to b us b an nt t p **q** s v n b

$$\mathbf{int}, \; \text{w}\, \mathbf{bool} @\text{loc}$$

s n on t apab t to wr t a boo an s r qu r o t r ot ann □

E P

r v s p rsona s tr at nt t n w s t w a wa s r p to a ann at t s t me □

P ar nt r a s H r w onstrat t us u n ss o n w t p at or o *r st r n s* n s tt n up s ar nt r a s a on r nt s t s Cons r a s st o t or

newreg put rc p, get rc g Bserver | Client$_1$ | Client | ...

ons st n o a ban a ount s rv r Bserver an a nu b r o nts s st s w t n t s op o two r st r na s put an get r st r at sp t p s p an g on w w w not aborat s pa r o t p na s a s rv n or a as t nt r a or ban a ounts r at b t s rv r or t var ous nts An xa p s rv r wou ta t or

Bserver s request x int, y@z

newloc b L$_b$ with ...put, get... in

ounts an t s rv r wou r a n st r t s ar nt r a

Server newreg put rc p, get rc g
s[[request y@z
goto z.y put, get]]

H r on r pt o a r qu st t s rv r s p orwar s t two r st r na s put an get A t p a nt wou oo

Client me[[newc r r goto s.request r@me |
r y, z newloc b $L^{y,z}$ with ... o ... in ...]]

H r t nt n r spons to a r qu st r v s two r st r na s w ar boun to y an z an t n a n w ban a ount s s t up w t a arat on t p

$L^{y,z}$ loc y g, z p

ot t at t s a a n s a na t p w w b nstant at at run t A so t t p o t r p ann us b nts r s or *r st r n s* rat r t an *nn s* H r t a b put rc p, get rc g n t t s t at a ban a ounts stab s b nts w o us t s rv r w s ar t sa nt r a □

3.2 Type environments

A t p u nt w ta t or M w r s a *t p n ron nt* a st o assu pt ons about t t p s to

(PT)

$\overline{\quad\quad}$
□ env

( )
□ env
, u base □ env   u ∉

(N — CH N)
□ env
□ w loc
, u A@w □ env   u ∉
u ∉ A

(R — CH N)
□ env
□ w loc
□ u rc□B□, B < A
, u A@w □ env   u@w ∉ dom( )
u ∉ A

(RCH N)
□ env

x st at t o at on w but t a x st s w r t at s a onta n an asso at on u A′@w′ or so w′ r nt t an w But to ntro u su a na to b s ar a on var ous o at ons t ust a r a b ar as a r st r na an t an on b ntro u at w w t a subt p o ts ar t p s s t port o t pr s u rc B an t on t on B < A o n n ra o a ann na s a x st at r nt o at ons but a t r o a t p s ar ons st nt n t at t av t ar t p B as a ow r boun

a t p nv ron nts asso at t p s to rs but w ar so w at ax about t us o var ab s n t s t p s In pr n p su a t p a onta n var ab s w ar not nown to t nv ron nt It w turn out t at w w not b ab to t p s st s r at v to su nv ron nts

INITION N IRON NT O AIN For an nv ron nt s o p a or so $_1$ < □

ROPO ITION 11 **Let Envs be the set of all valid environments. Then the preorder Envs, < has partial meets.**

Proof: F rst not that **Envs** or r b **<** s n a pr or r but not a part a or r For xa p $_1$, not th nv ron nts

k loc, l loc an l loc, k loc

r sp t v th n $_1$ **<** an **<** $_1$ but th ar r nt nv ron nts

uppos th r s a va nv ron nt su h that **<** $_i$ or **i** , w s ow ow to onstru t a va nv ron nt $_1$ onstru t on s b n u t on on th s o I t s pt th n th r su t s obv ous $_1$ ts th rw s t s o th or ′, **u** an w a assu $_1$ ′ x sts h n $_1$ s onstru t b xt n n $_1$ ′ th pr s xt ns on p n s on **u** an I **u** dom $_1$ ′ th n th onstru t on v s $_1$ ′, **u** o t us assu that **u** dom $_1$ ′

- s loc h onstru t on v s $_1$ ′ ts
- s **base** ar
- s rcA H r th r ar two as s
  - I **u** rcB app ars n $_1$ ′ th n th r su t s obta n b r p a n that ntr w th **u** rcB

r su t o r ov n t at ntr n t onstru t on v s , u
rc A A , u A@w, u A@w′

av t r a r to t at t s onstru t on s orr t t at s

- $_1$ env
- $_1$ < $_i$ or i ,
- I < $_i$ or i , t n < $_1$ □

(T-RN) , n rc A M / new n rc A M

(T-CN) , n A@k M / new n A@k M

(T-N) k K M, k K $_{dec}$ k K / new k K M

(T-PAR) M, N / M | N

(T-THR) $_k$ P proc, k loc / k⟦P⟧

Typing Systems

---

an n r n ru s T THR In or r to nsur that k⟦P⟧ s a w t p s st w ust show that t t r a s w t p to run at k t p n o t r a s ust b r at v to a o at on b aus t a us o ann s t s ann s ust x st at k r s a so a subt t n t t p n o na r at on F rst not that n t s an a subs qu nt ru s w assu that a boun na s n a on us on o not app ar r n an assu pt ons us n T N w n onstru t n k K w now that k s a tua n w to so t v t t p asso at ons n k K ar s p app n to t os n r s a so an p t assu pt on that k K s a tua a w or nv ron nt How v r not that w av to that K s a prop r arat on t p that s w n to nsur that t on onta ns r st r r sour na s s s a v b an a t ona u nt on va u s

k K $_{dec}$ k K

t ru T C OC n F ur t s nsur s that a ann na s nsta at n w o at ons av a r a b n r st r

F na t t p n ru s or t u nts on t r a s

$_w$ P proc

ar v n n F ur an o w s ou b a ar ro t p n s s t s or t π a u us For xa p T IN sa s that to nsur t pro ss u X P s w t p r at v to to run at o at on w w ust nsur that

- u s a ann w t r a apab t o t appropr at t p at w that s u r @w
- t r s ua s w t p n t nv ron nt au nt b assu n t var ab s n t patt rn X av t t p s ass n to t b t

(T-OUTPUT)
$\Gamma \vdash_w$ **P** : **proc**
$\Gamma \vdash$ **V** : @**w**
$\Gamma \vdash$ **u** : w⟨ ⟩@**w**

$\Gamma \vdash_w$ **u**!⟨**V**⟩**P** : **proc**

(T-IN)
Γ, **X** : @**w** $\vdash_w$ **P** : **proc**
$\Gamma \vdash$ **u** : r⟨ ⟩@**w**

$\Gamma \vdash_w$ **u**?(**X**) **P** : **proc**

(T-GO)
$\Gamma \vdash$ **u** : loc
$\Gamma \vdash_u$ **P** : **proc**

$\Gamma \vdash_w$ goto **u.P** : **proc**

(T-STOP)
Γ : **env**

$\Gamma \vdash_w$ stop : **proc**

(T-L-NEW)
Γ, **k** : K $\vdash_w$ **P** : **proc**
Γ, **k** : K $\vdash_k$ **C** : **proc**
Γ, **k** : K $\vdash_{dec}$ **k** : K

$\Gamma \vdash_w$ newloc **k** : K with **C** in **P** : **proc**

(T-C-NEW)
Γ, **n** : A@**w** $\vdash_w$ **P** : **proc**

$\Gamma \vdash_w$ newc **n** : A **P** : **proc**

(T-MATCH)
$\Gamma \vdash$ **u**, **v**
$\Gamma \vdash_w$ **Q** : **proc**
Γ, **u** : @**w** $\sqcap$ **v** : @**w** $\vdash_w$ **P** : **proc**

$\Gamma \vdash_w$ if **u** = **v** then **P** else **Q** : **proc**

(T-R-NEW)
Γ, **n** : G $\vdash_w$ **P** : **proc**

$\Gamma \vdash_w$ newreg **n** : G **P** : **proc**

(T-PAR)
$\Gamma \vdash_w$ **P** : **proc**
$\Gamma \vdash_w$ **Q** : **proc**

$\Gamma \vdash_w$ **P** | **Q** : **proc**

(T-REP)
$\Gamma \vdash_w$ **P** : **proc**

$\Gamma \vdash_w$ ∗**P** : **proc**

Figure: Typing Threads

n o n t p t at s Γ, **X** : @**w** $\vdash_w$ **P** : **proc**

ru s T-OUTPUT T-STOP T-PAR an T-REP ar n or n t sa ann ro s ar ru s or t π a u us ru T-GO s a natura on or t p n t pro ss goto **u.P** an not t at t r qu r nts ar a tua n p n nt o t urr nt o at on **w** t r ru s ov rn n t n rat on o n w na s at t t r n s o t p s A, K an G s ou b s xp anator F na t ru T-MATCH s ot vat at n t n o w r t s ar u to b ss nt a n apab t bas t p s st s Br w n stab s n t at if **u** = **v** then **P** else **Q** s w t p w t r sp t to w n to nsur t at bot **P** an **Q** ar w t p How v r n t as o **P** w an ta a vanta o t a t t at t rs **u** an **v** ar n a t t sa Cons qu nt an t p n n or at on asso at w t t an b a a a at o w n on stab s t at **P** s w t p w t r sp t to t au nt nv ron nt □ □**u** □@**w** □ □**v** □@**w** r t t p o **u** s au nt b t at o **v** na w t at o **v** s au nt w t t t p o **u** In apab t bas t p n s st s t s s portant as t nab s us to p r o a a u u at apab t s asso at w t part u ar rs

3.4 Properties of the typing system

ar a n nt r st n stab s n ub t on r u t on but t s r qu r s a s r s o pr nar r su ts w w rst out n o t n abbr v at abbr v at t u nt $□_w$ **P** proc to $□_w$ **P** F rst two stan ar prop rt s on wou xp t

ROPO ITION 12

- **(Weakening) Suppose , ′ are two well-defined environments such that ′ < . Then □ M implies ′ □ M.**
- **(Strengthening) Suppose If , u □ M and u does not occur in the free identifiers of M. Then □ M.**

Proof: tan ar ot ow v r t at orr spon n r su ts ust b rst stab s or t t p n s st s or va u s an pro ss s □

n stan ar prop rt w o s **not** o s Int r an

$$_1, \mathbf{u}_1 \quad _1, \mathbf{u} \quad , □ \mathbf{M} \quad p \quad s \quad _1, \mathbf{u} \quad , \mathbf{u}_1 \quad _1, □ \mathbf{M}$$

b aus on an not arb trar sw t p p o p p ot o an s ar or pro ss s an va u s us w an r arran va n v ron nts us n t nt t s abov w t out an n t r us n t n r n o t p n u nts s u nts w b us n p a o Int r an

an u t n stab s n t ub t u t on r s s n s own t t r u t on ru R CO pr s rv s w t p n s a ounts to s own t at $\Box_k$ c V Q | c X R p s $\Box_k$ Q | R{V/x} an prov n

$$\Box_k \; R\{V/x\}$$

s t non tr v a part A t r so ana s s o t pr s w w av

$$\Box\; X \quad \Box @k\; \Box_k\; R \text{ an } \quad \Box\; V \quad @w$$

an t ubst tut on r su t s ou b su nt to n r ro

How v r r t notat on or t onstru t nv ron nt X @k s ons rab o p x t t t p a b an o t a ow trans ss on t p s or o a or non o a ann s or o at ons or or stru tur va u s A or n to a t proo s or transpar nt w w so at t part u ar as s an tr at so o t n v ua

ROPO ITION 1 OC CH NN U TITUTION **Suppose** v A@**w and** w$_1$ loc**. Then, if x does not appear in**

U , x A@w U @w$_1$ **implies** U{V/x} @w$_1$

ROC , x A@w $\Box_{w_1}$ R **implies** $\Box_{w_1}$ R{V/x}

Proof: rou out t proo w t α′ not α{V/x} or an appropr at s nta t ob t α

r su t or va u s s as stab s b n u t on on t n r n o t u nt , x A@w U @w$_1$ bas as s w n t ax o TN s us w r t ar u nt p n s on w t r U s t var ab x or not A ot r as s o ow stra t orwar b n u t on ot t at b aus o t r str t ons on t or at on ru s or w t p nv ron nts w now t at x an not app ar n t t p A

ar t r su t or pro ss s prov b n u t on on t n r n o , x A@w $\Box_{w_1}$ R an an ana s s o t ast ru us xa n two t p a as s

- uppos , x A@w $\Box_{w_1}$ u X R b aus

, x A@w u r @w$_1$ an

, x A@w X @w $\Box_{w_1}$ R

App n t rst r su t to w obta n

u′ r w$_1$

In b aus w loc t nv ron nt a b wr tt n as x A@w X @w w s qu va nt to X @w x A@w us a b r wr tt n as

X @w x A@w $_{w_1}$ R

H r w an app n u t on to obta n

” X @w $_{w_1}$ R′

ow t nput ru T IN an b app to an ” to obta n t r qu r $_{w_1}$ u′ X R′ ot t at our onv nt ons about boun var ab s nsur s t at u′ X R′ s t sa as u X R ′

- uppos ,x A@w $_{w_1}$ if u$_1$ u then P else Q b aus

,x A@w u$_1$,u

,x A@w $_{w_1}$ Q an

,x A@w u$_1$ @w$_1$ u @w$_1$ $_{w_1}$ P

App n t rst r su t to an n u t on to w obta n

u′$_1$,u′

$_{w_1}$ Q′

nv ron nt n an b r wr tt n to t qu va nt or

u$_1$ @w$_1$ u @w$_1$ x A@w

ar u nt now p n s on w t r u$_1$ or u or bot o n w t x As an xa p ons r t as w n u$_1$ s x an u s r nt H r w ust b t sa as w$_1$ an ust b a o a ann t p A′@w su t at A A′ x sts n t nv ron nt an b r wr tt n as

u @w x A A′@w

A so b aus v A@w w now v A′@w s w n an t r or b a n n w av

v A′@w u @w x A A′@w $_{w_1}$ P

But v A′@w v A A′ @w an so w app n u t on to to obta n

v A′@w u @w $_{w_1}$ P′

ow T TCH an b app to an to obta n $_{w_1}$ if u′$_1$ u then P′ else Q′.

□

n ortunat t subst tut on o o at ons r qu r s a or o p

ROC $_1$ x K $_w$ **R implies** $_1$ **v/x** $_{w\{v/x\}}$ **R{v/x}**

Proof: ot t at t pr v ous L a nsur s t at **v/x** s a w n nv ron nt rst r su t s prov b n u t on on w t s on s b n u t on on t n r n o t u nt $_1$ x K **U** @**w** w av t ta s to t r a r

r su t or pro ss s s b n u t on on t n r n o $_1$ x K $_w$ **R** an an ana s s o t ast ru us v on r pr s ntat v xa p

uppos $_1$ x K $_w$ **newloc l** L **with C in P** b aus

$_1$ x K **l** L $_l$ **C**

$_1$ x K **l** L $_w$ **P**

$_1$ x K **l** L $_{dec}$ **l** L

s n t asso at v t o w an r arran to t or

$_1$ x K **l** L $_l$ **C**

to w n u t on an b app to v

$_1$ **l** L **v/x** $_l$ **C{v/x}**

as r qu r □

substt tut on o r st r na s n s a or u at on s ar to t at o o at ons For xa p ons r an att pt to prov

, x rc A w newloc k loc x B with C in P

s w b r u to an att pt to prov

, x rc A , k loc, x B@k w P

w s not o t or

ROPO ITION 1 T R N U TITUTION Suppose 1 v rc A and x does not appear in 1. Then

N IRON NT , 1 x rc A env implies 1 v/x env

U , 1 x rc A U @w implies 1 v/x U{v/x} @w {v/x} P

□$_k$ **Q** an

□$_k$ **P**{**V**/**x**}

rst s as s n to o ow ro pot s s w t s on w o ow ro or w an stab s

a □**V** @**k** an

b □□**X** □@**w** □$_k$ **P**

pot s s p s p s □$_k$ **c** **X** **P** w ans b s sa s but a so t at □$_k$ **c** r□ □@**k** n ot r an t pot s s a so p s t at □$_k$ **c** □**V** □**Q** w ans t at □**V** @**k** or so t p su t at □**c** w□ □@**k** How v r ropos t on p s t at **<** an part v o t sa propos t on v s a an w a n s

R C – CR T uppos □**k**[newc **n** A **P**] o stab s t u nt □ new **n** A@**k** **k**[**P**] t s su nt b T C – N to prov

$$\textbf{,n} \;\; A@\textbf{k} \;\square_k\; \textbf{P}$$

But t on wa to stab s t pot s s s b t ru T CN n F ur or w w n **,n** A@**k** □ **k**[**P**] w an on b stab s ro T THR or w s n ssar □

s nar os n w nts ar v n s t v now o na a r at r sour s

1 L t K b t t p loc **a** A, **b** B Cons r t s st **M**

CONT T C O UR sa t at a now n x r at on ov r s s t s s *ont t*

| M R N an , ′ env p s , ′ | M R N

| M R N an O p s | M | O R N | O

n | M R N p s | new n M R new n N

ot t at n t s ast aus w av us an abbr v at on to ov r t t r r nt or s o na s w an b ar o a ann s r st r na s an o at ons a r nt at b t or w an ta or ov r w assu t at n s n w to rst aus a so on ta ns a subt t t s p s t at t qu va n s ou b pr s rv v n t us r nv nts so n w na s It wou b unr asonab to r wr t t s as

| M R N an ′ < w r ′ env p s ′ | M R N

s wou a ow t us r to nv nt *n* apab t s on r sour s t as r v ro t s st s un r nv st at on

AR PR R ATION For an v n o at on k an an v n ann a su t at k loc an a rw @k w wr t M barb a@k t r x sts so M′ su t at M →* M′ | k⟦a P⟧ sa t at a now n x r at on ov r s st s s *r pr s r n*

| M R N an M barb a@k p s N barb a@k

s t r prop rt s t r n our *to ston* qu va n

INITION UCTION AR CON RU NC t $\cong^{rbc}$ b t ar st now n x r at on ov r s st s w s

- po ntw s s tr t at s | M $\cong^{rbc}$ N p s | N $\cong^{rbc}$ M
- ont xtua
- r u t on os
- barb pr s rv n □

w now ara t r s $\cong^{rbc}$ us n a ab trans t on s st an b s u at on qu va n t r b ust n our part u ar not on o b s u at ons ot t at now n x r at ons n ra s t or usua

4.1 A labelled transition characterisation of contextual equivalence

ab trans t on s st w pr s nt n t s s t on s n or b r nt wor b two o t

w a app ars n w av us t s v rs on on to a nta n s tr w t t output as ot a so t at apr or t r s no r at ons p r qu r b tw n t t p at w t va u s s nt an t t p at w t w b us But t turns out t at n t ont xt n w t s ru s w b app s n t on b ow t att r w b a sup rt p o t or r

r a n n ru s ar a ar ro stan ar tr at nts o t p a u us w t t poss b x pt on o T w stat s t at or an nput trans t on t nv ron nt a nv nt r s na s n or r to t p t n o n va u

onstrat t at t trans t on ru s ar n a t w n n t s ns t at t or a b nar r at on b tw n s p urat ons

ROPO ITION **Suppose ▷ M is a simple configuration. If ▷ M $\xrightarrow{\mu}$**

[redacted] to th r w th abov th n v s ar th r qu r h pot ss s n th [redacted] to obta n th r qu r □

[redacted]at n t p a t ons ▷ **M** $\xrightarrow{\mu}$ ′ ▷ **N** p t t r n b an **μ** a t ons as s p r str t ons on a [redacted]P L t us wr t

[redacted] $\xrightarrow{\mu}$ **N**

[redacted] ′ us n a var at on on th ru s ro [redacted]stra nts on ar not n or [redacted]n ss o n w na s ar st [redacted]

[redacted] $\xrightarrow{\mu}$ ′ ▷ **N** [redacted]

[redacted] th a [redacted]

[redacted] nt o s t [redacted]

[redacted] att r stat nt or [redacted]

[redacted]ITION . **Suppose** ▷ **M** [redacted]

- ▷ **M** $\xrightarrow{\tau}$ ′ ▷ **N** **if and only** [redacted]
- ▷ **M** $\xrightarrow{(\tilde{n})k.a\,V}$ ′ ▷ **N** **if and** [redacted]
  - **k** loc
  - **a** r @**k occurs in , for so**[redacted]
- ▷ **M** $\xrightarrow{(\tilde{n}\,\tilde{T})k.a?V}$ ′ ▷ **N** **if a**[redacted]
  - **k** loc
  - **a** w @**k, for some typ**[redacted]

h **V**o @**k**

H r w ar us n t stan ar notat on ro ans $\xrightarrow{\tau}{}^{*} \circ \xrightarrow{\mu} \circ \xrightarrow{\tau}{}^{*}$ w s $\xrightarrow{\tau}{}^{*}$ μ s τ an ot rw s t s a ows a s n nt rna ov to b at b ro or ov nt rna ov s

wr t | $M \approx^{bis} N$ $\triangleright M$ R $\triangleright N$ or so b s u at on R an sa t at M an N ar b s ar n t nv ron nt □

ot t at t r at on $\approx^{bis}$ or s a now n x r at on ov r s s t s b ons r n as a para t r to t r at on or ov r t sa s s a o t prop rt s n n t on . As an xa p w w prov t at $\approx^{bis}$ s ont xtua o own t r as w b p u n stab s n t s

If | $M \approx^{bis} N$ and $<$ ′, where dom dom ′**, then** ′ **|** $M \approx^{bis} N$**.**

Proof: tra t orwar o n u t on □

n xt a nsur s t at w n n w va u s ar xtru to t n v ron nt t t p s at w t b o nown ar sup rt p s o t t p at w t w r ar b t s st

If $\triangleright M \xrightarrow{(\tilde{n})k.c\,V}$ ′ $\triangleright M'$ **then** $M \equiv$ new **n** M'' **such that if** ′ **n then** $<$ **.**

Proof:

or ov r t s o pon nts an b r o pos to or a a n t o nt a t on r su ts p n on t a t t at t s st s part o a s p on urat on

CO PO ITION . CO PO ITION

(i) (a) If $\triangleright$ M $\xrightarrow{(\tilde{n})k.c\,V}$ $' \triangleright$ M$'$ and O $\xrightarrow{k.c?V}$ O$'$ then $\triangleright$ M | O $\triangleright$. new n M$'$ | O$'$ for some

(b) If $\triangleright$ M $\xrightarrow{(\tilde{n}\,\tilde{T})k.c?V}$ $' \triangleright$ M$'$ and O $\xrightarrow{(\tilde{n})k.c\,V}$ O$'$ then $\triangleright$ M | O $\triangleright$. new n M$'$ | O$'$

(ii) If $\triangleright$ M | O $\xrightarrow{\tau}$ $\triangleright$ M$'$ and O then one of the following hold

(a) $\triangleright$ M $\xrightarrow{\tau}$ $\triangleright$ M$''$ such that M$'$ ≡ M$''$ | O

(b) O $\longrightarrow$ O$'$ such that M$'$ ≡ M | O$'$

(c) $\triangleright$ M $\xrightarrow{(\tilde{n})k.c\,V}$ $' \triangleright$ M$''$ and O $\xrightarrow{k.c?V}$ O$'$ such that M$'$ ≡ . new n M$''$ | O$'$ for some

(d) $\triangleright$ M $\xrightarrow{(\tilde{n}\,\tilde{T})k.c?V}$ $' \triangleright$ M$''$ and O $\xrightarrow{(\tilde{n})k.c\,V}$ O$'$ such that M$'$ ≡ . new n M$''$ | O$'$

Proof: art s r at v stra t orwar on s ow t rst as as t ot r s s ar an pro b n u t on on t nu b r o **τ** a t ons n t r vat on ro t s st For t n u t v as t s o ows as b t n u t v t pot s s an t a t t at | an . new ar va uat on ont xts ons r t bas as n w $\triangleright$ M $\xrightarrow{(\tilde{n})k.c\,V}$ $' \triangleright$ M$'$

B ropos t on p w s t at M $\xrightarrow{(\tilde{n})k.c\,V}$ M$'$ B nsp t n t trans t on ru s w not t at t o own stru tura or s ust o

- M ≡ . new n . new m$'$ $'$. k⟦c V P⟧ | M$''$
- M$'$ ≡ . new m$'$ $'$. k⟦P⟧ | M$''$
- O ≡ . new n$'$ $'$. k⟦c . X Q M$'$ ≡ . new M0 M$'$ One ecti

- A is a subterm of M. In which case O does not contribute to the transition and (a) holds.
- A is a subterm of O. In which case M does not contribute to the transition and (b) holds.
- A is not a subterm of M or O. In which case by inspection of the rules we see that the only possibility is that A must be an instance of rule (R-COM). Let us suppose that A is of the form

$$k[\![c!\langle V\rangle P]\!] \mid k[\![c?(X{:}\ldots)Q]\!] \to k[\![P]\!] \mid k[\![Q\{\!|V/x|\!\}]\!]$$

There are two ways in which this could occur: either M provides the output action, say $\xrightarrow{(\tilde n)k.c!V}$ and N the corresponding input, in which case (c) holds, or vice versa, and (d) holds. We concentrate on the former case; the latter can be dealt with in a similar way. We know that it must be the case that, up to structural equivalence,

$$M \equiv (\text{new } n{:}\ldots)(\text{new } m'{:}\ldots)(k[\![c!\langle V\rangle P]\!] \mid M''')$$

$$O \equiv (\text{new } m{:}\ldots)(k[\![c?(X{:}\ldots)Q]\!] \mid O'')$$

such that k and c are not in n, m', m. Let M'' be the term

$$(\text{new } m'{:}\ldots)(k[\![P]\!] \mid M''')$$

and O' be

$$(\text{new } m{:}\ldots)(k[\![Q\{\!|V/x|\!\}]\!] \mid O'').$$

It is clear that $M' \equiv (\text{new } n{:}\ldots)(M'' \mid O')$ so it suffices to demonstrate that $O \xrightarrow{k.c?V} O'$ and $\ldots \rhd M \xrightarrow{(\tilde n)k.c!V} \ldots' \rhd M''$ for some $\ldots'$ such that $\ldots <: \ldots', n$, a fact which [illegible] from the transition rules for …

now that dom $'_0$ s s o nt ro dom $^-$ an that $^- \triangleright$ **M** s a s p on urat on so t ust b t as that $^-$ **k** loc an $^-$ onta ns **c** r **k** a so B ropos t on p a a n w s that $^- \triangleright$ **M** $\xrightarrow{\mu}$ $^-$ **V** **k** $\triangleright$ **M**$'$ B n t on o **R** w now that th r ust x st so

$$^- \triangleright \mathbf{N} \quad ^- \quad \mathbf{V} \quad \mathbf{k} \triangleright \mathbf{N}'$$

su h that

$$^- \quad \mathbf{V} \quad \mathbf{k} \,|\, \mathbf{M}' \approx^{bis} \mathbf{N}'.$$

h s an ropos t on p t s us that $^+ \triangleright$ **N** $^+_1 \triangleright$ **N**$'$ w th $^+_1 \triangleright$ **M**$'$ **R** $^+_1 \triangleright$ **N**$'$ as r qu r

h as n wh h **μ** s an nput trans t on an b tr at s ar but us n th th r ru n th ra ar or xt n n nv ron nts □

ROPO ITION 11 $\triangleright$new

$$\mathbf{n} \,|\, \mathbf{M} \approx^{bis} \mathbf{N} \text{ implies } | \text{ new } \mathbf{n} \quad \mathbf{M} \approx^{bis} \text{ new } \mathbf{n} \quad \mathbf{N}.$$

Proof: In a t u to L a t su s to show

$$\mathbf{n} \,|\, \mathbf{M} \approx^{bis} \mathbf{N} \quad p \quad s \quad | \text{ new } \mathbf{n} \quad \mathbf{M} \approx^{bis} \text{ new } \mathbf{n} \quad \mathbf{N}.$$

pro b n n a r at on **R** wh h onta ns $\approx^{bis}$ an r at s $\triangleright$ new **n** **M** an $\triangleright$ new **n** **N** wh n v r **, n** | **M** $\approx^{bis}$ **N** show that **R** or s a b s uat on

a an two on urat ons r at b **R** th s ar b s ar th n w an b sur that **R** sat s s th n ssar osur prop rt s hus w an assu that w hav hos n on urat ons o th or $\triangleright$i6.28008 Td (.)Tj -455.d

Proof: o t s b nn a r at on **R** su t at

▷ new $\mathbf{n}_0$ $_1$ **M | O** **R** ▷ new $\mathbf{n}_0$ **N | O**

an on t r x sts so ′, su t at a o t o own o

- ′ <
- $_1$ <
- <
- ′ $\mathbf{n}_0$ | **M** $\approx^{bis}$ **N**
- ′ $\mathbf{n}_0$ **O**

ust s ow t at **R** or s a b s u at on For t purpos s o xpos t on w w assu t at $\mathbf{n}_0$ s pt or n ra as o ows n a s ar ann r

a ▷ **M | O** **R** ▷ **N | O** w tn ss b ′ | **M** $\approx^{bis}$ **N** an ′ **O** an suppos t at ▷ **M | O** $\xrightarrow{\mu}$ $_0$ ▷ **M**′ I **μ** s not a **τ** a t on t n t ar r v s nt r ro **M** or **O** In t r as a at n **μ** trans t on an b oun ro **N** b aus ′ | **M** $\approx^{bis}$ **N** an ′ < uppos t n t at **μ** s a **τ** a t on so t at $_0$ s us L a art to obs rv t at on o our as s o

a ′ ▷ **M** $\xrightarrow{\tau}$ ′ ▷ **M**″ A a n at n trans t ons ar as oun b aus ′ | **M** $\approx^{bis}$ **N**

b **O** $\xrightarrow{\tau}$ **O**′ But t n ▷ **N | O** $\xrightarrow{\tau}$ ▷ **N | O**′ an b ub t u t on or w now t at ′ **O**′ a so so ▷ **M | O**′ **R** ▷ **N | O**′ as r qu r

′ ▷ **M** $\xrightarrow{(\tilde{n})k.c\,V}$ ″ ▷ **M**″ an **O** $\xrightarrow{k.c?V}$ **O**′ su t at **M**′ ≡ new **n** $_1$ **M**″| **O**′ or so $_1$ not t at t r ust x st so ▷ **N** $(\tilde{n})$**k.c** ″ ▷ **N**′ su t at ″ | **M**″ $\approx^{bis}$ **N**′ an or ov r b L a art w s t at ▷ **N | O** ▷ new **n** **N**′ | **O**′ or so But w now t at ″ s ′ **V** **k** an t at **n** ar a onta n n **V** so ″ s n ssar o t or $'_0$ **n** or so $'_0$ < ′ trans t v $'_0$ < now b L a t at $_1$ < an < In part u ar w av

$'_0$

an

$$O' \equiv \text{new}\, m \quad k[\![Q\{\!|V/x|\!\}]\!] \mid O''$$

w t k, c not n m B nsp t n t t p n ru s w s t at

$$' \; c \quad r \quad @k$$

an

$$' \; V \quad @k \; m \quad X \quad @k \;_k Q.$$

or r t s us t at < b aus w now onta ns c r @k an t att r a on w t t a t t at

$$' \; V \quad @k \; m \quad V \quad @k$$

an or t s us t at ′ V @k ′₀ n O′ so w an on u

$$\triangleright \text{new}\, n \quad {}_1 \quad M' \mid O' \quad R \quad \triangleright \text{new}\, n \quad N' \mid O'$$

as r qu r

$$\triangleright M \quad (\tilde{n} \; \tilde{T})$$

an on **C M** $_{\alpha,}$ □ **D N** w r **D** – s a anon a ont xt ro w bot **N** an ′ ar n so s ns *r o r*

or a proo an b r ov r as an nstan o t or o p at or p an s t r or o tt □

5 Controlling mobility

now ons r a r r a u us n w ov nt o pro ss s a b ontro As xp a n n t Intro u t on n PI an pro ss w s n poss ss on o t na o a o at on a trav to t at p a an b n x ut n arb trar o t r xt n PI w t a v r s p ans o ob t ontro an nv st at t r su t n ont xtua qu va n

5.1 Migration rights

H nn ss an av a r a propos a s p a ss ontro an s or PI n t or o t *o* apab t o an r w xt n t s a to a ow so w at or x b t

o at on t p s n PI ar o t or

$$\mathsf{loc}\ \mathbf{u}_1 \quad A_1, \ldots, \mathbf{u_n} \quad A_\mathbf{n}$$

w r t **u**$_\mathbf{i}$ A$_\mathbf{i}$ an b s n as apab t s at t at o at on ntro u an xtra t p o apab t now b a ow n o at on t p s to b a so o t or

$$\mathsf{loc}\ \mathsf{moves}, \mathbf{a}_1 \quad A_1, \ldots, \mathbf{a_n} \quad A_\mathbf{n}$$

w r **S** s a s t o rs I a o at on **k** s nown at t s t p t n

ta s ar stra t orwar

- r n t apab t s n F ur to r a

$$\text{Capab t s} \quad \mathbf{u} \; A \mid \mathbf{move_u}$$

- p nv ron nts an now a so n u ntr s o t or **u** $\mathbf{move_w}$ a ru s to t t p u nts or nv ron nts an va u s a or n s F ur
- F na w an t t p n r n o t rat on pr t v b r p a n t ru T O ro F ur w t

$$\frac{\begin{array}{c}\mathbf{u}\ \ \text{loc}\ \mathbf{move_w}\\ {}_u\ \mathbf{P}\ \ \mathbf{proc}\end{array}}{{}_w\ \text{goto}\ \mathbf{u.P}}\ (\text{T- O - O})$$

a no an to t r u t on s ant s nor t n t on o ont xtua qu va n or t an ua It s stra t orwar to t at or ub t u t on a so o s or t s xt n a u us t w nab us to onstrat t subt t nvo v n v op n b av oura qu va n s n t pr s n o ontro ob t ons r t sub an ua n w on t *n* *rs* ov apab t **move**$_*$ w r $_*$ s a w ar s a ow t s apab t rants rat on r ts to *r* *t* us n an nv ron nt onta n n

$$\mathsf{l}\ \ \mathsf{loc}\ \mathbf{move}_*,\ \ \mathsf{u}_1\ \ \mathrm{A}_1, \ldots$$

$$\mathsf{k}\ \ \mathsf{loc}\ \mathsf{u}_1\ \ \mathrm{A}_1, \ldots$$

a s t s av a ss to **l** w no s t s av a ss to **k**

For t s r str t an ua w v n t o ow n two subs t ons two r nt n ra sat ons to t u abstra t on r su t or

k⟦stop⟧ r sp t v an suppos s su t at **k** loc **move**$_*$ n
| **N**$_1$ $\approx^m_{bis}$ **N** b aus no t p a t ons ar poss b ro t s s s
t s □

H r t **N** , **N** r pr s nt t s st s

new **k** loc **move**$_*$, **b** rw **l**⟦**a** **k**⟧ | **k**⟦**b**⟧ an

new **k** loc **move**$_*$, **b** rw **l**⟦**a** **k**⟧ | **k**⟦**0**⟧

r sp t v an t $_1$ not t nv ron nt

l loc, **l** **move**$_*$, **b** rc rw , **a** rw loc

ot t at r w st on a ow barbs at o at ons to w w av rat on r ts s ou b n ra s to a so a ow barbs at o at ons n **T** But t wou not an t qu va n as t s o a barbs an a wa s b r p a b barbs at pr n o at ons w t nv ron nt ar s w t rat on r ts

qu st on now s w t r w an v s a b s u at on bas ar a t r sat on o $\Box^{\mathcal{T}}_{rbc}$

obv ous approa s to o t n t ons o t t p a t ons $\xrightarrow{\mu}_{m}$ to obta n a t ons $\xrightarrow{\mu}_{\mathcal{T}}$ w a ow obs rvat ons at a s t **k** t r t nv ron nt as rat on r ts to **k** as b or **or k** **T** t t s a t ons w an o n t on to obta n a n w b av oura qu va n w w not b $\approx^{\mathcal{T}}_{bis}$ n ortunat t s o s not o n w t t ont xtua qu va n $\Box^{\mathcal{T}}_{rbc}$

L t **N** , **N** b t s st s n b

h⟦**a** **b**@**k**⟧ | **k**⟦**b** ⟧ an **h**⟦**a** **b**@**k**⟧ | **k**⟦stop⟧

an t nv ron nt

h loc, **h** move$_*$, **k** loc, **a** rw @**h**

n **k** s n **T** on an t at **N** $\approxeq^{\mathcal{T}}_{bis}$ **N** s s b aus $\triangleright$**N** an p r or t a t on **h.a** **b**@**k** o ow b **k.b** w an not b at b $\triangleright$ **N**

How v r | **N** $\Box^{\mathcal{T}}_{rbc}$ **N** b aus t s not poss b n a ont xt to st n u s b tw n t A ont xt an b oun to au nt t now o t nv ron nt at **h** w t t a t t at **b** x sts at **k** But t s not poss b to trans r t s n or at on ro **h** to w r t an b put to us na **k** □

s xa p onstrat s t at v n w t our v r r str t ov apab t t r ar prob s w t t ow o n or at on Know about t s st arnt at **l** an not n ssar b pass to **k** t

- T {k1, . . . , kn}
- < ki or a ≤ i ≤ n

so t s us k0 to not rst o pon nt o t stru tur

- A on r t on ▷ M ov r T ons sts o an nv ron nt stru tur an a s st M su t at t r x sts so nv ron nt w t
  - M
  - <
  - dom dom □

w wr t to an t a o nv ron nts , k1, . . . , kn su t at a o pon nt ki s qua to t nv ron nt w w t p a o t t para t r T r as t an usua b r ov r ro ont xt un rstan , an

ROPO ITION 11

- If $\overline{} \triangleright M$ is a configuration and $\overline{} \triangleright M \xrightarrow{\alpha} \overline{}' \triangleright M'$ then $\overline{}' \triangleright M'$ is also a configuration.
- For every $\overline{}$ and every action α there exists a unique structure $\overline{}$ after α with the property that $\overline{} \triangleright M \xrightarrow{\alpha} \overline{}' \triangleright M'$ implies $\overline{}'$ is $\overline{}$ after α.

Proof: ar to t at o ropos t on . . □

vo ut on ro $\overline{}$ to $\overline{}$ after α nvo v s two st n t n s o n r as

Proof: tra t orwar unrav n o t n t ons ☐

o now w an on ntrat on r at n t r at on

For notat ona onv n n b ow w us —′ as an abbr v at on or — after **α**

- I **α** s **m k.a v** an **k** loc **move**$_*$ t n **C**$_{\alpha}$

  k$_0$⟦goto **k.a X.**if **X** new **m v** then goto **k**$_0$. **r**$_0$ **v**

- The action contexts for outputs receive a value v and test its identity against all known identifiers. In Figure 13 this testing is expressed using the notation X new m v , which is defined by

X n if v n □ m
X □ if v m
X_1 new

Proof:

How v r t poss b at n r u t ons ar onstra n b t barbs o $\mathbf{M}_0$ n t xt n nv ron nt t as t barb $\boldsymbol{\delta}_{\mathrm{succ}}@\mathbf{k}_0$ but t o s not av $\boldsymbol{\delta}_{\mathrm{fail}}@\mathbf{k}_0$ E t v t r u t on ust av t or

D

ob t ontro pr s nt r s not nt n to b a n t v tr at nt rat r rst st p towar s nt n t natur o ont xtua qu va n n t s s tt n A ar pro r ss on o t s wor t n wou b to ntro u a o n ra n ob t ontro ans nto or s ar an to a apt t as pr s nt r to un rstan ont xtua qu va n In anot r v n w an nv st at ow t para t r ts qu va n us w a o t r s to a ow t st n at an n t a nown o at on At t ot r xtr w ou x to b p s wou on a ow t sts to b p a at *r s* o at ons t r b an n t natur o obs rvab t an s p n t s ant s ons rab s a b t appropr at o or t st n qu va n s

r as b n a r at a o nt r st n o n str but s st s us n a u n r nt ars p as s so ar as ar b n on s n o t an ua s to v su n t s r pt ons o ob pro ss s w t t p s st s v n to onstra n b av our n a sa ann r r qu va n as b n us t as t p a b n ntro u as so sort o ont xtua qu va n v r s ar to t on oun n t pr s nt pap r roo s o orr tn ss o proto o s or an ua trans at ons av b n arr out w t r sp t to t s ont xtua qu v a n s nt n p a or o b s u at on as b n su st as a proo t o or stab s n ont xtua qu va n n t a a u us But as ar as w now t on x st n xa p o an op rat ona ar a t r sat on o b av oura qu va n n t str but s tt n s oun n us D to not t o ow n st o nv ron nts

$$\mathsf{k}_0 \ \mathsf{loc}, \ \mathbf{move}@\mathsf{k}_0, \ \mathsf{r}_0 \ \mathsf{rw} \ \mathsf{k}_0 \ @\mathsf{k}_0$$
$$\mathsf{r}_1 \ \mathsf{rw} \ \mathsf{k}_1 \ @\mathsf{k}_1, \ldots, \mathsf{r_n} \ \mathsf{rw} \ \mathsf{k}_n \ @\mathsf{k_n}$$

A 1 G N R TRU ION **Let**

[ ] Luca Cardelli and Andrew D. Gordon. Mobile ambients. *Theoretical Computer Science*, 0(1). 1 – 1 , June 000.

[ ] G. Castagna and F. Zappa. The seal calculus revisited. In *22th Conference on the Foundations of Software Technology and Theoretical Computer Science*. pringer-Verlag, 00 . to appear.

[ ] Cédric Fournet, Georges Gonthier, Jean-Jacques Lévy, Luc Maranget, and Didier Rémy. A calculus of mobile agents. In *7th International Conference on Concurrency Theory (CONCUR'96)*, pages 0 – 1, Pisa, Italy, August - 1 . Springer-Verlag. LNCS 111 .

[ ] M. Hennessy. *Algebraic Theory of Processes*. The MIT Press, Cambridge, Mass., 1 .

[ ] M. Hennessy and J. Rathke. Typed behavioural equivalences for processes in the presence of subtyping. In *Proc. CATS2002, Computing: Australasian Theory Symposium, Melbourne 2002*, 00 . Also available as a University of Sussex technical report.

[ ] Matthew Hennessy and James Riely. Resource access control in systems of mobile agents. *Information and Computation*, 1 . –1 0, 00 .

[10] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 1 ( ). – , 1 .

[11] M. Merro, J. Kleist, and U. Nestmann. Mobile Objects as Mobile Processes. *To appear in Journal of Information and Computation*, 00 .

[1] Massimo Merro and Matthew Hennessy. Bisimulation congruences in safe ambients. *ACM SIGPLAN Notices*, 1(1). 1– 0, January 00 .

[1] R. Milner. *Communication and Concurrency*. Prentice Hall, 1 .

[1] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, (Parts I and II). *Information and Computation*, 100. 1– , 1 .

[1] Benjamin Pierce and Davide Sangiorgi. Typing and subtyping for mobile processes. *Mathematical Structures in Computer Science*, ( ). 0 – , 1 . Extended abstract in LICS ' .

[1] Peter Sewell. Global/local subtyping and capability inference for a distributed pi-calculus. In *ICALP 98*, volume 1 of *LNCS*. Springer, 1 .

[1] Asis Unyapoth and Peter Sewell. Nomadic pict. Correct communication infrastructure for mobile computation. In *Conference Record of POPL'01: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, . f– ,(–)]gg 0.0 (Sed(Cos f– m')]TJA 1 1(. 0 (in)K. c eK (infere (Symp) .