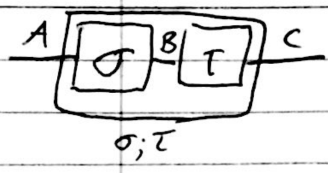
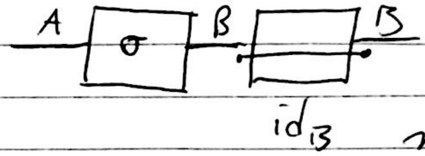
Ghica 3

Composition = sync + replication + hiding



- associative σ, τ, ν
- identity?



mirroring
copy-cat
strategies

(done in chess to play against grandmaster)

$K_A : A_0 \to A_1$

$inl(a) = m_0$
$inr(a) = m_1$

- define via a next-move function $\hat{K}_A$

$$?_1^{\,a\langle b\rangle} \mapsto ?_0^{\,b\langle c\rangle}$$

$A_0 \to A_1$
$?^P \to ?^O$

$$n_j^O \, d\langle e\rangle \mapsto n_{1-j}^P \, ?\langle f\rangle$$

p move needs to be justified by O move

$A_0 \to A_1$

$\vdots$

m^O

m^P

$\vdots$

n^O

n^P

seems natural that points to moves that were copied

$$p \cdot m_{1-j}\, a\langle b\rangle \cdot m_j\, c\langle d\rangle \cdot p' \cdot n_j^O\, d\langle e\rangle \mapsto n^P_{1-j}\, b\langle f\rangle$$

<u>Lemma</u> $\kappa_A : A_1 \to A_2$

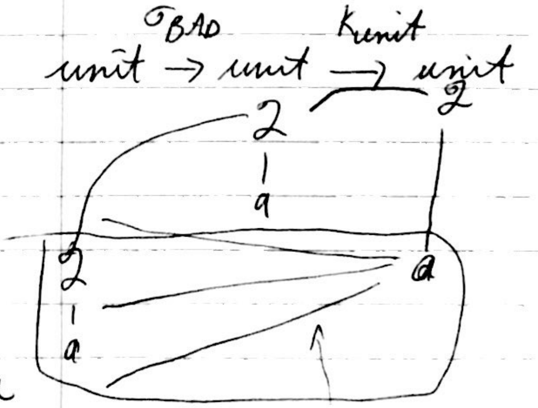
$\kappa_A \restriction A_1 = \kappa_A \restriction A_2$

<u>Lemma</u> κ_A is a strategy

<u>Theorem</u>? $\kappa_A ; \sigma = \sigma ; \kappa_B = \sigma$?

Not true!

<u>Counterexample</u>



a bit like "fork" in unix

no guarantee of order of these interaction

a could happen in any of these 3 orders

How to fix identity

A) Discipline plays with extra conditions

(rule out σ_{BAD} for example)

B) Add closure conditions on strategies

Careful - bec. language specific

concurrency → asynchronous games

Extra condition(s) because of "static concurrency"

$C_1 \| C_2$ fork / join

[don't allow fork ("dynamic unrestricted concurrency")]

Fork/join: If a $\overbrace{\text{thread}}^{\text{Question}}$ $\overbrace{\text{finishes}}^{\text{is answered}}$ then

all its $\underbrace{\text{sub-threads}}_{\text{justified Question}}$ must have finished / been answered

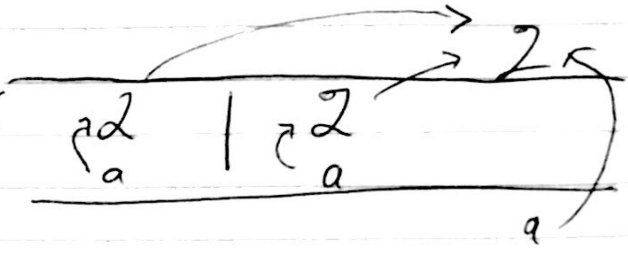
$(C_1 \| C_2); C_3$

both finished before C_3

(every instance of enabling is justification)

Typical strategy $\sigma_{par} : com_1 \to com_2 \to com$

can happen in any order hence bar between them



justification pointers

break up into two conditions

<u>Def</u> Strict scoping — think of ending scope of m

$$p \cdot m\, a\langle - \rangle \cdot p' \in P$$

and $m \in A$ then $a \notin p'$

<u>Def</u> Strict nesting

$$p_1 \cdot m_1\, a\langle b\rangle \cdot p_2 \cdot m_2\, b\langle c\rangle \cdot p_3 \cdot n\, b\langle d\rangle \in P$$

and $m_1 \in A$ then $\exists n_2 \in A$ s.t. $n_2\, c\langle - \rangle \in p_3$

<u>Thm</u> $A \xrightarrow{\sigma} B \xrightarrow{\tau} C$ have SN, SS plays (strict nested, strict scoped)

then $\sigma ; \tau$ has SN, SS plays

When can two moves synchronize in a play?

$$\begin{tikzcd} m_1 & m_2 \arrow[l, bend right] \end{tikzcd}$$

(means m_2 <u>caused</u> by m_1)

$O_{move} \cdot P_{move}$

~~(p p~~

<u>Def asynchronous strategy</u> $\sigma : A$

of strongly nested, strongly scoped plays

$$p \cdot \underbrace{m_0\ a_0 \langle b_0 \rangle}_{p_0} \cdot \underbrace{m_1\ a_1 \langle b_1 \rangle}_{p_1} \in \sigma$$

and $m_0 \in P_A$ or $m_1 \in O_A$

and $p \cdot p_1 \cdot p_0 \in P_A$

then $p \cdot p_1 \cdot p_0 \in \sigma$

$id_A = strat(\kappa_A)$ (least strategy that respects closure conditions)

Concurrent idealized Algol

PCF (ignore recursion)

+

local state

newvar $\qquad$ new x in M $\equiv$ newvar$(\lambda x.\, M)$

assign

dereferencing

+

concurrency

par

+

binary sems

new sem $\qquad$ ~~sem~~ x in M $\equiv$ newsem$(\lambda x.\, M)$

grab

release

$$\Gamma \vdash M : \Theta$$

"

$$x_0 : \Theta_0, \dots, x_k : \Theta_k$$

Challenge $\quad c : \mathrm{com} \vdash \mathrm{new}\ x = 0\ \mathrm{in}\ c ; !x \equiv c ; 0$

alternatively

$$\lambda c.\ \mathrm{new}\ x = 0\ \mathrm{in}\ c ; !x \equiv \lambda c.\ c ; 0$$

before game semantics, needed very sophisticated operational reasoning on parametricity

- $\lambda f.\ \mathrm{new}\ x = 0\ \mathrm{in}\ f(x := !x+1)(!x) \equiv \lambda f.\ \mathrm{new}\ y = 0\ \mathrm{in}\ f(x := !x-1)(-!x)$

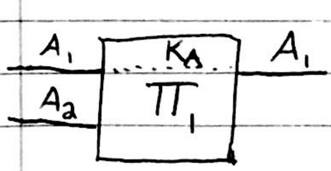
(something like two modules are equivalent)

A model of λ-calculus $\quad$ Cartesian Closed Category

- Product $\quad A \times B$
- have a unit s.t. $I \times A \cong A \times I \cong A$

  (empty arena)

- projections $\quad \pi_i : A_1 \times A_2 \to A_i \quad i = 1, 2$



$\pi_i = \text{least strat}(k_i)$

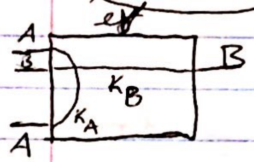
- pairing $\quad \langle \sigma, \tau \rangle : A_1 \times A_2 \to B$

  $\sigma : A_1 \to B \qquad \langle \sigma, \tau \rangle = \sigma \cup \tau$

  $\tau : A_2 \to B$

- exponential arena $A \to B$
- transpose $\dfrac{\sigma : A \times B \to C}{\lambda\sigma : A \to B \to C}$
- evaluation strategy

$$\overset{ev}{\cancel{e\sigma}} : (A \to B) \times A \to B$$



$$ev = strat\,(\kappa_A \cup \kappa_B)$$

Concurrent Idealised algol (general recipe)

$$[\![Nat]\!] = Nat$$

$$[\![Var]\!] = Var$$

$$[\![\theta \to \theta']\!] = [\![\theta]\!] \to [\![\theta']\!]$$

$$[\![\Gamma]\!] = [\![\theta_0]\!] \times \dots \times [\![\theta_l]\!]$$

$$[\![\Gamma, x_k : \theta_k, \Gamma' \vdash x_k : \theta_k]\!] = \pi_k = strat(\kappa_{[\![\theta_k]\!]})$$

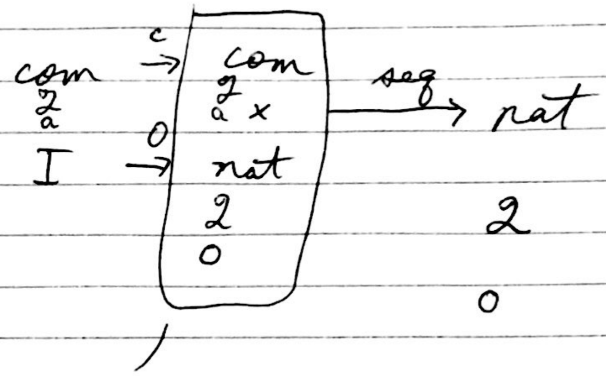
$$[\![\Gamma \vdash \lambda x. M : \theta \to \theta']\!] = \lambda [\![\Gamma, x : \theta \vdash M : \theta']\!]$$

$$[\![\Gamma \vdash MN : \theta]\!] = \langle [\![M : \theta' \to \theta]\!], [\![N : \theta']\!] \rangle ; ev$$

$$= ([\![M]\!] \cup [\![N]\!]) ; strat\,(\kappa_\theta \cup \kappa_{\theta'})$$

$$[\![\vdash M_0 : \theta_0]\!] = \sigma_{M_0}$$

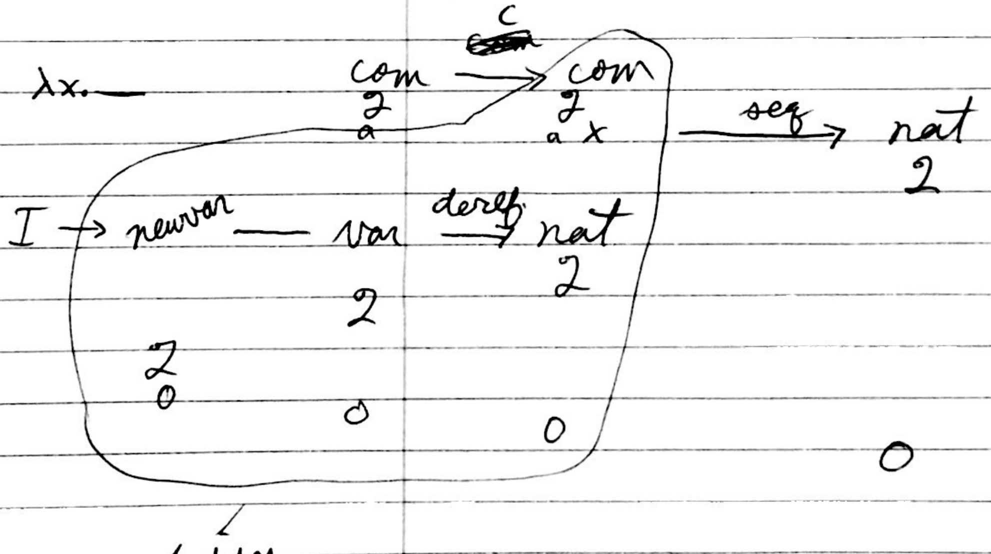
$[\![\lambda c. c ; 0]\!] = \lambda [\![c ; 0]\!]$



hidden due
to sequential composition

$\lambda \; strat(? \; ?_c \; a_c \; 0)$

$\lambda \; [\![new \; x = 0 \; in \; c ; \; !x]\!]$



hidden

$= strat(? \cdot ?_c \cdot a_c \cdot 0)$