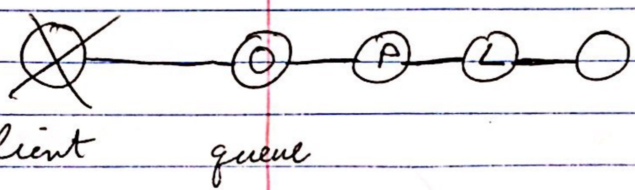
Balzer 2

$$\frac{\Gamma \vdash C}{\Gamma, A \vdash C}\ \text{Weakening} \qquad \frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C}\ \text{Contraction}$$

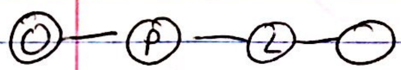
in a process calculi, what do they mean?

can think of weakening as "dropping a resource"
(don't care about cleaning up)

contraction - "duplicating a resource"

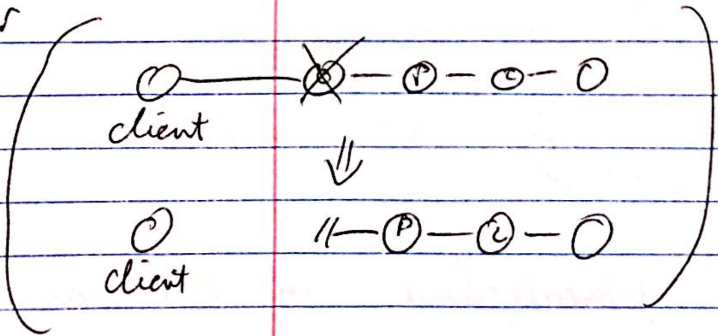


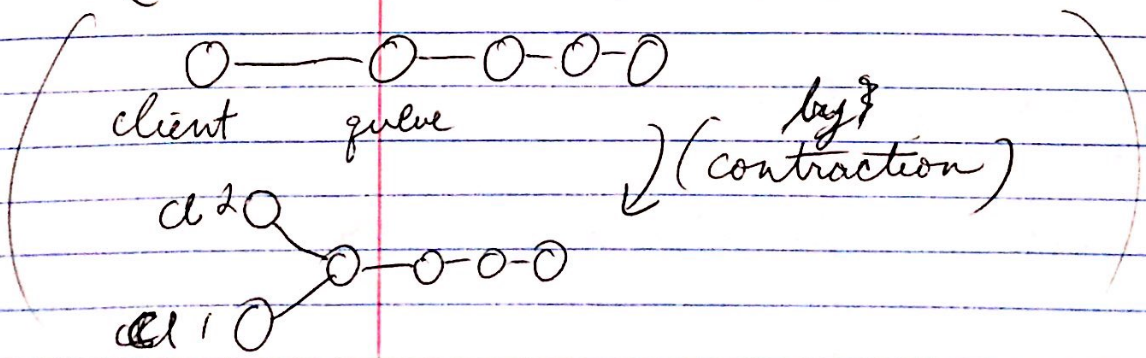
if weakening were allowed, could



(this would be admissable)
(from point of view of client's client)

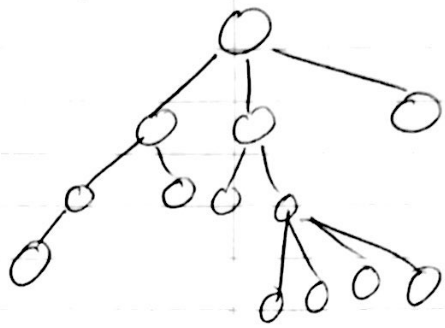
or





when remove weakening, contraction,

our process graph turns into a tree



parent: client

child: providing process / provider

if every ~~child~~ providing process has exactly one client,

what do we know about preservation?

problem w/ multiple clients, protocol changes for one, but another client doesn't see it

by design here, only one client

(can't do dining philosophers in linear setting)

Curry Howard correspondence between intuitionistic linear logic and session-typed π-calculus

[Pfenning 2010]
[Wadler 2012 classical linear logic]

<u>Logic</u>	<u>Programming</u>
linear propositions	session types
proofs	programs
cut-reduction	communication (message exchange)

have to split resources

$A, B, C :: A \multimap B$	multiplicative implication
$A \otimes B$	multiplicative conjunction
$A \& B$	additive conjunction
$A \oplus B$	additive disjunction
$!A$	"of course" persistent truth

additive } choice of resources (same)

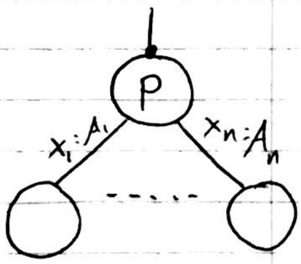
no par (in intuitionistic setting)

require: & and ⊕ have to have at least one label

$$\boxed{\overbrace{x_1 : A_1, \dots, x_n : A_n}^{\Delta} \vdash P :: (x : A)}$$

process P offers session of type A along channel x

Process P offers a session of type A
along channel x
using sessions of types $A_1, \dots, A_n$
that are offered along channels $x_1, \dots, x_n$



expressed from point of view of provider
(multiple children providers for a single client (parent))

Δ - linear context

External choice

$$\frac{\Delta \vdash P_1 :: (x:A) \qquad \Delta \vdash P_2 :: (x:B)}{\Delta \vdash \text{case } x \text{ of } (P_1, P_2) :: (x : A \,\&\, B)} \;(\&R)$$

additive nature - don't have to split resources
have to make a choice

two left rules (client makes a choice)

$$\frac{\Delta, x:A \vdash Q :: (z:C)}{\Delta, x:A\&B \vdash x\ \mathrm{inl} ; Q :: (z:C)} \;(\&L_1)$$

$$\frac{\Delta, x:B \vdash Q :: (z:C)}{\Delta, x:A\&B \vdash x\ \mathrm{inr} ; Q :: (z:C)} \;(\&L_2)$$

programming purposes : n-ary choice

n premises for R

n rules for L

Convention : use 'x' for right rule channel (on right)
'z' for left (on right side)

Internal Choice

$$\frac{\Delta \vdash Q :: (x : A)}{\Delta \vdash x.\text{inl} ; Q :: (x : A \oplus B)} \; (\oplus R_1)$$

$$\frac{\Delta \vdash Q :: (x : B)}{\Delta \vdash x.\text{inr} ; Q :: (x : A \oplus B} \; (\oplus R_2)$$

$$\frac{\Delta, x : A \vdash P_1 :: (z : C) \qquad \Delta, x : B \vdash P_2 :: (z : C)}{\Delta, x : A \oplus B \vdash \text{case } x \text{ of } (P_1, P_2) :: (z : C)} \; (\oplus L)$$

Channel Input

$$\frac{\Delta, y : A \vdash P :: (x : B)}{\Delta \vdash y \leftarrow \text{recv } x ; P :: (x : A \multimap B)} \; (\multimap R)$$

$$\frac{\Delta \vdash Q :: (y : A) \qquad \Delta', x : B \vdash Q' :: (z : C)}{\Delta, \Delta', x : A \multimap B \vdash \text{send } x \, (y \leftarrow Q) ; Q' :: (z : C)} \; (\multimap L)$$

(spawn new process Q provide type A)

Channel Output

$$\frac{\Delta \vdash Q :: (y : A) \qquad \Delta' \vdash Q' :: (x : B)}{\Delta, \Delta' \vdash \text{send } x \, (y \leftarrow Q) ; Q' :: (x : A \otimes B)} \; (\otimes R)$$

$$\frac{\Delta, x : B, y : A \vdash P :: (z : C)}{\Delta, x : A \otimes B \vdash y \leftarrow \text{recv } x ; P :: (z : C)} \; (\otimes L)$$

terminate a session

Termination

$$\frac{}{\cdot \vdash \text{close } x :: (x : 1)} \ (1R)$$

$$\frac{\Delta \vdash P :: (z : C)}{\Delta, x : 1 \vdash \text{wait } x ; P :: (z : C)} \ (1L)$$

can only terminate self w/ nothing on context

ensures no trees w/o root (parent)