Lecture 21

Untyped Lambda Calculus as a model of computation

What we have:

- variables
- (functions =) abstractions
- application (= fn call)

$$e ::= x \mid \lambda x.e_1 \mid (e_1\ e_2)$$

How expressive is this?

Recall

$$(\beta)\quad ((\lambda x. e_1)\ e_2) \to_\beta e_1[e_2/x]$$

$$(\mathrm{op})\quad \frac{e_1 \to_\beta e_1'}{e_1\ e_2 \to_\beta e_1'\ e_2}$$

$$(\mathrm{arg})\quad \frac{e_2 \to_\beta e_2'}{e_1\ e_2 \to_\beta e_1\ e_2'}$$

$$(\xi)\quad \frac{e \to_\beta e'}{\lambda x. e \to_\beta \lambda x. e'}$$

Redex (Reducible Expression)

- any expression of the form

$$((\lambda x. e_1)\ e_2)$$

Context and redex

$$\mathcal{C}[\ ] ::= [\ ] \mid e\,\mathcal{C}[\ ] \mid \mathcal{C}[\ ]\,e \mid \lambda x.\mathcal{C}[\ ]$$

<u>Now</u> all 4 rules captured by:

$$\mathcal{C}[(\lambda x.e_1)\ e_2] \longrightarrow_\beta \mathcal{C}[e_1[e_2/x]]$$

—

β-reducible : Can factor into CONTEXT $\mathcal{C}[\ ]$ and redex r

<u>Reflexive-Transitive Closure</u> of $\to_\beta$

$\rhd_\beta$: 0 or more β-reductions

(i.e) $(\longrightarrow_\beta)^*$

[Allow α-conversions

Allow α-conversions
(renaming of bound variables freely whenever we wish).

β-normal form (β-nf)

- e is in β-nf if e does not contain a β-redex.

is in β-nf DIFFERENT FROM has a β-nf.

CONFLUENCE OF $\triangleright_\beta$

CHURCH-ROSSER THM OF $\triangleright_\beta$

IF

$$\begin{tikzcd}
 & e \arrow[dl, "\triangleright_\beta"'] \arrow[dr, "\triangleright_\beta"] & \\
e_1 \arrow[dr, "\triangleright_\beta"'] & & e_2 \arrow[dl, "\triangleright_\beta"] \\
 & e_3 &
\end{tikzcd}$$

THEN … FOR SOME e_3

β-nfs are UNIQUE (UPTO α-cnv)

- If
$$\begin{tikzcd} & e \arrow[dl, "\beta"'] \arrow[dr, "\beta"] & \\ e_1 & & e_2 \end{tikzcd}$$
e_1 and e_2 in β-nf, then
$$e_1 \equiv_\alpha e_2 .$$

- If $e \rhd_\beta e_1$ e_1 in β-nf
and $e \rhd_\beta e_2$
then $e_2 \rhd_\beta e_1$.

Note: THIS DOES NOT MEAN EVERY EXPRESSION MUST TERMINATE IN A β-nf.

Eg: $\Delta \equiv \lambda x.(x\ x)$

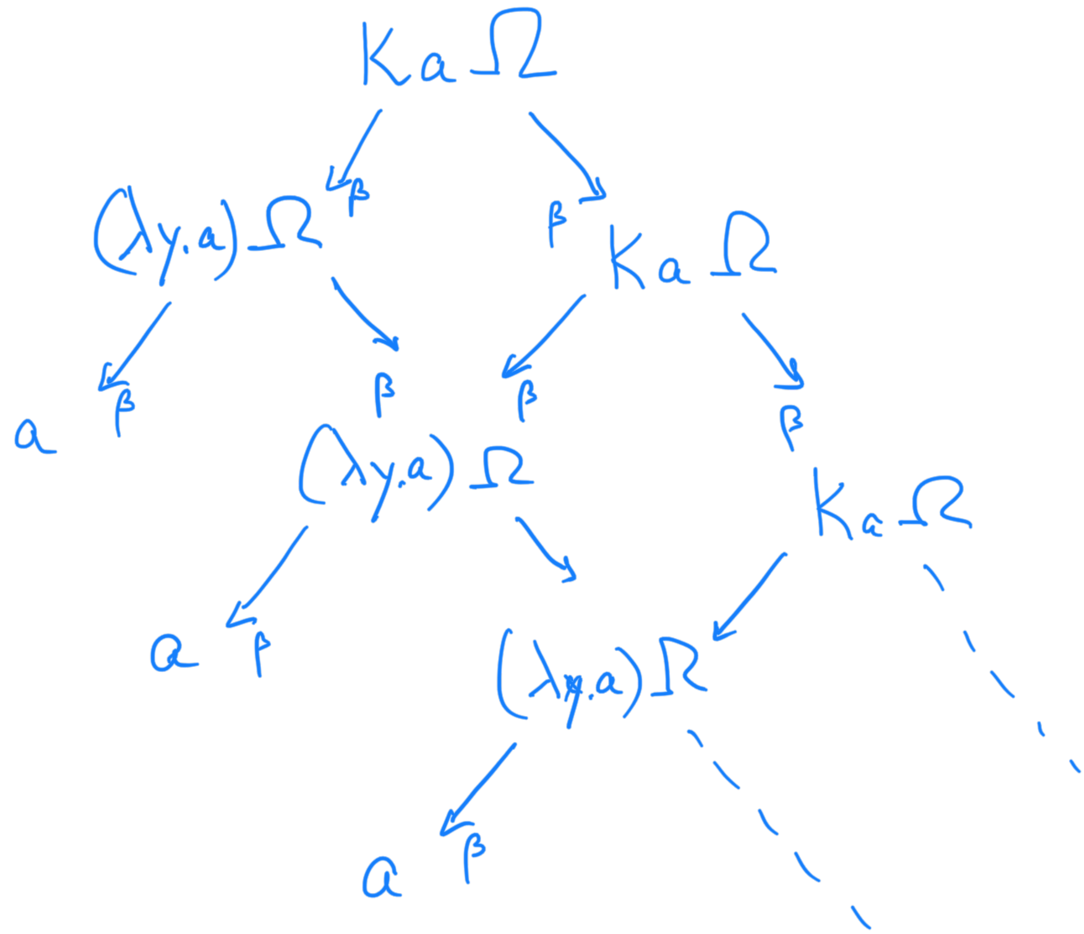
Consider $\Omega \equiv (\Delta\ \Delta)$

$$\Delta\ \Delta \equiv (\lambda x.(x\ x))\ \Delta$$
$$\to_\beta (x\ x)[\Delta/x]$$
$$\equiv \Delta\ \Delta$$

- $\Omega \to_\beta \Omega \to_\beta \Omega \to_\beta \cdots\cdots$

$\therefore\ \Omega \to_\beta \Omega \to_\beta \Omega \to_\beta \cdots$

Now consider $K\,a\,\Omega$ for any expression a (let a be in β-nf).



SO CONFLUENCE (CHURCH-ROSSER) ONLY SAYS THAT IF AN EXPRESSION HAS A β-nf, THEN IT IS UNIQUE.

- BUT THERE MAY BE BOTH
  - TERMINATING

- NON-TERMINATING PATHS

- WHICH TO TAKE?

(*) IF THERE IS A TERMINATING PATH, THEN LEFTMOST OUTERMOST REDUCTION WILL TERMINATE.

(LAZY SAFER THAN EAGER)

A NOTION OF EQUALITY

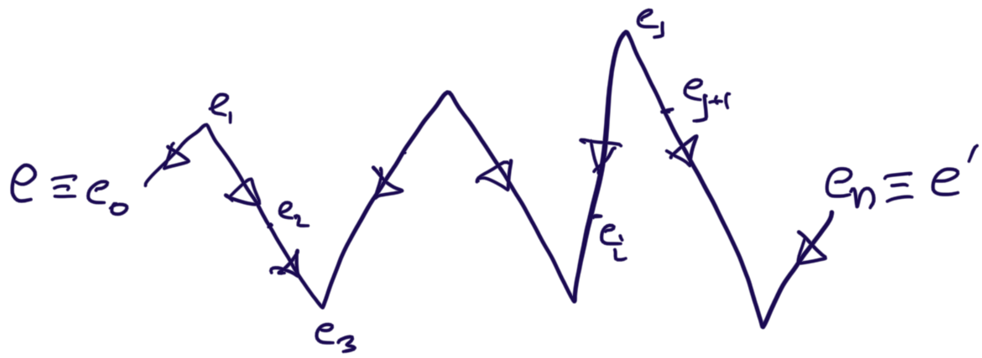
$e =_\beta e'$ if for some $e_0, e_1, \ldots e_n$ we have:

$e \equiv e_0$, $e_n \equiv e'$

and for each $0 \leq i < n$

- $e_i \rhd_\beta e_{i+1}$

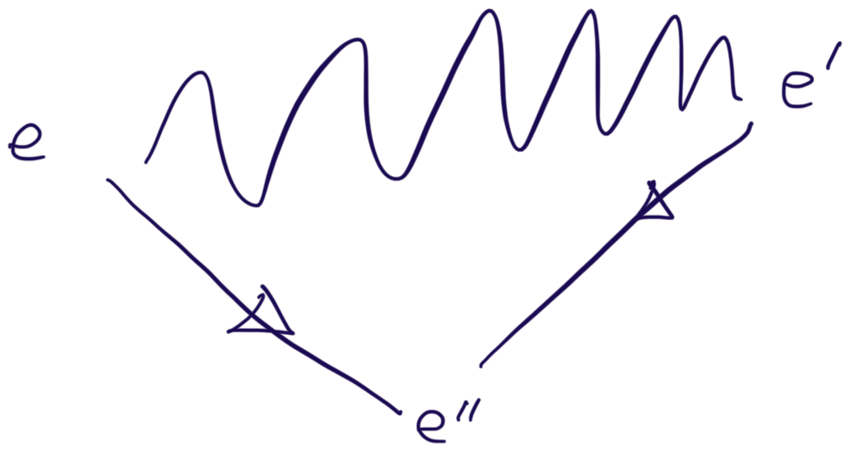
or - $e_{i+1} \rhd_\beta e_i$



CHURCH-ROSSER THEOREM OF $=_\beta$

If $e =_\beta e'$ then

there exists e'' s.t
$e \triangleright_\beta e''$ and $e' \triangleright_\beta e''$



STRATEGY FOR PROVING EQUALITY OF expression that have a β-nf.

• FIND THEIR β-nf

- CHECK IF THESE ARE EQUAL (mod $\equiv_\alpha$)

Modelling the booleans

2 "values"

$$T \triangleq \lambda x. \lambda y. x$$

$$F \triangleq \lambda x. \lambda y. y$$

How do we show $T \neq F$

• Show that if $T =_\beta F$ then all expressions are equal.

Suppose $T =_\beta F$

then for all e_1, e_2

$$
\begin{tikzcd}
T\, e_1\, e_2 \arrow[d, "\beta"] & =_\beta & F\, e_1\, e_2 \arrow[d, "\beta"] \\
e_1 \arrow[dr, dashed] & & e_2 \arrow[dl, dashed] \\
& e &
\end{tikzcd}
$$

All terms are equal!
(Useless theory).

Using T, F — need an
"if ___ then e_1 else e_2"

Define $D \triangleq \lambda t. \lambda a. \lambda b. (t\, a\, b)$

$$
\begin{aligned}
D\, T\, e_1\, e_2 &=_\beta T\, e_1\, e_2 \\
&=_\beta e_1
\end{aligned}
$$

—

$$DF\, e_1\, e_2 =_\beta +\, e_1\, e_2$$

$$=_\beta e_2$$

Can this be generalised to

- Sets of cardinality n
  (n - a finite integer)
- n-ary case analysis. ?

Modelling Pairs

$\langle e_1, e_2 \rangle$ for any e_1, e_2

Define $P \equiv \lambda a. \lambda b. \lambda t. (t\ a\ b)$

$$P\, e_1\, e_2 =_\beta \lambda t. (t\ e_1\ e_2)$$

Using pairs — need to
have proj1
proj2

$proj2$

Such that

$$proj1\ \langle e_1, e_2 \rangle =_\beta e_1 \qquad (1)$$

$$proj2\ \langle e_1, e_2 \rangle =_\beta e_2 \qquad (2)$$

Define

$$proj_1 \equiv \lambda p.\left(p\ \underline{(\lambda x.\lambda y.x)}\right) \qquad T$$

$$proj_2 \equiv \lambda p.\left(p\ \underline{(\lambda x.\lambda y.y)}\right) \qquad F$$

EXERCISE

Check (1) & (2) HOLD.

EXERCISE

HOW CAN ONE GENERALISE to k-tuples for any finite k?

CHURCH NUMERALS

$$\underline{0} \equiv \lambda f.\, \lambda x.\ x$$

$$\underline{1} \equiv \lambda f.\, \lambda x. (f\ x)$$

$$\underline{n} \equiv \lambda f.\, \lambda x.\ \underbrace{f\,(f \ldots (f}_{n\ f\text{'s.}}\ x) \ldots)$$

$$\text{add} \equiv \lambda m.\, \lambda n.\, \lambda h.\, \lambda z.\ (m\ h)(n\ h\ z)$$

$$\text{mult} \equiv \lambda m.\, \lambda n.\, \lambda h.\, \lambda z.\ (m\ (n\ h)\ z)$$

$$=_{\eta} \lambda m.\, \lambda n.\, \lambda h.\ m\ (n\ h)$$

$$\text{exp} \equiv \lambda m.\, \lambda n.\ n\ m$$