3. Reduction

① When we studied operational semantics, we ~~looked at~~ used transition relation $\to$ to model one step computation. We do something similar for the lambda calculus. We define a binary relation $\to \subseteq \langle exp \rangle \times \langle exp \rangle$, called contraction relation, which models or formalises ~~a~~ single-step computation. Then, we will ~~define~~ call the reflexive transitive closure of $\to$, i.e. $\to^*$, reduction relation.

② Here is the definition of the contraction relation using inference rules:

(i) β-reduction — called redex. or β-redex

$$\frac{}{(\lambda v.e)\,e' \to (e/_{v\to e'})}$$

Renaming

$$\frac{e_0 \to e_1 \qquad e_1 \equiv e_1'}{e_0 \to e_1'}$$

Contextual Closure

$$\frac{e_0 \to e_1}{e_0' \to e_1'}$$

e_1' is obtained from e_0' by replacing one occurrence of e_0 in e_0' by e_1.

③ The real change happens by the first rule., β-reduction. The second rule means that the contraction relation is defined on α-equivalence classes. The third rule says that ~~Any~~ any subexpression of a given e can be contracted by the β-reduction rule. Here are a few examples that may help you to see what goes on here.

$$(\lambda x.y)(\lambda z.z) \to y$$
$$(\lambda x.xx)(\lambda z.z) \to (\lambda z.z)(\lambda z.z) \to (\lambda z.z)$$
$$(\lambda x.\underline{(\lambda y.yx)z})(zw) \to (\lambda x.zx)(zw) \to z(zw)$$
$$\underline{(\lambda x.(\lambda y.yx)z)(zw)} \to (\lambda y.y(zw))z \to z(zw)$$

④ Because of the third rule, the contraction relation is not deterministic. That is, $e_0 \to e_1$ and $e_0 \to e_2$ do not imply that $e_1 \equiv e_2$. For a counterexample, look at. However, ~~the ta~~ this nondeterminism comes from the nondeterminism in computation strategy, not from any the nondeterministic constructs of the lambda calculus (which do not exist). The following theorem ~~provides for captures this~~ shows one consequence of this, and expresses that the contraction relation is essentially deterministic.

Prop 10.2. (Church-Rosser Theorem) If $e \to^* e_0$ and $e \to^* e_1$, then there exists e_2 s.t. $e_0 \to^* e_2$ and $e_1 \to^* e_2$. □

α-equiv. e_0, e_1 regarded the same
reflexive and transitive closure of $\to$

⑤ An expression e is a β-normal form if it cannot be contracted. Intuitively, such an expression denotes the ~~outcome~~ final result of some computation, and the reduction relation $\to^*$ (performs computation by) transforming a given expression to a normal form.

Prop 10.2 implies that every expression can be reduced to at most one normal-form expression modulo α-equivalence.