Computation and Normal Forms in LC

Ken Q Pu, Faculty of Science, Ontario Tech University

Lambda Calculus is a formal language

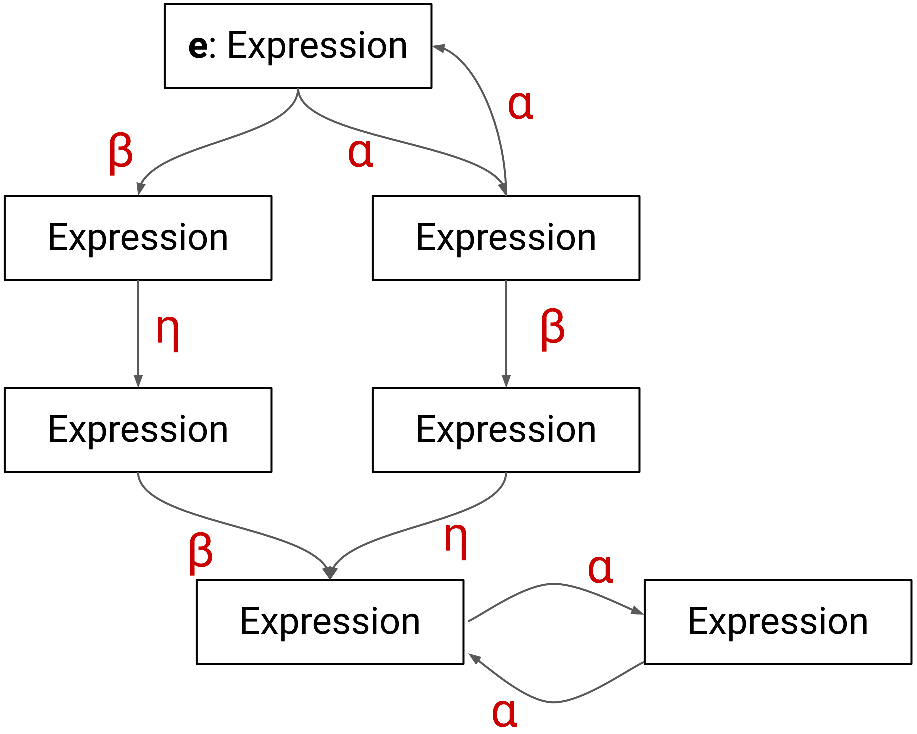
The expressions are strings in a formal language **LC**:

- "<name>" $\in$ **LC**.
- If $e \in$ **LC**, then "λ<name>. e" $\in$ **LC**
- If $e1 \in$ **LC** and $e2 \in$ **LC**, then "e1 e2" $\in$ **LC**.
- If $e \in$ **LC**, then "(e) $\in$ **LC**"

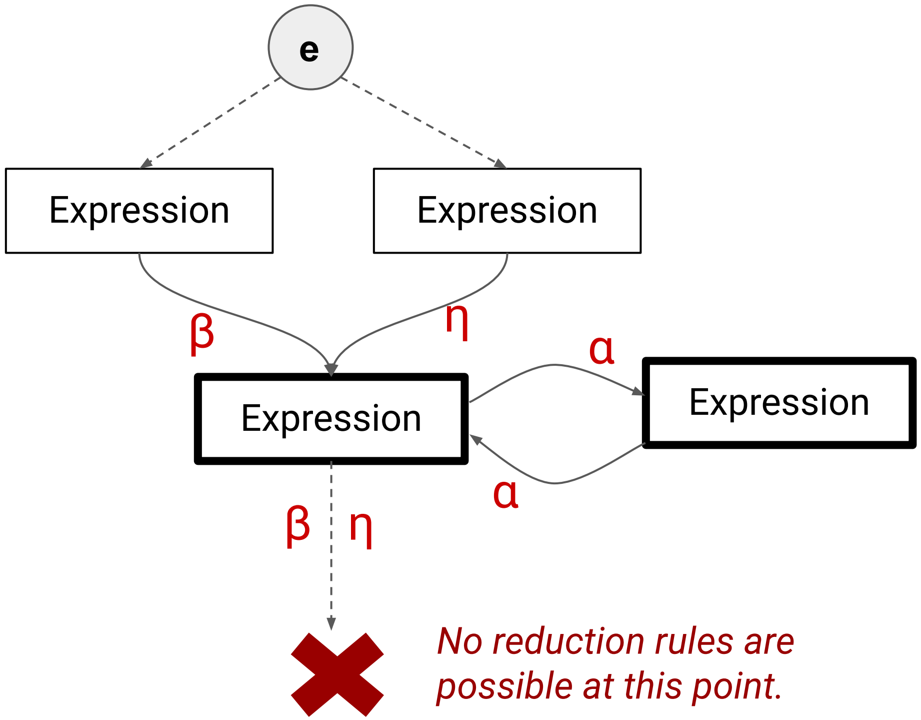
Ken Q Pu, Faculty of Science, Ontario Tech University

Rewrite rules are string operations

- Alpha conversion
- Beta reduction
- Eta reduction



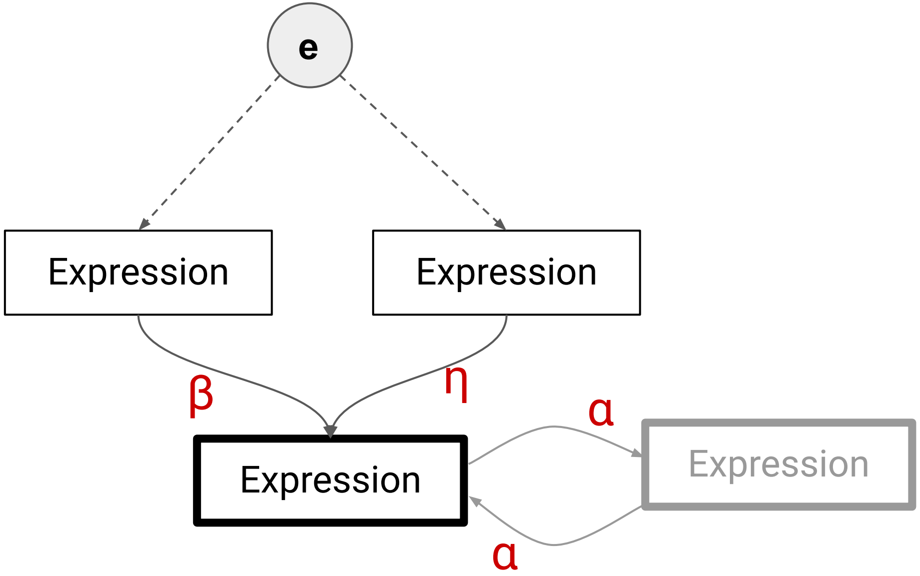
Normal forms



These expressions only differ in parameter names. They are equivalent.

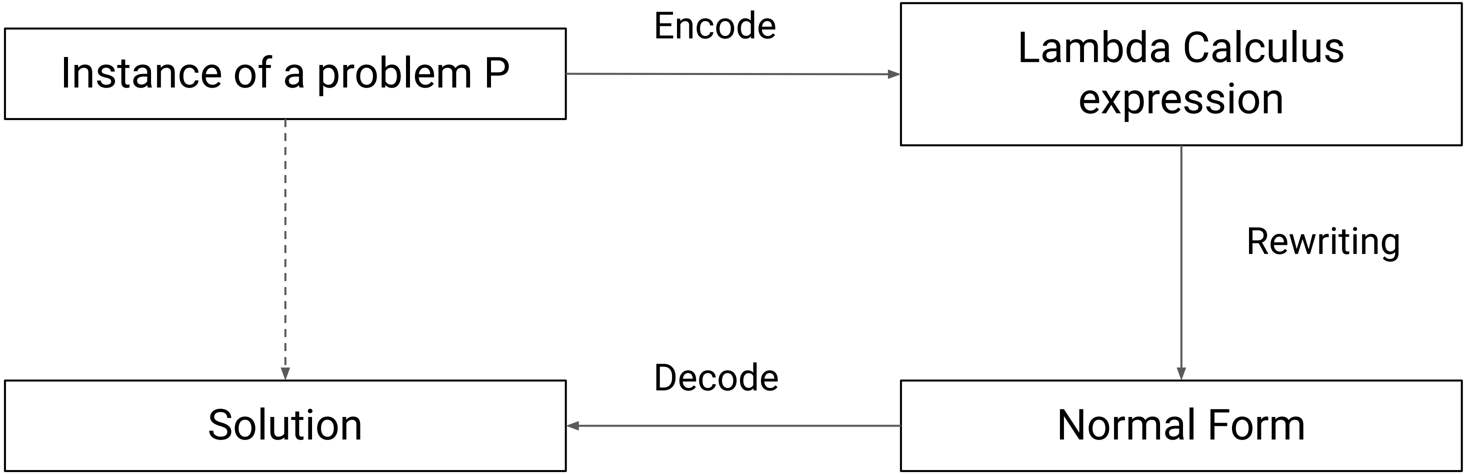
No reduction rules are possible at this point.

Normal forms



The *normal form* of **e** is an expression that can be derived from **e** using alpha, beta, and eta rules, but no further beta reductions are possible.

Computation in Lambda Calculus



Challenges with computation

- Ambiguity in selecting rewriting rules
- Termination

Ambiguity of reduction

(\x.\y. y) ((\z.z) (\z. z z))

\y. y

(\x.\y. y) (\z. z z)

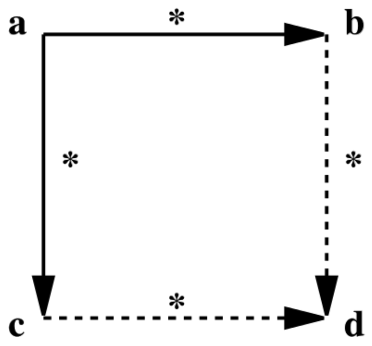
\y. y

Ken Q Pu, Faculty of Science, Ontario Tech University

Uniqueness of normal forms

Church-Rosser Theorem, 1936

If a $\longrightarrow_*$ b and a $\longrightarrow_*$ c, then there exists some expression d such that b $\longrightarrow_*$ d and c $\longrightarrow_*$ d.



Consequence of Church-Rosser Theorem

Every expression has at most **one** normal form modulo alpha conversion.

Termination

Consequence of Church-Rosser Theorem

Every expression has at most **one** normal form modulo alpha conversion.

$$\boxed{\texttt{(\textbackslash z. z z) (\textbackslash z. z z)}}$$

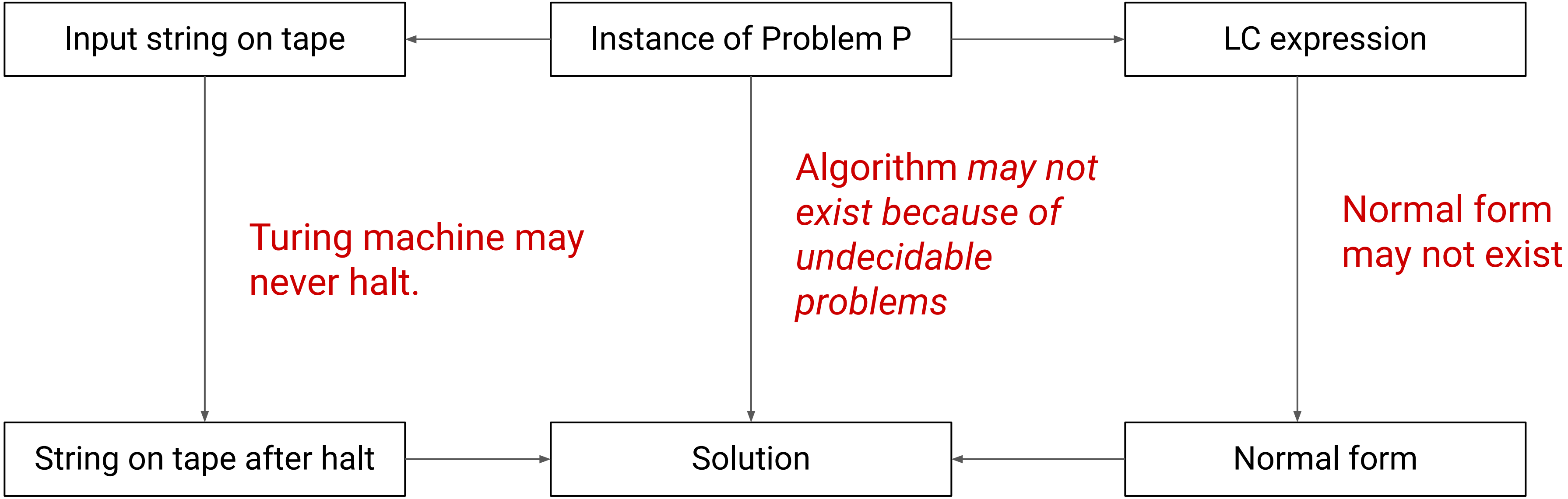
$$\Big\downarrow \beta$$

$$\boxed{\texttt{(\textbackslash z. z z) (\textbackslash z. z z)}}$$

$$\Big\downarrow \beta$$

forever

Comparison with Turing Machine



Ken Q Pu, Faculty of Science, Ontario Tech University

Challenge

Consider the following decision problem:

Given a LC expression, does it have a normal form?

Can you prove that there is no general algorithm to solve this problem?

Namely, this problem is undecidable.