Reduction in Λ

Reduction in Arithmetic Expression

We know the following **arithmetic expressions** are **equivalent**.

$$\mathbf{0+5, 2\times 2+1, 5, 1+1+3\times 1, 1\times 2+1+2\times 1, \cdots}$$

But there is somthing special about $\mathbf{5}$.

- Every other expression can be **reduced** to it,
- but $\mathbf{5}$ cannot be reduced to anything else.
- $\mathbf{5}$ is the **reduced** (**normal**) element of the class.

Reduction is a Binary Relation

- Let $\mathcal{E}$ be the collection of **arithmetic expressions** over $\mathbb{N}$ and $\{+, \times, (,)\}$.
- The **reduction** '$\rightarrow$' may be viewed as a **binary relation** on $\mathcal{E}$ i.e. $\rightarrow \subseteq \mathcal{E} \times \mathcal{E}$.
- Following is an example of **reduction**, $\rightarrow$:

$$(\mathbf{1} \times \mathbf{2} + \mathbf{1} + \mathbf{2} \times \mathbf{1}\ ,\ \mathbf{2} + \mathbf{1} + \mathbf{2}) \in \rightarrow$$

We in practice write: $\mathbf{1} \times \mathbf{2} + \mathbf{1} + \mathbf{2} \times \mathbf{1} \ \rightarrow \mathbf{2} + \mathbf{1} + \mathbf{2}$ i.e. $\mathbf{1} \times \mathbf{2} + \mathbf{1} + \mathbf{2} \times \mathbf{1}$ is **reduced** to $\mathbf{2} + \mathbf{1} + \mathbf{2}$ in one step.

Reflexive and Transitive Binary Relations

- A **binary relation** R over a set A is called **reflexive** if,

$$(\mathbf{a}, \mathbf{a}) \in \mathbf{A}, \text{ \textbf{for all} } a \in A.$$

- A **binary relation** R over a set A is called **transitive** if,

$$(\mathbf{a}, \mathbf{b}), (\mathbf{b}, \mathbf{c}) \in \mathbf{A} \Rightarrow (\mathbf{a}, \mathbf{c}) \in \mathbf{A}, \text{ \textbf{for all} } a, b, c \in A.$$

Reflexive-Transitive Closure of a Binary Relation

- Let R be a **binary relation** over the set A.
- The **reflexive-transitive closure** of R is also a **binary relation** R^* over A such that,
  - $\mathbf{R} \subseteq \mathbf{R}^*$,
  - $\mathbf{R}^*$ is both **reflexive** and **transitive**.
  - $\mathbf{R}^*$ is the **smallest relation** satisfying the first two conditions.

Reflexive-Transitive Closure of a Binary Relation

Given a **binary relation** R over A, the basic idea of its **reflexive-transitive closure** R^* are the following,

- For all $\mathbf{a \in A}$, $\mathbf{(a, a) \in R^*}$ - R^* is **reflexive**, and
- $\mathbf{(a, b) \in R^*}$, if there are $\mathbf{a_1, a_2, \cdots, a_n \in A}$, such that $\mathbf{a = a_1}$ and $\mathbf{b = a_n}$, $\mathbf{(a_i, a_{i+1}) \in R}$, for all $\mathbf{a_i}$, $\mathbf{1 \leq i \leq n}$.

$\mathbf{a}$ is **R^*-related** to $\mathbf{b}$, if either $\mathbf{a = b}$ or through some finite number of stages they are **R-related**.

Reflexive-Transitive Closure of Reduction

We can talk about **reduction** in finite number of steps if we take the **reflexive-transitive closure** of the **reduction relation** '$\rightarrow$' over $\mathcal{E}$.

$$
\begin{aligned}
1 \times 2 + 1 + 2 \times 1 &\rightarrow 2 + 1 + 2 \times 1 \\
&\rightarrow 2 + 1 + 2 \\
&\rightarrow 3 + 2 \\
&\rightarrow 5
\end{aligned}
$$

We write $\mathbf{1 \times 2 + 1 + 2 \times 1 \rightarrow^{*} 5}$.

β-Reduction in Λ

The **β-reduction** of **λ-terms** is defined as follows.

$$\beta = \{((\lambda \mathbf{x}.\mathbf{u})\mathbf{v}, \mathbf{u}[\mathbf{x} = \mathbf{v}]) : \ \mathbf{u}, \mathbf{v} \in \mathbf{\Lambda}\}$$

We write one-step β-*reduction* as

$$(\lambda \mathbf{x}.\mathbf{u})\mathbf{v} \ \rightarrow_{\beta} \ \mathbf{u}[\mathbf{x} = \mathbf{v}]$$

- $(\lambda \mathbf{x}.\mathbf{u})\mathbf{v}$ is called a **β-redex** as we can perform **β-reduction** on it.
- $\mathbf{u}[\mathbf{x} = \mathbf{v}]$ is called the **β-contractum** of the previous **β-radex**.

Example of β-Reduction

$$(\lambda \mathbf{x} y.\mathbf{x}(y\mathbf{x}))(\lambda \mathbf{a.aa})(\lambda ab.a)$$

$$\rightarrow_\beta \quad (\lambda \mathbf{y}.(\lambda a.aa)(\mathbf{y}(\lambda a.aa)))(\lambda \mathbf{ab.a})$$

$$\rightarrow_\beta \quad (\lambda a.aa)((\lambda \mathbf{a} b.\mathbf{a})(\lambda \mathbf{a.aa}))$$

$$\rightarrow_\beta \quad (\lambda \mathbf{a.aa})(\lambda \mathbf{b}.(\lambda \mathbf{a.aa}))$$

$$\rightarrow_\beta \quad (\lambda \mathbf{b}.(\lambda a.aa))(\lambda \mathbf{b}.(\lambda \mathbf{a.aa}))$$

$$\rightarrow_\beta \quad \lambda a.aa$$

R-T Closure of β-Reduction

The **reflexive-transitive closure** of '$\rightarrow_\beta$ is '$\rightarrow^*_\beta$ and we write

$$(\lambda \mathbf{xy.x(yx)})(\lambda \mathbf{a.aa})(\lambda \mathbf{ab.a}) \ \rightarrow^*_\beta \lambda \mathbf{a.aa}$$

β-Reduction in Λ

- A **λ-term** is in **β-normal form** if it does not contain any **β-redex**.
- Some **λ-term** cannot be **reduced** to a **normal form**.

$$
\begin{aligned}
(\lambda x.xx)(\lambda y.yy) &\rightarrow_\beta (\lambda y.yy)(\lambda y.yy) \\
&\rightarrow_\beta (\lambda y.yy)(\lambda y.yy) \\
&\rightarrow_\beta \cdots
\end{aligned}
$$

Church-Rosser Property

If two **λ-terms** are equal, then they can be **reduced** ($\rightarrow_\beta^*$) to a common term (upto renaming variables - α-**equivalence**).

$$\mathbf{u} = \mathbf{v} \;\Rightarrow\; \exists \mathbf{w},\; \mathbf{u} \rightarrow_\beta^* \mathbf{w} \text{ and } \mathbf{v} \rightarrow_\beta^* \mathbf{w}$$

A term can have **at most one** β-**normal form** i.e. if the computation **terminates** it always gives the **same value**.

Another Computation : Pevious Example

$$
\begin{aligned}
& (\lambda \mathbf{x} y.\mathbf{x}(y\mathbf{x}))(\lambda \mathbf{a.aa})(\lambda ab.a) \\
\rightarrow_\beta \; & (\lambda y.(\lambda \mathbf{a.aa})(\mathbf{y}(\lambda \mathbf{a.aa})))(\lambda ab.a) \\
\rightarrow_\beta \; & (\lambda \mathbf{y}.(\mathbf{y}(\lambda a.aa))(\mathbf{y}(\lambda a.aa)))(\lambda \mathbf{ab.a}) \\
\rightarrow_\beta \; & ((\lambda ab.a)(\lambda a.aa))((\lambda \mathbf{a}b.\mathbf{a})(\lambda \mathbf{a.aa})) \\
\rightarrow_\beta \; & ((\lambda \mathbf{a}b.\mathbf{a})(\lambda \mathbf{a.aa}))(\lambda ba.aa) \\
\rightarrow_\beta \; & (\lambda \mathbf{b}a.aa)(\lambda \mathbf{ba.aa}) \\
\rightarrow_\beta \; & \lambda a.aa
\end{aligned}
$$

Both computation **terminate** with **identical result**.

Terminating & Nonterminating Computations

Termination

$$(\lambda \mathbf{a} b.b)((\lambda \mathbf{a.aa})(\lambda \mathbf{a.aa})) \;\rightarrow_\beta\; \lambda b.b.$$

nonTermination

$$(\lambda ab.b)((\lambda \mathbf{a.aa})(\lambda \mathbf{a.aa})) \;\rightarrow_\beta\; (\lambda ab.b)((\lambda a.aa)(\lambda a.aa))$$

$$\rightarrow_\beta \;\cdots$$