**CS520, Handout 11, 3 October 2006**

# Normalization and Strong Normalization

Reading: [Pierce, Chapter 12]

## Deterministic vs. Non-Deterministic

***Deterministic*** evaluation strategy: The next evaluation step is uniquely determined. Example: **call-by-value** is deterministic, and so is **call-by-name** [Pierce, page 57].

***Non-deterministic*** evaluation strategy: The next evaluation step is ***not*** uniquely determined. Example: **full beta-reduction** is non-deterministic [Pierce, page 56].

Consider the term

$$t \triangleq \textbf{two}\ (\textbf{two id}) \quad \text{where}$$

$$\textbf{two} \triangleq \lambda \mathrm{x}.\ \lambda \mathrm{y}.\ \mathrm{x}\ (\mathrm{x}\ \mathrm{y}) \qquad \text{(Church numeral 2)}$$

$$\textbf{id} \triangleq \lambda \mathrm{x}.\ \mathrm{x} \qquad \text{(identity)}$$

***Quick Exercises***

1. Carry out the **call-by-value** evaluation of $t$.
2. Carry out the **call-by-name** evaluation of $t$.
3. How many evaluations of $t$ are there according to **full beta-reduction**?

## Normalization vs. Strong Normalization

***Normalizing*** evaluation strategy for a term $t$: **There is an evaluation of $t$** according to this strategy that halts in a normal form.

***Strongly normalizing*** evaluation strategy for a term $t$: **Every evaluation of $t$** according to this strategy halts in a normal form.

For a deterministic evaluation strategy, **normalizing $=$ strongly normalizing**.

For a non-deterministic evaluation strategy, **normalizing $\neq$ strongly normalizing**.



The following theorem [Pierce, page 152] is proved for **call-by-value** evaluation of terms in $\boldsymbol{\lambda}_{\rightarrow}$, the simply-typable pure lambda-calculus [Pierce, Fig 9-1, p 103].

**Theorem [Normalization].** If $\vdash t : T$ is derivable, i.e., $t$ is a simply-typable closed term, then $t$ is normalizable.

**Quick Question:** The theorem is restricted to **closed** terms. If it were generalized to all terms, including open terms, the hypothesis would have to change from "$\vdash t : T$" to "$\Gamma \vdash t : T$" where the context $\Gamma$ assigns a type for every variable occurring free in $t$. Why would the theorem become invalid?

The following theorem [Pierce, page 152] is proved for **call-by-value** evaluation of terms in $\boldsymbol{\lambda}_{\rightarrow}$, the simply-typable pure lambda-calculus [Pierce, Fig 9-1, p 103].

**Theorem [Normalization].** If $\vdash t : T$ is derivable, i.e., $t$ is a simply-typable closed term, then $t$ is normalizable.

**Quick Question:** The theorem is restricted to **closed** terms. If it were generalized to all terms, including open terms, the hypothesis would have to change from "$\vdash t : T$" to "$\Gamma \vdash t : T$" where the context $\Gamma$ assigns a type for every variable occurring free in $t$. Why would the theorem become invalid?

**Answer:** Some open terms that cannot be evaluated further according to call-by-value, such as $(\lambda \mathrm{x}.\, \mathrm{x})\, \mathrm{y}$, are stuck rather than in normal form.

The following theorem, not in [Pierce], holds and can be established using the same proof technique.

**Theorem.** If $\Gamma \vdash t : T$ is derivable, i.e., $t$ is a simply-typable term, not necessarily closed, then the call-by-value evaluation of $t$ always terminates.



***Several Comments:***

- The theorem trivially extends to strong normalization, because **normalization** = **strong normalization** for call-by-value evaluation.
- The theorem remains true for the extended $\boldsymbol{\lambda}\mathbf{NB}_{\rightarrow}$, the simply-typable lambda-calculus augmented with natural numbers, booleans values, and their prim ops (those listed in [Pierce, Chap 8]).
- The theorem remains true for $\boldsymbol{\lambda}\mathbf{NB}_{\rightarrow}$ when further extended with the features of [Pierce, Chap 11], but excluding recursion [Pierce, Sect 11.11].

  **Quick question:** Why should we exclude recursion?
- The theorem remains true for **full beta reduction**, with the conclusion strenghtened from "$t$ **is normalizable**" to "$t$ **is strongly normalizable**".