## Head-normal form

**Definition**

A term $M$ is **in head-normal form** (hnf) if and only if it has the shape

$$\lambda x_1 \ldots \lambda x_m . y M_1 \ldots M_k, \quad m, k \geq 0.$$

We sometimes call $M$ a **head-normal form**.

A term **has a head-normal form** if and only if it reduces to a term in head-normal form.



## Head-reduction

**Definition**

A term $M$ has a **head redex** if and only if it has the form

$$M \equiv \lambda x_1 \ldots x_m . (\lambda y . N_0) N_1 \ldots N_k, \quad m \geq 0, k > 0$$

where the $N_i$ are arbitrary terms. The redex $(\lambda y . N_0) N_1$ is called the **head redex**.

The **head reduction** of a term only reduces head redexes.

**Fact**

If a term has a hnf, then the head reduction always terminates.



## Results about head-normal forms

**Proposition**

Every normal form is a head-normal form.

**Proposition**

If $\lambda x_1 \ldots \lambda x_m . y M_1 \ldots M_k \twoheadrightarrow_\beta N$ then $N \equiv \lambda x_1 \ldots \lambda x_m . y N_1 \ldots N_k$ and $M_i \twoheadrightarrow_\beta N_i$ for $i = 1, \ldots, n$.

**Proposition**

If $M$ has no head-normal form, then $M N_1 \ldots N_k$ has no hnf for arbitrary terms $N_1, \ldots, N_k$.



## Undefined terms

**Definition**

A term is **defined** if and only if it can be reduced to a head-normal form; otherwise, it is **undefined**.

## Russell's paradox

**Russell's paradox** is illustrated by the set $R = \{x \mid x \notin x\}$

**Sets and unary predicates**

The **characteristic function** $\chi_A$ of a set $A$ is defined by

$$\chi_A(x) = \begin{cases} \textbf{true} & x \in A \\ \textbf{false} & x \notin A \end{cases}$$

Given a unary predicate $P$, the corresponding set is $\{x \mid P(x)\ \textbf{true}\}$

**Russell's paradox in $\lambda$-terms**

$$P \equiv \lambda x.\textbf{not}(xx)$$



**Well-typed judgements** The well-typed judgements are defined inductively by the rules:

$$\Gamma, x : A \vdash x : A$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \Rightarrow B}$$

$$\frac{\Gamma \vdash M : A \Rightarrow B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$



**Definition**

A term is called **solvable** if and only if there exists variables $x_1, \ldots, x_m$ and terms $N_1, \ldots N_k$, for $m, k \geq 0$, such that

$$(\lambda x_1 \ldots \lambda x_m.M)N_1 \ldots N_k = I$$

**Proposition**

A term is solvable if and only if it has a head-normal form.



## Simply-typed $\lambda$-calculus

**Types**

$$A ::= \mathrm{int}, \mathrm{bool}, \ldots \mid A \Rightarrow A$$

**Terms**

$$M ::= x \mid \lambda x.M \mid MM$$

**Judgements** $\Gamma \vdash M : A$ where $\Gamma = x_1 : A_1, \ldots, x_n : A_n$.

## Types

**Examples**

$2$ : **integer** $[1, 2, 3]$ : **integer list** $[\_, \_, \_]$ : **A list**

**Type checking** The expression $3 +$ **Ian** is not well-typed.

**Type soundness** If a well-typed program states that an expression is an integer then it really is an integer.

**Type inference** Much of the typing structure can be generated by the machine.

**Application** AnnoDomini (Tofte) uses type inference to find Year 2000 problems in COBOL programs.



## Facts

**Subject reduction** (Type soundness) If $\Gamma \vdash M : A$ and $M \twoheadrightarrow N$ then $\Gamma \vdash N : A$.

**Weak normalisation** Every well-typed term has a normal form.

**Definition** A term $M$ is **strongly normalisable** when there is no infinite reduction sequence beginning with $M$.

**Strong normalisation** Every well-typed term is strongly normalisable.