The λ-Calculus

Silvain Rideau

December 7, 2015

The λ-calculus was introduced by Church in one of the first attempt to formalize computation. Nowadays, it is not really used as a model of computation, nevertheless, it plays a major role in studying the semantics of higher order languages and in proof theory (through the study of *typed* λ-calculus).
These notes were greatly inspired by [Kri93]

Definition 0.1 (L)**:**
Let V be a (countable) set of variables and $A = V \cup \{(,), \lambda\}$. The set L of λ-terms is the smallest set of words on A such that:

- $V \subseteq L$;
- *If t and $u \in L$, then $(t)u \in L$;*
- *If $x \in V$ and $t \in L$, then $\lambda x\, t \in L$.*

Elements of L are called λ-terms. They should be thought of as functions. The λ-term $(t)u$ is the function t applied to u and $\lambda x\, t$ is the function $x \mapsto t$.
I will not prove it but there is a unique decomposition theorem for elements of L:

Proposition 0.2:
Let t be a λ-term, then one and only one of the following cases holds:

- $t = x$ *for some* $x \in V$;
- $t = (v)u$ *for some* v *and* $u \in L$;
- $t = \lambda x\, u$ *for some* $x \in V$ *and* $u \in L$.

Moreover, in those three cases x, u and v are unique (whenever they appear).

As usual we define the set of free and bound variables by induction on L:

Definition 0.3:
Let t be a λ-term, we define $\mathrm{fvar}(t)$ *and* $\mathrm{bvar}(t)$ *by induction on t:*

- *If* $t = x$, *then* $\mathrm{fvar}(t) = x$ *and* $\mathrm{bvar}(t) = \varnothing$;

- *If* $t = (v)u$*, then* $\mathrm{fvar}(t) = \mathrm{fvar}(t) \cup \mathrm{fvar}(u)$ *and* $\mathrm{bvar}(t) = \mathrm{bvar}(t) \cup \mathrm{bvar}(u)$*;*
- *If* $t = \lambda x\, u$*, then* $\mathrm{fvar}(t) = \mathrm{fvar}(t) \setminus \{x\}$ *and* $\mathrm{bvar}(t) = \mathrm{bvar}(t) \cup \{x\}$*.*

We define $\mathrm{var}(t) = \mathrm{fvar}(t) \cup \mathrm{bvar}(t)$*.*

1 Substitution and α-equivalence

As before, we also need to define substitution:

Definition 1.1 (Simple substitution)**:**
Let t *and* $s \in L$*,* $x \in V$*, we define* $t_{s/x}$ *by induction on* t*:*

- *If* $t = x$*, then* $t_{s/x} = s$*;*
- *If* $t = y \neq x$*, then* $t_{s/x} = y$*;*
- *It* $t = (v)u$*,* $t_{s/x} = (v_{s/x})u_{s/x}$*;*
- *If* $t = \lambda x\, u$*, then* $t_{s/x} = t$*;*
- *If* $t = \lambda y\, u$ *and* $y \neq x$*, then* $t_{s/x} = \lambda y\, u_{s/x}$*.*

As usual, this notion of substitution does not give us the expected result if $\mathrm{bvar}(t) \cap \mathrm{fvar}(s) \neq \varnothing$. For example $(\lambda y\, yx)_{y/x}$ should be $\lambda z\, zy$ and not $\lambda y\, yy$. When studying propositional and predicate logic, we where always capable of avoiding having to cope with this problem, but here, we will have to. The solution is to consider the set Λ of λ-terms up to renaming of bound variables.
We will denote by $|t|$ the length of t (as a word).

Lemma 1.2:
Let $t \in L$*,* x *and* $y \in V$*, then* $|t| = |t_{y/x}|$*.*

Proof. We proceed by induction on t. If t is a variables, so is $t_{y/x}$, so they both have length one. If $t = (u)v$, then $|t| = 2 + |u| + |v|$ and by induction $|t_{y/x}| = 2 + |u_{y/x}| + |v_{y/x}| = 2 + |u| + |v| = |t|$. Finally, if $t = \lambda x\, u$ then $t_{y/x} = t$ and the lemma is obvious or $|t| = 2 + |u| = 2 + |u_{y/x}| = |t_{y/x}|$. ■

Lemma 1.3:
Let t*,* $s \in L$ *and* $x \in V$*. If* x *is not free in* t *then* $t_{s/x} = t$*.*

Proof. This is immediate by induction on t. ■

Lemma 1.4:
Let t*,* $s \in L$ *and* $x \in V$*. Assume that* $x \in \mathrm{fvar}(t)$ *and* $\mathrm{bvar}(t) \cup \mathrm{fvar}(s) = \varnothing$*, then* $\mathrm{fvar}(t_{s/x}) = (\mathrm{fvar}(t) \setminus \{x\}) \cup \mathrm{fvar}(s)$ *and* $\mathrm{bvar}(t_{s/x}) = \mathrm{bvar}(t) \cup \mathrm{bvar}(s)$*.*

Proof. We proceed by induction on t. If $t = x$, then $t_{s/x} = s$, $\mathrm{fvar}(t_{s/x}) = \mathrm{fvar}(s) = (\mathrm{fvar}(t) \setminus \{x\}) \cup \mathrm{fvar}(s)$ and $\mathrm{bvar}(t_{s/x}) = \mathrm{bvar}(s) = \mathrm{bvar}(t) \cup \mathrm{bvar}(s)$. If $t = y \neq x$, then $x \notin \mathrm{fvar}(t)$.

If $t = (u)v$, then $t_{s/x} = (u_{s/x})v_{s/x}$ and $\mathrm{fvar}(t_{s/x}) = \mathrm{fvar}(u_{s/x}) \cup \mathrm{fvar}(v_{s/x}) = ((\mathrm{fvar}(u) \cup \mathrm{fvar}(v)) \setminus \{x\}) \cup \mathrm{fvar}(s) = (\mathrm{fvar}(t) \setminus \{x\}) \cup \mathrm{fvar}(s)$ as x must be free in u or v (or both). Moreover, $\mathrm{bvar}(t_{s/x}) = \mathrm{bvar}(u_{s/x}) \cup \mathrm{fvar}(v_{s/x}) = \mathrm{bvar}(u) \cup \mathrm{bvar}(v) \cup \mathrm{bvar}(s) = \mathrm{bvar}(t) \cup \mathrm{bvar}(s)$.
If $t = \lambda y\, u$ with $y \neq x$, then $t_{s/x} = \lambda y\, u_{s/x}$, $\mathrm{fvar}(t_{s/x}) = \mathrm{fvar}(u_{s/x}) \setminus \{y\} = ((\mathrm{fvar}(u) \setminus \{x\}) \cup \mathrm{fvar}(s)) \setminus y = (\mathrm{fvar}(u) \setminus \{x, y\}) \cup \mathrm{fvar}(s) = (\mathrm{fvar}(t) \setminus \{x\}) \cup \mathrm{fvar}(s)$, as y is not free in s, and $\mathrm{bvar}(t_{x,s}) = \mathrm{bvar}(u_{x/s}) \cup \{y\} = \mathrm{bvar}(u) \cup \mathrm{bvar}(s) \cup \{y\} = \mathrm{bvar}(t) \cup \mathrm{bvar}(s)$. ■

Lemma 1.5:
Let $t, s, u \in L$ and $x \neq y \in V$. Assume that $x \notin \mathrm{fvar}(u)$ and $y \notin \mathrm{fvar}(s) \cap \mathrm{bvar}(t)$ then $(t_{s/x})_{u/y} = (t_{u/y})_{s_{u/y}/x}$.

Proof. We proceed by induction on t. If $t = x$, then $(t_{s/x})_{u/y} = s_{u/y}$ and $(t_{u/y})_{s_{u/y}/x} = t_{s_{u/y}/x} = s_{u/y}$. If $t = y$, $(t_{s/x})_{u/y} = t_{u/y} = u$ and $(t_{u/y})_{s_{u/y}/x} = u_{s_{u/y}/x} = u$ by Lemma **(1.3)**. If $t = z$ distinct from x and y, $(t_{s/x})_{u/y} = z = (t_{u/y})_{s_{u/y}/x}$.
If $t = (v)w$, then $(t_{s/x})_{u/y} = ((v_{s/x})_{u/y})(w_{s/x})_{u/y} = ((v_{u/y})_{s_{u/y}/x})(w_{u/y})_{s_{u/y}/x} = (t_{u/y})_{s_{u/y}/x}$.
If $t = \lambda x\, v$, then $(t_{s/x})_{u/y} = t_{u/y}$ and $(t_{u/y})_{s_{u/y}/x} = (\lambda x\, v_{u/y})_{s_{u/y}/x} = \lambda x\, v_{u/y} = t_{u/y}$. If $t = \lambda y\, v$, then, by hypothesis, $y \notin \mathrm{fvar}(s)$, $(t_{s/x})_{u/y} = (\lambda y\, v_{s/x})_{u/y} = \lambda y\, v_{s/x} = t_{s/x}$ and $(t_{u/y})_{s_{u/y}/x} = t_{s_{u/y}/x} = t_{s/x}$ by Lemma **(1.3)**. Finally, if $t = \lambda z\, v$ where z is neither x nor y, then $(t_{s/x})_{u/y} = \lambda z\, (v_{s/x})_{u/y} = \lambda z\, (v_{u/y})_{s_{u/y}/x} = (t_{u/y})_{s_{u/y}/x}$. ■

Corollary 1.6:
Let $t, s, u \in L$ and $x, y \in V$. Assume that $x \notin \mathrm{fvar}(u)$ and $y \notin \mathrm{fvar}(s)$, then $(t_{s/x})_{u/y} = (t_{u/y})_{s/x}$.

Proof. As $y \notin \mathrm{fvar}(s)$, by Lemma **(1.3)**, $s_{u/y} = s$ and by Lemma **(1.5)**, $(t_{s/x})_{u/y} = (t_{u/y})_{s_{u/y}/x} = (t_{u/y})_{s/x}$. ■

Definition 1.7 (α-conversion)**:**
Let t and t' be λ-terms. We define $t \equiv_\alpha t'$ by induction on $|t|$:

- $x \equiv_\alpha x$;
- *If $t \equiv_\alpha t'$ and $u \equiv_\alpha u'$, then $(t)u \equiv_\alpha (t')u'$;*
- *If $t_{y/x} \equiv_\alpha t_{y/x'}$ for some $y \notin \mathrm{var}(t) \cup \mathrm{var}(t')$, then $\lambda x\, t \equiv_\alpha \lambda x'\, t'$.*

It is not immediately obvious that the above relation is an equivalence relation. To prove that it is transitive, we will need the following:

Lemma 1.8:
Let t and $t' \in L$ be such that $t \equiv_\alpha t'$, then $\mathrm{fvar}(t) = \mathrm{fvar}(t')$.

Proof. Let us proceed by induction on $|t|$. If $t = x$, then $t' = x$ and $\mathrm{fvar}(t) = \{x\} = \mathrm{fvar}(t')$.
If $t = (u)v$, then $t' = (u')v'$ with $u \equiv_\alpha v$ and $u' \equiv_\alpha v'$ and by induction $\mathrm{fvar}(t) = \mathrm{fvar}(u) \cup \mathrm{fvar}(v) = \mathrm{fvar}(u') \cup \mathrm{fvar}(v') = \mathrm{fvar}(t')$.
Finally, if $t = \lambda x\, u$, then $t' = \lambda x'\, u'$ and for some y not appearing in t and t', $u_{y/x} \equiv_\alpha u'_{y/x'}$. We have $\mathrm{fvar}(t) = \mathrm{fvar}(u) \setminus \{x\}$, $\mathrm{fvar}(t') = \mathrm{fvar}(u') \setminus \{x'\}$ and $\mathrm{fvar}(u_{y/x}) = \mathrm{fvar}(u'_{y/x'})$. It

follows that $(\mathrm{fvar}(u) \setminus \{x\}) \cup \{y\} = (\mathrm{fvar}(u') \setminus \{x'\}) \cup \{y\}$ and because y cannot appear in u, $\mathrm{fvar}(t) = \mathrm{fvar}(u) \setminus \{x\} = \mathrm{fvar}(u') \setminus \{x'\} = \mathrm{fvar}(t')$. ∎

Lemma 1.9:
Let t and t' be two λ-terms, x, x', y and $y' \notin \mathrm{var}(t) \cup \mathrm{var}(t')$. If $t_{y/x} \equiv_\alpha t'_{y/x'}$ then $t_{y'/x} \equiv_\alpha t'_{y'/x'}$.

Proof. We proceed by induction on $|t|$.
Let us first assume that $t = x$, then $t_{y/x} = y \equiv_\alpha t'_{y/x'}$, but the only term α-equivalent to y is y itself, so $t'_{y/x'} = y$. As y does not appear in t', we must have $t' = x'$ and $t_{y'/x} = y' = t'_{y'/x'}$. If $t = z \neq x$, then $t'_{y/x'} = z \neq y$ and hence $t' = z$ too. So $t_{y'/x} = z = t'_{y'/x'}$.
If $t = (u)v$, then $t'_{y/x'} = (u_{y/x})v_{y/x}$, so $t' = (u')v'$ and $u'_{y/x'} \equiv_\alpha u_{y/x}$ and $v'_{y/x'} \equiv_\alpha v_{y/x}$. By induction $u'_{y'/x'} \equiv_\alpha u_{y'/x}$ and $v'_{y'/x'} \equiv_\alpha v_{y'/x}$ and so $t_{y'/x} \equiv_\alpha t'_{y'/x'}$.
Finally, assume $t = \lambda z\, u$. Let us first assume that $z \neq x$ and x'. Then $t'_{y/x'} = \lambda z\, u_{y/x}$ so $t' = \lambda z'\, u'$ and for some $w \notin \mathrm{var}(t) \cup \mathrm{var}(t') \cup \{x, x'\}$, $(u_{y/x})_{w/z} \equiv_\alpha (u'_{y/x'})_{w/z'}$. By Corollary **(1.6)**, $(u_{y/x})_{w/z} = (u_{w/z})_{y/x}$ and $(u'_{y/x'})_{w/z'} = (u'_{w/z'})_{y/x'}$. By induction $(u_{w/z})_{y'/x} \equiv_\alpha (u'_{w/z'})_{y'/x'}$ and by Corollary **(1.6)** again, $(u_{y'/x})_{w/z} \equiv_\alpha (u'_{y'/x'})_{w/z'}$ and $t_{y'/x} \equiv_\alpha t'_{y'/x'}$. If $z = x \neq x'$, then $u_{w/z} \equiv (u'_{y/x'})_{w/z}$. If $x' \in \mathrm{fvar}(u')$, then $y \in \mathrm{fvar}(u'_{y/x'})$ so $y \in \mathrm{fvar}((u'_{y/x'})_{w/z}) = \mathrm{fvar}(u)$ by Lemma **(1.8)**, a contradiction. So $x' \notin \mathrm{fvar}(u')$ and $u_{w/z} \equiv u'_{w/z}$, i.e. $t_{y'/x} = t \equiv_\alpha t' = t_{y'/x'}$. By symmetry, the case $z = x' \neq x$ is also taken care of. There remains the case $z = x = x'$ but in that case $t_{y'/x} = t \equiv_\alpha t' = t'_{y'/x'}$. ∎

Corollary 1.10:
The relation $\equiv_\alpha$ is an equivalence relation.

Proof. Reflexivity, symmetry and transitivity are proved by induction. The proof of reflexivity and symmetry is essentially straightforward. Let us prove transitivity.
Let t, t' and t'' such that $t \equiv_\alpha t'$ and $t' \equiv_\alpha t''$. Let us proceed by induction on $|t|$. If $t = x$, then $t' = x$ hence $t'' = x$. If $t = (u)v$, then $t' = (u')v'$ with $u \equiv_\alpha u'$ and $v \equiv_\alpha v'$ and $t'' = (u'')v''$ with $u' \equiv_\alpha u''$ and $v' \equiv_\alpha v''$. By induction $u \equiv_\alpha u''$ and $v \equiv_\alpha v''$ and $t \equiv_\alpha t''$. If $t = \lambda y\, u$, then $t' = \lambda y'\, u'$ and for some $z \notin \mathrm{var}(t) \cup \mathrm{var}(t')$, $u_{z/y} \equiv_\alpha u'_{z/y'}$. Then we also have $t'' = \lambda y''\, u''$ and for some $z' \notin \mathrm{var}(t') \cup \mathrm{var}(t'')$, $u'_{z'/y'} \equiv_\alpha u''_{z'/y''}$. By Lemma **(1.9)**, we may assume that $z = z' \notin \mathrm{var}(t) \cup \mathrm{var}(t') \cup \mathrm{var}(t'')$. By induction, we have $u_{z/y} \equiv_\alpha u''_{z/y''}$ and hence $t \equiv_\alpha t''$. ∎

Definition 1.11 (Λ):
We define $\Lambda = L/\equiv_\alpha$.

When $t \in L$, we denote by $\underline{t} \in \Lambda$ its $\equiv_\alpha$-class.
We now wish to show that simple substitution induces a well defined notion of substitution on Λ. But first, let us show that the length is preserved under $\equiv_\alpha$:

Lemma 1.12:
Let t and $t' \in L$ be such that $t \equiv_\alpha t'$, then $|t| = |t'|$.

Proof. We proceed by induction on $|t|$. If $t = x$ then $t' = x$ and $|t| = 1 = |t'|$. If $t = u(v)$, then $t' = u'(v')$ such that $u' \equiv_\alpha u$ and $v' \equiv_\alpha v$. Then $|t| = 2 + |u| + |v| = 2 + |u'| + |v'| = |t'|$. Finally, if

$t = \lambda x\, u$, then $t' = \lambda x'\, u'$ and for some $y \notin \mathrm{var}(t) \cup \mathrm{var}(t')$, $u_{y/x} \equiv_\alpha u'_{y'/x'}$. It follows that, by Lemma **(1.2)**, $|t| = 2 + |u| = 2 + |u_{y/x}| = 2 + |u'_{y'/x'}| = 2 + |u'| = |t'|$. ∎

Lemma 1.13:
Let t, t' and $s \in L$ and $x \in V$. If $t \equiv_\alpha t'$ and $\mathrm{fvar}(s) \cap (\mathrm{bvar}(t) \cup \mathrm{bvar}(t')) = \varnothing$, then $t_{s/x} \equiv_\alpha t'_{s/x}$.

Proof. We proceed by induction on the length of t. The proof is completely straightforward if $t = x \in V$ or $t = (u)v$. Let us assume that $t = \lambda y\, u$ and $t' = \lambda y'\, u'$. If $y = x$, then x is not free in t. By Lemma **(1.8)**, x is not free in t' either and hence, by Lemma **(1.3)**, $t_{s/x} = t \equiv_\alpha t' = t'_{s/x}$. The case $y' = x$ is symmetric. So we can now assume that x is neither y nor y'. Let $z \notin \mathrm{var}(t) \cup \mathrm{var}(t') \cup \mathrm{var}(s)$. By definition of $t \equiv_\alpha t'$, $u_{z/y} \equiv_\alpha u'_{z/y'}$ and hence, by induction, $(u_{z/y})_{s/x} \equiv_\alpha (u'_{z/y'})_{s/x}$. Because y is bounded in t, it cannot be free in s (similarly for y') and $x \neq z$ so by Corollary **(1.6)**, $(u_{s/x})_{z/y} = (u_{z/y})_{s/x} \equiv_\alpha (u'_{z/y'})_{s/x} = (u'_{s/x})_{z/y'}$, so $t_{s/x} = \lambda y\, u_{s/x} \equiv_\alpha \lambda y'\, u'_{s/x} = t'_{s/x}$. ∎

Lemma 1.14:
Let t, s and $s' \in L$ and $x \in V$. If $s \equiv_\alpha s'$, then $t_{s/x} \equiv_\alpha t_{s'/x}$.

Proof. We proceed by induction on t. The proof is completely straightforward if $t = x \in V$ or $t = (u)v$. Let us assume that $t = \lambda y\, u$. If $y = x$, then $t_{s/x} = t = t_{s'/x}$. So we can assume $y \neq x$. By induction, $u_{s/x} \equiv_\alpha u_{s'/x}$. Let $z \notin \mathrm{fvar}(t) \cup \mathrm{fvar}(s) \cup \mathrm{fvar}(s')$. By Lemma **(1.13)**, $(u_{s/x})_{z/y} \equiv_\alpha (u_{s'/x})_{z/y}$ and hence $t_{s/x} \equiv_\alpha t_{s'/x}$. ∎

Corollary 1.15:
Let t, t', s and $s' \in L$ and $x \in V$. If $t \equiv_\alpha t'$ and $\mathrm{fvar}(s) \cap (\mathrm{bvar}(t) \cup \mathrm{bvar}(t')) = \varnothing$, then $t_{s/x} \equiv_\alpha t'_{s'/x}$.

Proof. By Lemma **(1.13)**, $t_{s/x} \equiv_\alpha t'_{s/x}$ and by Lemma **(1.14)**, $t'_{s/x} \equiv_\alpha t'_{s'/x}$. We conclude by transitivity of $\equiv_\alpha$. ∎

Lemma 1.16:
Let $t \in L$ and $X \subseteq V$ be finite. There exists $t' \in L$ such that $\mathrm{bvar}(t') \cap X = \varnothing$ and $t' \equiv_\alpha t$.

Proof. We proceed by induction on t. The proof is completely straightforward if $t = x \in V$ or $t = (u)v$. Let us assume that $t = \lambda x\, u$. By induction, there exists u' such that $u' \equiv_\alpha u$ and $\mathrm{bvar}(u') \cap X = \varnothing$. Let $y \notin X \cup \mathrm{var}(u')$, we want to show that $t \equiv_\alpha \lambda y\, u'_{y/x}$. Let $z \notin \mathrm{var}(t) \cup \mathrm{var}(u') \cup \{y\}$. By Lemma **(1.13)**, $u_{z/x} \equiv_\alpha u'_{z/x}$ and, by Lemma **(1.5)**, $(u'_{y/x})_{z/y} = (u'_{z/y})_{y_{z/y}/x} = u'_{z/x}$ as y is not free in u'. So $u_{z/x} \equiv_\alpha (u'_{y/x})_{z/y}$ and $t \equiv_\alpha \lambda y\, u'_{y/x}$. ∎

We can now define a substitution operation on Λ:

Definition 1.17:
Let $\underline{t}$, $\underline{s} \in \Lambda$ and $x \in V$. We define $\underline{t}_{\underline{s}/x}$ to be the class of $t'_{s'/x}$ where $t' \in \underline{t}$ and $s' \in \underline{s}$ are such that $\mathrm{fvar}(s') \cap \mathrm{bvar}(t') = \varnothing$.

Note that this is well defined by Corollary **(1.15)** and Lemma **(1.16)**.

What we have done in this section is formalize the idea that whenever we pick a λ-term, we can always assume that the bound variables avoid any set of variables we wish (usually, the variables that are free in other λ-terms). In fact, we have essentially made the bound variables nameless.
Note that the map $(\underline{u}, \underline{v}) \mapsto \overline{(u)v}$ and the map $\underline{u} \mapsto \underline{\lambda x\, u}$ are well defined. So the notations $(\underline{u})\underline{v}$ and $\lambda x\, \underline{u}$ make sense. The map $x \mapsto \underline{x}$ is injective, so we will identify x and $\underline{x}$. The maps $\mathrm{fvar}(t)$ and $|t|$ are also well defined on Λ, by Lemmas **(1.12)** and **(1.8)**. Lemmas **(1.3)** and **(1.5)** and **(1.6)** also remain true of the substitution on Λ. Moreover Lemma **(1.5)** can be improved slightly:

Lemma 1.18:
Let $\underline{t}$, $\underline{s}$, $\underline{u} \in \Lambda$ and $x \neq y \in V$. Assume that $x \notin \mathrm{fvar}(\underline{u})$ then $(\underline{t}_{\underline{s}/x})_{\underline{u}/y} = (\underline{t}_{\underline{u}/y})_{\underline{s}_{\underline{u}/y}/x}$.

Proof. By Lemma **(1.16)**, we can find $u' \in \underline{u}$ such that $\mathrm{bvar}(u') \cap (\mathrm{fvar}(\underline{s}) \cup \mathrm{fvar}(\underline{u})) = \varnothing$, $s' \in \underline{s}$ such that $\mathrm{bvar}(s') \cap \mathrm{fvar}(\underline{u}) = \varnothing$ and $t' \in \underline{t}$ such that $\mathrm{bvar}(t') \cap (\mathrm{fvar}(\underline{s}) \cup \mathrm{fvar}(\underline{u}) \cup \{y\})$. Then $(\underline{t}_{\underline{s}/x})_{\underline{u}/y}$ is the class of $(t'_{s'/x})_{u'/y}$ and $(\underline{t}_{\underline{u}/y})_{\underline{s}_{\underline{u}/y}/x}$ is the class of $(t'_{u'/y})_{s'_{u'/y}/x}$ (to be complete, by Lemma **(1.4)**, $\mathrm{bvar}(t'_{s'/x}) \subseteq \mathrm{bvar}(t') \cup \mathrm{bvar}(s')$, $\mathrm{bvar}(t'_{u'/y}) \subseteq \mathrm{bvar}(t') \cup \mathrm{bvar}(u')$ and $fvar(s_{u/y}) \subseteq \mathrm{fvar}(s) \cup \mathrm{fvar}(u)$, so we do have $\mathrm{bvar}(t'_{s'/x}) \cup \mathrm{fvar}(u') = \varnothing$ and $\mathrm{bvar}(t'_{u'/y}) \cap \mathrm{fvar}(s'_{u'/y}) = \varnothing$). We can now conclude by Lemma **(1.5)**. ■

To conclude this section, let us prove this very reasonable lemma:

Lemma 1.19:
Let $\underline{t} \in \Lambda$ and $x, y \in V$. Assume that $y \notin \mathrm{fvar}(\underline{t})$, then $\lambda y\, \underline{t}_{y/x} = \lambda x\, \underline{t}$.

Proof. We may assume that $y \notin \mathrm{bvar}(t)$. Then $\underline{t}_{y/x} = \underline{t_{y/x}}$ and we have to show that $\lambda y\, t_{y/x} \equiv_\alpha \lambda x\, t$. Let $z \notin \mathrm{var}(t) \cup \{x, y\}$, by Lemma **(1.5)**, $(t_{y/x})_{z/y} = (t_{z/y})_{y_{z/y}/x} = t_{z/x}$ so $(t_{y/x})_{z/y} \equiv_\alpha t_{z/x}$, which concludes the proof. ■

2 β-reduction

Definition 2.1 (β-reduction)**:**

- *Let $\underline{t} \in \Lambda$. We say that $\underline{t}$ is a redex if there exists $x \in V$, $\underline{u}$ and $\underline{v} \in \Lambda$ such that $\underline{t} = (\lambda x\, \underline{u})\underline{v}$.*
- *Let $\beta = \{((\lambda x\, \underline{u})\underline{v}, \underline{u}_{\underline{v}/x}) : x \in V, \underline{u} \text{ and } \underline{v} \in \Lambda\}$.*
  We will write $\underline{t}\beta\underline{u}$ instead of $(\underline{t}, \underline{u}) \in \beta$.
- *Let $\underline{t} \to_\beta \underline{u}$ be defined by induction on $|\underline{t}|$:*
  - *If $\underline{t}\beta\underline{u}$ then $\underline{t} \to_\beta \underline{u}$;*
  - *If $\underline{t} = (\underline{s})\underline{w}$, $\underline{u} = (\underline{v})\underline{w}$ and $\underline{s} \to_\beta \underline{v}$, then $\underline{t} \to_\beta \underline{u}$;*
  - *If $\underline{t} = (\underline{w})\underline{s}$, $\underline{u} = (\underline{w})\underline{v}$ and $\underline{s} \to_\beta \underline{v}$, then $\underline{t} \to_\beta \underline{u}$;*
  - *If $\underline{t} = \lambda x\, \underline{s}$, $\underline{u} = \lambda x\, \underline{v}$ and $\underline{s} \to_\beta \underline{v}$, then $\underline{t} \to_\beta \underline{u}$.*
- *Let $\underline{t} \to_\beta^\star \underline{u}$ hold if there exists $k \in \mathbb{N}$ and $(\underline{v}_i)_{0 \leqslant i \leqslant k}$ such that $\underline{v}_i \to_\beta \underline{v}_{i+1}$, $\underline{v}_0 = \underline{t}$ and $\underline{v}_k = \underline{u}$.*

The relation $\underline{t} \to_\beta \underline{u}$ means that we can obtain u from t by reducing a redex somewhere in t. The relation $\to_\beta^\star$ is its reflexive and transitive closure.

Example 2.2:

1. $(\lambda x\, x)t \to_\beta t$;
2. Let $\Delta = \lambda x\, (x)x$, then $(\Delta)t \to_\beta (t)t$, in particular $(\Delta)\Delta \to_\beta (\Delta)\Delta$;
3. Let $\Omega = \lambda x\, ((x)x)x$, then $(\Omega)\Omega \to_\beta ((\Omega)\Omega)\Omega \to_\beta (((\Omega)\Omega)\Omega)\Omega \to_\beta \cdots$. In fact let Ω^k be defined by $\Omega^1 = \Omega$ and $\Omega^{k+1} = (\Omega^k)\Omega$, then $(\Omega)\Omega \to_\beta^\star \Omega^k$ for all $k > 0$.
4. $$\begin{array}{lll} ((\lambda x\, x)\lambda y\, (y)y)(\lambda z\, z)\lambda w\, w & \to_\beta & (\lambda y\, (y)y)(\lambda z\, z)\lambda w\, w \\ & \to_\beta & ((\lambda z\, z)\lambda w\, w)(\lambda z\, z)\lambda w\, w \\ & \to_\beta & (\lambda w\, w)(\lambda z\, z)\lambda w\, w \\ & \to_\beta & (\lambda z\, z)\lambda w\, w \\ & \to_\beta & \lambda w\, w. \end{array}$$

   But we also have

$$\begin{array}{lll} ((\lambda x\, x)\lambda y\, (y)y)(\lambda z\, z)\lambda w\, w & \to_\beta & ((\lambda x\, x)\lambda y\, (y)y)\lambda w\, w \\ & \to_\beta & (\lambda y\, (y)y)\lambda w\, w \\ & \to_\beta & (\lambda w\, w)\lambda w\, w \\ & \to_\beta & \lambda w\, w. \end{array}$$

Definition 2.3:
Let $\underline{t} \in \Lambda$, we define the set $\mathrm{sub}(\underline{t})$ of subterms of t by induction on (the length of) $\underline{t}$:

- *If $\underline{t} = x$, then $\mathrm{sub}(\underline{t}) = \{x\}$;*
- *If $\underline{t} = (\underline{u})\underline{v}$, then $\mathrm{sub}(\underline{t}) = \mathrm{sub}(\underline{u}) \cup \mathrm{sub}(\underline{v}) \cup \{\underline{t}\}$;*
- *If $\underline{t} = \lambda x\, \underline{u}$, then $\mathrm{sub}(\underline{t}) = \mathrm{sub}(\underline{u}) \cup \{\underline{t}\}$.*

We say that $\underline{t}$ contains $\underline{u}$ if $\underline{u} \in \mathrm{sub}(\underline{t})$.

Definition 2.4 (Normal form)**:**

- *A term $\underline{t} \in \Lambda$ is normal if it does not contain any redexes.*
- *A term $\underline{t} \in \Lambda$ is normalizable if there exists a normal term $\underline{t}'$ such that $\underline{t} \to_\beta^\star \underline{t}'$.*
- *A term $\underline{t} \in \Lambda$ is strongly normalizable if there is no infinite sequence $(\underline{t}_i)_{i \in \mathbb{N}}$ such that $\underline{t}_0 = \underline{t}$ and $\underline{t}_i \to_\beta \underline{t}_{i+1}$.*

Note that a normal term $\underline{t}$ cannot be reduced further: if $\underline{t} \to_\beta^\star \underline{t}'$ then $\underline{t} = \underline{t}'$. The converse is false as $\Delta\Delta$ only reduces to itself but is not in normal form.
The β-reduction should really be conceived as a form of computation and normal forms are the result of that computation. A strongly normalizable λ-term is a computation that always terminates no matter how we proceed with the computation (note that β-reduction is a very

concurrent form of computation, reductions can happen 'at the same time' in distinct places independently). A normalizable term is a computation that terminates if we are careful. Note that a strongly normalizable term is in particular normalizable.

Example 2.5:

1. The terms $\lambda x\, x$, $\lambda x\, y$, Δ and Ω are normal.
2. The term $(\Delta)\Delta$ is not normalizable, nor is $(\Omega)\Omega$.
3. The term $(\lambda x\, y)(\Delta)\Delta$ is normalizable but not strongly normalizable.

Eventually we will try to isolate and describe certain classes of strongly normalizing λ-terms, but before we do that, we will try to formalize and prove the idea that the order in which we reduce redexes does not matter:

Theorem 2.6 (Church-Rosser)**:**

Let $\underline{t}$, $\underline{s}_1$ and $\underline{s}_2 \in \Lambda$. Assume that $\underline{t} \to_\beta^\star \underline{s}_1$ and $\underline{t} \to_\beta^\star \underline{s}_2$, then there exists $\underline{u} \in \Lambda$ such that $\underline{s}_1 \to_\beta^\star \underline{u}$ and $\underline{s}_2 \to_\beta^\star \underline{u}$.

We say that $\to_\beta^\star$ is confluent.
The idea of the proof is the following: first prove that if $\underline{t} \to_\beta \underline{s}_1$ and $\underline{t} \to_\beta \underline{s}_2$, then there exists $\underline{u} \in \Lambda$ such that $\underline{s}_1 \to_\beta^\star \underline{u}$ and $\underline{s}_2 \to_\beta^\star \underline{u}$ (we say that $\to_\beta$ is locally confluent) and then prove that the theorem follows. The main problem of this proof is that in general locally confluent relations may not have confluent transitive closure. What is true, however, is that a confluent relation has a confluent transitive closure. But $\to_\beta$ is not confluent. So we introduce another reduction relation $\to_\rho$ who is confluent and whose transitive closure is $\to_\beta^\star$.

Definition 2.7:

- *Let $\underline{t} \to_\rho \underline{t}'$ be defined by induction on (the length of) $\underline{t}$:*
  - $x \to_\rho x$;
  - *If $\underline{u} \to_\rho \underline{u}'$ and $\underline{v} \to_\rho \underline{v}'$, then $(\underline{u})\underline{v} \to_\rho (\underline{u}')\underline{v}'$;*
  - *If $\underline{u} \to_\rho \underline{u}'$, then $\lambda x\, \underline{u} \to_\rho \lambda x\, \underline{u}'$.*
  - *It $\underline{u} \to_\rho \underline{u}'$ and $\underline{v} \to_\rho \underline{v}'$, then $(\lambda x\, \underline{u})\underline{v} \to_\rho \underline{u}'_{\underline{v}'/x}$.*
- *Let $\underline{t} \to_\rho^\star \underline{t}'$ hold if there exists $k \in \mathbb{N}$ and $(\underline{u}_i)_{0 \leqslant i \leqslant k}$ such that $\underline{u}_i \to_\beta \underline{u}_{i+1}$, $\underline{v}_0 = \underline{t}$ and $\underline{v}_k = \underline{u}$.*

The intuitive meaning of $\underline{t} \to_\rho \underline{u}$ is that a number of redexes *occurring* in $\underline{t}$ have been reduced to obtain $\underline{u}$.

Lemma 2.8:
The relation $\to_\rho$ is reflexive.

Proof. This is immediate by induction. ∎

Lemma 2.9:
Let $\underline{u}$, $\underline{u}'$, $\underline{v}$ and $\underline{v}' \in \Lambda$. If $\underline{u} \to_\rho \underline{u}'$ and $\underline{v} \to_\rho \underline{v}'$, then $\underline{u}_{\underline{v}/x} \to_\rho \underline{u}'_{\underline{v}'/x}$.

Proof. We proceed by induction on $\underline{u}$. If $\underline{u} = y \in V$, then $\underline{u}' = y \in V$. If $y \neq x$, then $\underline{u}_{\underline{v}/x} = y = \underline{u}_{\underline{v}'/x}$. If $y = x$, $\underline{u}_{\underline{v}/x} = \underline{v} \to_\rho \underline{v}' = \underline{u}'_{\underline{v}'/x}$.
If $\underline{u} = \lambda y\, \underline{t}$, then $\underline{u}' = \lambda y\, \underline{t}'$ and $\underline{t} \to_\rho \underline{t}'$. We may assume $y \neq x$. By induction $\underline{t}_{\underline{v}/x} \to_\rho \underline{t}'_{\underline{v}'/x}$ and hence $\underline{u}_{\underline{v}/x} = \lambda y\, \underline{t}_{\underline{v}/x} \to_\rho \lambda y\, \underline{t}'_{\underline{v}'/x} = \underline{u}'_{\underline{v}'/x}$.
If $\underline{u} = (\underline{t})\underline{s}$, then there are two possibilities. Let us first assume that $\underline{u}' = (\underline{t}')\underline{s}'$, $\underline{t} \to_\rho \underline{t}'$ and $\underline{s} \to_\rho \underline{s}'$. Then, by induction $\underline{t}_{\underline{v}/x} \to_\rho \underline{t}'_{\underline{v}'/x}$ and $\underline{s}_{\underline{v}/x} \to_\rho \underline{s}'_{\underline{v}'/x}$ so $\underline{u}_{\underline{v}/x} = (\underline{t}_{\underline{v}/x})\underline{s}_{\underline{v}/x} \to_\rho (\underline{t}'_{\underline{v}'/x})\underline{s}'_{\underline{v}'/x} = \underline{u}'_{\underline{v}'/x}$.
Otherwise $\underline{u} = (\lambda y\, \underline{t})\underline{s}$ and $\underline{u}' = \underline{t}'_{\underline{s}'/y}$ where $\underline{t} \to_\rho \underline{t}'$ and $\underline{s} \to_\rho \underline{s}'$. We may assume $y \neq x$ and $y \notin \mathrm{fvar}(\underline{v}')$. By induction, $\underline{t}_{\underline{v}/x} \to_\rho \underline{t}'_{\underline{v}'/x}$ and $\underline{s}_{\underline{v}/x} \to_\rho \underline{s}'_{\underline{v}'/x}$ so $\underline{u}_{\underline{v}/x} = (\lambda y\, \underline{t}_{\underline{v}/x})\underline{s}_{\underline{v}/x} \to_\rho (\underline{t}'_{\underline{v}'/x})_{\underline{s}'_{\underline{v}'/x}/y} = (\underline{t}'_{\underline{s}'/y})_{\underline{v}'/x} = \underline{u}'_{\underline{v}'/x}$. The one before last equality follows from Lemma **(1.18)** as $y \notin \mathrm{fvar}(\underline{v}')$. ∎

Lemma 2.10:
Let $\underline{t}$, $\underline{s}_1$ and $\underline{s}_2 \in \Lambda$. Assume that $\underline{t} \to_\rho \underline{s}_1$ and $\underline{t} \to_\rho \underline{s}_2$, then there exists $\underline{u} \in \Lambda$ such that $\underline{s}_1 \to_\rho \underline{u}$ and $\underline{s}_2 \to_\rho \underline{u}$.

Proof. We proceed by induction on $\underline{t}$. If $\underline{t} = x$, then $\underline{s}_1 = \underline{s}_2 = x$ and it suffices to take $\underline{u} = x$.
If $\underline{t} = \lambda x\, \underline{v}$, then $\underline{s}_i = \lambda x\, \underline{w}_i$ and $\underline{v} \to_\rho \underline{w}_i$. By induction there exists $\underline{u}$ such that $\underline{v}_i \to_\rho \underline{u}$, for $i = 1, 2$. Then $\underline{s}_i = \lambda x\, \underline{w}_i \to_\rho \lambda x\, \underline{u}$.
If $\underline{t} = (\underline{v})\underline{w}$, there are numerous cases to consider. First, $\underline{s}_i = (\underline{a}_i)\underline{b}_i$ for $i = 1, 2$ where $\underline{v} \to_\rho \underline{a}_i$ and $\underline{w} \to_\rho \underline{b}_i$. By induction, there exists $\underline{c}$ and $\underline{d}$ such that $\underline{a}_i \to_\rho \underline{c}$ and $\underline{b}_i \to_\rho \underline{d}$. Then $\underline{s}_i = (\underline{a}_i)\underline{b}_i \to_\rho (\underline{c})\underline{d}$.
If $\underline{t} = (\lambda x\, \underline{v})\underline{w}$ and $\underline{s}_i = (\underline{a}_i)_{\underline{b}_i/x}$ for $i = 1, 2$, where $\underline{v} \to_\rho \underline{a}_i$ and $\underline{w} \to_\rho \underline{b}_i$. By induction, there exists $\underline{c}$ and $\underline{d}$ such that $\underline{a}_i \to_\rho \underline{c}$ and $\underline{b}_i \to_\rho \underline{d}$. Then, by Lemma **(2.9)**, $\underline{s}_i = (\underline{a}_i)_{\underline{b}_i/x} \to_\rho \underline{c}_{\underline{d}/x}$.
Finally, by symmetry, there only remains the case $\underline{t} = (\lambda x\, \underline{v})\underline{w}$, $\underline{s}_1 = (\underline{a}_1)_{\underline{b}_1/x}$ and $\underline{s}_2 = (\lambda x\, \underline{a}_2)\underline{b}_2$ where $\underline{v} \to_\rho \underline{a}_i$ and $\underline{w} \to_\rho \underline{b}_i$. By induction, there exist $\underline{c}$ and $\underline{d}$ such that $\underline{a}_i \to_\rho \underline{c}$ and $\underline{b}_i \to_\rho \underline{d}$. We have $\underline{s}_1 \to_\rho \underline{c}_{\underline{d}/x}$ by Lemma **(2.9)** and $\underline{s}_2 \to_\rho \underline{c}_{\underline{d}/x}$ by definition. ∎

Lemma 2.11:
Let $\underline{t}$, $\underline{s}_1$ and $\underline{s}_2 \in \Lambda$. Assume that $\underline{t} \to_\rho^\star \underline{s}_1$ and $\underline{t} \to_\rho \underline{s}_2$, then there exists $\underline{u} \in \Lambda$ such that $\underline{s}_1 \to_\rho \underline{u}$ and $\underline{s}_2 \to_\rho^\star \underline{u}$.

Proof. By definition of $\underline{t} \to_\rho^\star \underline{s}_1$, there exists $\underline{v}_i$ such that $\underline{v}_i \to_\rho \underline{v}_{i+1}$, $\underline{v}_0 = \underline{t}$ and $\underline{v}_k = \underline{s}_1$. We proceed by induction on k. If $k = 0$, then we can take $\underline{u} = \underline{s}_2$.
Otherwise, by Lemma **(2.10)**, there exists $\underline{w}$ such that $\underline{v}_1 \to_\rho \underline{w}$ and $\underline{s}_2 \to_\rho \underline{w}$. So now we have $\underline{v}_1 \to_\rho^\star \underline{s}_1$ in $k - 1$ steps so, by induction, there exists $\underline{u}$ such that $\underline{s}_1 \to_\rho \underline{u}$ and $\underline{w} \to_\rho^\star \underline{u}$ and hence $\underline{s}_2 \to_\rho^\star \underline{u}$. ∎

Lemma 2.12:
Let $\underline{t}$, $\underline{s}_1$ and $\underline{s}_2 \in \Lambda$. Assume that $\underline{t} \to_\rho^\star \underline{s}_1$ and $\underline{t} \to_\rho^\star \underline{s}_2$, then there exists $\underline{u} \in \Lambda$ such that $\underline{s}_1 \to_\rho^\star \underline{u}$ and $\underline{s}_2 \to_\rho^\star \underline{u}$.

Proof. By definition of $\underline{t} \to_\rho^\star \underline{s}_2$, there exists $\underline{v}_i$ such that $\underline{v}_i \to_\rho \underline{v}_{i+1}$, $\underline{v}_0 = \underline{t}$ and $\underline{v}_k = \underline{s}_2$. We proceed by induction on k. If $k = 0$, then we can take $\underline{u} = \underline{s}_1$.

Otherwise, by Lemma **(2.11)**, there exists $\underline{w}$ such that $\underline{v}_1 \to_\rho^\star \underline{w}$ and $\underline{s}_1 \to_\rho \underline{w}$. So now we have $\underline{v}_1 \to_\rho^\star \underline{s}_2$ in $k-1$ steps so, by induction, there exists $\underline{u}$ such that $\underline{s}_2 \to_\rho^\star \underline{u}$ and $\underline{w} \to_\rho^\star \underline{u}$ and hence $\underline{s}_1 \to_\rho^\star \underline{u}$. ■

Lemma 2.13:
Let $\underline{t}, \underline{u} \in \Lambda$. If $\underline{t} \to_\beta \underline{u}$, then $\underline{t} \to_\rho \underline{u}$.

Proof. We proceed by induction on $\underline{t}$. If $\underline{t} \beta \underline{u}$, then $\underline{t} = (\lambda x\, \underline{s})\underline{v}$ and $\underline{u} = \underline{s}_{\underline{v}/x}$. By Lemma **(2.8)**, we have $\underline{s} \to_\rho \underline{s}$ and $\underline{v} \to_\rho \underline{v}$.It follows that $(\lambda x\, \underline{s})\underline{v} \to_\rho \underline{s}_{\underline{v}/x}$.
If $\underline{t} = \lambda x\, \underline{s}$, then $\underline{u} = \lambda x\, \underline{v}$ where $\underline{s} \to_\beta \underline{v}$. By induction $\underline{s} \to_\rho \underline{v}$ and hence $\lambda x\, \underline{s} \to_\rho \lambda x\, \underline{v}$.
The two remaining cases are $\underline{t} = (\underline{s})\underline{w}$ and $\underline{u} = (\underline{v})\underline{w}$ or $\underline{t} = (\underline{w})\underline{s}$ and $\underline{u} = (\underline{w})\underline{v}$ where $\underline{s} \to_\beta \underline{v}$. By induction, $\underline{s} \to_\rho \underline{v}$, by Lemma **(2.8)** $\underline{w} \to_\rho \underline{w}$ and hence $(\underline{s})\underline{w} \to_\rho (\underline{v})\underline{w}$ and $(\underline{w})\underline{s} \to_\rho (\underline{w})\underline{v}$.
■

Lemma 2.14:
Let $\underline{t}, \underline{u} \in \Lambda$. If $\underline{t} \to_\rho \underline{u}$, then $\underline{t} \to_\beta^\star \underline{u}$.

Proof. We proceed by induction on $\underline{t}$. If $\underline{t} = x$, then $\underline{u} = x$ and $x \to_\beta^\star x$. If $\underline{t} = \lambda x\, \underline{s}$, then $\underline{u} = \lambda x\, \underline{v}$ and $\underline{s} \to_\rho \underline{v}$. By induction $\underline{s} \to_\beta^\star \underline{v}$ and hence $\underline{t} = \lambda x\, \underline{s} \to_\beta^\star \lambda x\, \underline{v} = \underline{u}$ (this follows immediately from the fact that if $\underline{v}_i \to_\beta \underline{v}_{i+1}$, then $\lambda x\, \underline{v}_i \to_\beta \lambda x\, \underline{v}_{i+1}$).
If $\underline{t} = (\underline{s})\underline{v}$ and $\underline{u} = (\underline{a})\underline{b}$ where $\underline{s} \to_\rho \underline{a}$ and $\underline{v} \to_\rho \underline{b}$, by induction, $\underline{s} \to_\beta^\star \underline{a}$ and $\underline{v} \to_\beta^\star \underline{b}$, so $(\underline{s})\underline{v} \to_\beta^\star (\underline{a})\underline{v} \to_\beta^\star (\underline{a})\underline{b}$ (these two statement follow from the fact that if $\underline{v}_i \to_\beta \underline{v}_{i+1}$, then $(\underline{v}_i)\underline{v} \to_\beta (\underline{v}_{i+1})\underline{v}$ and $(\underline{a})\underline{v}_i \to_\beta (\underline{a})\underline{v}_{i+1}$).
The only remaining case is $\underline{t} = (\lambda x\, \underline{s})\underline{v}$ and $\underline{u} = \underline{a}_{\underline{b}/x}$ where $\underline{s} \to_\rho \underline{a}$ and $\underline{v} \to_\rho \underline{b}$. By induction, $\underline{s} \to_\beta^\star \underline{a}$ and $\underline{v} \to_\beta^\star \underline{b}$, and hence $(\lambda x\, \underline{s})\underline{v} \to_\beta^\star (\lambda x\, \underline{a})\underline{v} \to_\beta^\star (\lambda x\, \underline{a})\underline{b} \to_\beta \underline{a}_{\underline{b}/x}$. ■

Corollary 2.15:
Let $\underline{t}, \underline{u} \in \Lambda$. If $\underline{t} \to_\beta^\star \underline{u}$ if and only if $\underline{t} \to_\rho^\star \underline{u}$.

Proof. This follows immediately from Lemmas **(2.13)** and **(2.14)**. ■

Proof (Theorem **(2.6)**). This is immediate by Lemma **(2.12)** and Corollary **(2.15)**. ■

Definition 2.16 (β-conversion)**:**
We say that $\underline{t}$ and $\underline{u} \in \Lambda$ are β-equivalent (and we write $\underline{t} \equiv_\beta \underline{u}$) if there exists $\underline{v} \in \Lambda$ such that $\underline{t} \to_\beta^\star \underline{v}$ and $\underline{u} \to_\beta^\star \underline{v}$.

Proposition 2.17:
The relation $\equiv_\beta$ is an equivalence relation.

Proof. Reflexivity and symmetry is evident. Let us prove transitivity. Let $\underline{t}, \underline{s}, \underline{u} \in \Lambda$ be such that $\underline{t} \equiv_\beta \underline{s} \equiv_\beta \underline{u}$. By definition there exists $\underline{v}$ and $\underline{w} \in \Lambda$ such that $\underline{t} \to_\beta^\star \underline{v}$, $\underline{s} \to_\beta^\star \underline{v}$, $\underline{s} \to_\beta^\star \underline{w}$ and $\underline{u} \to_\beta^\star \underline{w}$. By Theorem **(2.6)**, there exists $\underline{a} \in \Lambda$ such that $\underline{v} \to_\beta^\star \underline{a}$ and $\underline{w} \to_\beta^\star \underline{a}$, so $\underline{t} \to_\beta^\star \underline{a}$ and $\underline{u} \to_\beta^\star \underline{a}$ and hence $t \equiv_\beta u$. ■

Remark 2.18:
Note that if u is normal and $t \equiv_\beta u$, then $t \to_\beta^\star u$. It follows that if both t and u are normal and β-equivalent they are equal. Normalizable terms are exactly those whose β-equivalence

class contains one (and only one) normal form.

3 Simply typed λ-calculus

We wish to characterize a subset of Λ that only contains strongly normalizable terms
Let us have a look at one pathological examples: $(\Delta)\Delta$ where $\Delta = \lambda x\,(x)x$. The reason this term is problematic is because x is applied to itself. If we follow our intuition that λ-terms are functions then X is both a element in a set A and a function in A^B for some set B and that should not be possible. To prevent $(\Delta)\Delta$ from being an admissible term, we are going to prevent Δ from appearing altogether (although Δ alone is strongly normalizable) by forcing terms to have *types*, that is only consider terms that have a well defined "domain" and "image".

Definition 3.1 (Type)**:**
Let W be a set (disjoint from V the set of variables). The set T of types is the smallest set of words on $W \cup \{(,), \to\}$ (where the union is supposed to be disjoint) such that:

- *$W \subseteq T$;*
- *If A and $B \in T$, then $(A \to B) \in T$.*

The elements of W are called the type variables or the basic types.

Definition 3.2 (Context)**:**
A context is a set Γ of pairs $(x : A)$ where $x \in V$ and $A \in T$ such that if $(x : A)$ and $(x : B) \in \Gamma$, then $A = B$.

A context is therefore a partial function from V to T.

Definition 3.3:
Let Γ be a context, $A \in T$ and $t \in \Lambda$. We define $\Gamma \vdash t : A$ to be the smallest relation such that:

- *If $x \in V$, then $\Gamma \cup \{x : A\} \vdash x : A$;*
- *If $x \in V$ does not appear in Γ and $\Gamma \cup \{x : A\} \vdash t : B$ then $\Gamma \vdash \lambda x\, t : A \to B$;*
- *If $\Gamma_1 \vdash t : A \to B$ and $\Gamma_2 \vdash u : A$ then $\Gamma_1 \cup \Gamma_2 \vdash (t)u : B$ (provided $\Gamma_1 \cup \Gamma_2$ is indeed a context).*

If $\Gamma \vdash t : A$, we say that t has type A in the context Γ. As before, we will call a witness that a typing statement holds a *typing derivation*. We can also write those using inference rules:

$$(\mathrm{Ax})\ \frac{}{\Gamma \cup \{x : A\} \vdash x : A}$$

$$(\to_I)\ \frac{\Gamma \cup \{x : A\} \vdash t : B}{\Gamma \vdash \lambda x\, t : A \to B} \qquad (\to_E)\ \frac{\Gamma_1 \vdash t : A \to B \qquad \Gamma_2 \vdash u : A}{\Gamma_1 \cup \Gamma_2 \vdash (t)u : B}$$

Note that, depending on the form of t, only one rule applies. So looking at t one can predict what was the last rule applied in a typing derivation of t.
If we only keep the types (and hence only look at what is in black), we recognize a fragment of the deduction rules for propositional logic. This analogy is called the Curry-Howard correspondence. Here is a table that summarizes the aspects of the analogy that will be relevant to us:

λ-calculus	Logic
Types	Formulas
Typing statements	Deductions
Typing derivations	Deduction derivations
β-reduction	Proof simplification
Normal λ-terms	Normal proofs

Note that removing the terms is a completely harmless thing to do. Given a deduction derivation, we can reconstruct a (unique) λ-term whose typing derivation would correspond to the deduction derivation we started from.
Let us now give some examples:

$$\dfrac{\dfrac{}{x : A \vdash x : A}\,(\mathrm{Ax})}{\vdash \lambda x\, x : A \to A}\,(\to_I)$$

This is both a typing derivation for the identity λ-term and a proof of reflexivity of $\to$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y : B \to C \vdash y : B \to C}\,(\mathrm{Ax}) \qquad \dfrac{\dfrac{}{x : A \to B \vdash y : A \to B}\,(\mathrm{Ax}) \quad \dfrac{}{z : A \vdash z : A}\,(\mathrm{Ax})}{\{x : A \to B, z : A\} \vdash (x)z : B}}{\{x : A \to B, y : B \to C, z : A\} \vdash (y)(x)z : C}\,(\to_I)}{\{x : A \to B, y : B \to C\} \vdash \lambda z\,(y)(x)z : A \to C}\,(\to_I)}{x : A \to B \vdash \lambda y \lambda z\,(y)(x)z : (B \to C) \to (A \to C)}\,(\to_I)}{\vdash \lambda x \lambda y \lambda z\,(y)(x)z : (A \to B) \to ((B \to C) \to (A \to C))}\,(\to_I)$$

This is a derivation of the transitivity of $\to$.

An obvious consequence of the fact that the typing rules correspond to deduction rules for propositional logic is that T can be seen as a subset of the set of propositional variables with variables in W and that if $\vdash t : \varphi$ then $\vDash \varphi$, i.e. φ is a tautology. However, the converse is not true even if we consider only formulas containing the connective $\to$. In fact, the Curry-Howard correspondence relates λ-calculus and intuitionistic logic (in the case of simply typed λ-calculus, the implicative fragment of intuitionistic logic).

Lemma 3.4:
Let $\Gamma \subseteq \Gamma'$ be contexts, $t \in \lambda$ and $A \in T$. If $\Gamma \vdash t : A$, then $\Gamma' \vdash t : A$.

Proof. We proceed by induction on t. If $t = x$ then the last rule applied must be (Ax), so $x : A \in \Gamma \subseteq \Gamma'$ so, applying (Ax), $\Gamma' \vdash x : A$.

If $t = (u)v$, then the last applied ruled must be $(\to_E)$ so there exists $C \in T$ and contexts Γ_1 and Γ_2 such that $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \vdash u : C \to A$ and $\Gamma_2 \vdash v : C$. By induction $\Gamma' \vdash u : C \to A$ and $\Gamma' \vdash v : C$ and, applying $(\to_E)$, we have $\Gamma' \vdash (u)v : A$.
Finally, if $t = \lambda x\, u$ (we may assume $x \notin \Gamma'$), then the last applied rule must be $(\to_I)$ and there exists C, $D \in T$ such that $A = C \to D$ and $\Gamma \cup \{x : C\} \vdash u : D$. By induction $\Gamma' \cup \{x : C\} \vdash u : D$ and, applying $(\to_I)$, $\Gamma' \vdash \lambda x\, u : C \to D$. ■

Lemma 3.5:
Let $t, u \in \Lambda$, Γ be a context, $A, B \in T$ and $x \in V$. Assume that $\Gamma \cup \{x : B\} \vdash t : A$ and $\Gamma \vdash u : B$, then $\Gamma \vdash t_{u/x} : A$.

Proof. We proceed by induction on t. If $t = x$, then because $\Gamma \cup \{x : B\} \vdash t : A$, we must have $A = B$. As $t_{u/x} = u$, we have $\Gamma \vdash t_{u/x} : A$. If $t = y \neq x$, then $t_{u/x} = t = y$ and we have $\Gamma \cup \{x : B\} \vdash y : A$, so $y : A \in \Gamma$ and $\Gamma \vdash t_{u/x} : A$.
If $t = (s)v$, then the last rule applied was $(\to_E)$ and there exists $C \in T$ and contexts Γ_1 and Γ_2 such that $\Gamma_1 \cup \Gamma_2 = \Gamma \cup \{x : B\}$, $\Gamma_1 \vdash s : C \to A$ and $\Gamma_1 \vdash v : C$. By Lemma **(3.4)**, we can assume that $\Gamma_1 = \Gamma_2 = \Gamma \cup \{x : B\}$. By induction, $\Gamma \vdash s_{u/x} : C \to A$ and $\Gamma \vdash v_{u/x} : C$. So applying the rule $(\to_E)$, we get that $\Gamma \vdash (s_{u/x})v_{u/x} : A$ and, as $t_{u/x} = (s_{u/x})v_{u/x}$, we are done.
If $t = \lambda y\, s$ (we may assume that $y \neq x$), then the last rule applied was $(\to_I)$, so $A = C \to D$ and $\Gamma \cup \{x : B, y : C\} \vdash s : D$. By induction $\Gamma \cup \{y : C\} \vdash s_{u/x} : D$ and applying rule $(\to_E)$, $\Gamma \vdash \lambda y\, s_{u/x} : C \to D$ and that concludes the proof as $t_{u/x} = \lambda y\, s_{u/x}$. ■

Proposition 3.6:
Let $t, u \in \Lambda$, Γ be a context and $A \in T$. Let us assume that $\Gamma \vdash t : A$ and $t \to_\beta u$, then $\Gamma \vdash u : A$.

Proof. We proceed by induction on t. If $t \beta u$, then $t = (\lambda x\, s)v$ and $u = s_{v/x}$ (where we might assume that x is not free in u). The end of the derivation of $\Gamma \vdash (\lambda x\, s)v : A$ must look like:

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma_1 \cup \{x : B\} \vdash s : A}\,(\to_I)}{\Gamma_1 \vdash \lambda x\, s : B \to A}\,(\to_E) \qquad \dfrac{\vdots}{\Gamma_2 \vdash v : B}}{\Gamma_1 \cup \Gamma_2 \vdash (\lambda s\, t)v : A}$$

Where $\Gamma = \Gamma_1 \cup \Gamma_2$ and B is some type. By Lemma **(3.4)**, we can assume that $\Gamma_1 = \Gamma_2 = \Gamma$. By Lemma **(3.5)**, we have that $\Gamma \vdash s_{v/x} : A$.
If $t = (s)w$, $u = (v)w$ and $s \to_\beta v$, we must have $\Gamma \vdash s : C \to A$ for some type C and $\Gamma \vdash w : C$. By induction $\Gamma \vdash v : C \to A$ and hence $\Gamma \vdash (v)w : A$.
If $t = (w)s$, $u = (w)v$ and $s \to_\beta v$, we must have $\Gamma \vdash w : C \to A$ for some type C and $\Gamma \vdash s : C$. By induction $\Gamma \vdash v : C$ and hence $\Gamma \vdash (w)v : A$.
Finally, if $t = \lambda x\, s$, $u = \lambda x\, v$ and $s \to_\beta v$, we must have $A = C \to D$ for some types C and D and $\Gamma \cup \{x : C\} \vdash s : D$. By induction $\Gamma \cup \{x : C\} \vdash v : D$ and hence $\Gamma \vdash \lambda x\, v : C \to D$. ■

Note that if you prove B under some hypothesis Γ_1 and $B \to A$ under some hypothesis Γ_2 then Modus Ponens (or $(\to_E)$) tells you that you can prove A assuming Γ_1 and Γ_2. But another proof of A (in some sense more straightforward), would be to look at the proof of $B \to A$ and each time you use the hypothesis B replace that by the proof of B. This is

exactly the transformation on proofs associated to β-reduction. Strong normalizability of the term whose typing derivation corresponds to some deduction derivation implies that this "simplification" algorithm terminates.

Corollary 3.7:
Let $t, u \in \Lambda$, Γ be a context and $A \in T$. Let us assume that $\Gamma \vdash t : A$ and $t \to_\beta^\star u$, then $\Gamma \vdash u : A$.

Proof. This is an immediate by Proposition **(3.6)** and induction on the number of steps required to reduce t into u. ■

We now want to prove the following:

Theorem 3.8:
Let $t \in \Lambda$, Γ be a context and $A \in T$. If $\Gamma \vdash t : A$ then t is strongly normalizable.

The proof is rather convoluted and involves interpreting types as set of (strongly normalizable) terms that contain all the terms that can be given this particular type. Let $\mathcal{N}$ be the set of all strongly normalizing terms.

Definition 3.9:
Let $|A|$ be defined by induction on A:

- *If $X \in W$, $|X| = \mathcal{N}$;*
- $|A \to B| = \{t \in \Lambda :$ *for all* $u \in |A|$, $(t)u \in |B|\}$.

We now want to show that for all type A, $|A| \subseteq \mathcal{N}$ and for all $t \in \Lambda$, if $\Gamma \vdash t : A$ then $t \in |A|$. But that will require setting up some tools.

Definition 3.10:
Let $S \subseteq \Lambda$. We say that S is saturated if for all $x \in V$, t_i and $u \in \Lambda$ and all $t \in \mathcal{N}$, if $(\dots((u_{t/x})t_1)\dots)t_n \in S$ then $(\dots((\lambda x u)t)t_1\dots)t_n \in S$.

The intuition behind saturation is that S is saturated if is closed backwards under the relation $\to_\beta$ (at least under certain specific β-expansions).

Lemma 3.11:
If $S_1 \subseteq \Lambda$ is saturated, then for all $S_2 \subseteq \Lambda$, $S_2 \to S_1 = \{t \in \Lambda :$ for all $u \in S_2$, $(t)u \in S_1\}$ is saturated.

Proof. Let $t \in \mathcal{N}$, t_i, $u \in \Lambda$ and $x \in V$ such that $(\dots((u_{t/x})t_1)\dots)t_n \in S_2 \to S_1$. By definition, for all $t_{n+1} \in S_2$, $((\dots((u_{t/x})t_1)\dots)t_n)t_{n+1} \in S_1$. Since S_1 is saturated, we also have $((\dots(((\lambda x\, u)t)t_1)\dots)t_n)t_{n+1} \in S_1$. It follows that $(\dots(((\lambda x\, u)t)t_1)\dots)t_n \in S_2 \to S_1$. ■

Lemma 3.12:
Let S_1, S_1', S_2 and $S_2' \subseteq \Lambda$. If $S_1' \subseteq S_1$ and $S_2 \subseteq S_2'$ then $S_1 \to S_2 \subseteq S_1' \to S_2'$.

Proof. Pick $t \in S_1 \to S_2$ and $u \in S_1' \subseteq S_1$. By definition $(t)u \in S_2 \subseteq S_2'$ and hence $t \in S_1' \to S_2'$.
■

Lemma 3.13:
$\mathcal{N}$ is saturated.

Proof. Let us begin with the following claim:

Claim 3.14: *Let $t \in \mathcal{N}$. The set $\mathrm{red}(t) = \{u : t \to_\beta^\star u\}$ is finite.*

Proof. Assume it is not. Let us construct by induction a sequence v_i such that $v_0 = t$, $\mathrm{red}(v_i)$ is infinite and $v_i \to_\beta v_i$ contradicting the strong normalizability of t. Let us assume that v_i has been built. We have $\mathrm{red}(v_i) = \bigcup_{v_i \to_\beta u} \mathrm{red}(u)$ and there are only finitely many u such that $v_i \to_\beta u$ as there are only finitely many redexes in v_i. So, if $\mathrm{red}(v_i)$ is infinite, there exists u such that $v_i \to_\beta u$ and $\mathrm{red}(u)$ is finite. Take $v_{i+1} = u$. ⧫

Let $t \in \mathcal{N}$, $t_i, u \in \Lambda$ and $x \in V$ such that $s = (\dots((u_{t/x})t_1)\dots)t_n \in \mathcal{N}$, we have to prove that $v = (\dots(((\lambda x\, u)t)t_1)\dots)t_n \in \mathcal{N}$. We proceed by induction on $(|\mathrm{red}(t)|, |\mathrm{red}(s)|)$ ordered lexicographically.
To prove that $v \in \mathcal{N}$ it suffices to prove that for v' such that $v \to_\beta v'$, $v' \in \mathcal{N}$. Let us consider what are the possible β-reduct of v:

- First of all we could have $v' = s$ which is by hypothesis in $\mathcal{N}$.
- Second, we could have $u \to_\beta u'$ and $v' = (\dots(((\lambda x\, u')t)t_1)\dots)t_n$. By Lemma **(2.13)**, we have $u \to_\rho u'$ and by Lemma **(2.9)**, $u_{t/x} \to_\rho u'_{t/x}$, so by Lemma **(2.14)**, $u_{t/x} \to_\beta^\star u'_{t/x}$. It follows that $s \to_\beta^\star (\dots((u'_{t/x})t_1)\dots)t_n = s'$ (in particular $s' \in \mathcal{N}$). We have $s' \in \mathrm{red}(s)$ so $\mathrm{red}(s') \subseteq \mathrm{red}(s)$, but $s \notin \mathrm{red}(s')$ (that would contradict the strong normalizability of s), so $|\mathrm{red}(s')| < card\mathrm{red}(s)$. It follows by induction that $v' \in \mathcal{N}$.
- There exists $t'_i \in \Lambda$ such that $v' = (\dots((\dots(((\lambda x\, u)t)t_1)\dots)t'_i)\dots)t_n$. Then $s \to_\beta (\dots((\dots((u_{t/x})t_1)\dots)t'_i)\dots)t_n = s'$, $|\mathrm{red}(s')| < |\mathrm{red}(s)|$ and by induction $v' \in \mathcal{N}$.
- There exists $t' \in \Lambda$ such that $v' = (\dots(((\lambda x\, u)t')t_1)\dots)t_n \in \mathcal{N}$. By Lemma **(2.13)**, we have $t \to_\rho t'$ and by Lemma **(2.9)**, $u_{t/x} \to_\rho u_{t'/x}$, so by Lemma **(2.14)**, $u_{t/x} \to_\beta^\star u_{t'/x}$. It follows that $s \to_\beta^\star (\dots((u_{t'/x})t_1)\dots)t_n = s'$. In particular $s' \in \mathcal{N}$. Moreover $|\mathrm{red}(t)| < |\mathrm{red}(t')|$ and by induction (on t' and s'), we have that $v' \in \mathcal{N}$.

That concludes the proof. ■

Let $\mathcal{N}_0$ be the set of terms of the form $(\dots((x)t_1)\dots)t_n$ where $x \in V$ and $t_i \in \mathcal{N}$.

Lemma 3.15:
We have:

- $\mathcal{N}_0 \subseteq \mathcal{N}$;
- $\mathcal{N}_0 \subseteq \mathcal{N} \to \mathcal{N}_0$;
- $\mathcal{N}_0 \to \mathcal{N} \subseteq \mathcal{N}$.

Proof.

- Let $u = (\dots((x)t_1)\dots)t_n \in \mathcal{N}_0$. We proceed by induction on $\sum_i |\mathrm{red}(t_i)|$. Let $u' \in \Lambda$ be such that $u \to_\beta u'$. Then there exists t_i' such that $u' = (\dots((\dots((x)t_1)\dots)t_i')\dots)t_n$ and $t_i \to_\beta t_i'$. Then $|\mathrm{red}(t_i')| < |\mathrm{red}(t_i)|$ and hence $u' \in \mathcal{N}$ by induction. We have proved that every β-reduct of u is strongly normalizable so $u \in \mathcal{N}$.
- Pick any $u = (\dots((x)t_1)\dots)t_n \in \mathcal{N}_0$ and $t_{n+1} \in \mathcal{N}$. Then, by definition, $(u)t_{n+1} = ((\dots((x)t_1)\dots)t_n)t_{n+1} \in \mathcal{N}_0$. It follows that $u \in \mathcal{N} \to \mathcal{N}_0$.
- Pick any $u \in \mathcal{N}_0 \to \mathcal{N}$. By definition, $x \in \mathcal{N}_0$ and $(u)x \in \mathcal{N}$. We proceed by induction on $|\mathrm{red}((u)x)|$. Let $u' \in \Lambda$ be such that $u \to_\beta u'$, then $(u)x \to_\beta (u')x$ and hence $|\mathrm{red}((u')x)| < |\mathrm{red}((u)x)|$. By induction $u' \in \mathcal{N}$ and hence $u \in \mathcal{N}$.

That concludes the proof. ∎

Lemma 3.16:
Let $A \in T$. We have:

- *$|A|$ is saturated;*
- *$\mathcal{N}_0 \subseteq |A| \subseteq \mathcal{N}$.*

Proof. We proceed by induction on A. If $A \in W$, then $|A| = \mathcal{N}$ is saturated by Lemma **(3.13)** and $\mathcal{N}_0 \subseteq |A| \subseteq \mathcal{N}$, by Lemma **(3.15)**.
If $A = B \to C$, then $|A| = |B| \to |C|$ and $|C|$ is saturated by induction. Hence, so is $|A|$, by Lemma **(3.11)**. Moreover $\mathcal{N}_0 \subseteq |B|$ and $|C| \subseteq \mathcal{N}$ by induction so $|B| \to |C| \subseteq \mathcal{N}_0 \to \mathcal{N} \subseteq \mathcal{N}$ by Lemmas Lemma **(3.12)** and Lemma **(3.15)**. Similarly, $\mathcal{N}_0 \subseteq |C|$ and $|B| \subseteq \mathcal{N}$ so $\mathcal{N}_0 \subseteq \mathcal{N} \to \mathcal{N}_0 \subseteq |b| \to |C|$. ∎

Lemma 3.17:
Let $t, u_i \in \Lambda$, $\Gamma = \{(x_i : A_i) : 0 < i \leqslant n\}$ be a context and $A \in T$. Assume that $x_i \notin \bigcup_{j<i} \mathrm{fvar}(t_j)$, $\Gamma \vdash t : A$ and $u_i \in |A_i|$, then $s = (\dots(t_{u_1/x_1})\dots)_{t_n/x_n} \in |A|$.

The hypothesis $x_i \notin \bigcup_{j<i} \mathrm{fvar}(t_j)$ is just there to insure that the substitution is a simultaneous one (like we defined for propositional/predicate logic).

Proof. We proceed by induction on t. If $t = x$, then $\Gamma \vdash x : A$ so $x = x_i$ for some i, $A = A_i$ and $s = t_i \in |A_i|$ (this can be seen easily by induction on n and it is where the hypothesis on x_i is used).
If $t = \lambda x\, v$, we may assume that $x \neq x_i$ and $x \notin \mathrm{fvar}(t_i)$ for all i. Moreover, we must have $A = C \to D$ and $\Gamma \cup \{x : C\} \vdash v : D$. Pick $u \in |C|$. By induction $((\dots(v_{u_1/x_1})\dots)_{u_n/x_n})_{u/x} \in cardC$. Let $w = (\dots(v_{u_1/x_1})\dots)_{u_n/x_n}$. Because $|C|$ is saturated (see Lemma **(3.16)**), we have $s = (\lambda x\, w)u \in |C|$.
Finally, if $t = (v)w$, then there is some $C \in T$ such that $\Gamma \vdash v : C \to A$ and $\Gamma \vdash w : C$. By induction $v' = (\dots(v_{u_1/x_1})\dots)_{t_n/x_n} \in |C \to A|$ and $w' = (\dots(w_{u_1/x_1})\dots)_{t_n/x_n} |C|$, so by definition of $|C \to A|$, $s = (v')w' \in |A|$. ∎

Proof. (Theorem **(3.8)**) Let $\Gamma = \{(x_i : A_i) : 0 < i \leqslant n\}$. By Lemma **(3.16)**, $x_i \in \mathcal{N}_0 \subseteq |A_i|$ and hence, by Lemma **(3.17)**, $t = (\dots(t)_{x_1/x_1}\dots)_{x_n/x_n} \in |A| \subseteq \mathcal{N}$ (the last inclusion is proved in Lemma **(3.16)**). ∎

As discussed earlier, the logic we get by only allowing the rule (Ax), $(\to_I)$ and $(\to_E)$ is weaker than propositional logic even if we restrict to those formulas that only contain $\to$. We can now make that remark formal. We can now prove that there are no λ-term whose type is $((A \to B) \to A) \to A$ although that last formula is a tautology (often called Pierce's law).

Lemma 3.18:
Let t be a normal λ-term. Then there exists variables x and x_i and normal terms u_j such that $t = \lambda x_1 \cdot \lambda x_k \, (\dots((x)u_1)\dots)u_n$.

Proof. We proceed by induction on t. If $t = x$ then t is of the right form. If $t = \lambda x_0 s$ then by induction $s = \lambda x_1 \cdot \lambda x_k \, (\dots((x)u_1)\dots)u_n$ and $t = \lambda x_0 \lambda x_1 \cdot \lambda x_k \, (\dots((x)u_1)\dots)u_n$ is of the right form. If $t = (s)v$, then s and v are normal and s cannot be of the form $lambdax\, w$ or else t would be a redex. By induction $s = (\dots((x)u_1)\dots)u_n$ and hence $t = ((\dots((x)u_1)\dots)u_n)v$ which is of the right form. ∎

Lemma 3.19:
Let $t \in \Lambda$, $A \in T$, $x \in V$ and Γ be a context. If $\Gamma \vdash t : A$ and $x \in \mathrm{fvar}(t)$ then there is some $C \in T$ such that $(x : C) \in \Gamma$.

Proof. We proceed by induction on t. If $t = x$, the only applicable rule is (Ax) and hence $(x : A) \in \Gamma$. If $t = \lambda y \, s$, we may assume that $y \neq x$, $A = C \to D$ and we have $\Gamma \cup \{y : C\} \vdash s : D$. We have $x \in \mathrm{fvar}(t) = \mathrm{fvar}(s) \setminus \{y\}$ and hence by induction there is some $E \in T$ such that $(x : E) \in \Gamma \cup \{y : C\}$. But because $x \neq y$, we must have $(x : E) \in \Gamma$. If $t = (u)v$, then $\Gamma \vdash u : C \to A$ for some $C \in T$ and $\Gamma \vdash v : C$. As $x \in \mathrm{fvar}(t) = \mathrm{fvar}(u) \cup \mathrm{fvar}(v)$, x must be free in either u or v and by induction there is some $E \in T$ such that $(x : E) \in \Gamma$. ∎

Proposition 3.20:
Let $A, B \in T$. There is no λ-term whose type is $((A \to B) \to A) \to A$ in the empty context.

Proof. Assume such a $t \in \Lambda$ exists. By Theorem **(3.8)**, t is strongly normalizable. Let u be the (unique) normal form of t By Proposition **(3.6)**, we also have $\vdash u : ((A \to B) \to A) \to A$. So we may assume that $t = u$ is normal. By Lemma **(3.18)**, $t = \lambda x_1 \cdot \lambda x_k \, (\dots((x)u_1)\dots)u_n$ for some x and $x_i \in V$ and $u_i \in \mathcal{N}$. Because the only applicable typing rule is $(\to_I)$, we see that the type of t has to be of the form $A_1 \to (A_2 \dots \to (A_n \to A_{n+1})\dots)$ where $A_i \in T$ and hence $k = 0$ or 1. If $k = 0$, $t = (\dots((x)u_1)\dots)u_n$ and x is free in t. But, by Lemma **(3.19)**, t cannot be typed in the empty context. So we must have $k = 1$, $t = \lambda x(\dots((x)u_1)\dots)u_n$ (by Lemma **(3.19)**, the variable x must be bound in t, so it must be x_1) and $x : (A \to B) \to A \vdash (\dots((x)u_1)\dots)u_n : A$. The only applicable rule is $(\to_E)$ and hence there exists $C_i \in T$ such that $x : (A \to B) \to A \vdash u_i : C_i$ and $x : (A \to B) \to A \vdash C_1 \to (C_2 \to \dots(C_n \to A)\dots)$. It follows that $(A \to B) \to A = C_1 \to (C_2 \to \dots(C_n \to A)\dots)$ and so $n = 1$ and $C_1 = A \to B$.
We have obtained so far that there exists $u \in \mathcal{N}$ such that $x : (A \to B) \to A \vdash u : A \to B$. By Lemma **(3.18)**, $t = \lambda y_1 \cdot \lambda y_k \, (\dots((y)v_1)\dots)v_n$ for some y and $y_i \in V$ and $v_i \in \mathcal{N}$. As before $k = 0$ or 1. If $k = 0$, then $y = x$ and we must have $x : (A \to B) \to A \vdash D_1 \to (D_2 \dots \to (D_n \to (A \to B))\dots)$ for some $D_i \in T$ and thus $(A \to B) \to A = D_1 \to (D_2 \dots \to (D_n \to (A \to B))\dots)$. But that is impossible. It follows that $k = 1$ and we must have $\{x : (A \to B) \to A, y_1 : A\} \vdash (\dots((y)v_1)\dots)v_n : B$ where $y = x$ or y_1 and hence $\{x : (A \to$

$B) \to A, y_1 : A\} \vdash y : D_1 \to (D_2 \ldots \to (D_n \to B)\ldots)$. This is not possible if $y = x$ nor if $y = y_1$ so this is a contradiction. ■

Note that the "simplification" for proofs played a fundamental role in this argument.

4 Enriched λ-calculi

The fact that we end up with a weaker logic could come from the fact that we can only treat the connective →. One natural solution to this problem could be to enrich our λ-calculus and/or our type system to obtain a Curry-Howard correspondence with a larger fragment of propositional logic. I will outline four different possible enrichments one might want to consider. There are many more and some even allow for example to also obtain a Curry-Howard correspondence with fragments of predicate logic or higher order logic (we will consider one of those at the end).

4.1 Intersection types

One first thing we could do is add an intersection to our types:

Definition 4.1 (Intersection types)**:**
Let W be a set (disjoint from V the set of variables). The set $T_{\mathcal{D}}$ of intersection types is the smallest set of words on $W \cup \{(,), \to, \wedge\}$ (where the union is supposed to be disjoint) such that:

- $W \subseteq T_{\mathcal{D}}$;
- *If A and $B \in T_{\mathcal{D}}$, then $(A \to B)$ and $(A \wedge B) \in T_{\mathcal{D}}$.*

We keep the old typing rules:

$$(\mathrm{Ax})\ \frac{}{\Gamma \cup \{x : A\} \vdash_{\mathcal{D}} x : A}$$

$$(\to_I)\ \frac{\Gamma \cup \{x : A\} \vdash_{\mathcal{D}} t : B}{\Gamma \vdash_{\mathcal{D}} \lambda x\, t : A \to B} \qquad (\to_E)\ \frac{\Gamma_1 \vdash_{\mathcal{D}} t : A \to B \qquad \Gamma_2 \vdash_{\mathcal{D}} u : A}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{D}} (t)u : B}$$

And we add three new rules:

$$(\wedge_I)\ \frac{\Gamma_1 \vdash_{\mathcal{D}} t : A \qquad \Gamma_2 \vdash_{\mathcal{D}} t : B}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{D}} t : A \wedge B} \qquad (\wedge_{EL})\ \frac{\Gamma \vdash_{\mathcal{D}} t : A \wedge B}{\Gamma \vdash_{\mathcal{D}} t : A} \qquad (\wedge_{ER})\ \frac{\Gamma \vdash_{\mathcal{D}} t : A \wedge B}{\Gamma \vdash_{\mathcal{D}} t : R}$$

Most of the result we proved for simply typed λ-calculus still hold (with very similar proofs):

Proposition 4.2:
Let $t, u \in \Lambda$, Γ be a context and $A \in T_{\mathcal{D}}$. Let us assume that $\Gamma \vdash_{\mathcal{D}} t : A$ and $t \to_\beta^\star u$, then $\Gamma \vdash_{\mathcal{D}} u : A$.

Theorem 4.3:

Let $t \in \Lambda$, Γ be a context and $A \in T_{\mathcal{D}}$. If $\Gamma \vdash_{\mathcal{D}} t : A$ then t is strongly normalizable.

Remember that our initial goal when introducing types was to characterize certain classes of strongly normalizing terms. To do that we had to get rid of $(\Delta)\Delta = (\lambda(x)x)\lambda(x)x$ and when we introduced simple types, we got rid of $\lambda(x)x$ although this term is perfectly fine, being normal. Intersection types do not suffer from this flaw:

$$\dfrac{\dfrac{\dfrac{}{x : A \wedge (A \to B) \vdash_{\mathcal{D}} xA \wedge (A \to B)}\,(\mathrm{Ax})}{x : A \wedge (A \to B) \vdash_{\mathcal{D}} x : A \to B}\,(\wedge_{ER}) \quad \dfrac{\dfrac{}{x : A \wedge (A \to B) \vdash_{\mathcal{D}} xA \wedge (A \to B)}\,(\mathrm{Ax})}{x : A \wedge (A \to B) \vdash_{\mathcal{D}} x : A}\,(\wedge_{EL})}{\dfrac{x : A \wedge (A \to B) \vdash_{\mathcal{D}} (x)x : B}{\vdash_{\mathcal{D}} \lambda x\,(x)x : (A \wedge (A \to B)) \to B}\,(\to_I)}\,(\to_E)$$

In fact, we can prove that we got rid of this flaw altogether:

Theorem 4.4:

Let $t \in \Lambda$, Γ be a context and $A \in T_{\mathcal{D}}$. The following are equivalent:

(i) $\Gamma \vdash_{\mathcal{D}} t : A$;

(ii) t *is strongly normalizable.*

We can also show that Pierce's law is the $\mathcal{D}$-type of no λ-term and hence that it cannot be derived using only (Ax), $(\to_I)$, $(\to_E)$, $(\wedge_I)$, $(\wedge_{EL})$ and $(\wedge_{ER})$.

4.2 Intuitionistic propositional logic

Intersection types completely solve the problem of characterizing strongly normal terms. Nevertheless from a logical perspective, they are not completely satisfying as the $\mathcal{D}$-types represent a rather small fragment of all propositional formulas. One way to obtain a type system whose types are exactly the formulas of propositional logic is the following (but it requires changing the λ terms and the notion of reduction):

Definition 4.5:
Let V be a (countable) set of variables and $A = V \cup \{(,), \lambda, \langle, \rangle, ;, \pi_1, \pi_2, \mathcal{C}, |, i_1, i_2, \nabla\}$. The set $L_{\mathcal{I}}$ is the smallest set of words on A such that:

- $V \subseteq L_{\mathcal{I}}$;
- *If t and $u \in L_{\mathcal{I}}$, then $(t)u$ and $\langle t; u\rangle \in L_{\mathcal{I}}$;*
- *If $x \in V$ and $t \in L_{\mathcal{I}}$, then $\lambda x\, t \in L_{\mathcal{I}}$.*
- *If $t \in L_{\mathcal{I}}$, then $\pi_1 t$, $\pi_2 t$, $i_1 t$, $i_2 t$ and $\nabla t \in L_{\mathcal{I}}$;*
- *If t, u, $v \in L_{\mathcal{I}}$ then $\mathcal{C}t(u|v) \in L_{\mathcal{I}}$.*

As for the terms in L, we can define a notion of α-conversion and we define $\Lambda_{\mathcal{I}} = L_{\mathcal{I}}/\equiv_\alpha$. The construction $\langle t; u\rangle \in L_{\mathcal{I}}$ corresponds to pairs, $\pi_1 t$ and $\pi_2 t$ to the two projections. The other construction are somewhat more complex to describe but, for what it is worth, ∇t

corresponds to an error being raised by the program and $\mathcal{C}t(u|v)$ corresponds to a pattern matching on t and i_1 and i_2 are the two constructors for the pattern matching. This might become clearer once we have the reduction rules:

(β) $(\lambda x\, t)u \to_\beta t_{u/x}$;

(π_1) $\pi_1\langle t; u\rangle \to_\pi t$;

(π_2) $\pi_2\langle t; u\rangle \to_\pi u$;

$(\mathcal{C}_1)$ $\mathcal{C}i_1 t(u|v) \to_\mathcal{C} (u)t$;

$(\mathcal{C}_2)$ $\mathcal{C}i_2 t(u|v) \to_\mathcal{C} (v)t$;

(∇) $(\nabla t)u \to_\nabla \nabla t$.

We define a notion of reduction $t \to_\mathcal{I} u$ by allowing any of the above reductions to happen anywhere in t:

Definition 4.6:
Let t and $u \in \Lambda_\mathcal{I}$. We define $t \to_\mathcal{I} u$ by induction on t:

- *If $t \to_\square u$ where $\square \in \{\beta, \pi, \mathcal{C}, \nabla\}$, then $t \to_\mathcal{I} u$;*
- *If $t \to_\mathcal{I} u$ then for $x \in V$, $s, v \in \Lambda_\mathcal{I}$ and $\square \in \{\pi_1, \pi_2, i_1, i_2, \nabla\}$, $\lambda x\, t \to_\mathcal{I} \lambda x\, u$, $(t)s \to_\mathcal{I} (u)s$, $(s)t \to_\mathcal{I} (s)u$, $\langle t; s\rangle \to \langle u; s\rangle$, $\langle s; t\rangle \to \langle s; u\rangle$, $\square t \to_\mathcal{I} \square u$, $\mathcal{C}t(s|v) \to_\mathcal{I} \mathcal{C}u(s|v)$, $\mathcal{C}s(t|v) \to_\mathcal{I} \mathcal{C}s(u|v)$ and $\mathcal{C}v(s|t) \to_\mathcal{I} \mathcal{C}v(s|u)$.*

Definition 4.7:
Let W be a set (disjoint from V the set of variables). The set $T_\mathcal{I}$ is the smallest set of words on $W \cup \{(,), \to, \wedge, \vee, \bot\}$ (where the union is supposed to be disjoint) such that:

- $W \subseteq T_\mathcal{I}$;
- $\bot \in T_\mathcal{I}$
- *If A and $B \in T_\mathcal{I}$, then $(A \to B)$ $(A \wedge B)$ and $(A \vee B) \in T_\mathcal{I}$.*

And we consider the following typing rules:

$$(\mathrm{Ax})\ \frac{}{\Gamma \cup \{x : A\} \vdash_\mathcal{I} x : A}$$

$$(\to_I)\ \frac{\Gamma \cup \{x : A\} \vdash_\mathcal{I} t : B}{\Gamma \vdash_\mathcal{I} \lambda x\, t : A \to B} \qquad (\to_E)\ \frac{\Gamma_1 \vdash_\mathcal{I} t : A \to B \qquad \Gamma_2 \vdash_\mathcal{I} u : A}{\Gamma_1 \cup \Gamma_2 \vdash_\mathcal{I} (t)u : B}$$

$$(\wedge_I)\ \frac{\Gamma_1 \vdash_\mathcal{I} t : A \qquad \Gamma_2 \vdash_\mathcal{I} u : B}{\Gamma_1 \cup \Gamma_2 \vdash_\mathcal{I} \langle t; u\rangle : A \wedge B} \qquad (\wedge_{EL})\ \frac{\Gamma \vdash_\mathcal{I} t : A \wedge B}{\Gamma \vdash_\mathcal{I} \pi_1 t : A} \qquad (\wedge_{ER})\ \frac{\Gamma \vdash_\mathcal{I} A \wedge B}{\Gamma \vdash_\mathcal{I} \pi_2 t : B}$$

$$(\vee_{IL})\ \frac{\Gamma \vdash_\mathcal{I} t : A}{\Gamma \vdash_\mathcal{I} i_1 t : A \vee B} \qquad (\vee_{IR})\ \frac{\Gamma \vdash_\mathcal{I} t : B}{\Gamma \vdash_\mathcal{I} i_2 t : A \vee B}$$

$$(\vee_E)\ \frac{\Gamma_1 \vdash_{\mathcal{I}} u : A \to C \qquad \Gamma_2 \vdash_{\mathcal{I}} v : B \to C \qquad \Gamma_3 \vdash_{\mathcal{I}} t : A \vee B}{\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \vdash_{\mathcal{I}} \mathcal{C}t(u|v) : C}$$

$$(\bot_E)\ \frac{\Gamma \vdash_{\mathcal{I}} t : \bot}{\Gamma \vdash_{\mathcal{I}} \nabla t : A}$$

If we define $\neg A$ as $A \to \bot$ and $A \leftrightarrow B$ as $(A \to B) \wedge (B \to A)$ then it is very easy to see that we can derive every deduction rule from propositional logic except for (ExMid) (and this rule is not only hard to derive, it is in fact impossible to derive as we will see later on). Note that in this setting, it is more natural to have $\bot$ as a primitive and to define $\neg$ using $\to$ and $\bot$.
As previously we can prove the following[1]:

Proposition 4.8:
Let t, $u \in \Lambda_{\mathcal{I}}$, Γ be a context and $A \in T_{\mathcal{I}}$. Let us assume that $\Gamma \vdash_{\mathcal{I}} t : A$ and $t \to^{\star}_{\mathcal{I}} u$, then $\Gamma \vdash_{\mathcal{I}} u : A$.

Theorem 4.9:
Let $t \in \Lambda_{\mathcal{I}}$, Γ be a context and $A \in T_{\mathcal{I}}$. If $\Gamma \vdash_{\mathcal{I}} t : A$ then t is strongly normalizable (for $\to_{\mathcal{I}}$).

By a (rather horrible) discussion on the form of normal terms one can prove the following statement which is symptomatic of intuitionistic logic (and which explains why this logic is also referred to as constructive logic):

Proposition 4.10:
Let A_1, $A_2 \in T_{\mathcal{I}}$ and $t \in \Lambda_{\mathcal{I}}$. Assume that $\vdash_{\mathcal{I}} t : A_1 \vee A_2$ and t is normal. Then there exists $u \in \Lambda_{\mathcal{I}}$ and $j \in \{1, 2\}$ such that $t = i_j u$ and $\vdash_{\mathcal{I}} u : A_j$ holds.

Corollary 4.11:
There is no $t \in \Lambda_{\mathcal{I}}$ whose type in the empty context is $A \vee (A \to \bot)$.

If we call *intuitionistic logic* the logic whose rules are the term-free version of $\vdash_{\mathcal{I}}$, then we have just proved that $A \vee \neg A$ is a tautology of classical logic that does not hold in intuitionistic logic. What we can prove in intuitionistic logic is that Pierce's law is equivalent to the excluded middle and hence Pierce's law does not hold either in intuitionistic logic.

4.3 Classical propositional logic

If we want to recover full classical logic we just need to add that a rule that says that A can be deduced from $\neg\neg A$.
As the system of connectives $\{\to, \bot\}$ is complete, to keep things minimal, we will only consider those two basic type constructions and define the rest of the connectives using them.

Definition 4.12:
Let V be a (countable) set of variables and $A = V \cup \{(,), \lambda, \mathcal{E}\}$. The set $L_{\mathcal{P}}$ is the smallest set of words on A such that:

[1] The proofs are essentially variations on the previous proofs. They become more annoying to write as there are many more cases.

- $V \subseteq L_{\mathcal{P}}$;
- *If* t *and* $u \in L_{\mathcal{P}}$, *then* $(t)u \in L_{\mathcal{P}}$;
- *If* $x \in V$ *and* $t \in L_{\mathcal{P}}$, *then* $\lambda x\, t \in L_{\mathcal{P}}$.
- *If* $t \in L_{\mathcal{P}}$, *then* $\mathcal{E}t \in L_{\mathcal{P}}$;

As for the terms in L, we can define a notion of α-conversion and we define $\Lambda_{\mathcal{P}} = L_{\mathcal{P}}/\equiv_{\alpha}$. The construction $\mathcal{E}t$ corresponds to what is called a continuation operator. The reduction rule associated to $\mathcal{E}$ has a computational meaning but it is somewhat complicated to explain. We consider the following reduction rules:

(β) $(\lambda x\, t)u \to_{\beta} t_{u/x}$;

(η) $\lambda x\, (t)x \to_{\eta} t$ if $x \notin \mathrm{fvar}(t)$;

$(\mathcal{E})$ $(\mathcal{E}t)u \to_{\mathcal{E}} \mathcal{E}\lambda f\, (t)\lambda g\, (f)(g)u$ where $f, g \notin \mathrm{fvar}(t) \cup \mathrm{fvar}(u)$;

$(\mathcal{E}\eta)$ $\mathcal{E}\lambda f\, (f)t \to_{\mathcal{E}\eta} t$ if $f \notin \mathrm{fvar}(t)$.

In fact, we could have introduced the η-reduction rule from the beginning. It corresponds to some natural simplification of proof but it is not really useful and does make things a little more complicated. However, in $\lambda\mathcal{E}$-calculus, products and sums do not work as well if we don't add this new rule and the corresponding rule associated with $\mathcal{E}$.
We define a notion of reduction $t \to_{\mathcal{P}} u$ by allowing any of the above reductions to happen anywhere in t:

Definition 4.13:
Let t *and* $u \in \Lambda_{\mathcal{P}}$. *We define* $t \to_{\mathcal{P}} u$ *by induction on* t:

- *If* $t \to_{\square} u$ *where* $\square \in \{\beta, \eta, \mathcal{E}, \mathcal{E}\eta\}$, *then* $t \to_{\mathcal{P}} u$;
- *If* $t \to_{\mathcal{P}} u$ *then for* $x \in V$, *and* $s \in \Lambda_{\mathcal{P}}$ $\lambda x\, t \to_{\mathcal{P}} \lambda x\, u$, $(t)s \to_{\mathcal{P}} (u)s$, $(s)t \to_{\mathcal{P}} (s)u$ *and* $\mathcal{E}t \to_{\mathcal{P}} \mathcal{E}u$.

Definition 4.14:
Let W *be a set (disjoint from* V *the set of variables). The set* $T_{\mathcal{P}}$ *is the smallest set of words on* $W \cup \{(,), \to, \bot\}$ *(where the union is supposed to be disjoint) such that:*

- $W \subseteq T_{\mathcal{P}}$;
- $\bot \in T_{\mathcal{P}}$
- *If* A *and* $B \in T_{\mathcal{P}}$, *then* $(A \to B) \in T_{\mathcal{P}}$.

And we consider the following typing rules:

$$(\mathrm{Ax})\ \frac{}{\Gamma \cup \{x : A\} \vdash x : A}$$

$$(\to_I)\ \frac{\Gamma \cup \{x : A\} \vdash t : B}{\Gamma \vdash \lambda x\, t : A \to B} \qquad (\to_E)\ \frac{\Gamma_1 \vdash t : A \to B \qquad \Gamma_2 \vdash u : A}{\Gamma_1 \cup \Gamma_2 \vdash (t)u : B}$$

$$(\bot_E)\ \frac{\Gamma \vdash t : \bot}{\Gamma \vdash \nabla t : A} \qquad (\neg\neg_E)\ \frac{\Gamma \vdash t : (A \to \bot) \to \bot}{\Gamma \vdash \mathcal{E} t : A}$$

As previously we can prove the following:

Proposition 4.15:
Let t, $u \in \Lambda_{\mathcal{P}}$, Γ be a context and $A \in T_{\mathcal{P}}$. Let us assume that $\Gamma \vdash_{\mathcal{P}} t : A$ and $t \to_{\mathcal{P}}^{\star} u$, then $\Gamma \vdash_{\mathcal{P}} u : A$.

Theorem 4.16:
Let $t \in \Lambda_{\mathcal{P}}$, Γ be a context and $A \in T_{\mathcal{P}}$. If $\Gamma \vdash_{\mathcal{P}} t : A$ then t is strongly normalizable (for $\to_{\mathcal{P}}$).

If we want to recover all of propositional logic, we should redefine all the other connectives and their associated constructors. As previously, we define $\neg A$ as $A \to \bot$. Also, although we do have $\bot$ as logical connective/type constructor, we are missing the constructor ∇u that transforms a proof of $\bot$ into a proof of any type A. Let $\nabla u = \mathcal{E}\lambda x\, u$ for some $x \notin \mathrm{fvar}(u)$. Let us check that this term is typed the right way, i.e. if $\Gamma \vdash_{\mathcal{P}} u : \bot$ for some context Γ, then we should have $\Gamma \vdash_{\mathcal{P}} \nabla u : A$ for any $A \in T_{\mathcal{P}}$:

$$\begin{array}{c} \vdots \\ (\to_I)\ \dfrac{\Gamma \cup \{x : A \to \bot\} \vdash_{\mathcal{P}} u : \bot}{\Gamma \vdash_{\mathcal{P}} \lambda x\, u : (A \to \bot) \to \bot} \\ (\neg\neg_E)\ \dfrac{}{\Gamma \vdash_{\mathcal{P}} \mathcal{E}\lambda x\, u : A} \end{array}$$

where the first line comes from weakening the context (we proved this result for simply type λ-calculus, but it also holds here).
Let us also check that this term reduces in the intented way:

$$\begin{array}{rcl} (\nabla u)v & = & (\mathcal{E}\lambda x\, u)v \\ & \to_{\mathcal{E}} & \mathcal{E}\lambda f\, (\lambda x\, u)\lambda g\, (f)(g)u \\ & \to_{\beta} & \mathcal{E}\lambda f\, u \equiv_{\alpha} \lambda x\, u = \nabla u. \end{array}$$

Let us now take care of the conjonction. Let $A \wedge B = (A \to (B \to \bot)) \to \bot$, $\langle t; u\rangle = \lambda f\, ((f)t)u$, $\pi_1 t = \mathcal{E}\lambda f\, (t)\lambda x\lambda y\, (f)x$ and $\pi_2 t = \mathcal{E}\lambda f\, (t)\lambda x\lambda y\, (f)y$. Let us check that we get the right typing rules. Assumong $\Gamma_1 \vdash_{\mathcal{P}} t : A$ and $\Gamma_2 \vdash_{\mathcal{P}} u : B$ hold, then:

$$(\to_I)\ \frac{(\mathrm{Ax})\ \dfrac{\dfrac{}{f : A \to (B \to \bot) \vdash_{\mathcal{P}} f : A \to (B \to \bot)} \qquad \begin{array}{c}\vdots \\ \Gamma_1 \vdash_{\mathcal{P}} t : A\end{array}}{\Gamma_1 \cup \{f : A \to (B \to \bot)\} \vdash_{\mathcal{P}} (f)t : B \to \bot} \qquad \begin{array}{c}\vdots \\ \Gamma_2 \vdash_{\mathcal{P}} u : B\end{array}}{\Gamma_1 \cup \Gamma_2 \cup \{f : A \to (B \to \bot)\} \vdash_{\mathcal{P}} ((f)t)u : \bot}}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{P}} \lambda f\, ((f)t)u : (A \to (B \to \bot)) \to \bot = A \wedge B}$$

Now assume that $\Gamma \vdash t : A \wedge B = (A \to (B \to \bot)) \to \bot$ holds. We have:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{f : A \to \bot \vdash_{\mathcal{P}} f : A \to \bot}\,(\mathrm{Ax}) \quad \dfrac{}{\{x : A, y : B\} \vdash_{\mathcal{P}} a : A}\,(\mathrm{Ax})}{\Gamma \cup \{f : A \to \bot, x : A, y : B\} \vdash_{\mathcal{P}} (f)x : \bot}\,(\to_E)}{\Gamma \cup \{f : A \to \bot, x : A\} \vdash_{\mathcal{P}} \lambda y\,(f)x : B \to \bot}\,(\to_I)}{\Gamma \cup \{f : A \to \bot\} \vdash_{\mathcal{P}} \lambda x \lambda y\,(f)x : A \to (B \to \bot)}\,(\to_I)$$

and:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma \vdash_{\mathcal{P}} t : (A \to (B \to \bot)) \to \bot\end{array} \qquad \Gamma \cup \{f : A \to \bot\} \vdash_{\mathcal{P}} \lambda x \lambda y\,(f)x : A \to (B \to \bot)}{\Gamma \cup \{f : A \to \bot\} \vdash_{\mathcal{P}} (t)\lambda x \lambda y\,(f)x : \bot}\,(\to_E)}{\Gamma \vdash_{\mathcal{P}} \lambda f\,(t)\lambda x \lambda y\,(f)x : (A \to \bot) \to \bot}\,(\to_I)}{\Gamma \vdash_{\mathcal{P}} \mathcal{E}\lambda f\,(t)\lambda x \lambda y\,(f)x : A}\,\neg\neg_E$$

The term $\pi_2 t$ is typed in a similar way.
Let us now check that we have the right reduction:

$$\begin{array}{rcl}
\pi_1\langle t; u\rangle & = & \mathcal{E}\lambda f\,(\lambda f\,((f)t)u)\lambda x \lambda y\,(f)x \\
& \to_\beta & \mathcal{E}\lambda f\,((\lambda x \lambda y\,(f)x)t)u \\
& \to_\beta & \mathcal{E}\lambda f\,(\lambda y\,(f)t)u \\
& \to_\beta & \mathcal{E}\lambda f\,(f)t \\
& \to_{\mathcal{E}\eta} & t.
\end{array}$$

We have the symmetric reduction for $\pi_2\langle t; u\rangle$.
Conjonctions can also be recovered. Define $A \vee B = (A \to \bot) \to ((B \to \bot) \to \bot)$, $i_1 t = \lambda f \lambda g\,(f)t$, $i_2 t = \lambda f \lambda g\,(g)t$ and $\mathcal{C}t(u|v) = \mathcal{E}\lambda h\,((t)\lambda a\,(h)(u)a)\lambda b\,(h)(v)b$. We can check again that all these terms are typed correctly. Let us just show that we have the right reduction:

$$\begin{array}{rcl}
\mathcal{C}i_1 t(u|v) & = & \mathcal{E}\lambda h\,((\lambda f \lambda g\,(f)t)\lambda a\,(h)(u)a)\lambda b\,(h)(v)b \\
& \to_\beta & \mathcal{E}\lambda h\,(\lambda g\,(\lambda a\,(h)(u)a)t)\lambda b\,(h)(v)b \\
& \to_\beta & \mathcal{E}\lambda h\,(\lambda a\,(h)(u)a)t \\
& \to_\beta & \mathcal{E}\lambda h\,(h)(u)t \\
& \to_{\mathcal{E}\eta} & (u)t.
\end{array}$$

One last remark, the excluded middle is valid in classical propositional logic and so some $\lambda\mathcal{E}$-term should have type $A \vee \neg A = (A \to \bot) \to (((A \to \bot) \to \bot) \to \bot)$. The term $\lambda f \lambda g\,(g)f$ is such a term and here is a derivation proving it:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{g : (A \to \bot) \to \bot \vdash_{\mathcal{P}} g : (A \to \bot) \to \bot}\,(\mathrm{Ax}) \quad \dfrac{}{f : A \to \bot \vdash_{\mathcal{P}} g : A \to \bot}\,(\mathrm{Ax})}{\{f : A \to \bot, g : (A \to \bot) \to \bot \vdash_{\mathcal{P}} (g)f : \bot}\,(\to_E)}{f : A \to \bot \vdash_{\mathcal{P}} \lambda g\,(g)f : ((A \to \bot) \to \bot) \to \bot}\,(\to_I)}{\vdash_{\mathcal{P}} \lambda f \lambda g\,(g)f : (A \to \bot) \to (((A \to \bot) \to \bot) \to \bot)}\,(\to_I)$$

4.4 System $\mathcal{F}$

The last possible approach that will be discussed here is rather orthogonal to the two previous ones. Instead of enriching our λ-calculus with new construction to allow for new type construction, we will (as we did with intersection types) simply enrich the typing system and hence the corresponding logic. The main difference with intersection types is that the enrichment that we will consider now will correspond to an enrichement of propositional logic that we have not considered so far: quantification over propositional variables (often referred two as second order logic). From a programming language perspective it corresponds to type polymorphism.
First we need to define the types of system $\mathcal{F}$.

Definition 4.17:
Let W be a set (disjoint from V the set of variables). The set $T_{\mathcal{F}}$ is the smallest set of words on $W \cup \{(,), \to, \forall\}$ (where the union is supposed to be disjoint) such that:

- $W \subseteq T_{\mathcal{F}}$;
- *If A and $B \in T_{\mathcal{F}}$, then $(A \to B) \in T_{\mathcal{F}}$.*
- *If $X \in W$ and $sA \in T_{\mathcal{F}}$, then $\forall X\, A \in T_{\mathcal{F}}$.*

As for terms, we can define a notion of α-conversion on types (also denoted $\equiv_\alpha$), and we define $\mathcal{T}_{\mathcal{F}} = T_{\mathcal{F}}/\equiv_\alpha$. We can also define a notion of substitution on $\mathcal{T}_{\mathcal{F}}$ that we will denote by $A_{B/X}$.
If $\Gamma = \{(x_i : A_i) : i \in I\}$ is a context, we denote by $\mathrm{fvar}(\Gamma)$ the set $\bigcup_{i\in I} \mathrm{fvar}(A_i)$.
We keep the old typing rules:

$$(\mathrm{Ax})\ \frac{}{\Gamma \cup \{x : A\} \vdash_{\mathcal{F}} x : A}$$

$$(\to_I)\ \frac{\Gamma \cup \{x : A\} \vdash_{\mathcal{F}} t : B}{\Gamma \vdash_{\mathcal{F}} \lambda x\, t : A \to B} \qquad (\to_E)\ \frac{\Gamma_1 \vdash_{\mathcal{F}} t : A \to B \qquad \Gamma_2 \vdash_{\mathcal{F}} u : A}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{F}} (t)u : B}$$

And we add two new rules:

$$(\forall_I)\ \frac{\Gamma \vdash_c Ft : A \qquad X \notin \mathrm{fvar}(\Gamma)}{\Gamma \vdash_c Ft : \forall X\, A} \qquad (\forall_E)\ \frac{\Gamma \vdash_{\mathcal{F}} t : \forall X\, A}{\Gamma \vdash_{\mathcal{F}} t : A_{B/X}}$$

Although the proof become more complicated (especially the proof of Theorem **(4.19)**), we can prove the following:

Proposition 4.18:
Let t, $u \in \Lambda$, Γ be a context and $A \in \mathcal{T}_{\mathcal{F}}$. Let us assume that $\Gamma \vdash_{\mathcal{F}} t : A$ and $t \to_{\mathcal{P}}^{\star} u$, then $\Gamma \vdash_{\mathcal{F}} u : A$.

Theorem 4.19:
Let $t \in \Lambda$, Γ be a context and $A \in \mathcal{T}_{\mathcal{F}}$. If $\Gamma \vdash_{\mathcal{F}} t : A$ then t is strongly normalizable (for $\to_\beta$).

We can also encode $\bot$, $\wedge$ and $\vee$ in system $\mathcal{F}$. The encodings are quite similar to those in $\lambda\mathcal{E}$-calculus, but the underlying λ-terms are somewhat more reasonnable.
We define $\bot = \forall X\, X$. Assuming that $\Gamma \vdash t : \bot = \forall X\, X$, we have:

$$(\forall_E)\ \frac{\Gamma \vdash_{\mathcal{F}} t : \forall X\, X}{\Gamma \vdash_{\mathcal{F}} t : A}$$

We define $A \wedge B = \forall X\,(A \to (B \to X)) \to X$ (where X is free in A and B), $\langle t; u\rangle = \forall f\,(f(t))u$, $\pi_1 t = (t)\lambda a \lambda b\, a$ and $\pi_1 t = (t)\lambda a \lambda b\, b$. Let us check that these terms have the right type. Assume that $\Gamma_1 \vdash_{\mathcal{F}} t : A$ and $\Gamma_2 \vdash_{\mathcal{F}} u : t$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{f : A \to (B \to X) \vdash_{\mathcal{F}} f : A \to (B \to X)}(\mathrm{Ax}) \quad \vdots\ \Gamma_1 \vdash_{\mathcal{F}} t : A}{\Gamma_1 \cup \{f : A \to (B \to X)\} \vdash_{\mathcal{F}} (f)t : B \to X} \quad \vdots\ \Gamma_2 \vdash_{\mathcal{F}} u : B}{\Gamma_1 \cup \Gamma_2 \cup \{f : A \to (B \to X)\} \vdash_{\mathcal{F}} ((f)t)u : X}(\to_I)}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{F}} \lambda f\,((f)t)u : (A \to (B \to X)) \to X}(\forall_E) \qquad X \notin \Gamma_1 \cup \Gamma_2}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{F}} \lambda f\,((f)t)u : \forall X\,(A \to (B \to X)) \to X = A \wedge B}$$

Now assume that $\Gamma \vdash t : A \wedge B = (A \to (B \to \bot))$ holds. We have:

$$\dfrac{\dfrac{\vdots\ \Gamma \vdash_{\mathcal{F}} t : \forall X\,(A \to (B \to X)) \to X}{\Gamma \vdash_{\mathcal{F}} t : (A \to (B \to A)) \to A}(\forall_E) \quad \dfrac{\dfrac{\dfrac{}{\Gamma \cup \{a : A, b : B\} \vdash_{\mathcal{F}} a : A}(\mathrm{Ax})}{\Gamma \cup \{a : A\} \vdash_{\mathcal{F}} \lambda b\, a : B \to A}(\to_I)}{\Gamma \vdash_{\mathcal{F}} \lambda a \lambda b\, a : A \to (B \to A)}(\to_I)}{\Gamma \vdash_{\mathcal{F}} (t)\lambda a \lambda b\, a : A}(\to_E)$$

The term $\pi_2 t$ is typed in a similar way.
Let us also check that we have the correct reduction:

$$\begin{aligned} \pi_1\langle t; u\rangle = (\lambda f\,((f)t)u)\lambda a \lambda b\, a & \\ \to_\beta \quad & ((\lambda a \lambda b\, a)t)u \\ \to_\beta \quad & (\lambda b\, t)u \\ \to_\beta \quad & t \end{aligned}$$

We have the symmetric reduction for $\pi_2\langle t; u\rangle$.
Finally, we define $A \vee B = \forall X\,(A \to X) \to ((B \to X) \to X)$ (where X is free in A and B), $i_1 t = \lambda f \lambda g\,(f)t$, $i_2 t = \lambda f \lambda g\,(g)t$ and $\mathcal{C}t(u|v) = ((t)u)v$. We can check again that all these terms are typed correctly and that we have the corresponding reduction.
However, the logic associated to system $\mathcal{F}$ should once again more precisely be called "Intuitionistic second order propositinal logic" as the excluded middle is not valid in this logic. In fact the following proposition is also true in system $\mathcal{F}$ (and the fact that no term has type $A \vee \neg A$ in system $\mathcal{F}$ follows immediately):

Proposition 4.20:
Let $t \in \Lambda$ be normal, if there exists types A_1, $A_2 \in \mathcal{T}_{\mathcal{F}}$ such that $\vdash_{\mathcal{F}} t : A_1 \vee A_2$, then there exists $u \in \Lambda$ and $j \in \{1, 2\}$ such that $t = i_j u$ and $\vdash_{\mathcal{F}} u : A_j$.

References

[Kri93] J.-L. Krivine. *Lambda-calculus, types and models*. Ellis Horwood Series in Computers and Their Applications. Translated from the 1990 French original by René Cori. Ellis Horwood, New York; Masson, Paris, 1993, pp. viii+180.