Elambda-calculus : A higher order logic extension of untyped λ-calculus.

Mohamed Mezghiche* Choukri Ben-yelles †

Abstract

In this paper we define an extension of the classical λ-calculus by adding two constants P and Π which represent respectively the implication and the universal quantifier. We show that the obtained system, called Eλ, is a consistent system. The idea behind the proof of consistency is to introduce a notion of level of terms that allows an appropriate definition of substitution. The term $[N/x]M$ is defined only when level(N) $\leq$ level(x). This restriction on substitution induces a new reduction in Eλ which is proved to have Church-Rosser property. Higher order logic can be interpreted in two ways in the Eλ-system, following proposition-as-types paradigm or in a more direct form as is the matter in illative combinatory logic. We will also give a procedure to construct a proof (if any) for a given proposition.

Key words: λ-calculus, Higher order logic, combinatory logic, level of terms, Curry's paradox, substitution.

1 Introduction

The λ-calculus and combinatory logic with unrestricted introduction and elimination rules for implication are inconsistent. The proof of this is usually called Curry's paradox. Curry, in his work on illative combinatory logic, considered that there is a fundamental incompatibility between the unrestricted rules of implication and the property that the term λx.X exists for each term X. This implies that one must recognise that only some terms can represent propositions (canonical terms). Curry and his school started a program of defining several systems of illative combinatory logic where a search for an appropriate definition of canonicalness was an important part in the definition of the different systems (see [6], [3],[4] and [9]).

Following this view, Bunder introduced restrictions on the rules of illative constants so that first order logic could be interpreted in the resulting systems. In Bunder's systems the usual derivation of Curry's paradox is avoided but the consistency of the systems remains an open problem.

*Faculté des sciences, département d'informatique Université de Boumerdès, Algeria; Email: mmez@wissal.dz

†IUT-Valence France; Email: choukri.Ben-yelles@iut.valence.fr

In a recent work Barendregt, Bunder and Dekker [7] considered systems of illative combinatory logic that are sound for first order propositional and predicate calculus. They showed that the interpretation from ordinary logic into illative systems can be done in two ways: following the propositions-as-types paradigm in which derivations became λ-terms, or in a more direct form(derivations are not translated). They proved that both translations are complete.

An other attempt to turn λ-calculus into the alternative foundation was originally intended to be is the Map Theory defined by Grue [8]. Map theory differs from illative combinatory logic initiated by Curry. It is based on functions (maps) and identifies some of them to be well-founded. The definition of well-foundedness plays a centre role and is crucial to the expressive power of map theory.

In this paper, we have as aim, by extending λ-calculus in the spirit of Curry's school, to provide a basis for higher order logic and constructive mathematics.We define a new system Eλ by adding to the λ-calculus two constants P and Π (implication and universal quantifier). To avoid Curry's paradox and prove the consistency of our system we take another approach than Curry's. Our idea is that fundamental incompatibility lies in the unrestricted rules for implication and the operation of substitution. A notion of level of terms is introduced : it gives an appropriate definition of substitution and a way to avoid Curry's paradox. This restriction on substitution induces a new reduction in Eλ called E$\lambda\beta$-reduction. We show that E$\lambda\beta$-reduction has the Church-Rosser property and that the system Eλ is a consistent system in which higher order logic can be interpreted. We think that the system Eλ allows a natural representation of higher order logic. This answers the open question asked,recently, by Barendregt in [2].

2 The system Eλ

The set C_ω of terms of the system Eλ is the union of the sets C_0, C_1 C_2 .... Each set C_i (i$\geq$ 1) is built by successively adding to the set C_{i-1} some terms constructed by the introduction of two extra constants P and Π. The set C_0 is the set of classical λ-terms whose variables are x^0 ,y^0, z^0 ... and C_0 do not contain the constants P and Π. The definition of the set C_0 is the following:

Definition 2.1. *(C_0−terms). The set of C_0 (C_0-terms) is built up from an infinite set of variables $V_0 = \{x^0, y^0, z^0, ...\}$ using application and λ-abstraction.*
- *if $x \in V_0$ then $x \in C_0$*
- *if M, $N \in C_0$ then $(MN) \in C_0$*
- *if $M \in C_0$, $x \in V_0$ then $(\lambda x.M) \in C_0$.*

The steps of construction of the set C_i (i$\geq$ 1) from the set C_{i-1} are given by the following definition.

Definition 2.2. *(C_i−terms). The set of C_i-terms is built up from an infinite set of variables $V_i = V_{i-1} \cup \{x^i, y^i, z^i, ...\}$ (i$\geq$ 1) and the constants P and Π by the following inductive steps.*
- *if $x \in V_i$ then $x \in C_i$.*

- *if* $M \in C_i$, $N \in C_j$ *then* $(MN) \in C_{max(i,j)}$.
- *if* $x \in V_i$, $M \in C_j$ *then* $\lambda x.M \in C_{max(i,j)}$.
- *if* X *is a* C_i*-term then* $\Pi X \in C_{max(1,i)}$.
- *if* $X \in C_i$, $Y \in C_j$ *then* $PXY \in C_{max(i,j)+1}$.

Example 2.1. - x^i, z^i *are in* C_i

- $\lambda x^i.M^j \in C_{max(i,j)}$
- $P(x^i)(P(y^j)(x^i)) \in C_{max(i,j)+2}$
- $\Pi X^i \in C_{max(1,i)}$.

It is easy to see that : $C_0 \subset C_1 \subset C_2 \subset \ldots \subset C_i \ldots$.

Definition 2.3. *(C_ω-terms). The set of Eλ-terms C_ω is defined by $C_\omega = \cup_{i<\omega} C_i$.*

Now, we will give a notion of the level of Eλ-term. It introduce a restriction on the operation of substitution allowing a variable with level k to be substituted by terms with levels equal or less than k. It will, also, play a centre role in avoiding Curry's paradox and the proof of the consistency of the system Eλ.

Definition 2.4. *(level of an Eλ-term). The level of a Eλ-term is computed by the following rules:*
- $level(x^i) = i$
- $level(\lambda x.M) = max(level(x), level(M))$.
- $level(PXY) = max(level(X), level(Y)) + 1$.
- $level(\Pi X) = max(1, level(X))$.
- $level(XY) = max(level(X), level(Y))$.

Notation. In the sequel, to make the text more readable , we will write "$X \supset Y$" instead of "PXY". We also write X^k to say that X is a E-λ-term and his level is k.

Example 2.2. *Let $X \supset Z$ be a Eλ-term. If $level(X) = i$ and $level(Y) = 0$ then $level(X \supset Z) = max(i, 0) + 1 = i + 1$. Let Y be the term $(\lambda x.(x \supset y))(\lambda x.(x \supset y))$, then $level(Y) = max(level(\lambda x.(x \supset y)), level(\lambda x.(x \supset y)) = level(\lambda x.(x \supset y)) = max(level(x), level(x \supset y)) = level(x \supset y) = max(level(x), level(y)) + 1$.*

Remark 2.1. *We can observe that level(X)= i iff $X \in C_i$ and $X \notin C_{i-1}$.*

The notion of bound variable and free variable is as usual in Church's λ-calculus. But, there is some minor changes in the definition of congruence between two terms.

Definition 2.5. *(Congruence). In the Eλ system the act of changing bound variable is allowed only when the old variable and the new variable have the same level. We shall say P is congruent to Q ($P \equiv_\alpha Q$) iff Q can been obtained from P by a finite (perhaps empty) series of changes of bound variables.*

For example the two terms $(\lambda x.(xx) \supset Z)$ and $\lambda y.(yy) \supset Z$ are congruent iff level(x)=level(y) otherwise they are not congruent.

Definition 2.6. *($E\lambda$-system). The system $E\lambda$ is defined as follows:*
I) The set of term of $E\lambda$ is the set C_ω .

II) The notion of $E\lambda\beta$-reduction is defined by the following rules:

(ρ) $M \rightarrow_{E\lambda\beta} M$
(μ) $M \rightarrow_{E\lambda\beta} N \Rightarrow ZM \rightarrow_{E\lambda\beta} ZN$
(μ') $M \rightarrow_{E\lambda\beta} N \Rightarrow X \supset M \rightarrow_{E\lambda\beta} X \supset N.$
(μ'') $M \rightarrow_{E\lambda\beta} N \Rightarrow \Pi M \rightarrow_{E\lambda\beta} \Pi N.$
(ν) $M \rightarrow_{E\lambda\beta} N \Rightarrow MZ \rightarrow_{E\lambda\beta} NZ.$
(ν') $M \rightarrow_{E\lambda\beta} N \Rightarrow M \supset X \rightarrow_{E\lambda\beta} N \supset X.$
(τ) $M \rightarrow_{E\lambda\beta} N$, $N \rightarrow_{E\lambda\beta} T$, *then* $M \rightarrow_{E\lambda\beta} T$
($E\alpha$) $\lambda x.M \rightarrow_{E\lambda\beta} \lambda y.[y/x]\ M$ *if level(x) = level(y) and* $y \notin FV(M)$.
($E\beta$) $(\lambda x.M)N \rightarrow_{E\lambda\beta} [N/x]M$ *if level(N)$\leq$ level(x) or level*$(\lambda x.M) = 0$
($E\xi$) $M \rightarrow_{E\lambda\beta} N \Rightarrow \lambda x.M \rightarrow_{E\lambda\beta} \lambda x.N$

The relation of equivalence generated by this relation $\rightarrow_{E\lambda\beta}$ is denoted $=_{E\lambda\beta}$.

III)

1. *A statement of the system $E\lambda$ is just an $E\lambda$-term. A basis is a denumerable set of statements.*
2. *Let Γ be a basis and X a statement, then X is derivable from Γ if $\Gamma \vdash X$ can be produced by the following deduction system:*

(ax) $X \in \Gamma \Rightarrow \Gamma \vdash X$
($Eq_{E\lambda\beta}$) $\Gamma \vdash X,\ X =_{E\lambda\beta} Y$, *level(Y)$\leq$level(X)* $\Rightarrow \Gamma \vdash Y.$
(P_e) $\Gamma \vdash X \supset Y,\ \Gamma \vdash X \Rightarrow \Gamma \vdash Y.$
(P_i) $\Gamma,\ X \vdash Y \Rightarrow \Gamma \vdash X \supset Y.$
(Π_e) $\Gamma \vdash \Pi(\lambda x^i.X) \Rightarrow \Gamma \vdash (\lambda x^i.X)Y^k$ *with* $k \leq i$.
(Π_i) $\Gamma \vdash M \Rightarrow \Gamma \vdash \Pi(\lambda x.M)$ *with* $x \notin FV(\Gamma)$.

Now, let us give some examples showing how reducibility of a term depends on its level.

Example 2.3.

- *We have* $(\lambda x^i.(y^i \supset x^i))Z^0 \rightarrow_{E\lambda\beta} [Z/x^i](y^i \supset x^i)$ *by rule* ($E\beta$)
- *The level of the term $(\lambda x.xx)$ is equal to the level of the variable x. Hence the rule ($E\lambda\beta$) can be applied and we get :*
  $(\lambda x.xx)(\lambda x.xx) \rightarrow_{E\lambda\beta} (\lambda x.xx)(\lambda x.xx) \rightarrow_{E\lambda\beta} \cdots \rightarrow_{E\lambda\beta} (\lambda x.xx)(\lambda x.xx) \cdots$
- *The term $(\lambda x^i.(x^i \supset x^i))(\lambda y^i.y^i \supset y^i)$ is not reducible. The rule ($E\beta$) cannot be applied because level$(\lambda y^i.(y^i \supset y^i))$ = i+1 > level(x^i).*

In the system $E\lambda$ we are able to define the different logical connectors and quantifiers. We can also prove, for each connector that its introduction rule and elimination rule are derived rules in the system Eλ. Following the work of Curry's school in illative combinatory logic the different logical connectors and quantifiers can be defined as below. Besides the usual ones $\vee$, $\wedge$, $\forall$, $\perp$ and $\neg$) we define the less familiar ones Ξ, F and G whose significance is explained in [9]. If we interpret the term XY as a statement saying "Y $\in$ X" or "Y satisfies the predicate X", then the term ΞXY can be interpreted "X $\subseteq$ Y" or " for all x (Xx implies Yx)". The constant F which can be defined in terms of Ξ allows us to represent functions . FabX is interpreted as " X is a function from a into b". One feature of the system Eλ is that the equality relation can be defined within it. We will use here the same definition which was used first by Church and Curry.

Definition 2.7. *(logical connectors and quantifiers).*

$$\begin{array}{lcl}
(\forall\ x^i)X & \equiv & \Pi(\lambda x^i.X) \\
\perp^i & \equiv & (\forall\ x^i)x^i \\
\neg^{i+1} & \equiv & (\lambda x^i.(x^i \supset \perp)\ ) \\
\Xi^{i+1} & \equiv & (\lambda x^i y^i.(\forall z^i)(x^i z^i) \supset (y^i z^i)) \\
F^{i+1} & \equiv & (\lambda x^i y^i z^i.(\forall u^i)(x^i u^i) \supset (y^i (z^i u^i))) \\
G^{i+1} & \equiv & (\lambda x^i y^i z^i.(\forall u^i)(x^i u^i) \supset (y^i u^i (z^i u^i))) \\
\wedge^{i+3} & \equiv & \lambda x^i y^i.(\forall z^i)((x^i \supset (y^i \supset z^i)) \supset z^i) \\
\vee^{i+3} & \equiv & \lambda x^i y^i.(\forall z^i)((x^i \supset z^i) \supset (y^i \supset z^i)) \supset z^i \\
EQ^{i+1} & \equiv & \lambda x^i y^i.(\forall z^i)((z^i x^i) \supset (z^i y^i))
\end{array}$$

We can observe that each connector and each quantifier is an $E\lambda$-term with a given level. By construction it follows, for example, that there is an infinite number of connectors of negation : $\neg^1, \neg^2, \ldots \neg^i, \neg^{i+1}, \ldots$

For example, to define the term representing the negation of the term X, where level(X) = k, we must apply the connector $\neg^{i+1} \equiv \lambda x^i.x^i \supset \perp$ with k $\leq$ i to X. Then $(\lambda x^i.x^i \supset \perp)X \rightarrow_{E\lambda\beta} X \supset \perp$.

The $E\lambda$ can be considered as a system of "typed logics". The system $E\lambda$ restricted to terms of level $\leq i$ (terms in C_i) noted $E\lambda_i$ can be viewed as logical system of "type i".

This idea of "typing logic" is similar to that found in the work of A. Markov [11] and also found in the work of A. Chauvin [5].

We will show that given the two constants Π with P (or $\supset$) and their introduction and elimination rules, one can derive in the system Eλ the elimination rule and the introduction rule of the other connectors (F, Ξ, G, $\vee$, ...).

Proposition 2.1. *Let X and Y be $E\lambda$-terms such that level$(X) \leq i$ and level$(Y) \leq i$.*
i) $\Xi^{i+1}XY,\ XU \vdash\ YU$;
ii) $Xz^i \vdash\ Yz^i \Longrightarrow\ \vdash \Xi^{i+1}XY$

Proposition 2.2. *If level(a) $\leq i$, level(b) $\leq i$, level(X) $\leq i$ and level(Y) $\leq i$ then:*
i) $F^{i+1}abX,\ aY \vdash b(XY)$;
ii) $aY \vdash b(XY) \Rightarrow\ \vdash F^{i+1}abX$.

Proposition 2.3. *If level(A)* $\leq i$ *and level(B)* $\leq i$*, then :*
i) $\Gamma,\ A \wedge B \vdash A$*;*
ii) $\Gamma \vdash A,\ \Gamma \vdash B \Rightarrow \Gamma \vdash A \wedge B$*.*

Proposition 2.4. . *If* $level(X) = level(Y) = i$ *then,*

(i) $\vdash EQ^{i+1}XX$

(ii) $EQ^{i+1}XY, Z^iX \vdash Z^iY$

(iii) $EQ^{i+1}XY \vdash EQ^{i+2}XY$

Proof. It is immediate by the definition of EQ. The property holds also when $level(Y) \leq level(X)$.

Proposition 2.5. *if* $level(X) = level(Y) = i$ *then,*

(i) $EQ^{i+1}XY \vdash EQ^{i+1}YX$;

(ii) $EQ^{i+1}XY, EQ^{i+1}YZ \vdash EQ^{i+1}XZ$;

(iii) $EQ^{i+1}XY \vdash EQ^{i+1}(ZX)(ZY)$;

(iv) $EQ^{i+1}XY \vdash EQ^{i+1}(XZ)(YZ)$

Proof of (i).

1) $EQ^{i+1}XY \quad \vdash \quad EQ^{i+2}XY$ by (iii) of prposition 2.4
2) $EQ^{i+1}XY \quad \vdash \quad EQ^{i+1}XX$ by (i) of prposition 2.4
3) $EQ^{i+1}XY \quad \vdash \quad (\lambda u^i.(\forall z^i)((z^iu^i) \supset (z^iX))X$ by rule $Eq_{E\lambda\beta}$ (it is in the form $Z^{i+1}X$)
4) $EQ^{i+1}XY \quad \vdash \quad (\lambda u^i.(\forall z^i)((z^iu^i) \supset (z^iX))Y$ by (1)+ (3) and (ii) of the proposition 2.4.
5) $EQ^{i+1}XY \quad \vdash \quad (\forall z^i)((z^iY) \supset (z^iX)) =_{E\lambda\beta} EQ^{i+1}YX$.

3 $E\lambda\beta$-reduction is Church-Rosser

The proof of Church-Rosser property for E$\lambda\beta$-reduction is similar to Martin-löf and Tait proof for the $\lambda\beta$-reduction in the classical λ-calculus (see [9]).

We need some preliminary results before starting the description of the proof.

Proposition 3.1. *1. If level(M)* $\leq$*level(x), then level([M/x]P)* $\leq$ *level(P)*

2. *if* $P \rightarrow_{E\lambda\beta} Q$ *then level(P)* $\geq$ *level(Q)*
3. *if* $P \rightarrow_{E\lambda\beta} Q$ *then* $[P/x]M \rightarrow_{E\lambda\beta} [Q/x]M$
4. *if* $x \neq y$, $y \notin FV(P)$, $level(P) \leq level(x)$ *and* $level(Q) \leq level(y)$, *then* $[P/x][Q/y]M \equiv [[P/x]Q/y][P/x]M$.
5. *if* $level(M) \leq level(x)$ *and* $P \rightarrow_{E\lambda\beta} Q$ *then,* $[M/x]P \rightarrow_{E\lambda\beta} [M/x]Q$.

Now let us introduce in the $E\lambda$-system the notion of minimal complete development as defined by Hindley (see Appendix of [9])

Let R$_1$, R$_2$,...,R$_n$ ($n \geq 1$) be redexes in a term P. An R$_i$ is called minimal iff it properly contains no other R$_j$. We say $P \rightarrow_{mcd} Q$ iff Q is obtained from P by a process called a minimal complete developement (MCD) of the set $\{R_1, ..., R_n\}$.

Lemma 3.2. *For $E\lambda\beta$-reduction, if $P \rightarrow_{mcd} P'$ and $Q \rightarrow_{mcd} Q'$, then $[Q/x]P \rightarrow_{mcd} [Q'/x]P'$ if $level(Q) \leq level(x)$.*

Lemma 3.3. *For $E\lambda\beta$-reduction: if $P \rightarrow_{mcd} X$ and $P \rightarrow_{mcd} Y$, then there exists T such that $X \rightarrow_{mcd} T$ and $Y \rightarrow_{mcd} T$.*

Theorem 3.4. *The $E\lambda\beta$-reduction is Church-Rosser.*

4 Consistency of the system Eλ

4.1 Curry's paradox

Curry's paradox is derived in illative combinatory logic (Curry's systems) as follows. Let X be a statement of illative combinatory logic, then $\vdash$ X. The proof for a given X is : Take Y $=_{\lambda\beta}$ (λx.(xx)$\supset$X)(λx.(xx) $\supset$ X), then Y $=_{\lambda\beta}$ (Y $\supset$ X). Therefore the following derivation shows that $\vdash$ X :

1. Y $\vdash$ Y
2. Y $\vdash$ Y $\supset$ X since Y $=_{\lambda\beta}$ (Y $\supset$ X)
3. Y $\vdash$ X 1) + 2) + introduction of $\supset$
4. $\vdash$ Y $\supset$ X introduction of $\supset$
5. $\vdash$ Y since Y $=_{\lambda\beta}$ (Y $\supset$ X).
6. $\vdash$ X 4) + 5) + elimination of $\supset$.

From the steps of this proof, Curry deduced that there is an incompatibility between the unrestricted rules for $\supset$ and the property that λx.M exists for each term X. To use λ-calculus as a basis for logic and overcome this incompatibility Curry and his school search for systems where only certain terms (canonical terms) can represent logical propositions.
In Eλ system Curry's paradox is avoided thanks to the notion of level for each term and a restriction of the application of a term X to another term Y. The term (λx.X)Y can be reduced only if the level of x $\geq$ level(Y) or X is a λ-term (term with level 0). Thus the term Y such that Y $=_{E\lambda\beta}$ (Y $\supset$ X) can not be constructed in the Eλ system. Let Y be the term (λx.(xx)$\supset$ X)(λx.(xx) $\supset$ X). If level (λx.$(xx) \supset X$) = j and level(x) = i, then by the construction of the term Y we have i<j. In this case the rule (Eβ) cannot be applied and the equality Y $=_{E\lambda\beta}$ $(Y \supset X)$ does not hold.

4.2 Consistency

In the following we will prove that the system Eλ is consistent in the sense that no arbitrary term (statement) Y is provable : In $E\lambda$, not every statement (term) is a theorem. Some properties of the system Eλ (first pointed to the authors by Bunder) such that (each term has a unique level and it is finitely based, the E$\lambda\beta$- reduction cannot increase level of a given term) can made that all cut formulas in all proofs can be eliminated.

We will represent a deduction $\Gamma \vdash$ Y in the Eλ system in a natural deduction style by a tree, made of formulas, the root of which is Y; the topmost formula are of two sorts : the discharged formulas and the other called hypothesis of a deduction.

$$\dfrac{\dfrac{\begin{matrix}D_1\\ \vdots\\ X\\ \vdots\\ Y\end{matrix}}{X \supset Y}\,(Pi) \qquad \begin{matrix}D_2\\ \vdots\\ X\end{matrix}}{Y}\,(Pe)$$

is a deduction of Y, and

$$\dfrac{\dfrac{\dfrac{\begin{matrix}D_1\\ \vdots\\ X\end{matrix}}{\Pi(\lambda x.X)}\,(\Pi i)}{(\lambda x.X)Y}\,(\Pi e)}{[Y/x]\,X}\,(=_{E\lambda\beta})$$

is a deduction of $[Y/x]$X.

Generally such deductions involve cut formulas, i.e formulas formed by an introduction rule that are immediately removed by an elimination rule with some (perhaps empty) equality steps.
In the first deduction the cut formula is $X \supset Y$. In the second deduction the cut formula is $\Pi(\lambda$x.X). Let us recall that in the rule (Pe) $X \supset Y$ is called the major premiss and X is called the minor premiss. In the rule (Πe), ΠX is the major premiss.

Now we can characterize the inconsistency proofs in the system Eλ by a similar result obtained by Bunder for illative combinatory logic (see [4]).

Definition 4.1. *(cut elimination). A deduction D_2 is said to be a reduction of D_1 if D_2 is obtained from D_1 by successively removing cut formulas in the way described below.*

P-*contraction*

$$\dfrac{\dfrac{\begin{matrix}D_1\\ \vdots\\ X\\ \vdots\\ Y\end{matrix}}{X \supset Y}\,(Pi) \qquad \begin{matrix}D_2\\ \vdots\\ X\end{matrix}}{Y}\,(Pe)$$

reduce to

$$\begin{array}{c} D_2 \\ \vdots \\ X \\ \vdots \\ D_1 \\ \vdots \\ Y \end{array}$$

Π-*contraction*

$$\dfrac{\dfrac{\dfrac{\begin{array}{c} D_1 \\ \vdots \\ X \end{array}}{\Pi(\lambda x.X)}\,(\Pi i)}{(\lambda x.X)Y}\,(\Pi e)}{[Y/x]\,X}\,(=_{E\lambda\beta})$$

reduce to

$$\begin{array}{c} [Y/x]\,D_1 \\ \vdots \\ [Y/x]\,X \end{array}$$

We prove the following results.

Theorem 4.1. *$E\lambda$ is inconsistent only if all inconsistency proof contain a cut formula of the form $X \supset Y$ or $\Pi(\lambda x.X)$.*

Theorem 4.2. *In the system $E\lambda$ all deductions can be made cut free.*

The proof is by a double induction on the maximum level of cut formula and the number of formulas having this level.

The induction value of a deduction **D** is then defined as the pair $< m, n >$ such that m is the highest level of formula in **D** and n is the number of formulas of level m. The induction value $< m', n' >$ is less than the induction value $< m, n >$ if and only if :

i) m' < m or
ii) m = m' and n' < n

Let $< m, n >$ the induction value of a deduction **D** (m>0 , n>0). Take A the first cut formula of maximal level. If A is of type 1 (say $X \supset Y$), eliminate it by reducing the proof by applying the P-contraction. This may introduce a new cut formulae X or Y, but these will be of lower level than $X \supset Y$.
The new induction value is then :

i) $< m, n-1 >$ or
ii) $< m', n' >$ with $m' < m$;

in both cases it is less than $< m, n >$.

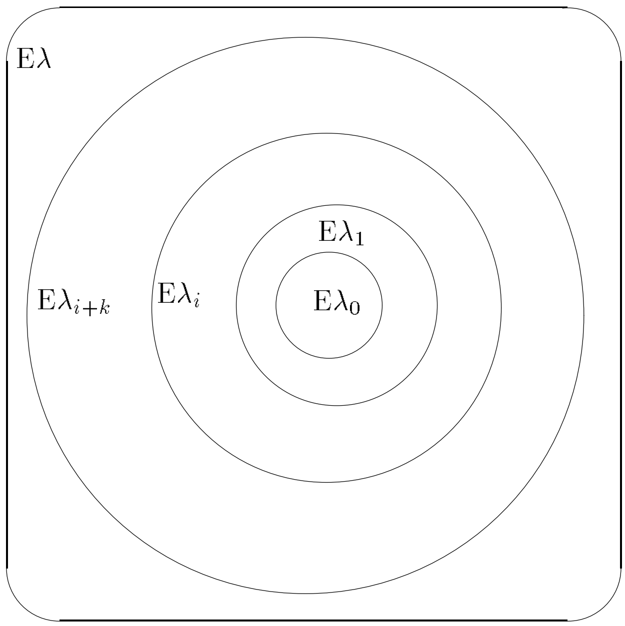
If the first formula of maximal level is of type 2 (say $\Pi(\lambda x.X)$), eliminate it by reducing the proof by applying Π-contraction. This may introduce a new cut formula at $[Y/x]$X and if level(Y) = level(x), this has the same level as the cut formula that has been eliminated. This reduction however shortens the proof, so, if we continue the procedure by eliminating the new cut formula, the process must terminate eventually either with no cut formula or with a cut formula of type 1 which can be eliminated to give at most cuts of lower level. Hence the theorem holds.

Corollary 4.3. *$E\lambda$ is consistent.*

Proof. It is immediate by theorem 4.1 and theorem 4.2.

What we really proved is that, in the system Eλ, not all term with level i are theorems ($\not\vdash$ X^i). But, we have no proof that for all level i not all terms X^i are theorems. It is the consequence of Gödel incompleteness theorem.

Now, we will introduce the notion of level of a proof of a statement X. As usual we consider a proof in the system $E\lambda$ as a sequence of statement $X_1, X_2, \dots X_n \equiv X$ where any X_i is deduced from X_j with $j \leq i$ by application of one of the deduction rules of the system. We write *Proof*(X) to denote the sequence $X_1, X_2, \dots X_n$. Then the level of the proof of X is given by the relation level(*Proof*(X)) = max(level(X_1, level(S_2), ..., level(X_n)). Let us write $\vdash_{E\lambda_i}$X if level(*Proof*(X))$\leq$ i. Eλ_i is the set of all terms X such that $level(X) \leq i$. It is immediate that $E\lambda_0 \subset E\lambda_1 \subset \dots \subset E\lambda_i \subset \dots$ and $E\lambda = \bigcup_{n<\omega} E\lambda_n$ where ω is the first limit ordinal.



It will be easy to show the following proposition.

Proposition 4.4. *1.* $\vdash_{E\lambda_i} X \Rightarrow \vdash_{E\lambda} X$

2. $\vdash_{E\lambda} X \Rightarrow \exists j \vdash_{E\lambda_j} X$

In the system Eλ we can deduce from the hypothesis EQ(`true`)(`false`) the absurdity ($\perp$) where `true` and `false` are respectively the terms λxy.x and λxy.y. Let us consider the term $(\lambda x.xT \perp)$ then we have $(\lambda x.xT \perp)\mathtt{true} =_{E\lambda\beta} T$ (here T is the term $\forall$x.(Pxx)). Let us note, before giving the proof, that it is easy to show that EQ`truefalse` imply P(`true`)(`false`).

Proposition 4.5. .
$EQ(\mathtt{true})(\mathtt{false}) \vdash \perp$

Proof.

1)	$\vdash$	EQ(`true`)(`false`)	hypothesis.
2)	$\vdash$	$(\forall x.Pxx) \equiv T$	T is a theorem.
3)	$\vdash$	$(\lambda x.xT\perp)\mathtt{true}$	by $(\lambda x.xT\perp)\mathtt{true} =_{E\lambda\beta}$ T and rule $Eq_{E\lambda\beta}$ (see definition 2.6.)
4)	$\vdash$	$(\lambda x.xT\perp)(\mathtt{false}))$	1 + 3 + (ii) of propostion 2.4.
5)	$\vdash$	$\perp$	by (4) and rule $Eq_{E\lambda\beta}$

5 Proposition as types and proofs in $E\lambda$

Let (a $\rightarrow$ (b $\rightarrow$ a)) be a type in the usual notation. In the $E\lambda$ it will be written $Fa(Fba)$ where F is the term $\lambda xyz.\forall u(xu \supset y(zu))$. Now, if we say that the term $Fa(Fba)$ is the type of a term X,then we write in $E\lambda$ $Fa(Fba)X$.To search the proof X of the proposition $Fa(Fba)$, we proceed as following.

$Fa(Fba)X$	$\vdash$	$Fa(Fba)X$
	$\vdash$	$\forall u(au \supset \forall v(bv \supset a(Xuv)))$
	$\vdash$	$(au \supset \forall v(bv \supset a(Xuv)))$
au	$\vdash$	$\forall v(bv \supset a(Xuv)))$
au	$\vdash$	$(bv \supset a(Xuv))$
au, bv	$\vdash$	$a(Xuv)$

We have $au = a(Xuv)$, then $Xuv = u$ and $X = \lambda uv.u$.

Another interesting example is the proof of the proposition:
$\forall P(P0 \rightarrow ((\forall n \in N(Pn \rightarrow P(succ\ n))) \rightarrow (\forall n \in N)Pn))$.
If we look for a term X such that :
$\vdash \forall P(P0 \rightarrow ((\forall n \in N(Pn \rightarrow P(succ\ n))) \rightarrow (\forall n \in N)Pn))X$;
we find that X is the recursion combinator R.

6 Conclusion

The system $E\lambda$ is a consistent higher order logic. Computer mathematics can be developed in it. This answers the open question asked by Barendregt in [2] for illative combinatory logic systems. In further work, we will deal with representing mathematics and developing proof-checker for the $E\lambda$ system.

Acknowledgements
We would like to thank W. Bunder, Gilles Dowek, R. Hindley, Thérèse Hardin, and J. Seldin for their many many helpful comments and suggestions.

References

[1] H. Barendregt. The lambda-calculus : Its syntax and semantics Series "studies in logic", North Holland Co. Amsterdam (1984).

[2] H.P. Barendregt. The impact of the lambda calculus in logic and computer science. (1997). Available by ftp.

[3] M.W Bunder. The inconsistency of higher order predicate calculus and set theory based on combinatory logic. Mathematical logic in Latin America(proceedings of the fourth symposium) A.I. Arruda et al.(Editors) North Holland company Amsterdam 1980, pp 99-107.

[4] M.W Bunder. Some consistency proofs and a charaterization of inconsistency proofs in illative combinatory logic, The Journal of Symbolic Logic, Volume 52, Number 1 (1987) pp 89-110.

[5] A. Chauvin, Une notion dynamique de fonction et sa logique; unpublished notes (1989).

[6] H. Curry, R. Hindley and J. Seldin. Combinatory logic II. North Holland compagny Amsterdam, 1972.

[7] W. Dekkers, M. Bunder, H.P. Barendregt. Completeness of two systems of illative combinatory logic for first-order propositional and predicate calculus. The Journal of Symbolic logic(1997).

[8] K. Grue. Map Theory, Theoretical computer science 101 (1990).

[9] R. Hindley and J. Seldin. Introduction to combinators and lambda-calculus. Cambridge University Press (1986).

[10] E. Mendelson. Introduction to Mathematical Logic, van Nostrand New York 1967.

[11] A. Markov, Essai de construction d'une logique de la mathématique constructive; Revue internationale de philosophie 25^{me} année 1971 fasicule 4 pp.477-507.