Typed λ-calculi with one binder

Fairouz Kamareddine
School of Mathematical and Computer Sciences, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, Scotland,
http://www.macs.hw.ac.uk/~fairouz

Abstract

Type theory was invented at the beginning of the twentieth century with the aim of avoiding the paradoxes which result from the self-application of functions. λ-calculus was developed in the early 1930s as a theory of functions. In 1940, Church added type theory to his λ-calculus giving us the influential simply typed λ-calculus where types were simple and never created by binders (or abstractors). However, realising the limitations of the simply typed λ-calculus, in the second half of the twentieth century we saw the birth of new more powerful typed λ-calculi where types are indeed created by abstraction. Most of these calculi use two binders λ and Π to distinguish between functions (created by λ-abstraction) and types (created by Π-abstraction). Moreover, these calculi allow β-reduction but not Π-reduction. That is, $(\pi_{x:A}.B)C \to B[x := C]$ is only allowed when π is λ but not when it is Π. This means that, modern systems do not allow types to have the same instantiation right as functions. In particular, when b has type B, the type of $(\lambda_{x:A}.b)C$ is taken immediately to be $B[x := C]$ instead of $(\Pi_{x:A}.B)C$. Extensions of modern type systems with both Π-reduction and type instantiation have appeared in (Kamareddine, Bloo and Nederpelt, 1999; Kamareddine and Nederpelt, 1996; Peyton-Jones and Meijer, 1997). This makes the λ and Π very similar and hence leads to the obvious question: why not use a unique binder instead of the λ and Π? This makes more sense since already, versions of de Bruijn's Automath unified λ and Π giving more elegant systems. This paper studies the main properties of type systems with unified λ and Π.

Contents

1 Introduction

In Church's simply typed λ-calculus, the function which takes $f : \mathbb{N} \to \mathbb{N}$ and $x : \mathbb{N}$ and returns $f(f(x))$ is given below together with its type:

doubling function on $\mathbb{N}$	$\lambda_{f:\mathbb{N}\to\mathbb{N}}.\lambda_{x:\mathbb{N}}.f(f(x))$
type of doubling function on $\mathbb{N}$	$(\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$

If we want the same function on booleans $\mathcal{B}$, we would need to write:

doubling function on $\mathcal{B}$	$\lambda_{f:\mathcal{B}\to\mathcal{B}}.\lambda_{x:\mathcal{B}}.f(f(x))$
type of doubling function on $\mathcal{B}$	$(\mathcal{B} \to \mathcal{B}) \to (\mathcal{B} \to \mathcal{B})$

Instead of repeating the work, we can bind the varying type α. So, if we let $\alpha : *$ stand for "α is a type" (any type), we can define the polymorphic doubling function in the polymorphic λ-calculus, as follows:

polymorphic doubling function $\qquad \lambda_{\alpha:*}.\lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)).$

Now, we can instantiate α to what we need:

- $\alpha = \mathbb{N}$ then: $(\lambda_{\alpha:*}.\lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)))\mathbb{N} = \lambda_{f:\mathbb{N}\to\mathbb{N}}.\lambda_{x:\mathbb{N}}.f(f(x))$.
- $\alpha = \mathcal{B}$ then: $(\lambda_{\alpha:*}.\lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)))\mathcal{B} = \lambda_{f:\mathcal{B}\to\mathcal{B}}.\lambda_{x:\mathcal{B}}.f(f(x))$.
- $\alpha = (\mathcal{B} \to \mathcal{B})$ then: $(\lambda_{\alpha:*}.\lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)))(\mathcal{B} \to \mathcal{B}) =$
  $\lambda_{f:(\mathcal{B}\to\mathcal{B})\to(\mathcal{B}\to\mathcal{B})}.\lambda_{x:(\mathcal{B}\to\mathcal{B})}.f(f(x))$.

So, types (like terms) can be abstracted over and can be passed as arguments. The types of the new polymorphic terms are given by a new binder usually written as $\forall$ or Π. We use Π. The type of the polymorphic doubling function is:

type of polymorphic doubling function $\qquad \Pi_{\alpha:*}.(\alpha \to \alpha) \to (\alpha \to \alpha).$

Hence, unlike simple types, modern non-simple types have similar features to functions. In particular, like functions, types can be:

- *Created by abstraction.* Functions are created via λ where $\lambda_{x:A}.B$ stands for the *function* from A to B which given $a \in A$ returns $B[x := a]$ (i.e., B where a is substituted for x); and types are created via Π where $\Pi_{x:A}.B$ stands for the *type* of the functions from A to $\cup_{a\in A}B[x := a]$ which given $a \in A$ return $fa \in B[x := a]$. For example, the type $\Pi_{A:*}.A \to A$ of the polymorphic identity function $\lambda_{A:*}.\lambda_{y:A}.y$, is obtained by taking any type A and returning the type $A \to A$ of the identity function on A, $\lambda_{y:A}.y$.
- *Instantiated.* For example, if A above is the set of natural numbers $\mathbb{N}$ then we are concerned with the identity function over $\mathbb{N}$ whose type is $A \to A$ where A is substituted by $\mathbb{N}$ (written $(A \to A)[A := \mathbb{N}]$), i.e., $\mathbb{N} \to \mathbb{N}$.

Looking at the behaviour of λ and Π, it seems questionable why one needs two different binders. In fact, in the literature, there were several attempts to unify the binders λ and Π in type systems:

- Sometimes, in his Automath, de Bruijn identified the abstractions obtained by λ and Π. He wrote $[x : A]B$ for both $\lambda_{x:A}.B$ and $\Pi_{x:A}.B$. But what are the properties of such type systems and is there a correspondence between ordinary type systems and those where abstractions are identified?

- (Kamareddine, Bloo and Nederpelt, 1999; Kamareddine and Nederpelt, 1996; Peyton-Jones and Meijer, 1997) argued that Π-reduction and β-reduction should be both allowed. I.e., $(\Pi_{x:A}B)C \to_{\Pi} B[x := C]$ and $(\lambda_{x:A}B)C \to_{\beta} B[x := C]$ should be both allowed. Moreover, Π-reduction was a main feature of Automath (de Bruijn, 1970). When de Bruijn did not identify λ and Π, he gave Π-terms the same instantiation power as λ-terms and allowed Π-reduction. In some sense, adding Π-reduction to a type system has similar effect as replacing the λ and Π by a unique binder.
- In a private communication, during his PhD studies, Laan attempted to unify binders in the cube, however, no progress was made there except stating (without any proof) a generation lemma and a weaker form of isomorphism.
- In (Coquand, 1985), Coquand first presented the calculus of constructions using de Bruijn's identification of binders. However, he did not investigate the connection with type systems where binders have not been identified, nor did he establish how contexts, terms and types behave under the exchange of binders.
- (de Groote, 1993) defined a system λ^{λ} which departs from the usual systems as in for example, the Barendregt cube of (Barendregt, 1992), in the sense that degrees are no longer restricted to 0, 1, 2 or 3. The system λ^{λ} uses the same binder λ for both λ and Π.

Despite the above mentioned work, modern type systems with unified binders have still not been investigated. Although (Kamareddine, 2002) gave a tutorial on functions and types in which unified binders also featured, this unification concentrated on the concepts of parameters, definitions, Π-reduction and explicit substitutions, and studied an extension containing all these concepts. This is unsatisfactory since there is no agreement on which system of explicit substitution should be used (or indeed whether one needs explicit substitution at all), and the same holds for systems of definitions. So, how can the idea of unifying binders be accepted if it is built on top of controversial calculi of definitions and explicit substitutions? This paper fills these gaps and gives the first extensive account of modern type systems (as we know them, without any controversial extensions) where the λ and the Π are unified. We carry our study in Barendregt's β-cube (Barendregt, 1992) which hosts eight influential type systems.

The paper is divided as follows: Section 2 presents the basic notions of reduction and typing and relates flat terms (where binders are unified) to ordinary terms. In Section 3 we review the β-cube and establish the properties of typing modulo flattened binders. We show that in any typing judgement of the β-cube, λs and Πs cannot be exchanged and hence, from the judgement itself, one can decide the status of any binder. So, why use different binders when the typing judgement carries the unique identity of a binder? In Section 4, we present the $\flat$-cube where both λ and Π are written as $\flat$. We show that this $\flat$-cube satisfies all the desirable properties except for the unicity of types. We also show that this $\flat$-cube is isomorphic to the β-cube in the sense that for any typing judgement in the $\flat$-cube, there corresponds a *unique* typing judgement in the β-cube. We show furthermore that despite the loss of the unicity of types, all the different types of the same term obey the same pattern. In Section 5, we discuss type checking and type inference. In Section 6, we discuss Coquand's calculus of constructions with unified binders. In Section 7 we conclude.

2 Notions of reduction and typing

In this section we present the basic notions of reduction and typing. We use two basic sets of terms: the set $\mathcal{T}$ of typed terms as written in modern type systems and the set $\mathcal{T}_\flat$ where λ and Π have been flattened into the single binder $\flat$.

Definition 1. [Terms and translations] We let π range over $\{\lambda, \Pi\}$.
1. We define the set of terms $\mathcal{T}$ by: $\mathcal{T} ::= * \mid \Box \mid \mathcal{V} \mid \pi_{\mathcal{V}:\mathcal{T}}.\mathcal{T} \mid \mathcal{T}\mathcal{T}$.
2. We define the set of $\flat$-terms (or terms when no confusion occurs) $\mathcal{T}_\flat$ by: $\mathcal{T}_\flat ::= * \mid \Box \mid \mathcal{V} \mid \flat_{\mathcal{V}:\mathcal{T}_\flat}.\mathcal{T}_\flat \mid \mathcal{T}_\flat\mathcal{T}_\flat$.
3. For $A \in \mathcal{T}$, we define $\overline{A} \in \mathcal{T}_\flat$ by: $\overline{s} \equiv s, \overline{x} \equiv x, \overline{AB} \equiv \overline{A}\,\overline{B}, \overline{\pi_{x:A}.B} \equiv \flat_{x:\overline{A}}.\overline{B}$.
4. Let $A \in \mathcal{T}_\flat$. We define: $[A]$ to be $\{A' \text{ in } \mathcal{T} \mid \overline{A'} \equiv A\}$. We also define $A^\lambda \in \mathcal{T}$ by: $s^\lambda \equiv s, x^\lambda \equiv x, (AB)^\lambda \equiv A^\lambda B^\lambda$ and $(\flat_{x:A}.B)^\lambda \equiv \lambda_{x:A^\lambda}.B^\lambda$.

Note that, if $A \in \mathcal{T}$ then $A \in [\overline{A}]$. Moreover, if $A \in \mathcal{T}_\flat$ then $A^\lambda \in [A]$.

Notation 2. *We let s, s', s_1, etc. range over the sorts $\{*, \Box\}$. We take $\mathcal{V}$ to be a set of variables over which, x, y, z, x_1, etc. range. We divide $\mathcal{V}$ into two disjoint subsets $\mathcal{V}^*$ and $\mathcal{V}^\Box$. We use x^s, y^s, etc., to range over $\mathcal{V}^s$. We assume that $\{*, \Box\} \cap \mathcal{V} = \emptyset$. We take A, A_1, A_2, B, a, b, etc. to range over both $\mathcal{T}$ and $\mathcal{T}_\flat$. We use $\textsc{fv}(A)$ to denote the free variables of A, and $A[x := B]$ to denote the substitution of all the free occurrences of x in A by B. We assume familiarity with the notion of* compatibility. *As usual, we take terms to be equivalent up to variable renaming and let $\equiv$ denote syntactic equality. We assume the Barendregt convention (BC) where names of bound variables are chosen to differ from free ones in a term and where different abstraction operators bind different variables. Hence, for example, we write $(\pi_{y:A}.y)x$ instead of $(\pi_{x:A}.x)x$ and $\pi_{x:A}.\pi_{y:B}.C$ instead of $\pi_{x:A}.\pi_{x:B}.C$. We also assume (BC) for contexts and typings so that for example, if $\Gamma \vdash \pi_{x:A}.B : C$ then x will not occur in Γ. We define subterms in the usual way. For $\Lambda \in \{\lambda, \Pi, \flat\}$, we write $\Lambda_{x_m:A_m} \ldots \Lambda_{x_n:A_n}.A$ as $\Lambda^{i:m..n}_{x_i:A_i}.A$.*

2.1 Reduction

Definition 3. [Reductions]
- β-reduction $\to_\beta$ is the compatible closure of $(\lambda_{x:A}.B)C \to_\beta B[x := C]$.
- $\flat$-reduction $\to_\flat$ is the compatible closure of $(\flat_{x:A}.B)C \to_\flat B[x := C]$.
- Π-reduction $\to_\Pi$ is the compatible closure of $(\Pi_{x:A}.B)C \to_\Pi B[x := C]$.
- We define the union of reduction relations as usual. For example, $\beta\Pi$-reduction $\to_{\beta\Pi}$, is the union of $\to_\beta$ and $\to_\Pi$.
- Let $r \in \{\beta, \Pi, \beta\Pi, \flat\}$. We define r-redexes in the usual way. Moreover:
  - $\twoheadrightarrow_r$ is the reflexive transitive closure of $\to_r$ and $=_r$ is the equivalence closure of $\to_r$. We write $\overset{+}{\twoheadrightarrow}_r$ to denote one or more steps of r-reduction.
  - If $A \to_r B$ (resp. $A \twoheadrightarrow_r B$), we also write $B\ {}_r\!\leftarrow A$ (resp. $B\ {}_r\!\twoheadleftarrow A$).

— We say that A is strongly normalising with respect to $\to_r$ (we use the notation $\mathrm{SN}_{\to_r}(A)$) if there are no infinite $\to_r$-reductions starting at A.
— We say that A is in r-normal form if there is no B such that $A \to_r B$.
— We use $\mathrm{nf}_r(A)$ to refer to the r-normal form of A if it exists.

Theorem 4 (Church-Rosser for $\mathcal{T}$ and $\to_{\beta/\beta\Pi/\flat}$). *Let $r \in \{\beta, \beta\Pi, \flat\}$. If $B_1 \ {}_r\!\twoheadleftarrow A \twoheadrightarrow_r B_2$ then there exists $C \in \mathcal{T}$ such that $B_1 \twoheadrightarrow_r C \ {}_r\!\twoheadleftarrow B_2$.*

Proof. For β see (Barendregt, 1992). For $\beta\Pi$ see (Kamareddine, Bloo and Nederpelt, 1999). For $\flat$, note that for $A \in \mathcal{T}_\flat$, $A^\lambda \in \mathcal{T}$. ⊠

Corollary 5. *For $r \in \{\beta, \beta\Pi, \flat\}$, r-normal forms are unique. Moreover, if $SN_{\to_\flat}(\flat^{i:1..n}_{x_i:B_i}.A)$, $SN_{\to_\flat}(\flat^{j:1..m}_{y_j:C_j}.A)$ and $n \neq m$ then $\flat^{i:1..n}_{x_i:B_i}.A \neq_\flat \flat^{j:1..m}_{y_j:C_j}.A$.*

The next lemma will be used to connect the different kinds of terms.

Lemma 6.

1. *If $A, B \in \mathcal{T}$ then $\overline{A[x := B]} \equiv \overline{A}[x := \overline{B}]$.*
2. *Let $A, B \in \mathcal{T}_\flat$ and $R \in \{\to, \twoheadrightarrow\}$. If $AR_\flat B$ then for all $A' \in [A]$ there is $B' \in [B]$ such that $A'R_{\beta\Pi}B'$.*
3. *Let $A, B \in \mathcal{T}$, $r \in \{\beta, \beta\Pi\}$ and $R \in \{\to, \twoheadrightarrow, =\}$. If AR_rB then $\overline{A}R_\flat\overline{B}$.*
4. *Let $A \in \mathcal{T}$. a) If $SN_{\to_{\beta\Pi}}(A)$ then $SN_{\to_\flat}(\overline{A})$.*
   b) If A is in $\beta\Pi$-normal form then $\overline{A}$ is in $\flat$-normal form.
5. *Let $r \in \{\beta, \beta\Pi\}$ and $A \in \mathcal{T}_\flat$. a) If $SN_{\to_\flat}(A)$ then $SN_{\to_r}(A')$ for all $A' \in [A]$.*
   b) If A is in $\flat$-normal form then A' is in r-normal form for all $A' \in [A]$.
6. *Let $A \in \mathcal{T}$. A is in $\beta\Pi$-normal form if and only if $\overline{A}$ is in $\flat$-normal form.*

Proof. 1. By induction on the structure of A.
2. $\to_\flat$: induction on $A \to_\flat B$ using 1. $\twoheadrightarrow_\flat$: induction on the length of $A \twoheadrightarrow_\flat B$.
3. $\to_r$ and $\twoheadrightarrow_r$: similar to 2. $=_r$: use Church-Rosser and the property for $\twoheadrightarrow_r$.
4. 2 maps any $\to_\flat$-path from $\overline{A}$ into the same length $\to_{\beta\Pi}$-path from $A \in [\overline{A}]$.
5. 3 maps any $\to_r$-path from $A' \in [A]$ into the same length $\to_\flat$-path from A.
6. This is a corollary of 4 and 5 above. ⊠

Remark 7. *In Lemma 6.2 and 6.4, we cannot replace $\beta\Pi$ by β. For example:*

- *$(\flat_{x:*}.x)y \to_\flat y$ and $(\Pi_{x:*}.x)y \in [(\flat_{x:*}.x)y]$ but $(\Pi_{x:*}.x)y$ is in β-normal form.*
- *$(\flat_{x:*}.xy)(\flat_{z:*}.z) \twoheadrightarrow_\flat y$ and $(\lambda_{x:*}.xy)(\Pi_{z:*}.z) \in [(\flat_{x:*}.xy)(\flat_{z:*}.z)]$ but $(\lambda_{x:*}.xy)(\Pi_{z:*}.z) \not\twoheadrightarrow_\beta C$ where $C \in [y]$.*
- *$SN_{\to_\beta}((\Pi_{x:*}.xx)(\Pi_{x:*}.xx))$ but it is not the case that $SN_{\to_\flat}((\flat_{x:*}.xx)(\flat_{x:*}.xx))$.*

The next lemma relates normal forms in $\mathcal{T}$ and $\mathcal{T}_\flat$.

Lemma 8. *1. If $SN_{\to_{\beta\Pi}}(\pi^{i:1..n}_{x_i:C_i}.A)$, $SN_{\to_{\beta\Pi}}(\pi^{j:1..m}_{y_j:D_j}.B)$, $\overline{A} \equiv \overline{B}$ and $n \neq m$ then $\pi^{i:1..n}_{x_i:C_i}.A \neq_{\beta\Pi} \pi^{j:1..m}_{y_j:D_j}.B$.*
2. Let $SN_{\to_{\beta\Pi}}(A)$. a) $\overline{\mathrm{nf}_{\beta\Pi}(A)} \equiv \mathrm{nf}_\flat(\overline{A})$. b) If $\overline{A} \equiv \overline{B}$ then $\overline{\mathrm{nf}_{\beta\Pi}(A)} \equiv \overline{\mathrm{nf}_{\beta\Pi}(B)}$.

Proof.

1. By Lemma 6.4 $\mathrm{SN}_{\to_\flat}(\flat^{i:1..n}_{x_i:\overline{C_i}}.\overline{A})$ and $\mathrm{SN}_{\to_\flat}(\flat^{j:1..m}_{y_j:\overline{D_j}}.\overline{B})$. If $\pi^{i:1..n}_{x_i:C_i}.A =_{\beta\Pi} \pi^{j:1..m}_{y_j:D_j}.B$ by Lemma 6.3, $\flat^{i:1..n}_{x_i:\overline{C_i}}.\overline{A} =_\flat \flat^{j:1..m}_{y_j:\overline{D_j}}.\overline{B} \equiv \flat^{j:1..m}_{y_j:\overline{D_j}}.\overline{A}$, contradicting Corollary 5.

2. a) By Lemma 6.4, $\mathrm{SN}_{\to_\flat}(\overline{A})$. By Lemma 6.6, $\overline{\mathrm{nf}_{\beta\Pi}(A)}$ is in $\flat$-normal form. By Lemma 6.3, $\overline{A} \twoheadrightarrow_\flat \overline{\mathrm{nf}_{\beta\Pi}(A)}$. By Corollary 5, $\overline{\mathrm{nf}_{\beta\Pi}(A)} \equiv \mathrm{nf}_\flat(\overline{A})$.
   b) By Lemma 6.4, $\mathrm{SN}_{\to_\flat}(\overline{A})$ and $\mathrm{SN}_{\to_\flat}(\overline{B})$. As $B \in [\overline{A}]$, by Lemma 6.5, $\mathrm{SN}_{\to_{\beta\Pi}}(B)$. Since $\mathrm{nf}_\flat(\overline{A}) \equiv \mathrm{nf}_\flat(\overline{B})$, by a) $\overline{\mathrm{nf}_{\beta\Pi}(A)} \equiv \overline{\mathrm{nf}_{\beta\Pi}(B)}$. ⊠

2.2 Typing

Definition 9. [Declarations, contexts, $\subseteq$]
1. A *declaration* d is of the form $x : A$. We define $\mathtt{var}(d) \equiv x$, $\mathtt{type}(d) \equiv A$, and $\mathrm{FV}(d) = \mathrm{FV}(A)$. We let $d, d', d_1, \ldots$ range over declarations.
2. A *context* Γ is a concatenation of declarations $d_1, d_2, \cdots, d_n$ such that if $i \neq j$ then $\mathtt{var}(d_i) \not\equiv \mathtt{var}(d_j)$. We define $\mathrm{DOM}(\Gamma) = \{\mathtt{var}(d) \mid d \in \Gamma\}$ and use $\langle\rangle$ to denote the *empty context*. We let $\Gamma, \Delta, \Gamma', \Gamma_1, \ldots$ range over contexts.
3. Assume Γ is a context such that $x \notin \mathrm{DOM}(\Gamma)$. We define the substitution of A for x on Γ, denoted $\Gamma[x := A]$, inductively as follows:
   $\langle\rangle[x := A] \equiv \langle\rangle$, and $(\Gamma', y : B)[x := A] \equiv \Gamma'[x := A], y : B[x := A]$.
4. We define $\subseteq$ between contexts as the least reflexive transitive relation closed under: $\Gamma, \Delta \subseteq \Gamma, d, \Delta$.

We extend the translations in Definition 1 to contexts as follows:
- In $\mathcal{T}$: $\overline{\langle\rangle} \equiv \langle\rangle$ $\quad \overline{\Gamma, x : A} \equiv \overline{\Gamma}, x : \overline{A}$.
- In $\mathcal{T}_\flat$: $[\Gamma] \equiv \{\Gamma' \mid \overline{\Gamma'} \equiv \Gamma\}$.

Since we want to assess unified binders in a variety of type systems, we chose to use the eight powerful systems of Barendregt's β-cube. In the β-cube of (Barendregt, 1992), eight well-known type systems are given in a uniform way. The weakest system is Church's simply typed λ-calculus $\beta\!\!\to$ (Church, 1940), and the strongest system is the Calculus of Constructions β_C (Coquand, 1988). The second order λ-calculus (Girard, 1972; Reynolds, 1974) figures on the β-cube between $\beta\!\!\to$ and β_C (cf. Figure 2). Moreover, via the Propositions-as-Types principle (see (Howard, 1980)), many logical systems can be described in the β-cube.

The β-cube has two sorts $*$ (the set of types) and $\Box$ (the set of kinds) where $* : \Box$. If $A : *$ (resp. $A : \Box$) we say A is a type (resp. a kind). All systems of the β-cube have the same typing rules (cf. Figure 1) but differ by the set $\boldsymbol{R}$ of pairs of sorts (s_1, s_2) allowed in the *type-formation* or Π-*formation* rule, (Π). Each system has its own set $\boldsymbol{R}$ such that $(*, *) \in \boldsymbol{R} \subseteq \{(*, *), (*, \Box), (\Box, *), (\Box, \Box)\}$. With rule (Π), the β-cube factorises the expressive power of β_C into three features: *polymorphism, type constructors*, and *dependent types*:
- $(*, *)$ is basic. All the β-cube systems have this rule.
- $(\Box, *)$ takes care of polymorphism. β_2 is the weakest system on the β-cube that features this rule.
- $(\Box, \Box)$ takes care of type constructors. $\beta_{\underline{\omega}}$ is the weakest system on the β-cube that features this rule.
- $(*, \Box)$ takes care of term dependent types. β_P is the weakest system on the β-cube that features this rule.

These features make the β-cube an excellent bed for testing unified binders. Since we will give another cube (the $\flat$-cube), we refer to each system of Figure 2 according

(axiom)	$\langle\rangle \vdash * : \Box$
(start)	$\dfrac{\Gamma \vdash A : s \qquad x^s \notin \mathrm{DOM}\,(\Gamma)}{\Gamma, x^s{:}A \vdash x^s : A}$
(weak)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s \qquad x^s \notin \mathrm{DOM}\,(\Gamma)}{\Gamma, x^s{:}C \vdash A : B}$
(Π)	$\dfrac{\Gamma \vdash A : s_1 \qquad \Gamma, x{:}A \vdash B : s_2 \qquad (s_1, s_2) \in \boldsymbol{R}}{\Gamma \vdash \Pi_{x:A}.B : s_2}$
(λ)	$\dfrac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash \Pi_{x:A}.B : s}{\Gamma \vdash \lambda_{x:A}.b : \Pi_{x:A}.B}$
(conv_β)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_\beta B'}{\Gamma \vdash A : B'}$
(appl)	$\dfrac{\Gamma \vdash F : \Pi_{x:A}.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[x{:=}a]}$

Fig. 1. Typing rules with two binders λ and Π

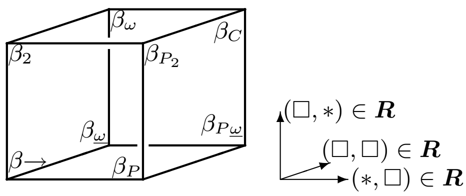

Fig. 2. Barendregt's β-cube

to the cube we are in. So, β_C resp. $\flat_C$, is the calculus of constructions of the β-cube resp. the ♭-cube. Now we give basic notions of type systems:

Definition 10. [Statements, judgements]
1. $\Gamma \vdash A : B$ is a *judgement* which states that A has type B in context Γ. When Γ is empty, we simply write $\vdash A : B$.
2. Γ is $\vdash$-*legal* (or simply *legal*) if there exist A, B where $\Gamma \vdash A : B$.
3. A is $\vdash$-*legal* (or simply *legal*) if there exist Γ, B where $\Gamma \vdash A : B \vee \Gamma \vdash B : A$.
4. A is $\Gamma\vdash$-*legal* (or simply Γ-*legal*) if there exists B where $\Gamma \vdash A : B \vee \Gamma \vdash B : A$.

2.3 Desired Lemmas for Type Systems

Lemma 11 (Free Variable Lemma for $\vdash$ and $\to_r$).
1. *If $x : A$ and $y : B$ are different elements in a legal context Γ, then $x \not\equiv y$.*
2. *If $\Gamma_1, x : A, \Gamma_2 \vdash B : C$ then $\mathrm{FV}(A) \subseteq \mathrm{DOM}\,(\Gamma_1)$ and $\mathrm{FV}(B), \mathrm{FV}(C) \subseteq \mathrm{DOM}\,(\Gamma_1, x : A, \Gamma_2)$.*

Lemma 12 (Start/Context Lemma for $\vdash$ and $\to_r$). *If Γ is $\vdash$-legal then*
1. $\Gamma \vdash * : \Box$ *and for all $x : A \in \Gamma$, $\Gamma \vdash x : A$.*

2. *If* $\Gamma \equiv \Gamma_1, x : A, \Gamma_2$ *then* $\Gamma_1 \vdash A : s$ *for some sort* s.

Lemma 13 (Thinning Lemma for $\vdash$ and $\to_r$).
If Γ *and* Δ *are* $\vdash$*-legal,* $\Gamma \subseteq \Delta$*, and* $\Gamma \vdash A : B$ *then* $\Delta \vdash A : B$.

Lemma 14 (Substitution Lemma for $\vdash$ and $\to_r$). *Let* $\Gamma, x : A, \Delta$ *be* $\vdash$*-legal*
If $\Gamma, x : A, \Delta \vdash B : C$ *and* $\Gamma \vdash a : A$ *then* $\Gamma, \Delta[x := a] \vdash B[x := a] : C[x := a]$.

Lemma 15 (Generation Lemma for $\vdash$ and $\to_r$).
1. *If* $\Gamma \vdash s : C$ *then* $s \equiv *$ *and* $C \equiv \Box$.
2. *If* $\Gamma \vdash x : C$ *then for some* s, A, $x : A \in \Gamma$, $C =_r A$, $x \equiv x^s$ *and* $\Gamma \vdash C : s$.
3. *If* $r = \beta$ *then*
   (a) *If* $\Gamma \vdash \lambda_{x:A}.B : C$ *then for some* D, s, $\Gamma \vdash \Pi_{x:A}.D : s$; $\Gamma, x{:}A \vdash B : D$; $\Pi_{x:A}.D =_\beta C$ *and if* $\Pi_{x:A}.D \not\equiv C$ *then* $\Gamma \vdash C : s'$ *for some sort* s'.
   (b) *If* $\Gamma \vdash \Pi_{x:A}.B : C$ *then there is* $(s_1, s_2) \in \boldsymbol{R}$ *such that* $\Gamma \vdash A : s_1$, $\Gamma, x{:}A \vdash B : s_2$, $C =_\beta s_2$ *and if* $C \not\equiv s_2$ *then* $\Gamma \vdash C : s$ *for some sort* s.
   (c) *If* $\Gamma \vdash Fa : C$ *then there are* A, B *such that* $\Gamma \vdash F : \Pi_{x:A}.B$, $\Gamma \vdash a : A$ *and* $C =_\beta B[x{:=}a]$ *and if* $C \not\equiv B[x{:=}a]$ *then* $\Gamma \vdash C : s$ *for some* s.
4. *If* $r = \flat$ *then*
   (a) *If* $\Gamma \vdash \flat_{x:A}.B : C$ *then* **only one** *of the following holds:*
      i *Either there are* s *and* D *where* $\Gamma \vdash \flat_{x:A}.D : s$; $\Gamma, x{:}A \vdash B : D$; $\flat_{x:A}.D =_\flat C$ *and if* $\flat_{x:A}.D \not\equiv C$ *then* $\Gamma \vdash C : s'$ *for some sort* s'.
      ii *Or there is* $(s_1, s_2) \in \boldsymbol{R}$ *such that* $\Gamma \vdash A : s_1$, $\Gamma, x{:}A \vdash B : s_2$, $C =_\flat s_2$ *and if* $C \not\equiv s_2$ *then* $\Gamma \vdash C : s$ *for some sort* s.
   (b) *If* $\Gamma \vdash Fa : C$ *then there are* A, B *such that* $\Gamma \vdash F : \flat_{x:A}.B$, $\Gamma \vdash a : A$ *and* $C =_\flat B[x{:=}a]$ *and if* $C \not\equiv B[x{:=}a]$ *then* $\Gamma \vdash C : s$ *for some* s.

Lemma 16 (Correctness of types for $\vdash$ and $\to_r$).
If $\Gamma \vdash A : B$ *then* ($B \equiv \Box$ *or* $\Gamma \vdash B : s$ *for some sort* s).

Lemma 17 (Subject Reduction for $\vdash$ and $\to_r$).
If $\Gamma \vdash A : B$ *and* $A \twoheadrightarrow_r A'$ *then* $\Gamma \vdash A' : B$.

Lemma 18 (Reduction preserves types for $\vdash$ and $\to_r$).
If $\Gamma \vdash A : B$ *and* $B \twoheadrightarrow_r B'$ *then* $\Gamma \vdash A : B'$.

Lemma 19 (Strong Normalisation for $\vdash$ and $\to_r$).
If A *is* $\vdash$*-legal then* $SN_{\to_r}(A)$.

Lemma 20 (Typability of subterms for $\vdash$ and $\to_r$).
If A *is* $\vdash$*-legal and* B *is a subterm of* A*, then* B *is* $\vdash$*-legal.*

Lemma 21 (Unicity of Types for $\vdash$ and $\to_r$).
1. *If* $\Gamma \vdash A : B_1$ *and* $\Gamma \vdash A : B_2$*, then* $B_1 =_r B_2$.
2. *If* $\Gamma \vdash A_1 : B_1$ *and* $\Gamma \vdash A_2 : B_2$ *and* $A_1 =_r A_2$*, then* $B_1 =_r B_2$.
3. *If* $\Gamma \vdash B_1 : s$, $B_1 =_r B_2$ *and* $\Gamma \vdash A : B_2$ *then* $\Gamma \vdash B_2 : s$.

3 The β-cube and typing modulo flattened binders

Definition 22. [The β-cube] The β-cube has terms $\mathcal{T}$ and the reduction relation $\rightarrow_\beta$. We use $\vdash_\beta$ to denote type derivation in the β-cube given by the rules of Figure 1. Sometimes, we annotate $\vdash_\beta$ with particular systems. For example, $\vdash_{\beta_C}$ is type derivation in β_C, the calculus of constructions of the β-cube.

All of Lemmas 11..21 hold for the β-cube (see (Barendregt, 1992)). Moreover, we have the next lemma, which enables us to freely interchange β and $\beta\Pi$ for $\vdash_\beta$-legal terms.

Lemma 23.

1. $\Gamma \not\vdash_\beta \Box : A$, $\Gamma \not\vdash_\beta AB : \Box$, $\Gamma \not\vdash_\beta \lambda_{x:A}.B : s$ *and* $\Gamma \not\vdash_\beta (\Pi_{x:A}.B)a : C$.
2. *If* $\Gamma \vdash_\beta A : B$ *then all of* Γ, A *and* B *are free of* Π*-redexes.*
3. *Let* A *be* $\vdash_\beta$*-legal and* $R \in \{\rightarrow, \twoheadrightarrow\}$. $AR_{\beta\Pi}A'$ *if and only if* $AR_\beta A'$.
4. *Let* A, A' *be* $\vdash_\beta$*-legal.* $A =_{\beta\Pi} A'$ *if and only if* $A =_\beta A'$.
5. *Let* A *be* $\vdash_\beta$*-legal. a)* A *is in* $\beta\Pi$*-normal form if and only if* A *is in* β*-normal form. b)* $\mathrm{nf}_{\beta\Pi}(A) \equiv \mathrm{nf}_\beta(A)$. *c)* $SN_{\rightarrow_{\beta\Pi}}(A)$ *if and only if* $SN_{\rightarrow_\beta}(A)$.
   d) If $\overline{A} \equiv \overline{A'}$ *and* A *is in* β*-normal form then* A' *is in* β*-normal form.*

Proof. 1. See (Barendregt, 1992).

2. First we show by induction on the derivation $\Gamma_1, x : D, \Gamma_2 \vdash_\beta E : F$ that if E and a are free of Π-redexes, $\Gamma_1, x : D, \Gamma_2 \vdash_\beta E : F$ and $\Gamma_1 \vdash_\beta a : D$, then $E[x := a]$ is free of Π-redexes. We only do the (appl) case. Take a and $E'b$ (hence E' and b) free of Π-redexes, $\Gamma_1 \vdash_\beta a : D$ and let $\Gamma_1, x : D, \Gamma_2 \vdash_\beta E'b : F'[y{:=}b]$ come from $\Gamma_1, x : D, \Gamma_2 \vdash_\beta E' : \Pi_{y:E''}.F'$ and $\Gamma_1, x : D, \Gamma_2 \vdash_\beta b : E''$.
   By IH, $E'[x := a]$ and $b[x := a]$ are free of Π-redexes.
   By Lemma 14, $\Gamma_1, \Gamma_2[x := a] \vdash_\beta E'[x := a]b[x := a] : F'[y{:=}b][x := a]$.
   By 1., $E'[x := a]b[x := a]$ is not a Π-redex.
   Hence, $(E'b)[x := a]$ is free of Π-redexes.
   Now, we show 2. by induction on $\Gamma \vdash_\beta A : B$. We only do the (appl) case. If $\Gamma \vdash_\beta Fa : B'[x{:=}a]$ comes from $\Gamma \vdash_\beta F : \Pi_{x:A'}.B'$ and $\Gamma \vdash_\beta a : A'$, by IH, Γ, F, a, A' and B' are free of Π-redexes. By 1., Fa is not a Π-redex. Hence, Γ and Fa are free of Π-redexes. Since $\Gamma \vdash_\beta a : A'$, $\Gamma, x : A' \vdash_\beta B' : s$ (by Lemmas 16 and 15 on $\Gamma \vdash_\beta F : \Pi_{x:A'}.B'$), and a, B' are free of Π-redexes, then by what we first proved above, $B'[x := a]$ is free of Π-redexes.
3. For $\rightarrow$, use 2. For $A \twoheadrightarrow_{\beta\Pi} A'$ implies $A \twoheadrightarrow_\beta A'$, use induction on the length of $A \twoheadrightarrow_{\beta\Pi} A'$ (by Lemmas 17 and 18, if $A \twoheadrightarrow_\beta C$ then C is $\vdash_\beta$-legal).
4. Use Church-Rosser and 3.
5. a) and c): Corollary of 3. d): Use a), Lemma 6.4, and Lemma 6.5.
   b): By Lemma 17 or 18 and a), $\mathrm{nf}_\beta(A)$ is $\vdash_\beta$-legal and in $\beta\Pi$-normal form. By Corollary 5, $\mathrm{nf}_\beta(A) \equiv \mathrm{nf}_{\beta\Pi}(A)$. ⊠

The normal forms of $\vdash_\beta$-legal terms follow an organised pattern:

Lemma 24. *Assume* $\Gamma \vdash_\beta A_1 : B_1$, $\Gamma \vdash_\beta A_2 : B_2$ *and* $\overline{A_1} \equiv \overline{A_2}$.

1. *If* A_1, A_2, B_1, B_2 *are in* β*-normal form then for some* $0 \leq n_1, n_2 \leq m$*:*

- $A_1 \equiv \lambda^{i:1..n_1}_{x_i:F_i}.\Pi^{i:n_1+1..m}_{x_i:F_i}.C$, $A_2 \equiv \lambda^{i:1..n_2}_{x_i:F_i}.\Pi^{i:n_2+1..m}_{x_i:F_i}.C$, *where* $C \equiv *$ *or* $C \equiv xL_1 \cdots L_k$ *for* $k \geq 0$,
- $B_1 \equiv \Pi^{i:1..n_1}_{x_i:F_i}.D$, $B_2 \equiv \Pi^{i:1..n_2}_{x_i:F_i}.D$, *where* $\Gamma, x_1 : F_1, \ldots, x_m : F_m \vdash_\beta C : D$.

2. $\mathrm{nf}_\beta(B_1) \equiv \Pi^{i:1..n_1}_{x_i:F_i}.D$ *and* $\mathrm{nf}_\beta(B_2) \equiv \Pi^{i:1..n_2}_{x_i:F_i}.D$ *where* $n_1, n_2 \geq 0$.
3. *If* $B_1 \equiv s_1$ *and* $B_2 \equiv s_2$ *then* $s_1 \equiv s_2$ *and* $\mathrm{nf}_\beta(A_1) \equiv \mathrm{nf}_\beta(A_2)$.

Proof. 1. By induction on the structure of A_1 in β-normal form.

- $A_1 \equiv \Box$ is not possible by Lemma 23.1.
- If A_1 is x or $*$ then take $n_1 = n_2 = m = 0$, $C \equiv A_1 \equiv A_2$ and $D \equiv B_1 \equiv B_2$ (by unicity of types $B_1 =_\beta B_2$ and as B_1, B_2 are in β-normal form, $B_1 \equiv B_2$).
- If for $1 \leq p \leq 2$, $A_p \equiv \Pi_{x_1:E_p}.G_p$ where $\overline{E_1} \equiv \overline{E_2}$ and $\overline{G_1} \equiv \overline{G_2}$, by generation, $\exists(s_p, s'_p)$ such that $\Gamma \vdash_\beta E_p : s_p$, $\Gamma, x_1 : E_p \vdash_\beta G_p : s'_p$ and $B_p \equiv s'_p$ (B_1, B_2 in β-normal form). By IH, $E_1 \equiv \Pi^{i:1..l}_{y_i:K_i}.R \equiv E_2$ (let $F_1 \equiv E_1 \equiv E_2$), $G_1 \equiv \Pi^{i:1..r}_{x_{i+1}:F_{i+1}}.H \equiv G_2$ and $s'_1 \equiv s'_2$ (let $D \equiv s'_1 \equiv s'_2$) where $H \equiv *$ or $H \equiv xL_1 \cdots L_k$ for $k \geq 0$, $\Gamma, x_1 : F_1, x_2 : F_2, \ldots, x_{r+1} : F_{r+1} \vdash_\beta H : D$ and $r \geq 0$. Let $m = r+1$. Then, $A_1 \equiv \Pi^{i:1..m}_{x_i:F_i}.H \equiv A_2$ and $B_1 \equiv B_2 \equiv D$.
- If for $1 \leq p \leq 2$, $A_p \equiv \lambda_{x_1:E_p}.G_p$ where $\overline{E_1} \equiv \overline{E_2}$ and $\overline{G_1} \equiv \overline{G_2}$, by generation, $\exists H_p, s_p$ where $\Gamma \vdash_\beta \Pi_{x_1:E_p}.H_p : s_p$, $\Gamma, x_1 : E_p \vdash_\beta G_p : H_p$ and $B_p \equiv \Pi_{x_1:E_p}.H_p$ (by Lemmas 19, 17 and 18, we take H_p in β-normal form). By generation, $\exists s'_p$ where $\Gamma \vdash_\beta E_p : s'_p$. By IH, $E_1 \equiv E_2$ (let $F_1 \equiv E_1 \equiv E_2$) and for $0 \leq n_p \leq m$:
  — $G_p \equiv \lambda^{i:1..n_p}_{x_{i+1}:F_{i+1}}.\Pi^{i:n_p+1..m}_{x_{i+1}:F_{i+1}}.C$ where C is $*$ or $xL_1 \cdots L_k$, for $k \geq 0$,
  — $H_p \equiv \Pi^{i:1..n_p}_{x_{i+1}:F_{i+1}}.D$, where $\Gamma, x_1 : F_1, x_2 : F_2, \ldots, x_{m+1} : F_{m+1} \vdash_\beta C : D$.
  Hence, $A_p \equiv \lambda_{x_1:F_1}.G_p \equiv \lambda^{i:1..n_p+1}_{x_i:F_i}.\Pi^{i:n_p+2..m+1}_{x_i:F_i}.C$ where $1 \leq n_p + 1 \leq m+1$.
- If $A_1 \equiv \lambda_{x_1:E_1}.G_1$, $A_2 \equiv \Pi_{x_1:E_2}.G_2$, $\overline{E_1} \equiv \overline{E_2}$ and $\overline{G_1} \equiv \overline{G_2}$, by generation:
  — $\exists H, s$ where $\Gamma \vdash_\beta \Pi_{x_1:E_1}.H : s$, $\Gamma, x_1 : E_1 \vdash_\beta G_1 : H$ and $B_1 \equiv \Pi_{x_1:E_1}.H$ (by Lemmas 19, 17 and 18, take H in β-normal form).
  — $\exists(s'_1, s'_2)$ where $\Gamma \vdash_\beta E_2 : s'_1$, $\Gamma, x_1 : E_2 \vdash_\beta G_2 : s'_2$ and $B_2 \equiv s'_2$.
  — $\exists s_1$ where $\Gamma \vdash_\beta E_1 : s_1$, and by IH, $E_1 \equiv E_2$. Let $F_1 \equiv E_1 \equiv E_2$.
  By IH, $G_1 \equiv \lambda^{i:1..n}_{x_{i+1}:F_{i+1}}.\Pi^{i:n+1..m}_{x_{i+1}:F_{i+1}}.C$, $G_2 \equiv \Pi^{i:1..m}_{x_{i+1}:F_{i+1}}.C$ (by Lemma 23.1, G_2 is not the form $\lambda_{x:E}.F$), $H \equiv \Pi^{i:1..n}_{x_{i+1}:F_{i+1}}.s'_2$ where $C \equiv *$ or $C \equiv xL_1 \cdots L_k$ for $k \geq 0$, and $\Gamma, x_1 : F_1, x_2 : F_2, \ldots, x_{m+1} : F_{m+1} \vdash_\beta C : s'_2$. Hence, $A_1 \equiv \lambda^{i:1..n+1}_{x_i:F_i}.\Pi^{i:n+2..m+1}_{x_i:F_i}.C$, $A_2 \equiv \Pi^{i:1..m+1}_{x_i:F_i}.C$, $B_1 \equiv \Pi^{i:1..n+1}_{x_i:F_i}.s'_2$ and $B_2 \equiv s'_2$.
- $A_1 \equiv \Pi_{x_1:E_1}.G_1$, $A_2 \equiv \lambda_{x_1:E_2}.G_2$ where $\overline{E_1} \equiv \overline{E_2}$ and $\overline{G_1} \equiv \overline{G_2}$ is similar.
- If for $1 \leq p \leq 2$, $A_p \equiv xL_{p1} \cdots L_{pk}$ where $k \geq 0$ and $\overline{L_{1i}} \equiv \overline{L_{2i}}$ for $1 \leq i \leq k$:
  — If $k = 0$, by generation $B_1 \equiv B_2$. Take $n_1 = n_2 = m = 0$, $C \equiv x$ and $D \equiv B_1 \equiv B_2$.
  — If $k \neq 0$, by generation $\Gamma \vdash_\beta xL_{p1} \cdots L_{p(k-1)} : \Pi_{y:C_p}.D_p$ and $\Gamma \vdash_\beta L_{pk} : C_p$. By IH, $xL_{11} \cdots L_{1(k-1)} \equiv xL_{21} \cdots L_{2(k-1)}$ and $\Pi_{y:C_1}.D_1 \equiv \Pi_{y:C_2}.D_2$. Hence, $C_1 \equiv C_2$ and $L_{1i} \equiv L_{2i}$ for $1 \leq i \leq k-1$. By IH on $\Gamma \vdash_\beta L_{pk} : C_p$, $L_{1k} \equiv L_{2k}$. Hence, $A_1 \equiv A_2 \equiv xL_1 \cdots L_k$ and by generation $B_1 \equiv B_2$.

2. By Lemma 19, $\mathrm{SN}_{\to_\beta}(A_p)$ and $\mathrm{SN}_{\to_\beta}(B_p)$ for $1 \leq p \leq 2$. By Lemmas 17 and 18, $\Gamma \vdash_\beta \mathrm{nf}_\beta(A_p) : \mathrm{nf}_\beta(B_p)$. By Lemmas 8.2 and 23.5, $\overline{\mathrm{nf}_\beta(A_1)} \equiv \overline{\mathrm{nf}_\beta(A_2)}$. By 1, $\mathrm{nf}_\beta(B_1) \equiv \Pi^{i:1..n_1}_{x_i:A_i}.D$ and $\mathrm{nf}_\beta(B_2) \equiv \Pi^{i:1..n_2}_{x_i:A_i}.D$ where $n_1, n_2 \geq 0$.

3. By Lemma 19, $\mathrm{SN}_{\to_\beta}(A_p)$ for $1 \le p \le 2$. By Lemma 17, $\Gamma \vdash_\beta \mathrm{nf}_\beta(A_p) : s_p$. By 1. above, $\mathrm{nf}_\beta(A_1) \equiv \mathrm{nf}_\beta(A_2)$ and $s_1 \equiv s_2$. ⊠

The next lemma relates legal contexts and terms of the same class: λs and Πs cannot be exchanged in legal contexts, nor in the types of a term, nor in the terms belonging to a type. This is basic for the isomorphism of both cubes.

Lemma 25. *Assume $\Gamma \vdash_\beta A_1 : B_1$ and $\Gamma \vdash_\beta A_2 : B_2$.*
1. *If $\overline{A_1} \equiv \overline{A_2}$ and $B_1 =_\beta B_2$ then $A_1 \equiv A_2$ and $B_1 \equiv B_2$.*
2. *If $B_1 \equiv s_1$, $B_2 \equiv s_2$ and $\overline{A_1} \equiv \overline{A_2}$ then $A_1 \equiv A_2$ and $s_1 \equiv s_2$.*
3. *If Γ_1 and Γ_2 are $\vdash_\beta$-legal and if $\overline{\Gamma_1} \equiv \overline{\Gamma_2}$ then $\Gamma_1 \equiv \Gamma_2$.*
4. *If $\overline{B_1} \equiv \overline{B_2}$ then $B_1 \equiv B_2$.*
5. *If $\overline{A_1} \equiv \overline{A_2}$ and $\overline{B_1} \equiv \overline{B_2}$ then $A_1 \equiv A_2$ and $B_1 \equiv B_2$.*
6. *If $B_1 \equiv s_1$, $B_2 \equiv s_2$, $\overline{A_1} =_\flat \overline{A_2}$ then $A_1 =_\beta A_2$.*

Proof.

1. By Lemma 24.1 for $1 \le p \le 2$, $A_p \equiv \lambda^{i:1..n_p}_{x_i:F_i}.\Pi^{i:n_p+1..m}_{x_i:F_i}.C$ and $B_p \equiv \Pi^{i:1..n_p}_{x_i:F_i}.D$. As B_p is $\vdash_\beta$-legal and $B_1 =_\beta B_2$, by Lemmas 19 and 23.(5 and 4), $\mathrm{SN}_{\to_{\beta\Pi}}(B_p)$ and $B_1 =_{\beta\Pi} B_2$. By Lemma 8.1, $n_1 = n_2$. So, $A_1 \equiv A_2$ and $B_1 \equiv B_2$.
2. By Lemma 24.3, $s_1 \equiv s_2$. By 1 above, $A_1 \equiv A_2$.
3. By induction on the length of Γ_1 using start/context Lemma 12 and 2 above.
4. If $B_1 \equiv \Box$ then $B_2 \equiv \Box$ and $B_1 \equiv B_2$. If $B_1 \not\equiv \Box$ then $B_2 \not\equiv \Box$ and by correctness of types, $\Gamma \vdash_\beta B_1 : s_1$ and $\Gamma \vdash_\beta B_2 : s_2$. Hence, by 2, $B_1 \equiv B_2$.
5. By 4 above, $B_1 \equiv B_2$. Hence, by 1 above, $A_1 \equiv A_2$.
6. By Church-Rosser $\overline{A_1} \twoheadrightarrow_\flat C \;{}_\flat\!\twoheadleftarrow \overline{A_2}$. By Lemma 6.2, $\forall i$, $1 \le i \le 2$ then $\exists D_i$ where $\overline{D_i} \equiv C$ and $A_i \twoheadrightarrow_{\beta\Pi} D_i$. Since A_i is $\vdash_\beta$-legal by Lemma 23.3 $A_i \twoheadrightarrow_\beta D_i$. By Lemma 17, $\Gamma \vdash D_i : s_i$. By 2 above, $D_1 \equiv D_2$. So, $A_1 =_\beta A_2$. ⊠

4 The $\flat$-cube: Identifying λ and Π in the cube

Definition 26. [The $\flat$-cube] The $\flat$-cube has $\mathcal{T}_\flat$ as the set of terms and $\flat$-reduction $\to_\flat$ for the reduction relation. We use $\vdash_\flat$ to denote type derivation in the $\flat$-cube given by the rules of Figure 3. If needed, we annotate $\vdash_\flat$ with particular systems. E.g., $\vdash_{\flat_C}$ is type derivation in $\flat_C$, the calculus of constructions of the $\flat$-cube.

Lemmas 11..14 hold for the $\flat$-cube and have the same proofs as in the β-cube. Next we prove Lemma 15 for the $\flat$-cube. Note the unicity clause in 3 which allows one to easily unpack the status of a $\flat$ as a λ or a Π:

Proof. **(Of Generation Lemma 15 for the $\flat$-cube)**
1. By induction on the derivation $\Gamma \vdash_\flat E : C$ where $E \equiv s$.
2. By induction on the derivation $\Gamma \vdash_\flat E : C$ where $E \equiv x$.

4(*a*). By induction on the derivation $\Gamma \vdash_\flat E : C$ where $E \equiv \flat_{x:A}.B$. We only do the interesting cases. Assume $\Gamma \vdash_\flat \flat_{x:A}.B : C$ comes from:
- ($\flat_1$): then *ii* holds. Moreover, *i* is impossible in this case since otherwise, there is D such that $\flat_{x:A}.D =_\flat s_2$ which is impossible by Church-Rosser.

(axiom)	$\langle\rangle \vdash * : \Box$
(start)	$\dfrac{\Gamma \vdash A : s \qquad x^s \notin \mathrm{DOM}\,(\Gamma)}{\Gamma, x^s{:}A \vdash x^s : A}$
(weak)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s \qquad x^s \notin \mathrm{DOM}\,(\Gamma)}{\Gamma, x^s{:}C \vdash A : B}$
($\flat_1$)	$\dfrac{\Gamma \vdash A : s_1 \qquad \Gamma, x{:}A \vdash B : s_2 \qquad (s_1, s_2) \in \boldsymbol{R}}{\Gamma \vdash \flat_{x:A}.B : s_2}$
($\flat_2$)	$\dfrac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash \flat_{x:A}.B : s}{\Gamma \vdash \flat_{x:A}.b : \flat_{x:A}.B}$
(conv$_\flat$)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_\flat B'}{\Gamma \vdash A : B'}$
(appl$\flat$)	$\dfrac{\Gamma \vdash F : \flat x{:}A.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[x{:=}a]}$

Fig. 3. Typing rules with one binder

— ($\flat_2$): then i holds where $C \equiv \flat_{x:A}.D$. If also ii holds then there is (s_1, s_2) such that $\flat_{x:A}.D =_\flat s_2$ which is impossible by Church-Rosser.

4(b). By induction on the derivation $\Gamma \vdash_\flat E : C$ where $E \equiv Fa$. ⊠

Also, Lemmas 16..18 and 20 hold for the $\flat$-cube and have the same proofs as those for the β-cube. Before showing strong normalisation Lemma 19 and before discussing unicity of types Lemma 21, we will establish the isomorphism of the $\flat$-cube and the β-cube. First, we write the rules of Figure 3, in a syntax-directed fashion as in Figure 4, which gives a type checking algorithm for the $\flat$-cube. We use $\vdash_{\flat\mathrm{tc}}$ to denote type derivation using the rules of Figure 4. Note that rules (tc4) and (tc5) do not overlap since by Church-Rosser we cannot have both $C =_\flat s_2$ and $C =_\flat \flat_{x:A}.D$. Below, we show that $\vdash_\flat$ and $\vdash_{\flat\mathrm{tc}}$ are equivalent.

Lemma 27. $\Gamma \vdash_\flat A : B$ *if and only if* $\Gamma \vdash_{\flat tc} A : B$.

Proof. "if": by induction on $\Gamma \vdash_{\flat\mathrm{tc}} A : B$ using Lemma 12.
"only if": by induction on $\Gamma \vdash_\flat A : B$ using 1 and 2 below which we show by induction on $\Gamma \vdash_{\flat\mathrm{tc}} A : B$.
1. If $\Gamma \vdash_{\flat\mathrm{tc}} A : B$, $\Gamma \subseteq \Gamma'$ and $\Gamma' \vdash_{\flat\mathrm{tc}} * : \Box$ then $\Gamma' \vdash_{\flat\mathrm{tc}} A : B$.
2. If $\Gamma \vdash_{\flat\mathrm{tc}} A : B$, $\Gamma \vdash_{\flat\mathrm{tc}} B' : s$ and $B =_\flat B'$ then $\Gamma \vdash_{\flat\mathrm{tc}} A : B'$. ⊠

Hence, we use $\vdash_\flat$ for both $\vdash_\flat$ and $\vdash_{\flat\mathrm{tc}}$. Next, we give an algorithm to construct for each $\Gamma \vdash_\flat A : B$, a triple $(\Gamma', A', B') \in [\Gamma] \times [A] \times [B]$ such that $\Gamma' \vdash_\beta A' : B'$.

(tc1) $$\langle\rangle \vdash * : \Box$$

(tc2) $$\frac{\Gamma \vdash A : s \qquad x^s \notin \textsc{dom}\,(\Gamma)}{\Gamma, x^s : A \vdash * : \Box}$$

(tc3) $$\frac{\Gamma \vdash C : s \qquad x^s : A \in \Gamma \qquad A =_\flat C}{\Gamma \vdash x^s : C}$$

(tc4) $$\frac{C =_\flat s_2 \quad \Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2 \quad (s_1, s_2) \in \boldsymbol{R} \quad C \not\equiv s_2 \Rightarrow \Gamma \vdash C : s}{\Gamma \vdash \flat_{x:A}.B : C}$$

(tc5) $$\frac{C =_\flat \flat_{x:A}.D \quad \Gamma \vdash \flat_{x:A}.D : s \quad \Gamma, x : A \vdash B : D \quad C \not\equiv \flat_{x:A}.D \Rightarrow \Gamma \vdash C : s'}{\Gamma \vdash \flat_{x:A}.B : C}$$

(tc6) $$\frac{\Gamma \vdash F : \flat_{x:A}.B \quad \Gamma \vdash a : A \quad C =_\flat B[x := a] \quad C \not\equiv B[x := a] \Rightarrow \Gamma \vdash C : s}{\Gamma \vdash Fa : C}$$

Fig. 4. Type checking in the syntax-directed version of the rules of the ♭-cube

Definition 28. Let $\Gamma \vdash_\flat A : B$. We define $(\Gamma \vdash_\flat A : B)^{-1} \in [\Gamma] \times [A] \times [B]$ by:

$$\begin{array}{ll}
(\langle\rangle \vdash_\flat * : \Box)^{-1} & = (\langle\rangle, *, \Box) \\
(\Gamma, x^s : A \vdash_\flat * : \Box)^{-1} & = (\Gamma', x^s : A', *, \Box) \text{ where } (\Gamma \vdash_\flat A : s)^{-1} = (\Gamma', A', s) \\
(\Gamma \vdash_\flat x^s : C)^{-1} & = (\Gamma', x^s, C') \text{ where } (\Gamma \vdash_\flat C : s)^{-1} = (\Gamma', C', s) \\
(\Gamma \vdash_\flat \flat_{x:A}.B : C)^{-1} & = \begin{cases} (\Gamma', \Pi_{x:A'}.B', C') & \text{if } C =_\flat s_2 \text{ and i.} \\ (\Gamma', \lambda_{x:A'}.B', C') & \text{if } C =_\flat \flat_{x:A}.D \text{ and ii.} \end{cases} \\
(\Gamma \vdash_\flat Fa : C)^{-1} & = (\Gamma', F'a', C') \text{ where iii.}
\end{array}$$

Where i., ii., and iii. are as follows:

i. all the following hold:
- $(\Gamma \vdash_\flat A : s_1)^{-1} = (\Gamma', A', s_1)$ for some s_1 where $(s_1, s_2) \in \boldsymbol{R}$,
- $(\Gamma, x : A \vdash_\flat B : s_2)^{-1} = (\Gamma'', x : A'', B', s_2)$ and
- if $C \equiv s_2$ then $C' \equiv s_2$ else
  if $C \not\equiv s_2$ then $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma''', C', s)$ for some s.

ii. all the following hold:
- $(\Gamma \vdash_\flat \flat_{x:A}.D : s)^{-1} = (\Gamma', \pi_{x:A'}.D', s)$ for some s,
- $(\Gamma, x : A \vdash_\flat B : D)^{-1} = (\Gamma'', x : A'', B', D'')$ and
- if $C \equiv \flat_{x:A}.D$ then $C' \equiv \Pi_{x:A'}.D'$, else
  if $C \not\equiv \flat_{x:A}.D$ then $(\Gamma \vdash_\flat C : s')^{-1} = (\Gamma''', C', s')$ for some s'.

iii. for some A, B where $C =_\flat B[x := a]$, all the following hold:
- $(\Gamma \vdash_\flat F : \flat_{x:A}.B)^{-1} = (\Gamma', F', \pi_{x:A'}.B')$,
- $(\Gamma \vdash_\flat a : A)^{-1} = (\Gamma'', a', A'')$ and
- if $C \equiv B[x := a]$ then $C' \equiv B'[x := a']$, else
  if $C \not\equiv B[x := a]$ then $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma''', C', s)$ for some s.

Lemma 29. *The following hold:*
1. *If* $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$ *then* $(\Gamma', A', B') \in [\Gamma] \times [A] \times [B]$.
2. *If* $\Gamma \vdash_\flat A : B$ *then there is* (Γ', A', B') *such that* $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$.
3. *If* $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$ *then* $\Gamma' \vdash_\beta A' : B'$.
4. *If* $\Gamma \vdash_\flat A : B$ *then* $(\Gamma \vdash_\flat A : B)^{-1}$ *is unique.*

Proof.

1. By induction on the derivation of $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$ according to Definition 28 (use Lemma 6.1 in the last clause).
2. By induction on the derivation $\Gamma \vdash_\flat A : B$ using the rules of Figure 4 (use 1., in (tc5) and (tc6)).
3. By induction on the derivation of $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$ according to Definition 28.
   - Case $(\langle\rangle \vdash_\flat * : \Box)^{-1} = (\langle\rangle, *, \Box)$, trivial.
   - Let $(\Gamma, x^s : A \vdash_\flat * : \Box)^{-1} = (\Gamma', x^s : A', *, \Box)$ where $(\Gamma \vdash_\flat A : s)^{-1} = (\Gamma', A', s)$. By IH, $\Gamma' \vdash_\beta A' : s$ and by Lemma 12, $\Gamma' \vdash_\beta * : \Box$. Since $\Gamma, x^s : A \vdash_\flat * : \Box$, then $x^s \notin \text{DOM}(\Gamma)$. By 1., $\Gamma' \in [\Gamma]$ and so $x^s \notin \text{DOM}(\Gamma')$. Hence, by (weak) $\Gamma', x^s : A' \vdash_\beta * : \Box$.
   - Let $(\Gamma \vdash_\flat x^s : C)^{-1} = (\Gamma', x^s, C')$ where $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma', C', s)$. By IH, $\Gamma' \vdash_\beta C' : s$. Since $\Gamma \vdash_\flat x^s : C$, by generation, there is $x^s : A \in \Gamma$ where $A =_\flat C$. Let $x^s : A' \in \Gamma'$ where $A' \in [A]$. By 1., $\Gamma' \in [\Gamma]$ and $C' \in [C]$. As Γ' is $\vdash_\beta$-legal, by Lemmas 12 and 15, $\Gamma' \vdash_\beta x^s : A'$ and $\Gamma' \vdash_\beta A' : s$. By Lemma 25.6 $A' =_\beta C'$. Hence, by (conv$_\beta$) $\Gamma' \vdash_\beta x^s : C'$.
   - Let $(\Gamma \vdash_\flat \flat_{x:A}.B : C)^{-1} = (\Gamma', \Pi_{x:A'}.B', C')$ where $C =_\flat s_2$ and:

     — $(\Gamma \vdash_\flat A : s_1)^{-1} = (\Gamma', A', s_1)$ for some s_1 where $(s_1, s_2) \in \boldsymbol{R}$.

     — $(\Gamma, x : A \vdash_\flat B : s_2)^{-1} = (\Gamma'', x : A'', B', s_2)$.

     — If $C \equiv s_2$ then $C' \equiv s_2$ else
     if $C \not\equiv s_2$ then $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma''', C', s)$ for some s.

     By IH, $\Gamma' \vdash_\beta A' : s_1$ and $\Gamma'', x : A'' \vdash_\beta B' : s_2$. By 1., $\Gamma', \Gamma'' \in [\Gamma]$, $A', A'' \in [A]$ and $B' \in [B]$. By Lemma 12 $\Gamma'' \vdash_\beta A'' : s''$. By Lemma 25.3 $\Gamma' \equiv \Gamma''$. By Lemma 25.2, $A' \equiv A''$. By (Π), $\Gamma' \vdash_\beta \Pi_{x:A'}.B' : s_2$. If $C \equiv s_2$ then $C' \equiv s_2$ and $\Gamma' \vdash_\beta \Pi_{x:A'}.B' : C'$. If $C \not\equiv s_2$ then since $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma''', C', s)$, by IH $\Gamma''' \vdash_\beta C' : s$ and by 1., $\Gamma''' \in [\Gamma]$ and $C' \in [C]$. By Lemma 25.3 $\Gamma' \equiv \Gamma'''$. As $C =_\flat s_2$, $C \twoheadrightarrow_\flat s_2$. By Lemmas 6.2 and 23.3 $C' \twoheadrightarrow_\beta s_2$. Since $\Gamma' \vdash_\beta C' : s$, $\Gamma' \vdash_\beta \Pi_{x:A'}.B' : s_2$ and $C' =_\beta s_2$, by (conv$_\beta$), $\Gamma' \vdash_\beta \Pi_{x:A'}.B' : C'$.
   - Let $(\Gamma \vdash_\flat \flat_{x:A}.B : C)^{-1} = (\Gamma', \lambda_{x:A'}.B', C')$ where $C =_\flat \flat_{x:A}.D$ and:

     — $(\Gamma \vdash_\flat \flat_{x:A}.D : s)^{-1} = (\Gamma', \pi_{x:A'}.D', s)$ for some s.

     — $(\Gamma, x : A \vdash_\flat B : D)^{-1} = (\Gamma'', x : A'', B', D'')$.

     — if $C \equiv \flat_{x:A}.D$ then $C' \equiv \Pi_{x:A'}.D'$, else
     if $C \not\equiv \flat_{x:A}.D$ then $(\Gamma \vdash_\flat C : s')^{-1} = (\Gamma''', C', s')$ for some s'.

     By IH, $\Gamma' \vdash_\beta \pi_{x:A'}.D' : s$ and $\Gamma'', x : A'' \vdash_\beta B' : D''$. By 1., $\Gamma', \Gamma'' \in [\Gamma]$, $A', A'' \in [A]$, $B' \in [B]$ and $D', D'' \in [D]$. By Lemma 15 $\Gamma' \vdash_\beta A' : s''$ and $\Gamma'' \vdash_\beta A'' : s'''$. By Lemma 25.(3, 2) $\Gamma' \equiv \Gamma''$ and $A' \equiv A''$. By Lemma 23.1 $\Gamma' \not\vdash_\beta \lambda_{x:A'}.D' : s$ so $\Gamma' \vdash_\beta \Pi_{x:A'}.D' : s$. By Lemma 15 $\Gamma', x : A' \vdash_\beta D' : s$. Since $D'' \not\equiv \Box$ (else $D' \equiv \Box$ and $\Gamma', x : A' \vdash_\beta \Box : s$ absurd by Lemma 23.1), by Lemma 16 $\Gamma', x : A' \vdash_\beta D'' : s_1$. By Lemma 25.2 $D' \equiv D''$. By (λ) $\Gamma' \vdash_\beta \lambda_{x:A'}.B' : \Pi_{x:A'}.D'$. If $C \equiv \flat_{x:A}.D$ then $C' \equiv \Pi_{x:A'}.D'$ and $\Gamma' \vdash_\beta \lambda_{x:A'}.B' : C'$. If $C \not\equiv \flat_{x:A}.D$, by IH, $\Gamma''' \vdash_\beta C' : s'$. By 1., $\Gamma''' \in [\Gamma]$ and $C' \in [C]$. By Lemma 25.3 $\Gamma' \equiv \Gamma'''$. By Lemma 25.6 $C' =_\beta \Pi_{x:A'}.D'$. Hence, by (conv$_\beta$), $\Gamma' \vdash_\beta \lambda_{x:A'}.B' : C'$.

- Let $(\Gamma \vdash_\flat Fa : C)^{-1} = (\Gamma', F'a', C')$ where for some A and B such that $C = B[x := a]$, all the following hold:
  — $(\Gamma \vdash_\flat F : \flat_{x:A}.B)^{-1} = (\Gamma', F', \pi_{x:A'}.B')$,
  — $(\Gamma \vdash_\flat a : A)^{-1} = (\Gamma'', a', A'')$ and
  — if $C \equiv B[x := a]$ then $C' \equiv B'[x := a']$, else
  if $C \not\equiv B[x := a]$ then $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma''', C', s)$ for some s.

  By IH, $\Gamma' \vdash_\beta F' : \pi_{x:A'}.B'$ and $\Gamma'' \vdash_\beta a' : A''$. By 1., $\Gamma', \Gamma'' \in [\Gamma]$, $A', A'' \in [A]$, $B' \in [B]$, $F' \in [F]$ and $a' \in [a]$. By Lemma 25.3 $\Gamma' \equiv \Gamma''$. By Lemma 16 $\Gamma' \vdash_\beta \pi_{x:A'}.B' : s'$ and by Lemma 23.1 $\pi = \Pi$. By Lemma 15 $\Gamma' \vdash_\beta A' : s''$. Moreover, $A'' \not\equiv \Box$ (else $A' \equiv \Box$ and $\Gamma' \vdash_\beta \Box : s''$ absurd by Lemma 23.1). By Lemma 16 $\Gamma' \vdash_\beta A'' : s'''$. By Lemma 25.2, $A' \equiv A''$. By (appl) $\Gamma' \vdash_\beta F'a' : B'[x := a']$. If $C \equiv B[x := a]$ then $C' \equiv B'[x := a']$ and $\Gamma' \vdash_\beta F'a' : C'$. If $C \not\equiv B[x := a]$ then by IH $\Gamma''' \vdash_\beta C' : s$ and by 1., $\Gamma''' \in [\Gamma]$ and $C' \in [C]$. By Lemma 25.3 $\Gamma' \equiv \Gamma'''$. By Lemmas 15 and 14 $\Gamma' \vdash_\beta B'[x := a'] : s'$. Recall that $C =_\flat B[x := a]$ and by Lemma 6.1 $\overline{B'[x := a']} \equiv B[x := a]$. Hence by Lemma 25.6 $C' =_\beta B'[x := a']$. Finally by (conv_β), $\Gamma' \vdash_\beta F'a' : C'$.

4. Let $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma_1, A_1, B_1)$ and $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma_2, A_2, B_2)$. By 1., $(\Gamma_1, A_1, B_1), (\Gamma_2, A_2, B_2) \in [\Gamma] \times [A] \times [B]$. By 3., $\Gamma_1 \vdash_\beta A_1 : B_1$ and $\Gamma_2 \vdash_\beta A_2 : B_2$. By Lemma 25.3 $\Gamma_1 \equiv \Gamma_2$ and by Lemma 25.5 $A_1 \equiv A_2$ and $B_1 \equiv B_2$. ⊠

The next theorem shows the isomorphism between the $\flat$-cube and the β-cube. It also says that given a typing judgement in terms of $\flat$, this judgement can be uniquely written in terms of λ and Π. This means that the semantic meaning of all the subterms of Γ, A and B of the judgement $\Gamma \vdash_\flat A : B$ is preserved.

Theorem 30.

1. *If* $\Gamma \vdash_\beta A : B$ *then* $\overline{\Gamma} \vdash_\flat \overline{A} : \overline{B}$.
2. *If* $\Gamma \vdash_\flat A : B$ *then*
   - *there exists a unique* $\Gamma' \in [\Gamma]$ *such that* Γ' *is* $\vdash_\beta$*-legal, and*
   - *there exist unique* $A' \in [A]$*, unique* $B' \in [B]$ *such that* $\Gamma' \vdash_\beta A' : B'$.

   Moreover, Γ', A' *and* B' *are constructed by* $^{-1}$ *where* $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$.

Proof. 1. By induction on the derivation $\Gamma \vdash_\beta A : B$.
2. By Lemma 29, let $(\Gamma', A', B') \in [\Gamma] \times [A] \times [B]$ such that $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$. Again by Lemma 29, $\Gamma' \vdash_\beta A' : B'$. By Lemma 25.3, Γ' is the unique $\vdash_\beta$-legal element of $[\Gamma]$. By Lemma 25.5, (A', B') is unique in $[A] \times [B]$ such that $\Gamma' \vdash_\beta A' : B'$. ⊠

Strong normalisation for the $\flat$-cube can be established directly as for the β-cube, or indirectly via Theorem 30. Below, we give the indirect proof.

Lemma 31 (Strong Normalisation for $\vdash_\flat$ and $\to_\flat$).
If A *is* $\vdash_\flat$*-legal then* $SN_{\to_\flat}(A)$.

Proof. Since A is legal, $\Gamma \vdash_\flat A : B$ or $\Gamma \vdash_\flat B : A$. If $\Gamma \vdash_\flat B : A$, by Lemma 16, $A \equiv \Box$ (and $\mathrm{SN}_{\to_\flat}(A)$) or $\Gamma \vdash_\flat A : s$. Hence, we only do the proof for $\Gamma \vdash_\flat A : B$. By Theorem 30, $\Gamma' \vdash_\beta A' : B'$ for $\overline{\Gamma'} \equiv \Gamma$, $\overline{A'} \equiv A$ and $\overline{B'} \equiv B$. By Lemma 19, $\mathrm{SN}_{\to_\beta}(A')$. By Lemma 23.5, $\mathrm{SN}_{\to_{\beta\Pi}}(A')$. By Lemma 6.4, $\mathrm{SN}_{\to_\flat}(A)$. ⊠

Hence, all the properties of the β-cube (except for unicity of types), hold indeed for the $\flat$-cube. What about unicity of types? This does not hold since $\flat_{x:A}.B$ represents both $\lambda_{x:A'}.B'$ and $\Pi_{x:A'}.B'$ which may both be typable. In other words, $\flat_{x:A}.B$ can have two types C and D where $C \neq_\flat D$. Here is an example:

Example 32.

1. *Using* $(\Box,\Box)$*:* $\vdash_\beta \lambda_{x:*}.x : \Pi_{x:*}.*$ *and* $\vdash_\flat \flat_{x:*}.x : \flat_{x:*}.*$.
   Using $(\Box,*)$*:* $\vdash_\beta \Pi_{x:*}.x : *$ *and* $\vdash_\flat \flat_{x:*}.x : *$.
   Note: $\flat_{x:*}.* \neq_\flat *$.
2. *Using* $(\Box,*)$ *and* $(\Box,\Box)$*:* $\vdash_\beta \lambda_{x:*}.\Pi_{y:*}.y : \Pi_{x:*}.*$ *and* $\vdash_\flat \flat_{x:*}.\flat_{y:*}.y : \flat_{x:*}.*$.
   Using $(\Box,*)$*:* $\vdash_\beta \Pi_{x:*}.\Pi_{y:*}.y : *$ *and* $\vdash_\flat \flat_{x:*}.\flat_{y:*}.y : *$.
   Using $(\Box,\Box)$*:* $\vdash_\beta \lambda_{x:*}.\lambda_{y:*}.y : \Pi_{x:*}.\Pi_{y:*}.*$ *and* $\vdash_\flat \flat_{x:*}.\flat_{y:*}.y : \flat_{x:*}.\flat_{y:*}.*$.
   Note: $\Pi_{x:*}.\lambda_{y:*}.y$ *is not typable and* $\forall A \not\equiv B \in \{\flat_{x:*}.*, *, \flat_{x:*}.\flat_{y:*}.*\}$, $A \neq_\flat B$.
3. *If* $A \equiv \flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.\flat_{x_3:C}.\flat_{x_4:x_2x_3}.x_2x_3$ *then* $\vdash_\flat A : B$ *for any* B *in the set* $\{\flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.\flat_{x_3:C}.\flat_{x_4:x_2x_3}.*, \flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.\flat_{x_3:C}.*, \flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.*, \flat_{x_1:*}.*, *\}$. *(Note that you need the necessary* $(s_1,s_2) \in \boldsymbol{R}$*.) In the* β*-cube, the only possible judgements* $\vdash_\beta A' : B'$ *where* $A' \in [A]$ *are as follows:*

$$
\begin{array}{llll}
\vdash_\beta \lambda_{x_1:*}.\lambda_{x_2:\Pi_{y:C}.*}.\lambda_{x_3:C}.\lambda_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}. & * \\
\vdash_\beta \lambda_{x_1:*}.\lambda_{x_2:\Pi_{y:C}.*}.\lambda_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}. & * \\
\vdash_\beta \lambda_{x_1:*}.\lambda_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}. & * \\
\vdash_\beta \lambda_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}. & * \\
\vdash_\beta \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & & *
\end{array}
$$

As can be seen, we can relate the types of the same $\flat$-term. First, a definition:

Definition 33. Let $\Lambda \in \{\Pi, \flat\}$.
- We say $A_1 \overset{\diamond}{\because}_\Lambda A_2$ iff $A_1 \equiv \Lambda^{i:1..n_1}_{x_i:F_i}.B$ and $A_2 \equiv \Lambda^{i:1..n_2}_{x_i:F_i}.B$, where $n_1, n_2 \geq 0$.
- Note that if $A_1 \overset{\diamond}{\because}_\Pi A_2$ then $\overline{A_1} \overset{\diamond}{\because}_\flat \overline{A_2}$
- Let $\mathrm{SN}_{\to_\flat}(A_1)$ and $\mathrm{SN}_{\to_\flat}(A_2)$. We say $A_1 \overset{\diamond}{=}_\flat A_2$ iff $\mathrm{nf}_\flat(A_1) \overset{\diamond}{\because}_\flat \mathrm{nf}_\flat(A_2)$.

Now, look at the types of the $\flat$-terms in Example 32. Note that the types of $\flat_{x:*}.x$ are related by $\overset{\diamond}{\because}_\flat$. That is, $\flat_{x:*}.* \overset{\diamond}{\because}_\flat *$. Similarly, all the types of $\flat_{x:*}.\flat_{y:*}.y$ are related by $\overset{\diamond}{\because}_\flat$. In fact, for all $A, B \in \{\flat_{x:*}.*, *, \flat_{x:*}.\flat_{y:*}.*\}$, we have $A \overset{\diamond}{\because}_\flat B$. So, it seems that $\overset{\diamond}{\because}_\flat$ will be the relation satisfied by all the types of the same $\flat$-term. We must however take this relation modulo $=_\flat$ as is usual in the cube, due to the conversion rule ($\mathrm{conv}_\flat$). First, we need the next lemma:

Lemma 34. *1. If* $\Gamma \vdash_\flat A : B$ *then* $\Box$ *does not occur in* A. *2.* $\Gamma \not\vdash_\flat Fa : \Box$.

Proof. 1. By induction on the derivation $\Gamma \vdash_\flat A : B$. 2. Assume $\Gamma \vdash_\flat Fa : \Box$. By Lemma 15, $\Gamma \vdash_\flat F : \flat_{x:A}.B$, $\Gamma \vdash_\flat a : A$ and $\Box =_\flat B[x := a]$. Hence, $B[x := a] \twoheadrightarrow_\flat \Box$. By Lemmas 16 and 15, $\Gamma \vdash_\flat \flat_{x:A}.B : s$ and $\Gamma, x : A \vdash_\flat B : s'$. By Lemmas 14 and 17, $\Gamma \vdash_\flat B[x := a] : s'$ and $\Gamma \vdash_\flat \Box : s'$ contradicting 1. ⊠

Now, Lemma 21 becomes:

Lemma 35 (Organised multiplicity of Types for $\vdash_\flat$ and $\rightarrow_\flat$).
1. *If* $\Gamma \vdash_\flat A : B_1$ *and* $\Gamma \vdash_\flat A : B_2$*, then* $B_1 \overset{\diamond}{=}_\flat B_2$.
2. *If* $\Gamma \vdash_\flat A_1 : B_1$ *and* $\Gamma \vdash_\flat A_2 : B_2$ *and* $A_1 =_\flat A_2$*, then* $B_1 \overset{\diamond}{=}_\flat B_2$.
3. *If* $\Gamma \vdash_\flat B_1 : s_1$, $B_1 =_\flat B_2$ *and* $\Gamma \vdash_\flat A : B_2$ *then* $\Gamma \vdash_\flat B_2 : s_1$.
4. *Assume* $\Gamma \vdash_\flat A : B_1$ *and* $(\Gamma \vdash_\flat A : B_1)^{-1} = (\Gamma', A', B'_1)$*. Then* $B_1 =_\flat B_2$ *if:*
   (a) *either* $\Gamma \vdash_\flat A : B_2$, $(\Gamma \vdash_\flat A : B_2)^{-1} = (\Gamma', A'', B'_2)$ *and* $B'_1 =_\beta B'_2$,
   (b) *or* $\Gamma \vdash_\flat C : B_2$, $(\Gamma \vdash_\flat C : B_2)^{-1} = (\Gamma', C', B'_2)$ *and* $A' =_\beta C'$.

Proof.

1. By Theorem 30, there are unique $\Gamma' \in [\Gamma]$, $A_1, A_2 \in [A]$, $B'_1 \in [B_1]$ and $B'_2 \in [B_2]$ such that $\Gamma' \vdash_\beta A_1 : B'_1$ and $\Gamma' \vdash_\beta A_2 : B'_2$ (by Lemma 25.3, we take the same Γ' in both judgements). By Lemma 24.2, $\mathrm{nf}_\beta(B'_1) \overset{\diamond}{\because}_\Pi \mathrm{nf}_\beta(B'_2)$. Hence, $\overline{\mathrm{nf}_\beta(B'_1)} \overset{\diamond}{\because}_\flat \overline{\mathrm{nf}_\beta(B'_2)}$. Since for $1 \leq i \leq 2$, B'_i is $\vdash_\beta$-legal, by Lemma 19, $\mathrm{SN}_{\rightarrow_\beta}(B'_i)$ and by Lemma 23.5, $\mathrm{SN}_{\rightarrow_{\beta\Pi}}(B'_i)$ and $\mathrm{nf}_\beta(B'_i) \equiv \mathrm{nf}_{\beta\Pi}(B'_i)$. Hence, $\overline{\mathrm{nf}_{\beta\Pi}(B'_1)} \overset{\diamond}{\because}_\flat \overline{\mathrm{nf}_{\beta\Pi}(B'_2)}$ and by Lemma 8.2, $\mathrm{nf}_\flat(\overline{B'_1}) \overset{\diamond}{\because}_\flat \mathrm{nf}_\flat(\overline{B'_2})$. So, $B_1 \overset{\diamond}{=}_\flat B_2$.
2. By Church-Rosser, there is an A_3 such that $A_1 \twoheadrightarrow_\flat A_3 \, {}_\flat\!\twoheadleftarrow A_2$. By subject reduction, $\Gamma \vdash_\flat A_3 : B_1$ and $\Gamma \vdash_\flat A_3 : B_2$. Hence by 1, $B_1 \overset{\diamond}{=}_\flat B_2$.
3. By (conv$\flat$), $\Gamma \vdash_\flat A : B_1$. By 1, $\mathrm{nf}_\flat(B_1) \overset{\diamond}{\because}_\flat \mathrm{nf}_\flat(B_2)$. For $1 \leq p \leq 2$, let $\mathrm{nf}_\flat(B_p) \equiv \flat^{i:1..n_p}_{x_i:F_i}.C$ where $n_p \geq 0$. If $B_2 \equiv \Box$ then $B_1 \twoheadrightarrow_\flat \Box$ and by subject reduction, $\Gamma \vdash_\flat \Box : s_1$, absurd by Lemma 34.1. Hence, $B_2 \not\equiv \Box$ and by correctness of types, $\exists s_2$ such that $\Gamma \vdash_\flat B_2 : s_2$. By subject reduction, $\Gamma \vdash_\flat \mathrm{nf}_\flat(B_1) : s_1$ and $\Gamma \vdash_\flat \mathrm{nf}_\flat(B_2) : s_2$. By n_1 (resp. n_2) generations, for $s' =_\flat s_1$ and $s'' =_\flat s_2$, $\Gamma, x_1 : F_1, \ldots, x_{n_1} : F_{n_1} \vdash_\flat C : s'$ and $\Gamma, x_1 : F_1, \ldots, x_{n_2} : F_{n_2} \vdash_\flat C : s''$.
   - If $n_1 \leq n_2$, by thinning, $\Gamma, x_1 : F_1, \ldots, x_{n_2} : F_{n_2} \vdash_\flat C : s'$. By 1, $s' \overset{\diamond}{\because}_\flat s''$.
   - If $n_1 \geq n_2$, by thinning, $\Gamma, x_1 : F_1, \ldots, x_{n_1} : F_{n_1} \vdash_\flat C : s''$. By 1, $s' \overset{\diamond}{\because}_\flat s''$.

   Hence, $s' \equiv s''$. Since $s' =_\flat s_1$ and $s'' =_\flat s_2$, we get $s_1 \equiv s_2$ and $\Gamma \vdash_\flat B_2 : s_1$.
4. (a) By Lemma 29 $\overline{B'_1} \equiv B_1 \wedge \overline{B'_2} \equiv B_2$. As $B'_1 =_\beta B'_2$, by Lemma 6.3 $B_1 =_\flat B_2$.
   (b) Γ' is unique by Lemma 25.3. By Lemma 29 $\Gamma' \vdash_\beta A' : B'_1 \wedge \Gamma' \vdash_\beta C' : B'_2 \wedge$ $\overline{B'_1} \equiv B_1 \wedge \overline{B'_2} \equiv B_2$. By Lemmas 21 and 6.3, $B'_1 =_\beta B'_2$ and $B_1 =_\flat B_2$.

$\boxtimes$

This lemma means that the $\flat$-cube works as expected. We do not want that a $\flat$-term which represents a λ-term gets the same type as a $\flat$-term which represents a Π-term. The type of $\lambda_{x:A}.B$ must have more abstractions than the type of $\Pi_{x:A}.B$. In fact, the type of a term decides if this term is λ- or a Π-term. Take Example 32.1, by Theorem 30, $\vdash_\flat \flat_{x:*}.x : \flat_{x:*}.*$ can only be written in one way using λ and Π instead of $\flat$. The $\flat$ in $\flat_{x:*}.*$ must be Π (By Lemma 23, $\not\vdash \lambda_{x:*}.* : s$). Also, the $\flat$ in $\flat_{x:*}.x$ must be λ (otherwise by generation, $\Pi_{x:*}.* =_\beta s$ absurd by Church-Rosser). In the $\flat$-cube, we keep all the possibilities of a term open, but we have a relationship between all the types of a term. As soon as a particular type is chosen, the term and its type can be uniquely unpacked with λs and Π.

Figure 5 states that the $\flat$-cube has all the properties of the β-cube except unicity of types which is replaced by an organised patterned multiplicity of types.

It is useful for the rest of the paper to classify terms according to degrees.

Properties	β-cube	$\flat$-cube
Church-Rosser	yes	yes
Correctness of types	yes	yes
Typability of subterms	yes	yes
Subject reduction	yes	yes
Unicity of types	yes	organised patterned multiplicity
Strong normalisation	yes	yes

Fig. 5. Comparing the β-cube and the $\flat$-cube

Definition 36. We follow (Barendregt, 1992) and define the degree of terms A denoted $\natural(A)$ by:

$\natural(\Box) = 3 \quad \natural(*) = 2 \quad \natural(x^{\Box}) = 1 \quad \natural(x^{*}) = 0 \quad \natural(\flat_{x:B}.A) = \natural(AB) = \natural(A)$.

We say that $A : B$ is OK if $\natural(B) = \natural(A) + 1$. We say that $A : B$ is HOK (hereditarily OK) if it is OK and for every $\flat_{x:C}$ occurring in A or in B, we have $x : C$ is OK.

The next lemma and its proof are adapted from (Barendregt, 1992).

Lemma 37. *1. If* $\Gamma \vdash_\flat A : \Box$ *then* $\natural(A) = 2$.
2. If $\Gamma \vdash_\flat A : B$ *and* $\natural(A) \notin \{0, 1\}$ *then* $B \equiv \Box$.
3. If $\natural(x) = \natural(a)$ *then* $\natural(B[x := a]) = \natural(B)$.
4. If $\Gamma \vdash_\flat A : B$ *then* $A : B$ *and every* $y : C$ *in* Γ *are* HOK.
5. If $\flat_{x:A}.B$ *is* $\vdash_\flat$*-legal then* $1 \leq \natural(A) \leq 2$ *and* $\natural(B) \leq 2$.
6. If A *and* A' *are* $\vdash_\flat$*-legal and* $A =_\flat A'$ *then* $\natural(A) = \natural(A')$.

Proof. 1. is by induction on the derivation $\Gamma \vdash_\flat A : \Box$.
2. is by induction on the derivation $\Gamma \vdash_\flat A : B$.
3. is by induction on the structure of B.
4. is by induction on the derivation $\Gamma \vdash_\flat A : B$. For (appl$\flat$) use 3 both for showing that $Fa : B[x := a]$ is OK and that for any $\flat_{y:C[x:=a]}$ occurring in $B[x := a]$ we have that $y : C[x := a]$ is OK. We only do the (conv$_\flat$) case. Let $\Gamma \vdash_\flat A : B'$ come from $\Gamma \vdash_\flat A : B$, $\Gamma \vdash_\flat B' : s$ and $B =_\flat B'$. By Lemma 35.3, $\Gamma \vdash_\flat B : s$. By IH, $\natural(A) + 1 = \natural(B)$.

- If $s \equiv \Box$ then by 1 above, $\natural(B) = \natural(B') = 2$. Hence, $\natural(A) + 1 = \natural(B')$.
- If $s \equiv *$ then by 2 above, $\natural(B) \in \{0, 1\}$. By IH, $\natural(B') = \natural(*) - 1 = 1$. Since $\natural(A) + 1 = \natural(B)$, we deduce $\natural(B) = 1 = \natural(B')$. Hence, $\natural(A) + 1 = \natural(B')$.

5. By Lemmas 16 and 15, $\Gamma, x : A \vdash_\flat B : D$ for some D. By Lemma 12, $\Gamma \vdash_\flat A : s$. By 4., $x : A$, $A : s$ and $\flat_{x:A}.B : C$ are OK. Hence, $1 \leq \natural(A) \leq 2$ and $\natural(B) \leq 2$.
6. First, show that if A is $\vdash_\flat$-legal and $A \twoheadrightarrow_\flat A'$ then $\natural(A) = \natural(A')$. Two cases:

- If $\Gamma \vdash_\flat A : B$, by Lemma 17, $\Gamma \vdash_\flat A' : B$ and by 4., $\natural(A) = \natural(B) - 1 = \natural(A')$.
- If $\Gamma \vdash_\flat B : A$, by Lemma 18, $\Gamma \vdash_\flat B : A'$ and by 4., $\natural(A) = \natural(B) + 1 = \natural(A')$.

If $A =_\flat A'$ then by Church-Rosser $A \twoheadrightarrow_\flat C \;{}_\flat\!\twoheadleftarrow A'$. So, $\natural(A) = \natural(C) = \natural(A')$. ⊠

5 Type Checking/inference in the $\flat$-cube

Given Γ, A and B, type checking deals with the question "does $\Gamma \vdash A : B$ hold?". Given Γ and A, type inference deals with the question "is there a B where $\Gamma \vdash A : B$ holds?" and infers such a B. The rules of Figure 4 give a type checking algorithm for the $\flat$-cube. In this section we deal with type inference and with the connection between type checking/inference in the β-cube and the $\flat$-cube. The next definition gives a type inference class algorithm in the $\flat$-cube.

Definition 38. [Type Inference classes in the $\flat$-cube] We define $\mathrm{tnf}(\Gamma, A) \subseteq \mathcal{T}_\flat$ as follows:

$$\begin{array}{ll} \mathrm{tnf}(\Gamma, \Box) & = \emptyset \\ \mathrm{tnf}(\Gamma, *) & = \{\Box \mid \Gamma \vdash_\flat * : \Box\} \\ \mathrm{tnf}(\Gamma, x^s) & = \{\mathrm{nf}_\flat(A) \mid x^s : A \in \Gamma \wedge \Gamma \vdash_\flat A : s\} \\ \mathrm{tnf}(\Gamma, \flat_{x:A}.B) & = \{\flat_{x:\mathrm{nf}_\flat(A)}.C \mid C \in \mathrm{tnf}(\Gamma, x : A, B) \wedge \Gamma \vdash_\flat \flat_{x:A}.C : s''\} \cup \\ & \quad \{s' \in \mathrm{tnf}(\Gamma, x : A, B) \mid \exists s \in \mathrm{tnf}(\Gamma, A) \text{ where } (s, s') \in \boldsymbol{R}\} \\ \mathrm{tnf}(\Gamma, Fa) & = \{\mathrm{nf}_\flat(B[x := a]) \mid \flat_{x:A}.B \in \mathrm{tnf}(\Gamma, F) \wedge A \in \mathrm{tnf}(\Gamma, a)\} \end{array}$$

Lemma 39. *1. If $B \in tnf(\Gamma, A)$ then B is in $\flat$-normal form and $\Gamma \vdash_\flat A : B$.*
2. If $\Gamma \vdash_\flat A : B$ then $\mathrm{nf}_\flat(B) \in tnf(\Gamma, A)$.
3. $tnf(\Gamma, A) = \emptyset$ if and only if for every B, $\Gamma \not\vdash_\flat A : B$.

Proof. 1. By induction on the structure of A.
2. By induction on the derivation $\Gamma \vdash_\flat A : B$ where in (weak) we need:
"if $\Gamma \subseteq \Gamma'$, A is $\Gamma \vdash_\flat$-legal and Γ' is $\vdash_\flat$-legal then $\mathrm{tnf}(\Gamma, A) = \mathrm{tnf}(\Gamma', A)$"
which can be shown by induction on the structure of A (for this note that if $C \in \mathrm{tnf}(\Gamma, x : D, E)$ then by 1., $\Gamma, x : D \vdash_\flat E : C$ and hence $\Gamma \vdash_\flat \flat_{x:D}.C : s'' \Leftrightarrow \Gamma' \vdash_\flat \flat_{x:D}.C : s''$).
3. follows from 1 and 2. ⊠

This means that A is typable in context Γ if and only if $\mathrm{tnf}(\Gamma, A) \neq \emptyset$. Moreover, the normal form of any possible Γ-type of A is in $\mathrm{tnf}(\Gamma, A)$. Finally, all the infered types are related by $\overset{\diamond}{\because}_\flat$ and, when we type $(\flat_{x:A}.B)a$, although we have many types for $\flat_{x:A}.B$, we only have one applicable type for a and hence, the number of types of $(\flat_{x:A}.B)a$ will not grow beyond the number of types of $\flat_{x:A}.B$:

Lemma 40. *1. If $B, C \in tnf(\Gamma, A)$ then $B \overset{\diamond}{\because}_\flat C$.*
2. Let Fa be $\Gamma \vdash_\flat$-legal. There is a unique $A \in tnf(\Gamma, a)$ where $\flat_{x:A}.B \in tnf(\Gamma, F)$.
3. If $s, \flat_{x:\mathrm{nf}_\flat(A)}.C \in tnf(\Gamma, \flat_{x:A}.B)$ then $C \equiv \flat^{i:1..n}_{x_i:A_i}.s$ where $n \geq 0$.
4. Let $|S|$ stand for the size of set S. We have: $|tnf(\Gamma, *)| \leq 1$, $|tnf(\Gamma, x)| \leq 1$, $|tnf(\Gamma, Fa)| \leq |tnf(\Gamma, F)|$ and $|tnf(\Gamma, \flat_{x:A}.B)| \leq |tnf(\Gamma, x : A, B)| + 1$.

Proof. 1. By Lemma 39.1, B, C are in $\flat$-normal form, $\Gamma \vdash_\flat A : B$ and $\Gamma \vdash_\flat A : C$. By Lemma 35.1, $B \overset{\diamond}{=}_\flat C$. Hence, $B \overset{\diamond}{\because}_\flat C$.
2. As Fa is $\Gamma \vdash_\flat$-legal, $\Gamma \vdash_\flat Fa : C$ or $\Gamma \vdash_\flat C : Fa$. By Lemma 16 we assume $\Gamma \vdash_\flat Fa : C$. By lemma 39.2, $\mathrm{tnf}(\Gamma, Fa) \neq \emptyset$. Hence, $\exists A \in \mathrm{tnf}(\Gamma, a)$ where $\flat_{x:A}.B \in \mathrm{tnf}(\Gamma, F)$. If $D \in \mathrm{tnf}(\Gamma, a)$ where $\flat_{x:D}.E \in \mathrm{tnf}(\Gamma, F)$, by 1., $\flat_{x:A}.B \overset{\diamond}{\because}_\flat \flat_{x:D}.E$ and so $A \equiv D$.

3. By 1 above, $\flat_{x:\mathrm{nf}_\flat(A)}.C \overset{\diamond}{\rightarrow}_\flat s$. Hence, $C \equiv \flat^{i:1..n}_{x_i:A_i}.s$ where $n \geq 0$.
4. For Fa use 2. For $\flat_{x:A}.B$ note that if $s, s' \in \mathrm{tnf}(\Gamma, \flat_{x:A}.B)$, by 1., $s \equiv s'$. ⊠

Using Theorem 30 and Lemma 24 we can establish the following:

Lemma 41. *If* $\Gamma \vdash_\flat A : B$ *then* $\mathrm{nf}_\flat(A) \equiv \flat^{i:1..m}_{x_i:F_i}.C$ *and* $\mathrm{nf}_\flat(B) \equiv \flat^{i:1..n}_{x_i:F_i}.D$ *where* $0 \leq n \leq m$, *and:*
$C \equiv *$ *or* $C \equiv xL_1 \cdots L_k$ *where* $k \geq 0$ *and* $\Gamma, x_1 : F_1, \cdots x_m : F_m \vdash_\flat C : D$.

Proof. By Theorem 30, $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$ where $\overline{\Gamma'} \equiv \Gamma$, $\overline{A'} \equiv A$, $\overline{B'} \equiv B$ and $\Gamma' \vdash_\beta A' : B'$. By Lemma 24, $\mathrm{nf}_\beta(A') \equiv \lambda^{i:1..n}_{x_i:F'_i}.\Pi^{i:n+1..m}_{x_i:F'_i}.C'$ and $\mathrm{nf}_\beta(B') \equiv \Pi^{i:1..n}_{x_i:F'_i}.D'$ where $0 \leq n \leq m$, $C' \equiv *$ or $C' \equiv xL'_1 \cdots L'_k$ where $k \geq 0$ and $\Gamma', x_1 : F'_1, \cdots x_m : F'_m \vdash_\beta C' : D'$. Since A', B' are $\vdash_\beta$-legal, by Lemma 23.5 $\mathrm{nf}_\beta(A') \equiv \mathrm{nf}_{\beta\Pi}(A')$ and $\mathrm{nf}_\beta(B') \equiv \mathrm{nf}_{\beta\Pi}(B')$. Let $\overline{C'} \equiv C$, $\overline{D'} \equiv D$, for $0 \leq i \leq k$ $\overline{L_i} \equiv L_i$, and for $0 \leq i \leq m$ $\overline{F'_i} \equiv F_i$. By Lemma 8.2 and Theorem 30, $\mathrm{nf}_\flat(A) \equiv \flat^{i:1..m}_{x_i:F_i}.C$ and $\mathrm{nf}_\flat(B) \equiv \flat^{i:1..n}_{x_i:F_i}.D$ where $0 \leq n \leq m$, $C \equiv *$ or $C \equiv xL_1 \cdots L_k$ where $k \geq 0$ and $\Gamma, x_1 : F_1, \cdots x_m : F_m \vdash_\flat C : D$. ⊠

Looking at Lemmas 40.3 and 41, one may wonder whether it is the case that: if $\Gamma \vdash_\flat \flat_{x:A}.B : \flat^{i:1..n}_{x_i:A_i}.D$ where $n \geq 1$ then for all k, if $0 \leq k \leq n$ we have: $\Gamma \vdash_\flat \flat_{x:A}.B : \flat^{i:1..k}_{x_i:A_i}.D$. This however does not hold. Here is an example:

Example 42. 1. $y : * \vdash_\flat \flat_{x:y}.x : \flat_{x:y}.y$ but $y : * \not\vdash_\flat \flat_{x:y}.x : y$.
2. If $(\Box, \Box) \in \boldsymbol{R}$ and $(\Box, *) \notin \boldsymbol{R}$ then $\vdash_\flat \flat_{x:*}.x : \flat_{x:*}.*$ but $\not\vdash_\flat \flat_{x:*}.x : *$.

Next, we show that type checking/inference in the $\flat$-cube is equivalent to type checking/inference in the β-cube.

Lemma 43. *Let* $r \in \{\beta, \flat\}$. *Let* Π_{cr} *resp.* Π_{ir} *stand for type checking resp. type inference in the* r*-cube.* $\Pi_{c\beta}$ *is equivalent to* $\Pi_{c\flat}$ *and* $\Pi_{i\beta}$ *is equivalent to* $\Pi_{i\flat}$.

Proof. Theorem 30 and Lemmas 29 and 39 help us prove the next equivalences:
1. $\Gamma \vdash_\flat A : B$ if and only if $\{(\Gamma', A', B') \in [\Gamma] \times [A] \times [B] \mid \Gamma' \vdash_\beta A' : B'\} \neq \emptyset$.
2. $\exists B$ where $\Gamma \vdash_\flat A : B$ if and only if
   $\{(\Gamma', A') \in [\Gamma] \times [A] \mid \exists B'$ where $\Gamma' \vdash_\beta A' : B'\} \neq \emptyset$.
3. $\Gamma \vdash_\beta A : B$ if and only if $\overline{\Gamma} \vdash_\flat \overline{A} : \overline{B}$ and $(\overline{\Gamma} \vdash_\flat \overline{A} : \overline{B})^{-1} = (\Gamma, A, B)$.
4. $\exists B$ where $\Gamma \vdash_\beta A : B$ if and only if
   $\{C \in \mathrm{tnf}(\overline{\Gamma}, \overline{A}) \mid (\overline{\Gamma} \vdash_\flat \overline{A} : C)^{-1} = (\Gamma, A, D)\} \neq \emptyset$.

By 1., $\Pi_{c\flat}$ reduces to $\Pi_{c\beta}$. By 2., $\Pi_{i\flat}$ reduces to $\Pi_{i\beta}$. By 3., $\Pi_{c\beta}$ reduces to $\Pi_{c\flat}$. By 4., $\Pi_{i\beta}$ reduces to $\Pi_{i\flat}$. ⊠

6 Comparing with Coquand's unification of binders

6.1 Coquand's calculus of constructions with unified binders

In (Coquand, 1985), Coquand first gave the calculus of constructions C^* with unified binders. He used de Bruijn's notation $[x : A]B$ for abstraction, but for uniformity, we write his calculus with the $\flat$-binder. Note that Coquand's calculus as presented in this section may look quite different from the usual notation for the

(axiomc)	$\langle\rangle \vdash *$	
(varc)	$\dfrac{\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash *}{\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash x : A^{1/2}}$	
(contc)	$\dfrac{\Gamma \vdash B^1 : * \quad x^* \notin \mathrm{DOM}(\Gamma)}{\Gamma, x^*{:}B^1 \vdash *}$	$\dfrac{\Gamma \vdash B^2 \quad x^\Box \notin \mathrm{DOM}(\Gamma)}{\Gamma, x^\Box{:}B^2 \vdash *}$
(Π_c)	$\dfrac{\Gamma, x : A^{1/2} \vdash B^1 : *}{\Gamma \vdash \flat_{x:A^{1/2}}.B^1 : *}$	$\dfrac{\Gamma, x : A^{1/2} \vdash B^2}{\Gamma \vdash \flat_{x:A^{1/2}}.B^2}$
(λ_c)	$\dfrac{\Gamma, x : A^{1/2} \vdash b : B}{\Gamma \vdash \flat_{x:A^{1/2}}.b : \flat_{x:A^{1/2}}.B}$	
(convc)	$\dfrac{\Gamma \vdash A : B^1 \quad \Gamma \vdash C^1 : * \quad B^1 =_\flat C^1}{\Gamma \vdash A : C^1}$	$\dfrac{\Gamma \vdash A^1 : B^2 \quad \Gamma \vdash C^2 \quad B^2 =_\flat C^2}{\Gamma \vdash A^1 : C^2}$
(applc)	$\dfrac{\Gamma \vdash F : \flat_{x:A^{1/2}}.B \qquad \Gamma \vdash a : A^{1/2}}{\Gamma \vdash Fa : B[x := a]}$	

Fig. 6. Coquand's typing rules for C^*

systems of the cube. In Section 6.3, we present Coquand's calculus in modern notation.

Coquand gave terms and contexts as follows:

$$
\begin{array}{llllllll}
\mathcal{T}^{0/1/2} & ::= & \mathcal{T}^0 & \mid & \mathcal{T}^1 \mid \mathcal{T}^2 & & & \\
\mathcal{T}^{0/1} & ::= & \mathcal{T}^0 & \mid & \mathcal{T}^1 & & & \\
\mathcal{T}^{1/2} & ::= & & & \mathcal{T}^1 \mid \mathcal{T}^2 & & & \\
\mathcal{T}^0 & ::= & \mathcal{V}^* & \mid & \flat_{\mathcal{V}:\mathcal{T}^{1/2}}.\mathcal{T}^0 & \mid \; \mathcal{T}^0\mathcal{T}^0 & \mid & \mathcal{T}^0\mathcal{T}^1 \\
\mathcal{T}^1 & ::= & \mathcal{V}^\Box & \mid & \flat_{\mathcal{V}:\mathcal{T}^{1/2}}.\mathcal{T}^1 & \mid \; \mathcal{T}^1\mathcal{T}^0 & \mid & \mathcal{T}^1\mathcal{T}^1 \\
\mathcal{T}^2 & ::= & * & \mid & \flat_{\mathcal{V}:\mathcal{T}^{1/2}}.\mathcal{T}^2 & & & \\
\Gamma & ::= & \langle\rangle & \mid & \Gamma, \mathcal{V} : \mathcal{T}^{1/2} & & &
\end{array}
$$

We use the same convention for metavariables as in Notation 2. We may decorate terms with superscripts to reflect the set they belong to (e.g., $A^0 \in \mathcal{T}^0$ and $A^{1/2} \in \mathcal{T}^{1/2}$). We write $A^2 \leq B^2$ if and only if "$A^2 \equiv \flat_{x_1:A_1^{1/2}} \ldots \flat_{x_l:A_l^{1/2}}.*$, $B^2 \equiv \flat_{x_1:B_1^{1/2}} \ldots \flat_{x_n:B_n^{1/2}}.*$ where $l \leq n$ and $A_i^{1/2} =_\flat B_i^{1/2}$ for $1 \leq i \leq l$".

Coquand gave the typing rules of C^* (cf. Figure 6) and proved Lemma 44 as well as the strong normalisation theorem for C^*. We use $\vdash_{C^*}$ for type derivation in Coquand's C^* according to the rules of Figure 6.

Lemma 44. *Let Φ range over A^2, $B^1 : A^2$ and $C^0 : B^1$. The following holds:*
1. *If $\Gamma_1, \Gamma_2 \vdash_{C^*} \Phi$ then $\Gamma_1 \vdash_{C^*} *$.*
2.(a) *If $\Gamma \vdash_{C^*} A^0 : B^1$ and $\Gamma \vdash_{C^*} A^0 : C^1$ then $B^1 =_\flat C^1$.*
(b) *If $\Gamma \vdash_{C^*} A^1 : B^2$ and $\Gamma \vdash_{C^*} A^1 : C^2$ then either $B^2 \leq C^2$ or $C^2 \leq B^2$.*
3. *Assume every occurrence of $x : D^{1/2}$ in Γ occurs also in Γ' where $\Gamma \vdash_{C^*} *$ and $\Gamma' \vdash_{C^*} *$. If $\Gamma \vdash_{C^*} \Phi$ then $\Gamma' \vdash_{C^*} \Phi$.*

(axiomc$^\Box$) $$\langle\rangle \vdash * : \Box$$

(varc$^\Box$) $$\frac{\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash * : \Box}{\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash x : A^{1/2}}$$

(contc$^\Box$) $$\frac{\Gamma \vdash B^i : s^{i+1} \qquad i \in \{1,2\} \qquad x^s \not\in \mathrm{DOM}\,(\Gamma)}{\Gamma, x^s{:}B^i \vdash * : \Box}$$

($\Pi_c^\Box$) $$\frac{\Gamma, x : A^{1/2} \vdash B^i : s^{i+1} \qquad i \in \{1,2\}}{\Gamma \vdash \flat_{x:A^{1/2}}.B^i : s^{i+1}}$$

($\lambda_c^\Box$) $$\frac{\Gamma, x : A^{1/2} \vdash b : B \qquad B \not\equiv \Box}{\Gamma \vdash \flat_{x:A^{1/2}}.b : \flat_{x:A^{1/2}}.B}$$

(convc$^\Box$) $$\frac{\Gamma \vdash A : B^i \qquad \Gamma \vdash C^i : s^{i+1} \qquad B^i =_\flat C^i \qquad i \in \{1,2\}}{\Gamma \vdash A : C^i}$$

(applc) $$\frac{\Gamma \vdash F : \flat_{x:A^{1/2}}.B \qquad \Gamma \vdash a : A^{1/2}}{\Gamma \vdash Fa : B[x := a]}$$

Fig. 7. The typing rules of $C^\Box$

4. *If* $\Gamma \vdash_{C^*} a : D^{1/2}$ *and* $\Gamma, x : D^{1/2}, \Gamma' \vdash_{C^*} \Phi$ *then* $\Gamma, \Gamma'[x := a] \vdash_{C^*} \Phi[x := a]$.

5.(a) *If* $\Gamma \vdash_{C^*} A^0 : B^1$ *then* $\Gamma \vdash_{C^*} B^1 : *$.

(b) *If* $\Gamma \vdash_{C^*} A^1 : B^2$ *then* $\Gamma \vdash_{C^*} B^2$.

6.(a) *If* $\Gamma, x : D^1, \Gamma' \vdash_{C^*} \Phi$ *and* $\Gamma \vdash_{C^*} E^1 : *$ *and* $D^1 =_\flat E^1$ *then* $\Gamma, x : E^1, \Gamma' \vdash_{C^*} \Phi$.

(b) *If* $\Gamma, x : D^2, \Gamma' \vdash_{C^*} \Phi$ *and* $\Gamma \vdash_{C^*} E^2$ *and* $D^2 =_\flat E^2$ *then* $\Gamma, x : E^2, \Gamma' \vdash_{C^*} \Phi$.

7. *If* $\Gamma \vdash_{C^*} B : A^{1/2}$ *and* $B \twoheadrightarrow_\flat B'$ *then* $\Gamma \vdash_{C^*} B' : A^{1/2}$.

Lemma 44.2 is related to Lemma 35.1. Below, we compare C^* to $\flat_C$ further.

6.2 The isomorphism between Coquand's calculus and $\flat_C$

We simplify the presentation of C^* by using a new calculus $C^\Box$ whose syntax is that of C^* together with the set $\mathcal{T}^3 ::= \Box$ and whose typing rules are those of Figure 7. As before, we use s to range over $\{*, \Box\}$ and from the superscript on s we can work out what s stands for: s^2 is $*$ and s^3 is $\Box$. We use $\vdash_{C^\Box}$ to denote type derivation in $C^\Box$.

We will show that $C^\Box$, C^* and $\flat_C$ are isomorphic. First, we need to define translations $|.|$ and $\langle . \rangle$ between statements of C^* and $C^\Box$ as follows:

$$|A^2| = A^2 : \Box \qquad |B : C| = B : C \qquad \langle B : C \rangle = \begin{cases} B & \text{if } C \equiv \Box \\ B : C & \text{otherwise} \end{cases}$$

Note that $|\langle B : C \rangle| = B : C$.

Lemma 45 (C^* isomorphic to $C^\Box$). *1. If* $\Gamma \vdash_{C^\Box} A : B$ *then* $\Gamma \vdash_{C^*} \langle A : B \rangle$.
2. Let Φ *range over* $A^2 : \Box$ *and* $B : C$. *If* $\Gamma \vdash_{C^*} \Phi$ *then* $\Gamma \vdash_{C^\Box} |\Phi|$.

Proof. 1. By induction on $\Gamma \vdash_{C^\square} A : B$. 2. By induction on $\Gamma \vdash_{C^*} \Phi$. ⊠

The next lemma (used in Lemma 47) shows that the $\mathcal{T}^i$s classify terms of $C^\square$ according to their degrees and that $\vdash_\flat$-legal terms $A \in \mathcal{T}_\flat$ belong to $\mathcal{T}^{\natural(A)}$ of $C^\square$.

Lemma 46. *1. For* $0 \le i \le 3$, $\mathcal{T}^i \subseteq \mathcal{T}_\flat$. *2. For* $0 \le i \le 3$, $\natural(A^i) = i$.
3. If A is $\vdash_\flat$-legal then $A \in \mathcal{T}^{\natural(A)}$.
4. If $\Gamma \vdash_\flat A : s$ *and* $\natural(A) = i$ *then* $A \in \mathcal{T}^i$, $s \in \mathcal{T}^{i+1}$ *and* $i \in \{1, 2\}$.

Proof. 1. Obviously $\mathcal{T}^3 \subseteq \mathcal{T}_\flat$. Then, prove by induction on the structure of A that if $A \in \mathcal{T}^{0/1/2}$ then $A \in \mathcal{T}_\flat$. 2. By induction on the structure of $A^i \in \mathcal{T}^i$.
3. By induction on the structure of A. We only treat $\flat_{x:B}.C$ and Fa. Since $A \not\equiv \square$ is $\vdash_\flat$-legal then $\Gamma \vdash_\flat A : D$ for some Γ, D (use Lemma 16 if needed).

- If $A \equiv \flat_{x:B}.C$ then by Lemma 37.5, $1 \le \natural(B) \le 2$ and $\natural(C) \le 2$. By IH, $B \in \mathcal{T}^{\natural(B)} \subseteq \mathcal{T}^{1/2}$ and $C \in \mathcal{T}^{\natural(C)} \subseteq \mathcal{T}^{0/1/2}$. Hence, $\flat_{x:B}.C \in \mathcal{T}^{\natural(C)} = \mathcal{T}^{\natural(\flat_{x:B}.C)}$.
- If $A \equiv Fa$, by generation, $\Gamma \vdash_\flat F : \flat_{x:B}.C$ and $\Gamma \vdash_\flat a : B$. By Lemma 37.5, $\natural(B), \natural(C) \le 2$. By Lemma 37.4, $\natural(a) = \natural(B) - 1 \le 1$, and $\natural(F) = \natural(C) - 1 \le 1$. By IH, $F \in \mathcal{T}^{\natural(F)} \subseteq \mathcal{T}^{0/1}$ and $a \in \mathcal{T}^{\natural(a)} \subseteq \mathcal{T}^{0/1}$. Hence, $Fa \in \mathcal{T}^{\natural(F)} = \mathcal{T}^{\natural(Fa)}$.

4. By Lemma 37.4, $i = \natural(A) = \natural(s) - 1 \in \{1, 2\}$. By 3, $s \in \mathcal{T}^{i+1}$ and $A \in \mathcal{T}^i$. ⊠

Lemma 47 ($C^\square$ isomorphic to $\flat_C$). $\Gamma \vdash_{\flat_C} A : B$ *if and only if* $\Gamma \vdash_{C^\square} A : B$.

Proof. "If" is by induction on the derivation $\Gamma \vdash_{C^\square} A : B$. Note by Lemma 46.1, for $0 \le i \le 3$, $\mathcal{T}^i \subseteq \mathcal{T}_\flat$. Also, in $\flat_C$, for any s, s', $(s, s') \in \boldsymbol{R}$. We only treat:

- (varc$^\square$). If $\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash_{C^\square} x : A^{1/2}$ comes from $\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash_{C^\square} * : \square$, by IH, $\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash_{\flat_C} * : \square$. By Lemma 12, $\Gamma_1, x : A^{1/2}, \Gamma_2 \vdash_{\flat_C} x : A^{1/2}$.
- ($\Pi_c^\square$). If $\Gamma \vdash_{C^\square} \flat_{x:A^{1/2}}.B^i : s^{i+1}$ comes from $\Gamma, x{:}A^{1/2} \vdash_{C^\square} B^i : s^{i+1}$ where $i \in \{1, 2\}$, by IH, $\Gamma, x{:}A^{1/2} \vdash_{\flat_C} B^i : s^{i+1}$. By Lemma 12, $\Gamma \vdash_{\flat_C} A^{1/2} : s'$. By ($\flat_1$), $\Gamma \vdash_{\flat_C} \flat_{x:A^{1/2}}.B^i : s^{i+1}$.
- ($\lambda_c^\square$). If $\Gamma \vdash_{C^\square} \flat_{x:A^{1/2}}.b : \flat_{x:A^{1/2}}.B$ comes from $\Gamma, x{:}A^{1/2} \vdash_{C^\square} b : B$ where $B \not\equiv \square$, by IH, $\Gamma, x{:}A^{1/2} \vdash_{\flat_C} b : B$. By Lemmas 12 and 16, $\Gamma \vdash_{\flat_C} A^{1/2} : s_1$ and $\Gamma, x{:}A^{1/2} \vdash_{\flat_C} B : s_2$. By ($\flat_1$), $\Gamma \vdash_{\flat_C} \flat_{x:A^{1/2}}.B : s_2$. Hence, by ($\flat_2$), $\Gamma \vdash_{\flat_C} \flat_{x:A^{1/2}}.b : \flat_{x:A^{1/2}}.B$.

The "only if" case is by induction on the derivation $\Gamma \vdash_{\flat_C} A : B$. We only treat:

- (weak). If $\Gamma, x^s{:}C \vdash_{\flat_C} A : B$ comes from $\Gamma \vdash_{\flat_C} A : B$, $\Gamma \vdash_{\flat_C} C : s$ and $x^s \notin \textsc{dom}(\Gamma)$, by IH, $\Gamma \vdash_{C^\square} A : B$ and $\Gamma \vdash_{C^\square} C : s$. By Lemma 46.4, $C \equiv C^i$ and $s \equiv s^{i+1}$ where $i \in \{1, 2\}$. By (contc$^\square$), $\Gamma, x^s{:}C \vdash_{C^\square} * : \square$. By Lemma 45.1, $\Gamma \vdash_{C^*} \langle A : B \rangle$ and $\Gamma, x^s{:}C \vdash_{C^*} *$. By Lemma 44.(1 resp. 3) $\Gamma \vdash_{C^*} *$ and $\Gamma, x^s{:}C \vdash_{C^*} \langle A : B \rangle$. By Lemma 45.2, $\Gamma, x^s{:}C \vdash_{C^\square} |\langle A : B \rangle| = A : B$.
- ($\flat_1$). If $\Gamma \vdash_{\flat_C} \flat_{x:A}.B : s_2$ comes from $\Gamma \vdash_{\flat_C} A : s_1$ and $\Gamma, x : A \vdash_{\flat_C} B : s_2$, by IH, $\Gamma \vdash_{C^\square} A : s_1$ and $\Gamma, x : A \vdash_{C^\square} B : s_2$. By Lemmas 37.5 and 46.3, $1 \le \natural(A) \le 2$ and $A \equiv A^{1/2}$. By Lemma 46.4, $B \equiv B^i$ and $s \equiv s^{i+1}$ where $i \in \{1, 2\}$. Hence, by ($\Pi_c^\square$), $\Gamma \vdash_{C^\square} \flat_{x:A}.B : s_2$.
- (conv$_\flat$). If $\Gamma \vdash_{\flat_C} A : C$ comes from $\Gamma \vdash_{\flat_C} A : B$, $\Gamma \vdash_{\flat_C} C : s$ and $B =_\flat C$ then by IH, $\Gamma \vdash_{C^\square} A : B$, $\Gamma \vdash_{C^\square} C : s$. Let $i = \natural(C)$. By Lemma 46.4, $s \equiv s^{i+1}$, $C \equiv C^i \in \mathcal{T}^i$ and $i \in \{1, 2\}$. By Lemma 37.6, $\natural(B) = \natural(C) = i$. By Lemma 46.3, $B \equiv B^i \in \mathcal{T}^i$. Hence by (convc$^\square$), $\Gamma \vdash_{C^\square} A : C$. ⊠

(axiomc$^\square$)	$\langle\rangle \vdash * : \square$
(var$^\square$)	$\dfrac{\Gamma_1, x : A, \Gamma_2 \vdash * : \square}{\Gamma_1, x : A, \Gamma_2 \vdash x : A}$
(cont$^\square$)	$\dfrac{\Gamma \vdash B : s \qquad x^s \notin \text{DOM}\,(\Gamma)}{\Gamma, x^s{:}B \vdash * : \square}$
($\Pi^\square$)	$\dfrac{\Gamma, x : A \vdash B : s}{\Gamma \vdash \flat_{x:A}.B : s}$
($\lambda^\square$)	$\dfrac{\Gamma, x : A \vdash b : B \qquad B \not\equiv \square}{\Gamma \vdash \flat_{x:A}.b : \flat_{x:A}.B}$
(conv$^\square$)	$\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash C : s \quad B =_\flat C}{\Gamma \vdash A : C}$
(appl)	$\dfrac{\Gamma \vdash F : \flat_{x:A}.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[x := a]}$

Fig. 8. The typing rules of $C_\flat$

6.3 Coquand's calculus in modern notation

We define the calculus $C_\flat$ whose terms are $\mathcal{T}_\flat$ and whose typing rules are those of Figure 8. We show that $C^\square$ (hence C^*) and $C_\flat$ are isomorphic.

Lemma 48. $\Gamma \vdash_{C^\square} A : B$ *if and only if* $\Gamma \vdash_{C_\flat} A : B$.

Proof. Recall by Lemma 46.1 that for $0 \leq i \leq 3$, $\mathcal{T}^i \subseteq \mathcal{T}_\flat$. Since the rules of $C^\square$ are rules of $C_\flat$, we only need to show: if $\Gamma \vdash_{C_\flat} A : B$ then $\Gamma \vdash_{C^\square} A : B$. This is by induction on the derivation $\Gamma \vdash_{C_\flat} A : B$ using Lemmas 12, 46 and 47. ⊠

7 Conclusion

In this paper, we used a unique binder à la de Bruijn instead of the usual two binders λ and Π. We studied eight of the most used type systems (those of Barendregt's β-cube) written in this notation and established an isomorphism between the two versions. We showed that although $\flat$ replaces both λ and Π, in any legal term, one can easily unpack the status of a $\flat$ (i.e., whether it should act as a λ or as a Π). We also showed that all the desirable properties of type systems still hold in the $\flat$-cube except for unicity of types. Moreover, we established a relationship $\overset{\diamond}{\because}_\flat$ between types where $A \overset{\diamond}{\because}_\flat B$ if and only if $A \equiv \flat_{x_1:A_1} \ldots \flat_{x_n:A_n}.C$ and $B \equiv \flat_{x_1:A_1} \ldots \flat_{x_m:A_m}.C$ where $n, m \geq 0$. We showed that if a term has two types A and B, then $\text{nf}_\flat(A) \overset{\diamond}{\because}_\flat \text{nf}_\flat(B)$. This result, together with the ability to unpack the status of a $\flat$ if needed, as well as all the other properties, make it desirable to write the single $\flat$ instead of the two different binders λ and Π. The Automath experience is another factor as to why unifying λ and Π is desirable. Just as the development of type theory meant that in the more expressive type systems, terms and types have the same syntax and act alike, we believe that this development should also mean that λ and Π act alike. In fact, λ and Π already act alike, so why not use the same name for them? This paper shows that there are no reasons why these binders should not be unified and that it is more natural that they are unified. Moreover, this unification brings elegance to the representation of powerful features. As an example, the type inclusion rule used in the Automath system AUT-QE to enable two different terms which stand for the same definition to have at least one common type, is written in the $\flat$- resp. β-cubes as follows (note the elegance of $(Q_\flat)$ compared to (Q_β)):

$$\frac{\Gamma \vdash A : \flat^{i:1..n}_{x_i:A_i}.*}{\Gamma \vdash A : \flat^{i:1..m}_{x_i:A_i}.*} \qquad 0 \le m \le n \tag{$Q_\flat$}$$

$$\frac{\Gamma \vdash \lambda^{i:1..k}_{x_i:A_i}.A : \Pi^{i:1..n}_{x_i:A_i}.*}{\Gamma \vdash \lambda^{i:1..m}_{x_i:A_i}.\Pi^{i:m+1..k}_{x_i:A_i}A : \Pi^{i:1..m}_{x_i:A_i}.*} \qquad 0 \le m \le n,\ A \not\equiv \lambda_{x:B}.C \tag{Q_β}$$

Acknowledgements

I am grateful for the comments received from Henk Barendregt and J.B. Wells. The anonymous referees gave valuable feedback which much improved the paper.

References

Barendregt, H.P. (1984) *The Lambda Calculus: its Syntax and Semantics*. Studies in Logic and the Foundations of Mathematics 103. North-Holland.

Barendregt, H.P. (1992) Lambda calculi with types. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum (editors), *Handbook of Logic in Computer Science, Volume 2*, pp. 117–309. Oxford University Press.

de Bruijn, N.G. (1970) The mathematical language AUTOMATH, its usage and some of its extensions. In M. Laudet, D. Lacombe, and M. Schuetzenberger (editors), *Symposium on Automatic Demonstration, Lecture Notes in Mathematics 125*, pp. 29–61, INRIA, Versailles, Springer Verlag, Berlin.

Church, A. (1940) A formulation of the simple theory of types. *The Journal of Symbolic Logic 5*, pp. 56–68, 1940.

Coquand. T (1985) *Une Théorie des Constructions*. PhD thesis, Université Paris 7.

Coquand, T. and Huet, G. (1988) The calculus of constructions. *Information and Computation 76*, pp. 95–120.

Girard, J.-Y. (1972) *Interprétation fonctionelle et élimination des coupures dans l'arithmétique d'ordre supérieur*. PhD thesis, Université Paris 7.

de Groote. P. (1993) Defining lambda-typed lambda-calculi by axiomatizing the typing relation. In P. Enjalbert, A. Finkel, and K.W. Wagner (editors), *STACS'93, Lecture Notes in Computer Science 665*, pp. 712–723.

Howard. W.A. (1980) The formulas-as-types notion of construction. In J.P. Seldin and J.R. Hindley (editors). *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pp. 479–490, Academic Press, New York.

Kamareddine, F. (2002) On Functions and Types, a tutorial. Invited talk. In W.I. Grosky and F. Plasil (editors), *29th Annual Conference on Current Trends in Theory and Practice of Informatics*, Milovy, Czech Republic, *Lecture Notes in Computer Science 2540*, pp. 74–93.

Kamareddine, F., Bloo, R. and Nederepelt, R. (1999) On π-conversion in the λ-cube and the combination with abbreviations. *Annals of Pure and Applied Logic 97*, pp. 27–45.

Kamareddine, F. and Nederepelt, R. (1996) Canonical typing and Π-conversion in the Barendregt Cube. *Journal of Functional Programming, 6(2)*, pp. 245–267.

Peyton-Jones, S. and Meijer, E. (1997) Henk: a typed intermediate language. *Types in Compilation Workshp.*

Reynolds, J.C. (1974) Towards a theory of type structure. *Lecture Notes in Computer Science. 19*, pp. 408–425, Springer-Verlag.