



HAL
open science

Impredicativity, Cumulativity and Product Covariance in the Logical Framework Dedukti

Thiago Felicissimo, Théo Winterhalter

► **To cite this version:**

Thiago Felicissimo, Théo Winterhalter. Impredicativity, Cumulativity and Product Covariance in the Logical Framework Dedukti. 2024. hal-04470850v1

HAL Id: hal-04470850

<https://hal.science/hal-04470850v1>

Preprint submitted on 21 Feb 2024 (v1), last revised 7 May 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Impredicativity, Cumulativity and Product Covariance in the Logical Framework Dedukti

Thiago Felicissimo ✉

Université Paris-Saclay, INRIA project Deducteam, Laboratoire de Méthodes Formelles,
ENS Paris-Saclay, 91190 France

Théo Winterhalter ✉

Université Paris-Saclay, INRIA project Deducteam, Laboratoire de Méthodes Formelles,
ENS Paris-Saclay, 91190 France

Abstract

Proof assistants such as Coq implement a type theory featuring three important features: impredicativity, cumulativity and product covariance. This combination has proven difficult to be expressed in the logical framework Dedukti, and previous attempts have failed in providing an encoding that is proven confluent, sound and conservative. In this work we solve this longstanding open problem by providing an encoding of these three features that we prove to be confluent, sound and to satisfy a restricted (but, we argue, strong enough) form of conservativity. Our proof of confluence is a contribution by itself, and combines classic and modern criteria from higher-order rewriting theory. Our proof of soundness also contributes a new strategy in which the result is shown in terms of an inverse translation function, fixing a common flaw made in some previous encoding attempts.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Equational logic and rewriting

Keywords and phrases Dedukti, Rewriting, Confluence, Dependent types, Cumulativity, Universes

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

As the number of proof systems grow, it becomes increasingly important to understand the relationship between their logics and to which extent they can be expressed in a unified setting. The research project centered around the logical framework DEDUKTI [7, 17] has precisely the intent of providing such a setting. By allowing for the encoding of popular logics such as predicate logic [17], higher-order logic [32, 17], set theory [18] and pure type systems [19, 23], it provides a common framework in which proofs coming from different proof systems can be rechecked, increasing the trust in their correctness. Moreover, DEDUKTI can then also be used for sharing these proofs with other systems, which has already allowed for exporting results to tools like Coq [16, 44], Agda [25] and HOL [44, 29].

The correctness of the verification provided by DEDUKTI relies however on metatheoretic results stating that the theorems that can be proven by a DEDUKTI encoding are exactly the same ones of the encoded logic. In the particular case of the cumulative calculus of constructions, a type theory combining impredicativity and cumulativity with product covariance, giving an encoding satisfying these properties has remained to this day a challenge. This issue is made especially relevant by the fact that this theory is quite popular, and is most notably implemented by the proof assistant COQ.

The current situation regarding encodings of this theory is summarised in Table 1. All encodings presented until now came with a proof of *soundness*, meaning that all facts that can be proven by the encoded logic can also be proven in the encoding. However, the proofs provided by Assaf, Assaf et al and Thiré have turned out to be incorrect, as they rely on ill-defined translation functions—see Section 9 for a detailed explanation. The situation is



© Thiago Felicissimo & Théo Winterhalter;
licensed under Creative Commons License CC-BY 4.0
42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:32



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

	Assaf* [5]	Assaf et al [8]	Thiré* [45]	Férey ⁺ [27]	This work
CONFLUENCE	✗	✓ [‡]	✗	✗	✓
SOUNDNESS	✗ [†]	✗ [†]	✗ [†]	✓	✓
CONSERVATIVITY	✗	✗	✗	✗	✓ [*]

[†]: The translation function is ill-defined (see the discussion in Section 9).

[‡]: Requires matching modulo ACU. *: Only in a restricted form.

*: Also handles other cumulative type systems. +: Also supports universe polymorphism.

■ **Table 1** Comparison with previous encodings

45 even more serious regarding *conservativity*, the property dual to soundness and which ensures
 46 that the encoding cannot prove more theorems than the encoded system. Indeed, none of
 47 the previous proposals have provided a proof of this fact, which is nevertheless essential to
 48 ensure that a proof checked by DEDUKTI is indeed correct in the original system.

49 One of the challenges in proving conservativity is that all known proof methods rely on
 50 confluence—which is moreover also essential to establish subject reduction. However, the
 51 combination of impredicativity, cumulativity and product covariance has proven difficult to
 52 be expressed in a confluent way in DEDUKTI. Indeed, almost all previous encodings have not
 53 succeeded in proving this property. A notable exception is the impressive work of Assaf et
 54 al [8], which however relies on matching modulo ACU (associativity-cumulativity with unit)
 55 a form of matching that is much less efficient and harder to implement than pure syntactical
 56 matching. For instance, the addition of ACU matching to the DKCHECK implementation
 57 doubled the size of the kernel [21] (see also the discussion by Blanqui [15]).

58 In this work we address this unsatisfying state of affairs by giving an encoding of the
 59 cumulative calculus of constructions, featuring cumulativity with product covariance, that
 60 we show to satisfy the necessary metaproperties to be used in practice.

61 Contrary to the previous proposals, our encoding does not require non-left-linear rewrite
 62 rules, which not only are less efficient but also make confluence proofs much harder [33].
 63 Our proof of confluence then relies on a sophisticated combination of classical results and
 64 techniques [36, 39], and automated checkers developed by the rewriting community [30, 28, 38].

65 With the confluence of our encoding in hand, we proceed to show soundness. In order
 66 to fix the problem with the translation function made in previous attempts, we contribute
 67 an adaptation of the technique of Winterhalter et al [47] and Oury [40] in which the well-
 68 typedness of the translation is stated and proved in terms of an inverse translation function.
 69 The direct translation function can then be extracted from our constructive proof of soundness.

70 We finish by showing that our encoding satisfies a restricted form of conservativity, namely
 71 only for so-called *object terms*. We argue that, in the encoding, these are the only terms that
 72 one writes in practice, and therefore that this restricted result is sufficient.

73 Outline of the paper

74 We start in Sections 2 and 3 by recalling the definitions of DEDUKTI and of the variant of
 75 the calculus of constructions we consider. We then proceed in Section 4 to present the theory
 76 used in our encoding, and in Section 5 by proving its desirable properties—in particular its
 77 confluence. We define the translation function we use in Section 6, and in Sections 7 and 8
 78 we establish the soundness and conservativity of our encoding respectively. We finish by
 79 discussing related work in Section 9, before concluding in Section 10. The proofs not given
 80 in the main body of the text can be found in the technical appendix at the end of the paper.

81 Supplementary material

82 We provide an artifact containing supplementary data used in some of our proofs [20].

$$\begin{array}{c}
\text{EMPTYCTX} \\
\frac{}{\cdot \vdash} \\
\\
\text{EXTCTX} \\
\frac{\Gamma \vdash A : \mathbf{Type}}{\Gamma, x : A \vdash} \\
\\
\text{VAR} \\
\frac{x : A \in \Gamma \quad \Gamma \vdash}{\Gamma \vdash x : A} \\
\\
\text{CONS} \\
\frac{\Gamma \vdash}{c : A \in \Sigma \quad \Gamma \vdash c : A} \\
\\
\text{SORT} \\
\frac{\Gamma \vdash}{\Gamma \vdash \mathbf{Type} : \mathbf{Kind}} \\
\\
\text{CONV} \\
\frac{A \equiv B \quad \Gamma \vdash t : A \quad \Gamma \vdash B : s}{\Gamma \vdash t : B} \\
\\
\text{PI} \\
\frac{\Gamma \vdash A : \mathbf{Type} \quad \Gamma, x : A \vdash B : s}{\Gamma \vdash (x : A) \rightarrow B : s} \\
\\
\text{ABS} \\
\frac{\Gamma \vdash A : \mathbf{Type} \quad \Gamma, x : A \vdash B : s \quad \Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A. t : (x : A) \rightarrow B} \\
\\
\text{APP} \\
\frac{\Gamma \vdash t : (x : A) \rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash t u : B[u/x]}
\end{array}$$

■ **Figure 1** Typing rules of DEDUKTI

2 Dedukti

83

84 We assume an underlying set $c, d, \dots \in \mathcal{C}$ of *constants*, $x, y, z, \dots \in \mathcal{V}$ of *variables* and
85 $A, B, \mathbf{t}, u, \dots \in \mathcal{M}$ of *metavariables* equipped with an *arity* (a natural number). The *metaterms*
86 of DEDUKTI [27] are defined by the following grammar.

87
88

$$\hat{\Lambda}_{\text{dk}} \ni t, u, A, B, \dots ::= x \mid c \mid \mathbf{Type} \mid \mathbf{Kind} \mid (x : A) \rightarrow B \mid \lambda x : A. t \mid t u \mid \mathbf{t}\{t_1, \dots, t_{\text{arity}(\mathbf{t})}\}$$

89

90 A *metavariable application* is written $\mathbf{t}\{t_1, \dots, t_k\}$ when $\text{arity}(\mathbf{t}) = k$, or just \mathbf{t} when
91 $\text{arity}(\mathbf{t}) = 0$. The metaterms **Type** and **Kind** are called *sorts* and referred to by the letter s .
92 We write $(x : A) \rightarrow B$ for the *dependent function type*, and whenever x does not appear free
93 in B we write $A \rightarrow B$ instead. We define $\text{fv}(t)$ as the set of free variables of t and $\text{mv}(t)$ as
94 the set of metavariables of t . When no ambiguity can arise, we allow ourselves to also write
95 t, u, A, B for variables. We adopt the convention of writing constants names in **blue font**.

95

96 A *substitution* θ is a finite set of pairs t/x or $(x_1 \dots x_k. t)/\mathbf{t}$, where $k = \text{arity}(\mathbf{t})$. We
97 write $t[\theta]$ for the application of a substitution θ to a metaterm t . The main cases of its
98 definition are $x[\theta] = t$ when $t/x \in \theta$, and $\mathbf{t}\{u_1, \dots, u_k\}[\theta] = \mathbf{t}\{u_1[\theta]/x_1, \dots, u_k[\theta]/x_k\}$ when
99 $(x_1 \dots x_k. t)/\mathbf{t} \in \theta$ —see for instance Férey [27] for the complete definition. A *rewrite system* \mathcal{R}
100 is a set of *rewrite rules*, which are pairs of the form $t \mapsto u$ where t is of the form $c \ t_1 \dots t_k$
101 and $\text{fv}(t) = \text{fv}(u) = \emptyset$ and $\text{mv}(u) \subseteq \text{mv}(t)$ and all occurrences of metavariables in t are of
102 the form $\mathbf{t}\{x_1, \dots, x_k\}$ with $x_1 \dots x_k$ pairwise disjoint (known as the pattern condition [37]).
103 When convenient, a rule can be given a name α , in which case we write $t \xrightarrow{\alpha} u$.

103

104 Metavariables are useful in order to define the notion of rewrite rules. However, apart
105 from this they will have no use for us, and in particular typing will only be defined for
106 metaterms without metavariables. Because of this, we define the set of DEDUKTI *terms* Λ_{dk}
107 as the metaterms t satisfying $\text{mv}(t) = \emptyset$. Given that terms will be the main object of study,
108 from now on we adopt the convention that the letters t, u, A, B, \dots refer to terms, unless they
109 explicitly appear inside of a rewrite rule—for instance, as in $c \ t_1 \dots t_k \mapsto u$.

109

110 We write $\longrightarrow_{\mathcal{R}}$ for the closure under context and substitution of \mathcal{R} , and $\longrightarrow_{\beta\mathcal{R}}$ for
111 $\longrightarrow_{\beta} \cup \longrightarrow_{\mathcal{R}}$ where \longrightarrow_{β} is the usual β -reduction. We then write $\longrightarrow_{\beta\mathcal{R}}^*$ for its transitive
112 closure, and $\equiv_{\beta\mathcal{R}}$ for its reflexive-symmetric-transitive closure, usually called *conversion* or
113 *definitional equality*. Most of the time \mathcal{R} is clear from the context, allowing us to write just
114 \longrightarrow for $\longrightarrow_{\beta\mathcal{R}}$ and \equiv for $\equiv_{\beta\mathcal{R}}$. We then say that the underlying rewrite system is confluent
when, for all terms t, u, v , if $u \xleftarrow{*} t \longrightarrow^* v$ then $u \longrightarrow^* w \xleftarrow{*} v$ for some term w .

23:4 Impredicativity, Cumulativity and Product Covariance in DEDUKTI

$$\begin{array}{c}
\text{SUB} \\
\frac{n \leq m}{n \subseteq m} \\
\\
\text{EQ} \\
\frac{A \equiv B}{A \subseteq B} \\
\\
\text{TRANS} \\
\frac{A \subseteq B \quad B \subseteq C}{A \subseteq C} \\
\\
\text{PRODCOV} \\
\frac{A \subseteq B}{\Pi x : C. A \subseteq \Pi x : C. B} \\
\\
\text{EMPTYCTX} \\
\frac{}{\cdot \vdash_{\text{CC}}} \\
\\
\text{EXTCTX} \\
\frac{\Gamma \vdash_{\text{CC}} \quad \Gamma \vdash_{\text{CC}} A : n}{\Gamma, x : A \vdash_{\text{CC}}} \\
\\
\text{VAR} \\
(x : A) \in \Gamma \quad \frac{\Gamma \vdash_{\text{CC}}}{\Gamma \vdash_{\text{CC}} x : A} \\
\\
\text{SORT} \\
\frac{\Gamma \vdash_{\text{CC}}}{\Gamma \vdash_{\text{CC}} n : \mathfrak{A}(n)} \\
\\
\text{PI} \\
\frac{\Gamma \vdash_{\text{CC}} A : n \quad \Gamma, x : A \vdash_{\text{CC}} B : m}{\Gamma \vdash_{\text{CC}} \Pi x : A. B : \mathfrak{R}(n, m)} \\
\\
\text{LAM} \\
\frac{\Gamma \vdash_{\text{CC}} A : n \quad \Gamma, x : A \vdash_{\text{CC}} t : B}{\Gamma \vdash_{\text{CC}} \lambda x : A. t : \Pi x : A. B} \\
\\
\text{APP} \\
\frac{\Gamma \vdash_{\text{CC}} t : \Pi x : A. B \quad \Gamma \vdash_{\text{CC}} u : A}{\Gamma \vdash_{\text{CC}} t u : B[u/x]} \\
\\
\text{CONV} \\
A \subseteq B \quad \frac{\Gamma \vdash_{\text{CC}} t : A \quad \Gamma \vdash_{\text{CC}} B : n}{\Gamma \vdash_{\text{CC}} t : B}
\end{array}$$

■ **Figure 2** Typing rules for CC

115 A *context* Γ is a finite sequence of entries of the form $x : A$. A *signature* Σ is a (possibly
116 infinite) sequence of entries of the form $c : A$. One central notion in DEDUKTI is that of
117 *theory*, which is a pair $\mathbb{T} = (\Sigma_{\mathbb{T}}, \mathcal{R}_{\mathbb{T}})$ where $\Sigma_{\mathbb{T}}$ is a signature and all constants appearing in
118 $\mathcal{R}_{\mathbb{T}}$ are declared in $\Sigma_{\mathbb{T}}$. Theories are used in DEDUKTI to define the object logics in which
119 we work (for instance, predicate logic). Given a theory \mathbb{T} , the typing rules of DEDUKTI are
120 given in Figure 1, where the signature Σ and the conversion relation \equiv are the ones defined
121 by the theory \mathbb{T} . Whenever \mathbb{T} is not clear from the context, we write $\mathbb{T} \triangleright \Gamma \vdash t : A$.

122 A signature entry $c : A$ is valid in \mathbb{T} when $\mathbb{T} \triangleright \cdot \vdash A : s$ for some sort s . A theory \mathbb{T} is
123 said to be *well typed* when each entry $c : A \in \Sigma_{\mathbb{T}}$ is valid in (Σ', \mathcal{R}') , where Σ' is the prefix of
124 $\Sigma_{\mathbb{T}}$ preceding $c : A$, and \mathcal{R}' is the restriction of $\mathcal{R}_{\mathbb{T}}$ to rules only containing constants in Σ' .

125 3 The Cumulative Calculus of Constructions with Product Covariance

126 We recall the definition of the cumulative calculus of constructions with product covariance [35,
127 31]. It can be seen as the underlying *cumulative type system* [34, 10] of the COQ proof
128 assistant [42], omitting the sorts `Set` and `SProp`. Its syntax is given by the following grammar.

$$\boxed{\Lambda_{\text{CC}}} \ni \quad t, u, A, B ::= x \mid n \mid \Pi x : A. B \mid \lambda x : A. t \mid t u$$

131 Here we have made the choice of representing universes directly by a natural number n .
132 The universe that is commonly referred to as `Prop` then corresponds to 0, whereas `Typen`
133 corresponds to $n + 1$, allowing us to manipulate them in a more uniform way. The typing
134 rules are then given in Figure 2, and are parametrized by the following *axiom* and *rule*
135 functions, as they are known in the *pure type system* literature [9].

$$\begin{array}{ll}
136 \quad \mathfrak{A} : \mathbb{N} \rightarrow \mathbb{N} & \mathfrak{R} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} \\
137 \quad \mathfrak{A}(0) := 2 & \mathfrak{R}(n, 0) := 0 \\
138 \quad \mathfrak{A}(1 + n) := 2 + n & \mathfrak{R}(n, 1 + m) := \max\{n, 1 + m\}
\end{array}$$

140 ▶ **Remark 1.** We choose to follow the implementation of COQ in placing 0 (`Prop`) in the
141 universe 2 (`Type1`). Some presentations choose instead to place it in 1 (`Type0`) [35], a technical
142 change that would have no impact in the strategy developed in this paper.

143 Compared with type systems that do not feature cumulativity, the conversion rule for CC
 144 does not only allow to exchange two types A and B when they are convertible, but also to
 145 coerce a term from type A to B when the latter is a subtype of the former. This subtyping
 146 relation, written $A \subseteq B$, is defined in the base case as $A \subseteq B$ when $A \equiv B$, or $n \subseteq m$ when
 147 $n \leq m$. The second rule allows us for instance to coerce a type $\Gamma \vdash A : 0$ to $\Gamma \vdash A : 1$. Then,
 148 what one calls *product covariance* is the rule allowing to deduce $\Pi x : C.A \subseteq \Pi x : C.B$ from
 149 $A \subseteq B$, which lets us for instance to coerce a function $\Gamma \vdash_{\text{CC}} f : \text{Nat} \rightarrow 0$ to $\Gamma \vdash_{\text{CC}} f : \text{Nat} \rightarrow 1$.

150 4 Introducing the theory \mathbb{T}_{CC}

151 We now introduce the DEDUKTI theory \mathbb{T}_{CC} we will use in our encoding. We build it
 152 incrementally in order to motivate as best as possible the choices we have made.

153 Our first step is declaring a type \mathfrak{S} along with constants 0 and S for zero and successor,
 154 allowing us to represent the CC sort n by the DEDUKTI term $S^n 0$ —which from now on we
 155 write \underline{n} . We then define many auxiliary constants that will be useful later, such as addition
 156 $+$, truncated predecessor P , and also constants \mathfrak{A} and \mathfrak{R} to represent the functions \mathfrak{A} and
 157 \mathfrak{R} from the definition of CC. We declare the associated rewrite rules so that they have the
 158 expected computational behavior, such as $\underline{n} + \underline{m} \longrightarrow^* \underline{n + m}$, $\underline{n} \vee \underline{m} \longrightarrow^* \underline{\max\{n, m\}}$, etc.

$$\begin{array}{llll}
 \mathfrak{S} : \mathbf{Type} & \mathfrak{A} : \mathfrak{S} \rightarrow \mathfrak{S} & P : \mathfrak{S} \rightarrow \mathfrak{S} & - : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} \quad (\text{infix}) \\
 0 : \mathfrak{S} & \mathfrak{A} 0 \mapsto S(S 0) & P 0 \mapsto 0 & 1_1 - 0 \mapsto 1_1 \\
 S : \mathfrak{S} \rightarrow \mathfrak{S} & \mathfrak{A}(S 1) \mapsto S(S 1) & P(S 1) \mapsto 1 & 1_1 - (S 1_2) \mapsto (P 1_1) - 1_2 \\
 \\
 + : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} & (\text{infix}) & \vee : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} & (\text{infix}) & \mathfrak{R} : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} \\
 0 + 1_2 \mapsto 1_2 & & 0 \vee 1_2 \mapsto 1_2 & & \mathfrak{R} 1_1 0 \mapsto 0 \\
 1_1 + 0 \mapsto 1_1 & & 1_1 \vee 0 \mapsto 1_1 & & \mathfrak{R} 1_1 (S 1_2) \mapsto 1_1 \vee (S 1_2) \\
 (S 1_1) + 1_2 \mapsto S(1_1 + 1_2) & & (S 1_1) \vee (S 1_2) \mapsto S(1_1 \vee 1_2) & & \\
 1_1 + (S 1_2) \mapsto S(1_1 + 1_2) & & & &
 \end{array}$$

159 Using \mathfrak{S} we can then encode the universes of CC. This is done by declaring a constant U ,
 160 such that the inhabitants of $U_{\underline{n}}$ can then be thought of as codes for the types of CC in n .
 161 The decoding function El then maps each such code to the DEDUKTI type of its elements.

$$\begin{array}{ll}
 U : (l : \mathfrak{S}) \rightarrow \mathbf{Type} & (\text{written } U_l) \quad \quad \quad El : (l : \mathfrak{S}) \rightarrow U_l \rightarrow \mathbf{Type} \quad (\text{written } El_l)
 \end{array}$$

162 Next we add constants to represent the codes inhabiting such universes. Because in CC
 163 each universe n inhabits $\mathfrak{A}(n)$, we add a constant u mapping each $l : \mathfrak{S}$ to its code in $U_{(\mathfrak{A} l)}$.
 164 An associated rewrite rule then ensures that u_l decodes to the type U_l as expected.

$$\begin{array}{ll}
 u : (l : \mathfrak{S}) \rightarrow U_{(\mathfrak{A} l)} & (\text{written } u_l) \quad \quad \quad El_{(_)} u_l \mapsto U_l
 \end{array}$$

165 A similar story happens for the function type: we add a constant π mapping a code
 166 $a : U_{l_a}$ and a family of codes $b : El_{l_a} a \rightarrow U_{l_b}$ to a code in $U_{(\mathfrak{R} l_a l_b)}$, so that if a represents
 167 A and b represents B , then the result represents the CC type $\Pi x : A.B$. However, for reasons
 168 that will become clear later, our constant also allows us to decompose the sorts l_a and l_b
 169 into a common factor l_0 to which we apply offsets l_1 and l_2 . In order to equate different
 170 decompositions of l_a and l_b , we also add a rewrite rule which removes two successors of l_1
 171 and l_2 and compensates it by adding one in l_0 . Finally, we add a rewrite rule defining the
 172 elements of $\pi_{l_1, l_2}^{l_0} a \lambda x.b$ as the DEDUKTI functions from the elements of a to the ones of b .

23:6 Impredicativity, Cumulativity and Product Covariance in Dedukti

$$\begin{aligned}
 \pi &: (l_0 \ l_1 \ l_2 : \mathfrak{S}) \rightarrow (A : \mathbf{U}_{(l_0+l_1)}) \\
 &\rightarrow (B : \mathbf{El}_{(l_0+l_1)} A \rightarrow \mathbf{U}_{(l_0+l_2)}) \rightarrow \mathbf{U}_{(\mathfrak{R}(l_0+l_1)(l_0+l_2))} \quad (\text{written } \pi_{l_1, l_2}^{l_0}) \\
 \pi_{(\mathfrak{S} \ 1_1), (\mathfrak{S} \ 1_2)}^{1_0} A \ B &\xrightarrow{\pi_{\mathfrak{S}}} \pi_{1_1, 1_2}^{(\mathfrak{S} \ 1_0)} A \ B \\
 \mathbf{El}_{(_)} (\pi_{1_1, 1_2}^{1_0} A \ \lambda x : \mathbf{C.B}\{x\}) &\mapsto (x : \mathbf{El}_{(1_0+1_1)} A) \rightarrow \mathbf{El}_{(1_0+1_2)} \mathbf{B}\{x\}
 \end{aligned}$$

173 The theory given until this point is a representation of CC *without* cumulativity, and
 174 straightforwardly applies well-known techniques from previous DEDUKTI encodings [19, 17].
 175 The interesting part is for the encoding of cumulativity. The main insight of our proposal
 176 comes from the following simple result regarding the relation \subseteq . In the following, given a
 177 context $\Delta = x_1 : B_1 \dots x_k : B_k$, let us write $\Delta \Rightarrow A$ for the CC term $\Pi x_1 : B_1 \dots x_k : B_k. A$.

178 ▶ **Lemma 2** (Case analysis of \subseteq). *If $A \subseteq B$ then either $A \equiv B$ or $A \longrightarrow^* \Delta \Rightarrow n$ and*
 179 *$B \longrightarrow^* \Delta \Rightarrow m$ for some context Δ and natural numbers n, m with $n \leq m$.*

180 Therefore, in order to simulate CC's cumulativity it suffices to add a *lift* \uparrow allowing the
 181 coercion of terms from a type $\Delta \Rightarrow n$ to $\Delta \Rightarrow n + 1$. However, to be able to state the type of
 182 \uparrow we first need to have an internal representation for types of the form $\Delta \Rightarrow n$ in DEDUKTI.
 183 We do this by first defining a type for *telescopes* whose canonical elements are either the
 184 empty telescope \blacklozenge , or the extension $A \blacktriangleleft \lambda x. D$ of a telescope D with a code A in universe \mathbf{U}_l .
 185 We can then define a function \Rightarrow that computes a DEDUKTI type corresponding to $\Delta \Rightarrow n$.

$$\begin{aligned}
 \mathbf{Tele} : \mathbf{Type} & & \Rightarrow : \mathbf{Tele} \rightarrow \mathfrak{S} \rightarrow \mathbf{Type} & \text{ (infix)} \\
 \blacklozenge : \mathbf{Tele} & & \blacklozenge \Rightarrow 1_1 \xrightarrow{\Rightarrow} \mathbf{U}_{1_1} \\
 \blacktriangleleft : (l : \mathfrak{S}) \rightarrow (A : \mathbf{U}_l) \rightarrow (\mathbf{El}_l A \rightarrow \mathbf{Tele}) & & (A \ 1_2 \blacktriangleleft \lambda x : _ . D\{x\}) \Rightarrow 1_1 \xrightarrow{\Rightarrow} (x : \mathbf{El}_{1_2} A) \rightarrow D\{x\} \Rightarrow 1_1 \\
 & & \rightarrow \mathbf{Tele} & \text{ (infix, written } \blacktriangleleft \text{)}
 \end{aligned}$$

186 With these definitions in place we can finally give the definition of \uparrow .¹

$$\uparrow : (l : \mathfrak{S}) \rightarrow (D : \mathbf{Tele}) \rightarrow (D \Rightarrow l) \rightarrow (D \Rightarrow (\mathfrak{S} \ l)) \quad (\text{written } \uparrow_l)$$

187 Because in CC the applications of cumulativity are silent, the main challenge in the
 188 encoding is to ensure that different DEDUKTI representations of the same CC term are
 189 convertible. The pioneering work of Assaf [4] first identified that, in a setting without
 190 product covariance, it suffices to add the following *full reflection* equations—here and in the
 191 rest of the article we write $\uparrow_{\underline{n}}^{\underline{m}} D \ t$ as a notation for $\uparrow_{\underline{m-1}} D (\dots (\uparrow_{\underline{n}} D \ t) \dots)$ when $n \leq m$.

$$\begin{aligned}
 \pi_{\underline{1+n}, \underline{m}}^{\underline{0}} (\uparrow_{\underline{n}} \blacklozenge \ a) (\lambda x. b) &\equiv \uparrow_{\frac{\mathfrak{R}(1+n, m)}{\mathfrak{R}(n, m)}} \blacklozenge (\pi_{\underline{n}, \underline{m}}^{\underline{0}} a (\lambda x. b)) \\
 \pi_{\underline{n}, \underline{1+m}}^{\underline{0}} a (\lambda x. \uparrow_{\underline{m}} \blacklozenge \ b) &\equiv \uparrow_{\frac{\mathfrak{R}(n, 1+m)}{\mathfrak{R}(n, m)}} \blacklozenge (\pi_{\underline{n}, \underline{m}}^{\underline{0}} a (\lambda x. b))
 \end{aligned}$$

195 The main difficulty in implementing these as rewrite rules is that the multistep lift $\uparrow_{\underline{n}}^{\underline{m}}$ is
 196 just a notation which computes the correct number of lifts \uparrow to be inserted only for a given
 197 concrete choice of n and m . For instance, if $n > m > 0$ in the second equation then no lifts
 198 should be inserted in the right hand side, whereas if $n > m = 0$ then we must insert $n - 1$ lifts.

¹ Note that our lift is *single-step*, in contrast with some previous encodings [5, 45, 27] which employed a *multi-step* lift, taking a type $A : \mathbf{U}_{l_1}$ to $\uparrow_{l_1}^{l_2} \blacklozenge \ t : \mathbf{U}_{l_2}$. The avoidance of the multi-step lift is essential in order to prevent its associated non-left-linear rules, such as $\uparrow_1^1 D \ t \mapsto t$.

199 If only we could have more information about n and m when applying the rule, we would
 200 be able to calculate the correct amount of lifts. Thankfully, because the sorts of a and b can
 201 be decomposed with the rule π_S , we know that for any $\pi_{n_1, n_2}^{n_0} a \lambda x. b$ in normal form we must
 202 have either $n_1 = 0$ or $n_2 = 0$. We can then proceed with a disjunction of cases, where in
 203 each situation we have enough information to apply the right number of lifts.

$$\begin{array}{ll}
 \pi_{(S\ 1),0}^0 (\uparrow_{-} \blacklozenge A) B \xrightarrow{\uparrow_{\pi}^1} \pi_{1,0}^0 A B & \uparrow : (l : \mathfrak{S}) \rightarrow (A : \mathbf{U}_0) \rightarrow \mathbf{U}_l \quad (\text{written } \uparrow_l) \\
 \pi_{0,1_2}^{(S\ 1_1)} (\uparrow_{-} \blacklozenge A) B \xrightarrow{\uparrow_{\pi}^2} \pi_{0,(S\ 1_2)}^{1_1} A B & \uparrow_0 A \mapsto A \\
 \pi_{(S\ 1_2),0}^{(S\ 1_1)} (\uparrow_{-} \blacklozenge A) B \xrightarrow{\uparrow_{\pi}^3} \uparrow_{(S\ (1_1+1_2))} \blacklozenge (\pi_{1_2,0}^{(S\ 1_1)} A B) & \uparrow_{(S\ 1)} A \mapsto \uparrow_1 \blacklozenge (\uparrow_1 A) \\
 \\
 \pi_{1_2,0}^{(S\ (S\ 1_1))} A (\lambda x : C. \uparrow_{-} \blacklozenge B\{x\}) \xrightarrow{\uparrow_{\pi}^4} \pi_{(S\ 1_2),0}^{(S\ 1_1)} A (\lambda x : C.B\{x\}) & \\
 \pi_{0,(S\ 1_2)}^{1_1} A (\lambda x : C. \uparrow_{-} \blacklozenge B\{x\}) \xrightarrow{\uparrow_{\pi}^5} \uparrow_{(1_1+1_2)} \blacklozenge (\pi_{0,1_2}^{1_1} A (\lambda x : C.B\{x\})) & \\
 \pi_{1,0}^{(S\ 0)} A (\lambda x : C. \uparrow_{-} \blacklozenge B\{x\}) \xrightarrow{\uparrow_{\pi}^6} \uparrow_{(S\ 1)} (\pi_{(S\ 1),0}^0 A (\lambda x : C.B\{x\})) &
 \end{array}$$

204 Note that in order to state the last rule we also define an auxiliary constant \uparrow which
 205 given a sort l , lifts a type from \mathbf{U}_0 to \mathbf{U}_l . The following proposition then ensures that we
 206 have correctly implemented Assaf's full reflection equations.

207 **► Proposition 3** (Simulation of Assaf's full reflection rules). *We have the following conversions.*

$$208 \quad \pi_{1+n,m}^0 (\uparrow_l \blacklozenge a) (\lambda x : C.b) \equiv \uparrow_{\mathfrak{R}(n,m)}^{\mathfrak{R}(1+n,m)} \blacklozenge (\pi_{n,m}^0 a (\lambda x : C.b)) \quad (1)$$

$$209 \quad \pi_{n,1+m}^0 a (\lambda x : C. \uparrow_l \blacklozenge b) \equiv \uparrow_{\mathfrak{R}(n,m)}^{\mathfrak{R}(n,1+m)} \blacklozenge (\pi_{n,m}^0 a (\lambda x : C.b)) \quad (2)$$

210 **Proof.** By a disjunction of cases in which each case corresponds to one of the rules \uparrow_{π}^i . ◀

212 **► Remark 4.** We note that these rules are also very similar to the ones identified by Assaf et
 213 al [8]. However they also differ in a crucial way by avoiding the use of non-left-linearity and
 214 matching modulo ACU, which render confluence proofs much harder and are less efficient.

215 The rules given until now would ensure the uniqueness of codes for a version of CC with
 216 “simple” cumulativity. However, in a setting with product covariance we also need to ensure
 217 that \uparrow properly commutes with abstraction and application. We therefore add the following
 218 two rules, which are variants of similar equations first identified by Thiré [45] and Férey [27].

$$\begin{array}{l}
 \uparrow_1 (_ _ \blacktriangleleft \lambda x : _ . D\{x\}) \lambda x : A. t\{x\} \xrightarrow{\uparrow_{\lambda}} \lambda x : A. \uparrow_1 D\{x\} t\{x\} \\
 \uparrow_1 (_ _ \blacktriangleleft \lambda x : _ . D\{x\}) t u \xrightarrow{\uparrow_{\otimes}} \uparrow_1 D\{u\} (t u)
 \end{array}$$

219 We now have almost finished presenting the theory \mathbb{T}_{cc} . The final step is adding the
 220 following rule explaining the relationship between the elements of $\uparrow_l \blacklozenge A$ and the ones of A ,
 221 which as expected should be the same. Here we have purposely avoided the expected rule
 222 $\text{El}_{(S\ 1)} (\uparrow_{(_)} \blacklozenge A) \mapsto \text{El}_1 A$ used in some previous proposals [5, 45]. This subtle difference is
 223 essential in order to allow the critical pairs between \uparrow_{π}^i and El_{π} to close. We add a similar
 224 rule for \uparrow , but once again we annotate El with $l_2 - l_1$ instead of 0 in order to ensure that
 225 critical pairs all close. Finally, we need a last rule similar to El_{\uparrow} ensuring the uniqueness of
 226 telescope representations, which will be key when proving the injectivity of \Rightarrow .

$$\text{El}_1 (\uparrow_{-} \blacklozenge A) \xrightarrow{\uparrow_{\text{El}}} \text{El}_{(P\ 1)} A \quad \text{El}_{l_2} (\uparrow_{l_1} A) \xrightarrow{\uparrow_{\text{El}}} \text{El}_{(l_2 - l_1)} A \quad (\uparrow_{-} \blacklozenge A)_{1 \blacktriangleleft D} \xrightarrow{\uparrow_{\blacktriangleleft}} A_{(P\ 1) \blacktriangleleft D}$$

227 **5** Basic properties of \mathbb{T}_{cc}

228 With the definition of the theory \mathbb{T}_{cc} in place, we now show that it satisfies the basic properties
 229 one expects, which will be essential for proving soundness and conservativity later. The first
 230 of them is the fact the the theory \mathbb{T}_{cc} is well-typed, in the sense defined in Section 2.

231 **► Proposition 5** (Well-typedness of \mathbb{T}_{cc}). *The theory \mathbb{T}_{cc} is well typed.*

232 **Proof.** Checked automatically with LAMBDAPI—see the artifact [20] for more details. ◀

233 5.1 Confluence

234 Unlike all previous proposals, our theory \mathbb{T}_{cc} only makes use of left-linear rules. By preventing
 235 the use of non-left-linearity, which interacts very badly with higher-order rewriting, we have
 236 made a first step for proving confluence. Yet, confluence still does not come for free. In
 237 order to show it, we split $\beta\mathcal{R}_{\text{cc}}$ into subsystems $\beta\mathcal{R}_1$ and \mathcal{R}_2 , allowing us to apply different
 238 techniques for showing their confluence. Note that the union $\beta\mathcal{R}_1 \cup \mathcal{R}_2$ is not disjoint: the
 239 rule \uparrow_{El} , needed for closing critical pairs in both subsystems, is shared between them.

$$240 \quad \mathcal{R}_1 := \{\uparrow_{\text{@}}, \uparrow_{\lambda}, \uparrow_{\blacktriangleleft}, \Rightarrow_{\blacktriangleleft}, \Rightarrow_{\blacktriangleright}, \uparrow_{\text{El}}\} \quad \mathcal{R}_2 := \mathcal{R}_{\text{cc}} \setminus \{\uparrow_{\text{@}}, \uparrow_{\lambda}, \uparrow_{\blacktriangleleft}, \Rightarrow_{\blacktriangleleft}, \Rightarrow_{\blacktriangleright}\}$$

242 The hardest part of our proof is showing the confluence of $\beta\mathcal{R}_1$, for two main reasons.
 243 First, even though all critical pairs of $\beta\mathcal{R}_1$ close (as shown in Figure 4), because the β rule
 244 is non-normalizing on untyped terms, we cannot apply Newman’s Lemma to reduce proving
 245 confluence to local confluence. Second, because the critical pairs are neither *trivial* [39]
 246 nor *development closed* [46], we cannot apply the classical criteria that avoid the use of
 247 termination. Thankfully, it turns out that we can still employ the well-known technique of
 248 showing that *orthogonal rewriting* with $\beta\mathcal{R}_1$ satisfies the diamond property—the proof can
 249 be found in the appendix. Confluence of $\beta\mathcal{R}_1$ then follows from this as a simple corollary.

250 **► Corollary 6.** *$\beta\mathcal{R}_1$ is confluent.*

251 **► Remark 7.** Alternatively, one can show the confluence of $\beta\mathcal{R}_1$ by applying a recent criterion
 252 by Dowek, Férey, Jouannaud and Liu [22, Theorem 38]. However, the proof we give is more
 253 elementary as it relies neither on orthogonal critical pairs nor on decreasing diagrams, and
 254 therefore we believe that it is accessible to a wider audience.

255 We can then move to the proof of confluence of \mathcal{R}_2 , which relies on termination.

256 **► Lemma 8.** *\mathcal{R}_2 is strongly normalizing.*

257 **Proof.** We translate from \mathcal{R}_2 into the first-order rewrite system $\hat{\mathcal{R}}_2$ obtained by forgetting
 258 about binders: $\lambda x : A.t$ is translated into $\hat{\lambda} A' t'$ and $\Pi x : A.B$ is translated into $\hat{\Pi} A' B'$,
 259 where A', B', t' are the translations of A, B, t . For instance, the rule \uparrow_{π}^4 is translated into
 260 the rule $\pi_{1_2,0}^{(S(S 1_1))} A (\hat{\lambda} C (\uparrow_{\blacktriangleleft} \blacktriangleright B)) \mapsto \pi_{(S 1_2),0}^{(S 1_1)} A (\hat{\lambda} C B)$. We can easily show that this
 261 interpretation preserves reduction sequences, therefore we reduce SN of \mathcal{R}_2 to the one of $\hat{\mathcal{R}}_2$.
 262 The latter can be shown with the use of the first-order termination checker AProVE [1, 28], and
 263 the proof can be verified by the formally certified tool CeTA [2, 43]—see the artifact [20]. ◀

264 **► Proposition 9.** *\mathcal{R}_2 is confluent.*

265 **Proof.** We use the tools CSI^h [3, 38] and SOL [30] to verify that all critical pairs of \mathcal{R}_2 are
 266 joinable—see the artifact [20] for details—so by Mayr and Nipkow’s critical pair criterion
 267 [36, Theorem 4.7] we conclude that \mathcal{R}_2 is locally confluent. Together with Lemma 8, this
 268 gives the confluence of \mathcal{R}_2 by applying Newman’s Lemma. ◀

269 Putting everything together, we obtain the confluence of $\beta\mathcal{R}_{cc}$.

270 ▶ **Theorem 10.** $\beta\mathcal{R}_{cc}$ is confluent.

271 **Proof.** By Corollary 6 and Proposition 9 we have the confluence of $\beta\mathcal{R}_1$ and \mathcal{R}_2 , and moreover
 272 the rewrite systems are left-linear and there are no critical pairs between them. Therefore, we
 273 conclude the confluence of their union by applying Van Oostrom and Raamsdonk's orthogonal
 274 combinations criterion [39, Theorem 3.13]. ◀

275 We obtain the following useful corollary, which we implicitly use in the rest of the article.

276 ▶ **Corollary 11** (Injectivity of undefined symbols). *If c is a constant that does not appear in
 277 the head of a rewrite rule, then $c t_1 \dots t_k \equiv c u_1 \dots u_k$ implies $t_i \equiv u_i$ for $i = 1..k$.*

278 5.2 Subject reduction

279 We start with subject reduction for β . Because we have already shown confluence of $\beta\mathcal{R}_{cc}$,
 280 we obtain directly the injectivity of function types: if $(x : A) \rightarrow B \equiv (x : A') \rightarrow B'$ then
 281 $A \equiv A'$ and $B \equiv B'$. This is sufficient in order to ensure that β satisfies subject reduction.

282 ▶ **Proposition 12** (SR_β). *If $\Gamma \vdash t : A$ and $t \rightarrow_\beta t'$ then $\Gamma \vdash t' : A$.*

283 **Proof.** Follows from the injectivity of function types [13, Lemma 31]. ◀

284 Moving to subject reduction for \mathcal{R}_{cc} , the first point we realize is that this property does
 285 not hold unconditionally. For instance, the rule

$$286 \pi_{0,1_2}^{(S \ 1_1)} (\uparrow _ \blacklozenge A) B \mapsto \pi_{0,(S \ 1_2)}^{1_1} A B$$

287 only preserves typing if $S (1_1[\theta] \vee (1_1[\theta] + 1_2[\theta])) \equiv 1_1[\theta] \vee S (1_1[\theta] + 1_2[\theta])$, yet both sides
 288 are already in normal form. Nevertheless, this is actually not a problem because whenever 1_1
 289 and 1_2 are substituted by terms of the form \underline{n} for some $n \in \mathbb{N}$ then we see that the equation
 290 holds. Starting from this insight, we now show that subject reduction holds in a restricted
 291 form, which turns out to be sufficient for our needs.

292 We say that a term is *guarded* when all occurrences of \uparrow are of the form $\uparrow_{\underline{n}}$ and all
 293 occurrences of π are of the form $\pi_{\underline{n}_1, \underline{n}_2}^{n_0}$ for some $n, n_0, n_1, n_2 \in \mathbb{N}$. The set of guarded terms
 294 satisfies the following basic stability properties.

295 ▶ **Proposition 13** (Stability of guarded terms under substitution and reduction).

- 296 1. *If t, u are guarded then $t[u/x]$ is guarded.*
- 297 2. *If t is guarded and $t \rightarrow t'$ then t' is guarded.*

298 We can now show that \mathcal{R}_{cc} satisfies subject reduction for guarded terms.

299 ▶ **Proposition 14** ($\text{SR}_{\mathcal{R}_{cc}}$). *If t is guarded and $\Gamma \vdash t : A$ and $t \rightarrow_{\mathcal{R}_{cc}} t'$ then $\Gamma \vdash t' : A$.*

300 **Proof.** We use Lambdapi to automatically verify that the rules preserve typing (the cor-
 301 rectness of this verification relies on the confluence of the rewrite system [41, 14], which we
 302 have by Theorem 10). The verification succeeds for all rules $l \mapsto r \in \mathcal{R}_{cc}$, except for those
 303 which do not preserve typing unconditionally. For these cases, Lambdapi reports conversion
 304 constraints on the substitution θ under which $\Gamma \vdash l[\theta] : A$ implies $\Gamma \vdash r[\theta] : A$.

- 305 1. Case \uparrow_{EI} . Preserves typing if $1_2[\theta] - 1_2[\theta] \equiv 0$. But by inversion of typing of the left-hand
 306 side we also get $1_1[\theta] \equiv 1_2[\theta]$, so the rule preserves typing whenever $1_1[\theta] - 1_1[\theta] \equiv 0$.

23:10 Impredicativity, Cumulativity and Product Covariance in Dedukti

- 307 2. Case \uparrow_{π}^2 . Preserves typing if $S (1_1[\theta] \vee (1_1[\theta] + 1_2[\theta])) \equiv 1_1[\theta] \vee S (1_1[\theta] + 1_2[\theta])$.
 308 3. Case \uparrow_{π}^3 . Preserves typing if $(1_1[\theta] + 1_2[\theta]) \vee 1_1[\theta] \equiv 1_1[\theta] + 1_2[\theta]$ and
 309 $S (1_1[\theta] + 1_2[\theta]) \vee 1_1[\theta] \equiv S (1_1[\theta] + 1_2[\theta])$.
 310 4. Case \uparrow_{π}^4 . Preserves typing if $S (1_1[\theta] + 1_2[\theta]) \vee 1_1[\theta] \equiv S ((1_1[\theta] + 1_2[\theta]) \vee 1_1[\theta])$.
 311 5. Case \uparrow_{π}^5 . Preserves typing if $1_1[\theta] \vee S (1_1[\theta] + 1_2[\theta]) \equiv S (1_1[\theta] + 1_2[\theta])$ and
 312 $\mathfrak{R} 1_1[\theta] (1_1[\theta] + 1_2[\theta]) \equiv 1_1[\theta] + 1_2[\theta]$.

313 Because t is guarded, it follows that $1_1[\theta]$ is a concrete sort in case 1, and both $1_1[\theta]$
 314 and $1_2[\theta]$ are concrete sorts in the other cases, so the result follows from the fact that these
 315 equations all hold for natural numbers. \blacktriangleleft

316 \blacktriangleright **Corollary 15** ($SR_{\beta\mathcal{R}_{CC}}$). *If t is guarded and $\Gamma \vdash t : A$ and $t \longrightarrow^* t'$ then $\Gamma \vdash t' : A$.*

317 \blacktriangleright **Remark 16.** Corollary 15 guarantees that the usual type inference algorithm for Dedukti [41]
 318 is sound when Γ and t are guarded. Indeed, by inspection on its definition, if the inputs Γ
 319 and t are guarded then only guarded terms are ever reduced.

6 The translation function

321 Defining a DEDUKTI encoding usually requires specifying a *translation function* from the
 322 syntax of the source system to the one of the framework. However, whereas cumulativity
 323 is *implicit* in CC, in DEDUKTI it is made *explicit* by the use of a lift (\uparrow). Therefore, when
 324 translating a CC term, the translation function needs to figure out when to insert such lifts,
 325 even though the initial term contains no information about cumulativity. To handle this, a
 326 first idea could be to define this function only for well-typed CC terms and use typing to
 327 retrieve the missing information. However, it is not clear how to define such a function in a
 328 unique and well-founded way—see the discussion in Section 9 for a detailed discussion.

329 To solve this problem, we adapt the approach of Winterhalter et al [47] of relying instead
 330 on an *inverse translation function* $| - |$, defined from a subset of the syntax of the framework
 331 to the syntax of CC. Because the syntax of DEDUKTI is more explicit than the one of CC,
 332 this function can be straightforwardly defined by structural induction. Then, we can use it
 333 to state and prove soundness and conservativity. Finally, the *direct* translation function can
 334 then be recovered as the underlying algorithm of our constructive proof of soundness.

335 We start by carving out a subset of DEDUKTI's syntax over which we define $| - |$. These
 336 are the *object terms* and *object contexts*, defined by the following grammars, and where n, m
 337 ranges over natural numbers and G ranges over arbitrary guarded terms.

$$338 \quad \boxed{\Lambda_o} \ni \quad t, u, A, B ::= x \mid \lambda x : \mathbf{El}_n A. t \mid \mathbf{u}_n \mid \pi_{n,m}^0 A \lambda x : G. B \mid \uparrow_n G t \mid t u$$

$$339 \quad \boxed{\text{Ctx}_o} \ni \quad \Gamma ::= \cdot \mid \Gamma, x : \mathbf{El}_n A$$

341 The *inverse translation function* can then be defined by structural induction over object
 342 terms and contexts, by the following clauses.

$$343 \quad | - | : \Lambda_o \rightarrow \Lambda_{CC} \qquad \| - \| : \text{Ctx}_o \rightarrow \text{Ctx}_{CC}$$

$$344 \quad |x| := x \qquad \| \cdot \| := \cdot$$

$$345 \quad |\mathbf{u}_n| := n \qquad \|\Gamma, x : \mathbf{El}_n A\| := \|\Gamma\|, x : |A|$$

$$346 \quad |\lambda x : \mathbf{El}_n A. t| := \lambda x : |A|. |t|$$

$$347 \quad |\pi_{n,m}^0 A (\lambda x : G. B)| := \Pi x : |A|. |B|$$

$$348 \quad |\uparrow_n G t| := |t|$$

$$349 \quad |t u| := |t| |u| \qquad (t \text{ u not of previous forms})$$

Crucially, object terms are all guarded, ensuring that whenever they are well typed then their reducts also are. In addition, object terms are stable under substitution, which moreover commutes with $|-|$, two basic properties that will be essential to our proofs.

► **Proposition 17** (Basic properties of Λ_o and $|-|$).

1. If $t \in \Lambda_o$ then t is guarded.
2. If $t, u \in \Lambda_o$ then $t[u/x] \in \Lambda_o$ and $|t|[|u/x|] = |t[u/x]|$.

7 Soundness

Our proof of soundness requires multiple intermediate steps. The first of them is showing the injectivity modulo lifting of **El** (Proposition 34) and the injectivity of \Rightarrow (Proposition 35), two technical results that will be essential for the subsequent parts. However, for space reasons, we give these proofs in the appendix.

Then, we can move to the proof of *coherence*, the central auxiliary result needed for soundness, ensuring that any two different **DEDUKTI** representations of the same **CC** term must be convertible. The actual statement of the theorem is however a bit more intricate.

► **Theorem 18** (Coherence). *Let $t_1, t_2 \in \Lambda_o$ with $\Gamma \vdash t_1 : A_1$ and $\Gamma \vdash t_2 : A_2$. If $|t_1| = |t_2|$ then at least one of the following holds:*

- (1) $t_1 \equiv t_2$
- (2) $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D t_2 : D \Rightarrow \underline{m}$ and $t_1 \equiv \uparrow_{\underline{n}}^{\underline{m}} D t_2$ for some D guarded
- (3) $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D t_1 : D \Rightarrow \underline{m}$ and $t_2 \equiv \uparrow_{\underline{n}}^{\underline{m}} D t_1$ for some D guarded

Proof. The proof is by induction on t_1 and t_2 , following the definition of $|-|$. We first treat the cases in which t_1 or t_2 is of the form $\uparrow_{\underline{n}} D u$. Then, for the other cases the definition of $|-|$ imposes that t_1 and t_2 have the same head structure, and therefore we only consider t_1 and t_2 of the same form.

We illustrate the case where $t_1 = u_1 v_1$ and $t_2 = u_2 v_2$. By inversion we have $\Gamma \vdash u_i : (x : A_i) \rightarrow B_i$ and $\Gamma \vdash v_i : A_i$. By the i.h. applied to u_1 and u_2 , we have three cases to consider:

- (a) $u_1 \equiv u_2$. We thus get $A_1 \equiv A_2$ and $B_1 \equiv B_2$. Looking at the induction hypothesis on v_1 and v_2 , in all cases we must have $v_1 \equiv v_2$. Indeed, if we are in cases (2) or (3) then we get $A_1 \equiv D \Rightarrow \underline{p}$ and $A_2 \equiv D \Rightarrow \underline{q}$, but together with $A_1 \equiv A_2$ this implies $p = q$, meaning that no lifts are inserted between v_1 and v_2 . We thus conclude that $t_1 \equiv t_2$.
- (b) $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D u_2 : D \Rightarrow \underline{m}$ and $u_1 \equiv \uparrow_{\underline{n}}^{\underline{m}} D u_2$. Now, $(x : A_1) \rightarrow B_1 \equiv D \Rightarrow \underline{m}$, so by Lemma 36 we have $D \longrightarrow^* a_l \blacktriangleleft \lambda x : C.D'$ with **El**_l $a \equiv A_1$ and $B_1 \equiv D' \Rightarrow \underline{m}$. Moreover, we also get that **El**_l $a \equiv A_2$ and $B_2 \equiv D' \Rightarrow \underline{n}$. We are again in a situation where v_1 and v_2 share a type, so by the same arguments as in case (a) the i.h. gives $v_1 \equiv v_2$. Therefore,

$$t_1 = u_1 v_1 \equiv (\uparrow_{\underline{n}}^{\underline{m}} (a_l \blacktriangleleft \lambda x : C.D') u_2) v_2 \equiv \uparrow_{\underline{n}}^{\underline{m}} D'[v_2/x] (u_2 v_2) = \uparrow_{\underline{n}}^{\underline{m}} D'[v_2/x] t_2$$

For typing, we have $\Gamma \vdash t_2 : B_2[v_2/x]$ so by conversion we have $\Gamma \vdash t_2 : D'[v_2/x] \Rightarrow \underline{n}$ and thus $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D'[v_2/x] t_2 : D'[v_2/x] \Rightarrow \underline{m}$.

- (c) $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D u_1 : D \Rightarrow \underline{m}$ and $u_2 \equiv \uparrow_{\underline{n}}^{\underline{m}} D u_1$. Symmetric to case (b). ◀

With coherence in hand, we can show that the conversion relation of **CC** can be reflected by the inverse translation function into the framework. As an intermediate lemma, we first need to show that individual reduction steps of **CC** can be simulated in **DEDUKTI**.

23:12 Impredicativity, Cumulativity and Product Covariance in Dedukti

392 ▶ **Lemma 19** (Simulation of reduction steps). *Let $t \in \Lambda_o$ with $\Gamma \vdash t : A$ and $|t| \longrightarrow u$ for some*
 393 *$u \in \Lambda_{cc}$. Then, there is some $t' \in \Lambda_o$ such that $|t'| = u$ and $t \longrightarrow^* t'$.*

394 **Proof.** By induction on t , following the definition of Λ_o . Almost all cases are either impossible,
 395 or follow by applying the i.h. to the subterm being reduced. The only interesting case is
 396 when $t = t_1 t_2$ and the reduction happens in the head. Then, the only possibility is that
 397 $t_1 = \uparrow_{n_k} D_k (\dots (\uparrow_{n_1} D_1 v) \dots)$ with $v = \lambda x : C.s$ and $|t| = (\lambda x : |C|. |s|) |t_2| \longrightarrow |s| [|t_2|/x]$. If
 398 $k = 0$ then the result is immediate, as t is a β redex. Otherwise, by typing constraints and
 399 Proposition 35 we can see that we have $D_1 \equiv \dots \equiv D_k$ and $n_{i+1} = n_i + 1$ for $i = 1..k - 1$,
 400 so by confluence we have some common reduct D_0 of all of them so that $t_1 \longrightarrow^* \uparrow_{n_1}^{n_k} D_0 v$.
 401 Then, by inversion of typing, v has both types $D_0 \Rightarrow \underline{n_1}$ and $(x : C) \rightarrow A'$ for some A' ,
 402 hence by uniqueness of types we have $D_0 \Rightarrow \underline{n_1} \equiv (x : C) \rightarrow A'$, which by Lemma 36 implies
 403 $D_0 \longrightarrow^* C' \lrcorner \lambda x : B.D'$ for some C', l, B, D' . Abbreviating $C' \lrcorner \lambda x : B.D'$ as D'_0 ,

$$404 \quad t \longrightarrow^* \uparrow_{n_1}^{n_k} D'_0 (\lambda x : C.s) t_2 \longrightarrow^* (\lambda x : C. \uparrow_{n_1}^{n_k} D' s) t_2 \longrightarrow \uparrow_{n_1}^{n_k} D' [t_2/x] s[t_2/x]$$

405 and we indeed have $\uparrow_{n_1}^{n_k} D' [t_2/x] s[t_2/x] \in \Lambda_o$, with $|\uparrow_{n_1}^{n_k} D' [t_2/x] s[t_2/x]| = |s| [|t_2|/x]$. ◀

406 ▶ **Proposition 20** (Reflection of type conversion). *Let $A, B \in \Lambda_o$ with $\Gamma \vdash A : \underline{U}_n$ and*
 407 *$\Gamma \vdash B : \underline{U}_m$. If $|A| \equiv |B|$ then $\text{El}_n A \equiv \text{El}_m B$.*

408 **Proof.** Take $k := \max\{n, m\}$; we have $\Gamma \vdash \uparrow_n^k \diamond A : \underline{U}_k$ and $\Gamma \vdash \uparrow_m^k \diamond B : \underline{U}_k$ and
 409 $|\uparrow_n^k \diamond A| = |A| \equiv |B| = |\uparrow_m^k \diamond B|$. By confluence we have $|\uparrow_n^k \diamond A| \longrightarrow^* C \ast \longleftarrow |\uparrow_m^k \diamond B|$
 410 for some C . By iterating Lemma 19 with subject reduction, we get $\uparrow_n^k \diamond A \longrightarrow^* A'$ and
 411 $\uparrow_m^k \diamond B \longrightarrow^* B'$ and $|A'| = C = |B'|$ for some A' and B' . We also have $\Gamma \vdash A' : \underline{U}_k$ and
 412 $\Gamma \vdash B' : \underline{U}_k$, so by Theorem 18 we get $A' \equiv B'$ — note that because A' and B' have the
 413 same type, there can be no lifts between them. Therefore, we have $\uparrow_n^k \diamond A \equiv \uparrow_m^k \diamond B$ and
 414 thus we conclude $\text{El}_n A \equiv \text{El}_k (\uparrow_n^k \diamond A) \equiv \text{El}_k (\uparrow_m^k \diamond B) \equiv \text{El}_m B$. ◀

415 We now have almost all auxiliary results needed for showing soundness. As a last step,
 416 we only need the following two easy lemmas.

417 ▶ **Lemma 21** (Computing the El of a translation). *Let $A \in \Lambda_o$ with $\text{El}_l A$ well typed.*

- 418 1. *If $|A| = n$ then $\text{El}_l A \longrightarrow^* \underline{U}_n$.*
- 419 2. *If $|A| = \Pi x : A_1.A_2$ then $\text{El}_l A \longrightarrow^* (x : \text{El}_{n_1} A'_1) \rightarrow \text{El}_{n_2} A'_2$ with $|A'_i| = A_i$.*

420 **Proof.** By definition of $|-|$ and typing constraints. ◀

421 ▶ **Lemma 22** (Telescope translation). *Let $A_1, A_2 \in \Lambda_o$ with $\Gamma \vdash A_i : \underline{U}_{n_i}$. If $|A_i| = \Delta \Rightarrow m_i$*
 422 *for some $m_1 \leq m_2$, then we have $\text{El}_{n_i} A_i \equiv D \Rightarrow \underline{m}_i$ for some guarded D with $\Gamma \vdash D : \text{Tele}$.*

423 **Proof.** By induction on Δ . ◀

424 ▶ **Theorem 23** (Soundness). *If $\Gamma \vdash_{cc} t : A$ then we have $\Gamma' \vdash t' : \text{El}_n A'$ for some $\Gamma' \in \text{Ctx}_o$*
 425 *and $t', A' \in \Lambda_o$ and $n \in \mathbb{N}$ with $\|\Gamma'\| = \Gamma$ and $|t'| = t$ and $|A'| = A$.*

426 **Proof.** We instead show the following two points, which together imply the theorem.

- 427 ■ If $\Gamma \vdash_{cc}$ then $\Gamma' \vdash$ for some $\Gamma' \in \text{Ctx}_o$ with $\|\Gamma'\| = \Gamma$.
- 428 ■ If $\Gamma \vdash_{cc} t : A$ and $\Gamma' \vdash$ for some $\Gamma' \in \text{Ctx}_o$ with $\|\Gamma'\| = \Gamma$ then $\Gamma' \vdash t' : \text{El}_n A'$ for some
 429 $n \in \mathbb{N}$ and $A', t' \in \Lambda_o$ with $|A'| = A$ and $|t'| = t$.

430 We prove them by induction on the derivation of $\Gamma \vdash_{\text{CC}} t : A$, and illustrate here
431 one of the interesting cases.

432 ■ Case

$$433 \quad A \subseteq B \quad \frac{\text{CONV} \quad \Gamma \vdash_{\text{CC}} t : A \quad \Gamma \vdash_{\text{CC}} B : n}{\Gamma \vdash_{\text{CC}} t : B}$$

434 By induction hypothesis we have $\Gamma' \vdash t' : \text{El}_m A'$ and $\Gamma' \vdash B' : \text{U}_n$ with $|t'| = t$, $|A'| = A$
435 and $|B'| = B$ (using Lemma 21 for the second derivation). By inversion we obtain
436 $\Gamma' \vdash A' : \text{U}_m$. We now use Lemma 2 to split $A \subseteq B$ into two cases:

- 437 ■ $A \equiv B$. We have $|A'| \equiv |B'|$ so by Proposition 20 we conclude $\text{El}_m A' \equiv \text{El}_n B'$, and
438 thus $\Gamma' \vdash t' : \text{El}_n B'$.
- 439 ■ $A \longrightarrow^* \Delta \Rightarrow p$ and $B \longrightarrow^* \Delta \Rightarrow q$ with $p \leq q$. We apply Lemma 19 on A' to get some
440 A'' such that $|A''| = \Delta \Rightarrow p$ and $A' \longrightarrow^* A''$. Similarly, we get B'' with $|B''| = \Delta \Rightarrow q$
441 and $B' \longrightarrow^* B''$. We can then apply Lemma 22 to obtain a guarded term D such that
442 $\Gamma' \vdash D : \text{Tele}$ and $\text{El}_m A'' \equiv D \Rightarrow p$ and $\text{El}_n B'' \equiv D \Rightarrow q$. We can now conclude with
443 $\Gamma' \vdash \uparrow_{\underline{p}}^q D t : \text{El}_n B'$. ◀

444 8 Conservativity

445 Now that we have seen that our encoding is sound, we can move to the proof of conservativity.
446 The usual statement of conservativity (using direct translation functions $[-] : \Lambda_{\text{CC}} \rightarrow \Lambda_{\text{dk}}$ and
447 $\llbracket - \rrbracket : \text{Ctx}_{\text{CC}} \rightarrow \text{Ctx}_{\text{dk}}$) would say that, given Γ, A satisfying $\Gamma \vdash_{\text{CC}} A : n$, if $\llbracket \Gamma \rrbracket \vdash t : \text{El}_n [A]$
448 then we have $\Gamma \vdash_{\text{CC}} t' : A$ for some t' . When rephrasing this statement with the inverse
449 translation function $|-|$, the full conservativity property would then assert that, for $\Gamma \in \text{Ctx}_o$
450 and $A \in \Lambda_o$ with $\llbracket \Gamma \rrbracket \vdash |A| : n$, if $\Gamma \vdash t : \text{El}_n A$ then $\llbracket \Gamma \rrbracket \vdash_{\text{CC}} t' : |A|$ for some t' .

451 In the following, we instead show *conservativity for object terms*, a restricted form of
452 conservativity in which the witness t of the typing judgment $\Gamma \vdash t : \text{El}_n A$ is required to be an
453 object term. We argue that this is enough because in practice the object terms are the only
454 ones a user of the encoding (or an automatic translator) would write. Nevertheless, it should
455 be possible to strengthen our result to obtain full conservativity, as discussed in the conclusion.

456 The first step in our proof is showing that $|-|$ preserves definitional equality. This is
457 however not immediate, because $|-|$ does not preserve reduction steps. Fortunately, we can
458 define an auxiliary function $|-|^\bullet$ extending $|-|$ that satisfies this property. We start by
459 defining the *extended object terms* Λ_o^\bullet which will be used as the domain of $|-|^\bullet$. Here we
460 write G, G' for any guarded terms, and n, n_0, n_1, n_2 for any natural numbers.

$$461 \quad \boxed{\Lambda_o^\bullet} \ni \quad t, u, A, B ::= x \mid (x : A) \rightarrow B \mid \lambda x : A. t \mid \text{U}_n \mid \text{El}_G A \mid \text{u}_n$$

$$462 \quad \mid \pi_{\underline{n_1, n_2}}^{n_0} A \lambda x : G.B \mid \uparrow_G G' t \mid \uparrow_{\underline{n}} t \mid t u$$

464 The function $|-|^\bullet$ is then defined by the following clauses.

$$465 \quad \begin{array}{lll} |-|^\bullet : \Lambda_o^\bullet \rightarrow \Lambda_{\text{CC}} & & |(x : A) \rightarrow B|^\bullet := \Pi x : |A|^\bullet. |B|^\bullet \\ 466 \quad |x|^\bullet := x & |\text{El}_G A|^\bullet := |A|^\bullet & |\lambda x : A. t|^\bullet := \lambda x : |A|^\bullet. |t|^\bullet \\ 467 \quad |\text{u}_n|^\bullet := n & |\uparrow_G G' t|^\bullet := |t|^\bullet & |\pi_{\underline{n_1, n_2}}^{n_0} A (\lambda x : G.B)|^\bullet := \Pi x : |A|^\bullet. |B|^\bullet \\ 468 \quad |\text{U}_n|^\bullet := n & |\uparrow_{\underline{n}} t|^\bullet := |t|^\bullet & |t u|^\bullet := |t|^\bullet |u|^\bullet \quad (t \text{ u not of previous forms}) \end{array}$$

23:14 Impredicativity, Cumulativity and Product Covariance in DEDUKTI

470 We can show that $|- \bullet$ satisfies many desirable properties, among them being the
471 preservation of reduction steps and thus also of definitional equality by $|- \bullet$.

472 ▶ **Lemma 24** (Basic properties of Λ_o^\bullet and $|- \bullet$).

- 473 1. Λ_o^\bullet is a superset of Λ_o , and $|- \bullet$ restricts to $|-$ in Λ_o .
- 474 2. If $t \in \Lambda_o^\bullet$ then t is guarded.
- 475 3. If $t, u \in \Lambda_o^\bullet$ then $t[u/x] \in \Lambda_o^\bullet$ and $|t|^\bullet[|u|^\bullet/x] = |t[u/x]|^\bullet$.
- 476 4. If $t \in \Lambda_o^\bullet$ and $t \longrightarrow^* u$ then $u \in \Lambda_o^\bullet$ and $|t|^\bullet \longrightarrow^* |u|^\bullet$.
- 477 5. If $t, u \in \Lambda_o^\bullet$ and $t \equiv u$ then $|t|^\bullet \equiv |u|^\bullet$.

478 Using these basic properties, we can now show conservativity.

479 ▶ **Theorem 25** (Conservativity for object terms). *Let $\Gamma \in \text{Ctx}_o$ and $A \in \Lambda_o$ with $\|\Gamma\| \vdash_{\text{CC}} |A| : n$
480 for some n . If $\Gamma \vdash t : \text{El}_n A$ with t an object term, then we have $\|\Gamma\| \vdash_{\text{CC}} |t| : |A|$.*

481 **Proof.** We instead show the following claim.

482 ▷ **Claim 26.** Let $\Gamma \vdash t : A$ with $\Gamma \in \text{Ctx}_o$ and $\|\Gamma\| \vdash_{\text{CC}}$. If t is an object term, then there
483 exists $A' \in \Lambda_o^\bullet$ with $A \equiv A'$ and $\|\Gamma\| \vdash_{\text{CC}} |t| : |A'|^\bullet$.

484 First note that this implies the statement of the theorem. Indeed, by the claim we have
485 $\|\Gamma\| \vdash_{\text{CC}} |t| : |B|^\bullet$ for some $B \in \Lambda_o^\bullet$ with $B \equiv \text{El}_n A$. Therefore $|B|^\bullet \equiv |A|^\bullet = |A|$, so we
486 conclude $\|\Gamma\| \vdash_{\text{CC}} |t| : |A|$ by the conversion rule.

487 We proceed with the proof of the claim, by induction on t , following the definition of
488 Λ_o . We illustrate the interesting case of λ -abstraction: $t = \lambda x : \text{El}_n A_1. u$. By inversion we
489 have $\Gamma \vdash A_1 : \text{U}_n$ and $\Gamma, x : \text{El}_n A_1 \vdash u : A_2$ for some A_2 with $A \equiv (x : \text{El}_n A_1) \rightarrow A_2$.
490 By i.h. we thus have $\|\Gamma\| \vdash_{\text{CC}} |A_1| : |B_1|^\bullet$ with $B_1 \equiv \text{U}_n$. Therefore, we have $|B_1|^\bullet \equiv n$,
491 so by conversion we can derive $\|\Gamma\| \vdash_{\text{CC}} |A_1| : n$, and so $\|\Gamma\|, x : |A_1| \vdash_{\text{CC}}$. By i.h. once
492 more, we have $\|\Gamma\|, x : |A_1| \vdash_{\text{CC}} |u| : |B_2|^\bullet$ for some B_2 with $B_2 \equiv A_2$. We can thus derive
493 $\|\Gamma\| \vdash_{\text{CC}} \lambda x : |A_1|. |u| : |(x : \text{El}_n A_1) \rightarrow B_2|^\bullet$ and $A \equiv (x : \text{El}_n A_1) \rightarrow B_2$. ◀

9 Related work

495 The first attempt to encode CC in DEDUKTI dates back to the work of Assaf. He first
496 identified the full-reflection equations (discussed in Section 4) in earlier work studying a
497 variant of the calculus of constructions with explicit cumulativity [4]. There, cumulativity is
498 made explicit by a family of lifts $\uparrow_i : \text{U}_i \rightarrow \text{U}_{i+1}$, which are sufficient in his setting because
499 the theory considered lacks product covariance.

500 These ideas were then employed in encoding a class of cumulative type systems (CTSs)
501 in DEDUKTI [5], containing in particular the type system CC. In order to handle product
502 covariance, he proposed the use of η -expansion at translation time: for instance, a variable
503 $f : \text{Nat} \rightarrow 0$ would be translated *at type* $\text{Nat} \rightarrow 1$ as $\lambda x. \uparrow_0 (f x)$. This however turned out to
504 invalidate conservativity, as observed by Thiré [45, Example 6.6].

505 Moreover, as mentioned in the introduction, the translation functions used by Assaf for
506 stating and proving soundness turn out to be ill-defined. He mutually defines functions
507 $[-]_\Gamma$ and $[-]_{\Gamma \vdash C}$ and $\llbracket - \rrbracket$, and among their defining clauses he states $\llbracket t \rrbracket_{\Gamma \vdash C} := \lambda x : \llbracket A \rrbracket. \llbracket t x \rrbracket_{\Gamma, x : A \vdash B}$ if $C \equiv \Pi x : A. B$ and t has a principal type convertible to $\Pi x : A. B'$ with
508 $B' \subsetneq B$. However, the term A is only determined up to conversion, yet the function is defined
509 over unquotiented terms, and the preservation of conversion is only shown at a later stage.
510 Worse, because A is recovered using typing information, it might not be structurally smaller
511 than t , and no well-founded order is given to justify the recursive call of $\llbracket - \rrbracket$ on A .
512

513 Regarding confluence, Assaf actually relies in his presentation on an axiomatization of the
 514 conversion relation required for the encoding. Because in DEDUKTI the conversion must be
 515 implemented by rewrite rules, each instantiation of his encoding then also needs to provide a
 516 rewrite system correctly implementing these equational axioms. In the particular case of CC,
 517 Assaf provides rules for implementing them, yet they are not confluent since some critical
 518 pairs are not joinable. This problem was later fixed in his joint work with Dowek, Jouannaud
 519 and Liu [8], though it required the use of rewriting modulo ACU, which is less efficient
 520 and harder to implement than pure syntactic matching. The problems with soundness and
 521 conservativity remained unaddressed.

522 Some years after the work of Assaf, the problem regained attention and new encodings
 523 were proposed by Thiré [45], also supporting a class of CTSs, and Férey [27], also supporting
 524 universe polymorphism. Starting from Thiré’s observation that η -expanding at translation
 525 time breaks conservativity, they decided to instead rely on a generalized cast operator
 526 mapping a term $t : \text{El}_a a$ to ${}^{l_b} \uparrow_a^b e t : \text{El}_b b$, where e is a term witnessing the inclusion of a
 527 in b . Unfortunately, the use of a multi-step lift then required non-left-linear rules to ensure
 528 that two consecutive casts can be composed or that identity casts can be removed. Despite
 529 the impressive work of Férey on confluence criteria for non-left-linear systems [26], they were
 530 unable to show the confluence of their encodings.

531 The translation function employed by Thiré unfortunately inherited the issue of Assaf’s
 532 function, as it also makes recursive calls on terms obtained through typing information without
 533 giving a decreasing measure. The proposal of Férey uses however a different technique, and
 534 instead defines the translation function over typing derivations. Finally, conservativity is
 535 only stated as a conjecture for both of the encodings.

536 10 Conclusion

537 In this work we have given an encoding of CC in DEDUKTI satisfying the necessary properties
 538 for being used in practice, solving a longstanding open problem. Our proof of confluence
 539 combines many confluence criteria and heavily uses the automated tools developed by the
 540 community. Yet, at the present moment, none of the available tools are able to fully show
 541 our result by themselves. Proving the confluence of our system automatically can thus be an
 542 interesting challenge for the next generation of today’s confluence checkers.

543 Our work has also identified a problem with the definition of the translation function
 544 in some previous attempts at encoding CC in DEDUKTI. To solve this issue, we have then
 545 contributed an adaptation of the technique of Winterhalter et al [47] in which soundness is
 546 instead stated and proved using an inverse translation function.

547 Regarding conservativity, we have proven a restricted form concerning only object terms.
 548 Even though we believe that for practical needs our result is sufficient, we conjecture
 549 that full conservativity can be obtained by adapting the logical relations technique of
 550 Assaf [6]. Alternatively, we could modify our encoding and employ the technique described
 551 by Felicissimo [23], which allows for easy conservativity proofs at the cost of increasing the
 552 amount of type annotations in the syntax. There is already ongoing work on removing
 553 these annotations by incorporating bidirectional typing into DEDUKTI [24], yet the encoding
 554 presented here would not be covered by the presently available framework.

555 Finally, we believe that our work can be a starting point for incorporating COQ’s universe-
 556 polymorphism. Among previous work, only Férey considers the combination of CC with
 557 universe polymorphism. Combining his ideas with ours is a promising direction to explore.

558 **Acknowledgements**

559 The authors would like to thank François Thiré and Yoan Gérard for helpful remarks about a
 560 first draft, and Gaspard Férey, Jean-Pierre Jouannaud, Frédéric Blanqui and Gilles Dowek
 561 for informative discussions around the subject of this paper.

562 **References**

-
- 563 1 AProVE. URL: <https://aprove.informatik.rwth-aachen.de/>.
- 564 2 CeTA. URL: <http://cl-informatik.uibk.ac.at/software/ceta/>.
- 565 3 CSIho. URL: <http://cl-informatik.uibk.ac.at/software/csi/ho/>.
- 566 4 Ali Assaf. A calculus of constructions with explicit subtyping. In *20th International Conference*
 567 *on Types for Proofs and Programs (TYPES 2014)*, volume 39, 2014.
- 568 5 Ali Assaf. *A framework for defining computational higher-order logics*. These, École polytechnique,
 569 September 2015. URL: <https://pastel.archives-ouvertes.fr/tel-01235303>.
- 570 6 Ali Assaf. Conservativity of Embeddings in the lambda Pi Calculus Modulo Rewriting. In Thorsten
 571 Altenkirch, editor, *13th International Conference on Typed Lambda Calculi and Applications (TLCA 2015)*,
 572 volume 38 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31–44, Dagstuhl, Germany,
 573 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TLCA.2015.31>,
 574 doi:10.4230/LIPIcs.TLCA.2015.31.
- 575 7 Ali Assaf, Guillaume Burel, Raphaël Cauderlier, David Delahaye, Gilles Dowek, Catherine
 576 Dubois, Frédéric Gilbert, Pierre Halmagrand, Olivier Hermant, and Ronan Saillard. Dedukti:
 577 a logical framework based on the $\lambda\Pi$ -calculus modulo theory. Unpublished, 2016.
- 578 8 Ali Assaf, Gilles Dowek, Jean-Pierre Jouannaud, and Jiaxiang Liu. Untyped Confluence In
 579 Dependent Type Theories. working paper or preprint, April 2017. URL: <https://hal.inria.fr/hal-01515505>.
- 580 9 Hendrik Pieter Barendregt, Wil Dekkers, and Richard Statman. *Lambda calculus with types*.
 581 Cambridge University Press, 2013.
- 582 10 Bruno Barras. Auto-validation d’un système de preuves avec familles inductives. *These de*
 583 *doctorat, Université Paris, 7, 1999*.
- 584 11 Bruno Barras and Benjamin Gregoire. On the role of type decorations in the calculus of
 585 inductive constructions. volume 3634, pages 151–166, 08 2005. doi:10.1007/11538363_12.
- 586 12 M. Bezem, J.W. Klop, R. de Vrijer, and Terese. *Term Rewriting Systems*. Cambridge
 587 Tracts in Theoretical Computer Science. Cambridge University Press, 2003. URL: <https://books.google.fr/books?id=7QQ5u-4tRUKC>.
- 588 13 Frédéric Blanqui. *Théorie des types et réécriture. (Type theory and rewriting)*. PhD thesis,
 589 University of Paris-Sud, Orsay, France, 2001. URL: <https://tel.archives-ouvertes.fr/tel-00105522>.
- 590 14 Frédéric Blanqui. Type safety of rewrite rules in dependent types. In *5th International*
 591 *Conference on Formal Structures for Computation and Deduction*, 2020.
- 592 15 Frédéric Blanqui. Encoding type universes without using matching modulo AC. In *Proceedings*
 593 *of the 7th International Conference on Formal Structures for Computation and Deduction*,
 594 Leibniz International Proceedings in Informatics 228, 2022.
- 595 16 Frédéric Blanqui. Hol-light library in coq. URL: <https://github.com/Deducteam/coq-hol-light>.
- 596 17 Frédéric Blanqui, Gilles Dowek, Emilie Grienemberger, Gabriel Hondet, and François Thiré.
 597 A modular construction of type theories. *Logical Methods in Computer Science*, Volume
 598 19, Issue 1, February 2023. URL: [http://dx.doi.org/10.46298/lmcs-19\(1:12\)2023](http://dx.doi.org/10.46298/lmcs-19(1:12)2023), doi:
 599 10.46298/lmcs-19(1:12)2023.
- 600 18 Valentin Blot, Gilles Dowek, and Thomas Traversié. An Implementation of Set Theory
 601 with Pointed Graphs in Dedukti. In *LFMTP 2022 - International Workshop on Logical*
 602
 603
 604
 605
 606

- 607 *Frameworks and Meta-Languages : Theory and Practice*, Haifa, Israel, August 2022. URL:
608 <https://inria.hal.science/hal-03740004>.
- 609 19 Denis Cousineau and Gilles Dowek. Embedding pure type systems in the lambda-pi-calculus
610 modulo. In Simona Ronchi Della Rocca, editor, *Typed Lambda Calculi and Applications*, pages
611 102–117, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- 612 20 Deducteam. An encoding of impredicativity, cumulativity and product covariance in the logical
613 framework dedukti. URL: <https://github.com/Deducteam/cc-in-dk>.
- 614 21 Deducteam. Pull request for ACU matching in DkCheck. URL: <https://github.com/Deducteam/Dedukti/pull/219>.
- 616 22 Gilles Dowek, Gaspard Férey, Jean-Pierre Jouannaud, and Jiaxiang Liu. Confluence of left-
617 linear higher-order rewrite theories by checking their nested critical pairs. *Mathematical*
618 *Structures in Computer Science*, 32(7):898–933, 2022. doi:10.1017/S0960129522000044.
- 619 23 Thiago Felicissimo. Adequate and Computational Encodings in the Logical Framework
620 Dedukti. In Amy P. Felty, editor, *7th International Conference on Formal Structures for*
621 *Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in*
622 *Informatics (LIPIcs)*, pages 25:1–25:18, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-
623 Zentrum für Informatik. URL: <https://drops.dagstuhl.de/opus/volltexte/2022/16306>, doi:
624 10.4230/LIPIcs.FSCD.2022.25.
- 625 24 Thiago Felicissimo. Generic bidirectional typing for dependent type theories, 2023. arXiv:
626 2307.08523.
- 627 25 Thiago Felicissimo, Frédéric Blanqui, and Ashish Kumar Barnawal. Translating Proofs
628 from an Impredicative Type System to a Predicative One. In Bartek Klin and Elaine
629 Pimentel, editors, *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*,
630 volume 252 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:19,
631 Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/opus/volltexte/2023/17480>, doi:10.4230/LIPIcs.CSL.2023.19.
- 633 26 Gaspard Férey and Jean-Pierre Jouannaud. Confluence in non-left-linear untyped higher-order
634 rewrite theories. In *Proceedings of the 23rd International Symposium on Principles and*
635 *Practice of Declarative Programming*, PPDP '21, New York, NY, USA, 2021. Association for
636 Computing Machinery. doi:10.1145/3479394.3479403.
- 637 27 Gaspard Férey. *Higher-Order Confluence and Universe Embedding in the Logical Frame-*
638 *work*. These, Université Paris-Saclay, June 2021. URL: [https://tel.archives-ouvertes.fr/tel-](https://tel.archives-ouvertes.fr/tel-03418761)
639 [tel-03418761](https://tel.archives-ouvertes.fr/tel-03418761).
- 640 28 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and Stephan Falke. Automated
641 termination proofs with approve. In *Rewriting Techniques and Applications: 15th International*
642 *Conference, RTA 2004, Aachen, Germany, June 3-5, 2004. Proceedings 15*, pages 210–220.
643 Springer, 2004.
- 644 29 Yoan Gérard. Mathématiques inversées de Coq. 2021. URL: [https://inria.hal.science/](https://inria.hal.science/hal-04319183)
645 [hal-04319183](https://inria.hal.science/hal-04319183).
- 646 30 Makoto Hamana. How to prove your calculus is decidable: Practical applications of second-
647 order algebraic theories and computation. *Proc. ACM Program. Lang.*, 1(ICFP), aug 2017.
648 doi:10.1145/3110266.
- 649 31 Hugo Herbelin. Type inference with algebraic universes in the calculus of inductive con-
650 structions. *Unpublished. Available at: pauillac.inria.fr/herbelin/publis/univalgcc.pdf*,
651 2005.
- 652 32 Gabriel Hondet and Frédéric Blanqui. Encoding of Predicate Subtyping with Proof Irrelevance
653 in the Lambdapi-Calculus Modulo Theory. In Ugo de'Liguoro, Stefano Berardi, and Thorsten
654 Altenkirch, editors, *26th International Conference on Types for Proofs and Programs (TYPES*
655 *2020)*, volume 188 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:18,
656 Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: <https://drops.dagstuhl.de/opus/volltexte/2021/13885>, doi:10.4230/LIPIcs.TYPES.2020.6.
- 657 33 Jan Willem Klop. *Combinatory reduction systems*. PhD thesis, Rijksuniversiteit Utrecht, 1963.

- 659 34 Marc Lasson. *Réalisabilité et paramétricité dans les systèmes de types purs*. Theses, Ecole
660 normale supérieure de lyon - ENS LYON, November 2012. URL: [https://theses.hal.science/
661 tel-00770669](https://theses.hal.science/tel-00770669).
- 662 35 Zhaohui Luo. *An extended calculus of constructions*. PhD thesis, University of Edinburgh,
663 1990.
- 664 36 Richard Mayr and Tobias Nipkow. Higher-order rewrite systems and their confluence. *Theoretical computer science*, 192(1):3–29, 1998.
- 665 37 Dale Miller. A logic programming language with lambda-abstraction, function variables, and
666 simple unification. *Journal of logic and computation*, 1(4):497–536, 1991.
- 667 38 Julian Nagele, Bertram Felgenhauer, and Aart Middeldorp. CSI: New evidence — a progress
668 report. In Leonardo de Moura, editor, *Proceedings of the 26th International Conference on
669 Automated Deduction (CADE-26)*, volume 10395 of *Lecture Notes in Artificial Intelligence*,
670 pages 385–397, 2017. doi:10.1007/978-3-319-63046-5_24.
- 671 39 Vincent Oostrom and Femke Raamsdonk. Weak orthogonality implies confluence: The
672 higher-order case. In Gerhard Goos, Juris Hartmanis, Anil Nerode, and Yu. V. Matiyasevich,
673 editors, *Logical Foundations of Computer Science*, volume 813, pages 379–392. Springer Berlin
674 Heidelberg, Berlin, Heidelberg, 1994. Series Title: Lecture Notes in Computer Science. URL:
675 http://link.springer.com/10.1007/3-540-58140-5_35, doi:10.1007/3-540-58140-5_35.
- 676 40 Nicolas Oury. Extensionality in the calculus of constructions. In *Theorem Proving in Higher
677 Order Logics: 18th International Conference, TPHOLs 2005, Oxford, UK, August 22-25, 2005.
678 Proceedings 18*, pages 278–293. Springer, 2005.
- 679 41 Ronan Saillard. *Type checking in the Lambda-Pi-calculus modulo: theory and practice*. PhD
680 thesis, Mines ParisTech, France, 2015.
- 681 42 The Coq Development Team. Typing rules for Coq. URL: [https://coq.inria.fr/doc/V8.16.
682 1/refman/language/cic.html#id6](https://coq.inria.fr/doc/V8.16.1/refman/language/cic.html#id6).
- 683 43 René Thiemann and Christian Sternagel. Certification of termination proofs using ceta. In
684 *International Conference on Theorem Proving in Higher Order Logics*, pages 452–468. Springer,
685 2009.
- 686 44 François Thiré. Sharing a library between proof assistants: Reaching out to the HOL family. In
687 Frédéric Blanqui and Giselle Reis, editors, *Proceedings of the 13th International Workshop on
688 Logical Frameworks and Meta-Languages: Theory and Practice, LFMTP@FSCD 2018, Oxford,
689 UK, 7th July 2018*, volume 274 of *EPTCS*, pages 57–71, 2018. doi:10.4204/EPTCS.274.5.
- 690 45 François Thiré. *Interoperability between proof systems using the logical framework Dedukti*.
691 PhD thesis, ENS Paris-Saclay, 2020.
- 692 46 Vincent Van Oostrom. Developing developments. *Theoretical Computer Science*, 175(1):159–
693 181, 1997.
- 694 47 Théo Winterhalter, Matthieu Sozeau, and Nicolas Tabareau. Eliminating Reflection from
695 Type Theory. In *CPP 2019 - 8th ACM SIGPLAN International Conference on Certified
696 Programs and Proofs*, pages 91–103, Lisbonne, Portugal, January 2019. ACM. URL: <https://hal.science/hal-01849166>,
697 doi:10.1145/3293880.3294095.

699 **A** Basic metaproperties of Dedukti

700 In the following, let us write $\Gamma \sqsubseteq \Gamma'$ when Γ is a subsequence of Γ' . We recall the following
701 basic metaproperties of DEDUKTI.

702 ▶ **Proposition 27** (Basic metaproperties of DEDUKTI).

703 **Weakening** *If $\Gamma \sqsubseteq \Gamma'$ and $\Gamma' \vdash t : A$ then $\Gamma \vdash t : A$.*

704 **Substitution** *If $\Gamma, x : B, \Gamma' \vdash t : A$ and $\Gamma \vdash u : B$ then $\Gamma, \Gamma'[u/x] \vdash t[u/x] : A[u/x]$.*

705 *For all the following points, suppose that the underlying theory is well typed.*

$$\begin{array}{l}
\mathfrak{S} : \mathbf{Type} \quad \mathfrak{A} : \mathfrak{S} \rightarrow \mathfrak{S} \quad \mathfrak{P} : \mathfrak{S} \rightarrow \mathfrak{S} \quad - : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} \quad (\text{infix}) \\
0 : \mathfrak{S} \quad \mathfrak{A} 0 \mapsto S(S 0) \quad \mathfrak{P} 0 \mapsto 0 \quad 1_1 - 0 \mapsto 1_1 \\
S : \mathfrak{S} \rightarrow \mathfrak{S} \quad \mathfrak{A}(S 1) \mapsto S(S 1) \quad \mathfrak{P}(S 1) \mapsto 1 \quad 1_1 - (S 1_2) \mapsto (\mathfrak{P} 1_1) - 1_2 \\
\\
+ : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} \quad (\text{infix}) \quad \vee : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} \quad (\text{infix}) \quad \mathfrak{R} : \mathfrak{S} \rightarrow \mathfrak{S} \rightarrow \mathfrak{S} \\
0 + 1_2 \mapsto 1_2 \quad 0 \vee 1_2 \mapsto 1_2 \quad \mathfrak{R} 1_1 0 \mapsto 0 \\
1_1 + 0 \mapsto 1_1 \quad 1_1 \vee 0 \mapsto 1_1 \quad \mathfrak{R} 1_1 (S 1_2) \mapsto 1_1 \vee (S 1_2) \\
(S 1_1) + 1_2 \mapsto S(1_1 + 1_2) \quad (S 1_1) \vee (S 1_2) \mapsto S(1_1 \vee 1_2) \\
1_1 + (S 1_2) \mapsto S(1_1 + 1_2) \\
\\
U : (l : \mathfrak{S}) \rightarrow \mathbf{Type} \quad (\text{written } U_l) \quad u : (l : \mathfrak{S}) \rightarrow U_{(\mathfrak{A} l)} \quad (\text{written } u_l) \\
El : (l : \mathfrak{S}) \rightarrow U_l \rightarrow \mathbf{Type} \quad (\text{written } El_l) \quad El_{(_)} u_1 \mapsto U_1 \\
\\
\pi : (l_0 l_1 l_2 : \mathfrak{S}) \rightarrow (A : U_{(l_0 + l_1)}) \\
\rightarrow (B : El_{(l_0 + l_1)} A \rightarrow U_{(l_0 + l_2)}) \rightarrow U_{(\mathfrak{R}(l_0 + l_1)(l_0 + l_2))} \quad (\text{written } \pi_{l_1, l_2}^{l_0}) \\
El_{(_)} (\pi_{1_1, 1_2}^{1_0} A \lambda x : C.B\{x\}) \xrightarrow{\pi_S} (x : El_{(1_0 + 1_1)} A) \rightarrow El_{(1_0 + 1_2)} B\{x\} \\
\pi_{(S 1_1), (S 1_2)}^{1_0} A B \mapsto \pi_{1_1, 1_2}^{(S 1_0)} A B \\
\\
Tele : \mathbf{Type} \quad \Rightarrow : Tele \rightarrow \mathfrak{S} \rightarrow \mathbf{Type} \quad (\text{infix}) \\
\blacklozenge : Tele \quad \blacklozenge \Rightarrow 1_1 \xrightarrow{\Rightarrow} U_{1_1} \\
\blacktriangleleft : (l : \mathfrak{S}) \rightarrow (A : U_l) \rightarrow (El_l A \rightarrow Tele) \quad (A_{1_2} \blacktriangleleft \lambda x : _ . D\{x\}) \Rightarrow 1_1 \xrightarrow{\Rightarrow} (x : El_{1_2} A) \rightarrow D\{x\} \Rightarrow 1_1 \\
\rightarrow Tele \quad (\text{infix, written } l \blacktriangleleft) \\
\\
\uparrow : (l : \mathfrak{S}) \rightarrow (D : Tele) \rightarrow (D \Rightarrow l) \rightarrow (D \Rightarrow (S l)) \quad (\text{written } \uparrow_l) \\
El_1 (\uparrow \blacklozenge A) \xrightarrow{\uparrow El} El_{(\mathfrak{P} 1)} A \\
(\uparrow \blacklozenge A) 1 \blacktriangleleft D \xrightarrow{\uparrow \blacktriangleleft} A_{(\mathfrak{P} 1)} \blacktriangleleft D \\
\\
\uparrow_1 (_ _ \blacktriangleleft \lambda x : _ . D\{x\}) \lambda x : A. \mathfrak{t}\{x\} \xrightarrow{\uparrow \lambda} \lambda x : A. \uparrow_1 D\{x\} \mathfrak{t}\{x\} \\
\uparrow_1 (_ _ \blacktriangleleft \lambda x : _ . D\{x\}) \mathfrak{t} u \xrightarrow{\uparrow \mathfrak{t}} \uparrow_1 D\{u\} (\mathfrak{t} u) \\
\\
\pi_{(S 1), 0}^0 (\uparrow \blacklozenge A) B \xrightarrow{\uparrow \pi^1} \pi_{1, 0}^0 A B \quad \uparrow : (l : \mathfrak{S}) \rightarrow (A : U_0) \rightarrow U_l \quad (\text{written } \uparrow_l) \\
\uparrow_0 A \mapsto A \\
\pi_{0, 1_2}^{(S 1_1)} (\uparrow \blacklozenge A) B \xrightarrow{\uparrow \pi^2} \pi_{0, (S 1_2)}^{1_1} A B \quad \uparrow_{(S 1)} A \mapsto \uparrow_1 \blacklozenge (\uparrow_1 A) \\
\pi_{(S 1_2), 0}^{(S 1_1)} (\uparrow \blacklozenge A) B \xrightarrow{\uparrow \pi^3} \uparrow_{(S (1_1 + 1_2))} \blacklozenge (\pi_{1_2, 0}^{(S 1_1)} A B) \quad El_{1_2} (\uparrow_1 A) \xrightarrow{\uparrow El} El_{(1_2 - 1_1)} A \\
\\
\pi_{1_2, 0}^{(S (S 1_1))} A (\lambda x : C. \uparrow \blacklozenge B\{x\}) \xrightarrow{\uparrow \pi^4} \pi_{(S 1_2), 0}^{(S 1_1)} A (\lambda x : C.B\{x\}) \\
\pi_{0, (S 1_2)}^{1_1} A (\lambda x : C. \uparrow \blacklozenge B\{x\}) \xrightarrow{\uparrow \pi^5} \uparrow_{(1_1 + 1_2)} \blacklozenge (\pi_{0, 1_2}^{1_1} A (\lambda x : C.B\{x\})) \\
\pi_{1, 0}^{(S 0)} A (\lambda x : C. \uparrow \blacklozenge B\{x\}) \xrightarrow{\uparrow \pi^6} \uparrow_{(S 1)} (\pi_{(S 1), 0}^0 A (\lambda x : C.B\{x\}))
\end{array}$$

■ **Figure 3** Definition of the theory $\mathbb{T}_{CC} = (\Sigma_{CC}, \mathcal{R}_{CC})$

23:20 Impredicativity, Cumulativity and Product Covariance in Dedukti

$$\begin{array}{ccc}
\uparrow_1 (_ _ \leftarrow \lambda x : C.D\{x\}) (\lambda x : A.t\{x\}) u & \xrightarrow{\uparrow_\circ} & \uparrow_1 D\{u\} ((\lambda x : A.t\{x\}) u) \\
\downarrow \uparrow_\lambda & & \downarrow \beta \\
(\lambda x : A.\uparrow_1 D\{x\} t\{x\}) u & \xrightarrow{\beta} & \uparrow_1 D\{u\} t\{u\} \\
\uparrow_1 ((\uparrow__ \blacklozenge B)_{1'} \leftarrow \lambda x : C.D\{x\}) (\lambda x : A.t\{x\}) & \xrightarrow{\uparrow_\lambda} & \lambda x : A.\uparrow_1 D\{x\} t\{x\} \\
\downarrow \uparrow_\blacklozenge & \nearrow \uparrow_\lambda & \\
\uparrow_1 (B_{(P\ 1')} \leftarrow \lambda x : C.D\{x\}) (\lambda x : A.t\{x\}) & & \\
\uparrow_1 ((\uparrow__ \blacklozenge B)_{1'} \leftarrow \lambda x : C.D\{x\}) t\ u & \xrightarrow{\uparrow_\circ} & \uparrow_1 D\{u\} (t\ u) \\
\downarrow \uparrow_\blacklozenge & \nearrow \uparrow_\circ & \\
\uparrow_1 (B_{(P\ 1')} \leftarrow \lambda x : C.D\{x\}) t\ u & & \\
((\uparrow_{1'} \blacklozenge A)_{1'} \leftarrow \lambda x : C.D\{x\}) \Rightarrow 1 & \xrightarrow{\Rightarrow_\blacklozenge} & (x : \text{El}_{1'} (\uparrow_{1'} \blacklozenge A)) \rightarrow D\{x\} \Rightarrow 1 \\
\downarrow \uparrow_\blacklozenge & & \downarrow \uparrow_{\text{El}} \\
(A_{(P\ 1')} \leftarrow \lambda x : C.D\{x\}) \Rightarrow 1 & \xrightarrow{\Rightarrow_\blacklozenge} & (x : \text{El}_{(P\ 1')} A) \rightarrow D\{x\} \Rightarrow 1
\end{array}$$

■ **Figure 4** Critical pairs of \mathcal{R}_1

706 **Validity** If $\Gamma \vdash t : A$ then either $A = \mathbf{Kind}$ or $\Gamma \vdash A : s$ for some sort s

707 For all the following points, suppose furthermore that $\beta\mathcal{R}$ is confluent.

708 **Subject reduction for β** If $\Gamma \vdash t : A$ and $t \rightarrow_\beta t'$ then $\Gamma \vdash t' : A$.

709 **Uniqueness of types** If $\Gamma \vdash t : A$ and $\Gamma \vdash t : B$ then $A \equiv B$

710 **Inversion of constant applications** If $c : (x_1 : A_1) \rightarrow \dots \rightarrow (x_k : A_k) \rightarrow B \in \Sigma$ and $\Gamma \vdash$
711 $c\ u_1..u_k : B'$ then we have $\Gamma \vdash u_i : A_i[u_j/x_j]_{j=1..i-1}$ for $i = 1..k$ and $B' \equiv B[u_j/x_j]_{j=1..k}$

712 **Proof.** We refer to the literature [13, 27, 41] for detailed proofs—even if there the definition
713 of the typing system is not exactly the same, the proofs for the variant used here are
714 straightforward adaptations of their proofs. ◀

715 **B** Basic metaproperties of CC

716 We recall the following basic properties of CC.

717 ▶ **Proposition 28** (Basic properties of CC).

718 **Church-Rosser** If $t \equiv u$ then $t \xrightarrow{*} v \xleftarrow{*} u$ for some v

719 **Weakening** If $\Gamma \subseteq \Gamma'$ and $\Gamma' \vdash$ and $\Gamma \vdash t : A$ then $\Gamma' \vdash t : A$

720 **Substitution** If $\Gamma \vdash t : A$ and $\Gamma, x : A, \Gamma' \vdash u : B$ then $\Gamma, \Gamma'[t/x] \vdash u[t/x] : B[t/x]$

721 **Validity** If $\Gamma \vdash t : A$ then $\Gamma \vdash A : s$ for some sort s

722 **Subject Reduction** If $\Gamma \vdash t : A$ and $t \rightarrow t'$ then $\Gamma \vdash t' : A$

723 **Proof.** See for instance [5], [11] or [34]. ◀

724 **C** Omitted proofs of Section 4

725 ▶ **Lemma 2** (Case analysis of \subseteq). If $A \subseteq B$ then either $A \equiv B$ or $A \xrightarrow{*} \Delta \Rightarrow n$ and
726 $B \xrightarrow{*} \Delta \Rightarrow m$ for some context Δ and natural numbers n, m with $n \leq m$.

727 **Proof.** By induction on the definition of \subseteq .

728 ■ Cases SUB and EQ : Trivial.

729 ■ Case TRANS : By i.h. we consider four cases :

730 ■ Subcase $A \equiv B$ and $B \equiv C$: Then $A \equiv C$.

731 ■ Subcase $A \equiv B$ and $B \longrightarrow^* \Delta \Rightarrow n$ and $C \longrightarrow^* \Delta \Rightarrow m$ with $n \leq m$: By confluence
732 we have $A \longrightarrow^* P \ast \longleftarrow \Delta \Rightarrow n$. It is easy to see that P must be of the form $\Delta' \Rightarrow n$
733 for some reduct Δ' of Δ . Therefore, we also have $C \longrightarrow^* \Delta' \Rightarrow m$.

734 ■ Subcase $A \longrightarrow^* \Delta \Rightarrow n$ and $B \longrightarrow^* \Delta \Rightarrow m$ with $n \leq m$ and $B \equiv C$: Symmetric to
735 the previous one.

736 ■ Subcase $A \longrightarrow^* \Delta \Rightarrow n$ and $B \longrightarrow^* \Delta \Rightarrow m$ with $n \leq m$, and $B \longrightarrow^* \Delta' \Rightarrow n'$ and
737 $C \longrightarrow^* \Delta' \Rightarrow m'$ with $n' \leq m'$: By confluence, we have $\Delta \Rightarrow m \longrightarrow^* P \ast \longleftarrow \Delta' \Rightarrow n'$.
738 It is easy to see that P is of the form $\Delta'' \Rightarrow m$ for some reduct Δ'' of Δ and Δ' , and that
739 $m = n'$. Hence we get $A \longrightarrow^* \Delta'' \Rightarrow n$ and $C \longrightarrow^* \Delta'' \Rightarrow m'$ with $n \leq m = n' \leq m'$.

740 ■ Case PRODCOV : By i.h. we consider two subcases

741 ■ Subcase $A \equiv B$: Then we also have $\Pi x : C.A \equiv \Pi x : C.B$

742 ■ Subcase $A \longrightarrow^* \Delta \Rightarrow n$ and $B \longrightarrow^* \Delta \Rightarrow m$ with $n \leq m$: Then by taking $\Delta' := x :$
743 C, Δ we have $\Pi x : C.A \longrightarrow^* \Delta' \Rightarrow m$ and $\Pi x : C.B \longrightarrow^* \Delta' \Rightarrow m$ with $n \leq m$. ◀

744 ▶ **Proposition 3** (Simulation of Assaf's full reflection rules). *We have the following conversions.*

$$745 \quad \pi_{\underline{1+n}, \underline{m}}^0 (\uparrow_l \diamond a) (\lambda x : C.b) \equiv \uparrow_{\mathfrak{R}(n, m)}^{\mathfrak{R}(1+n, m)} \diamond (\pi_{\underline{n}, \underline{m}}^0 a (\lambda x : C.b)) \quad (1)$$

$$746 \quad \pi_{\underline{n}, \underline{1+m}}^0 a (\lambda x : C.\uparrow_l \diamond b) \equiv \uparrow_{\mathfrak{R}(n, m)}^{\mathfrak{R}(n, 1+m)} \diamond (\pi_{\underline{n}, \underline{m}}^0 a (\lambda x : C.b)) \quad (2)$$

748 **Proof.** Each statement is shown separately, by a disjunction of cases in which each case
749 corresponds to one of the rules involving \uparrow and π .

750 1. ■ $m = 0$: Then the identity follows directly from the rule \uparrow_{π}^1 .

751 ■ $1 + n \leq m$ (which implies $m \neq 0$) : Then we have $m = k + 1 + n$ for some $k \in \mathbb{N}$ and

$$752 \quad \pi_{\underline{1+n}, \underline{k+1+n}}^0 (\uparrow_l \diamond a) (\lambda x : C.b) \equiv \pi_{\underline{0}, \underline{k}}^{1+n} (\uparrow_l \diamond a) (\lambda x : C.b)$$

$$753 \quad \equiv \pi_{\underline{0}, \underline{1+k}}^n a (\lambda x : C.b) \quad (\text{by } \uparrow_{\pi}^2)$$

$$754 \quad \equiv \pi_{\underline{n}, \underline{m}}^0 a (\lambda x : C.b)$$

756 ■ $1 + n > m > 0$: Then we have $n \geq m$ and thus $n = k + m$ for some $k \in \mathbb{N}$ and

$$757 \quad \pi_{\underline{1+k+m}, \underline{m}}^0 (\uparrow_l \diamond a) (\lambda x : C.b) \equiv \pi_{\underline{1+k}, \underline{0}}^m (\uparrow_l \diamond a) (\lambda x : C.b)$$

$$758 \quad \equiv \uparrow_{k+m} \diamond (\pi_{\underline{k}, \underline{0}}^m a (\lambda x : C.b)) \quad (\text{by } \uparrow_{\pi}^3)$$

$$759 \quad \equiv \uparrow_{\underline{n}} \diamond (\pi_{\underline{n}, \underline{m}}^0 a (\lambda x : C.b))$$

761 2. ■ $m \geq n$: Then we have $m = k + n$ for some $k \in \mathbb{N}$ and

$$762 \quad \pi_{\underline{n}, \underline{1+m}}^0 a (\lambda x : C.\uparrow_l \diamond b) = \pi_{\underline{n}, \underline{1+k+n}}^0 a (\lambda x : C.\uparrow_l \diamond b)$$

$$763 \quad \equiv \pi_{\underline{0}, \underline{1+k}}^n a (\lambda x : C.\uparrow_l \diamond b)$$

$$764 \quad \equiv \uparrow_{n+k} \diamond (\pi_{\underline{0}, \underline{k}}^n a (\lambda x : C.b)) \quad (\text{by } \uparrow_{\pi}^5)$$

$$765 \quad \equiv \uparrow_{\underline{m}} \diamond (\pi_{\underline{n}, \underline{m}}^0 a (\lambda x : C.b))$$

23:22 Impredicativity, Cumulativity and Product Covariance in Dedukti

767 \dashv $m < n$ with $m = 0$: Then we have $n = 1 + k$ for some $k \in \mathbb{N}$ and

$$\begin{aligned}
 768 \quad \pi_{\underline{n}, \underline{1}}^0 a (\lambda x : C. \uparrow_l \diamond b) &= \pi_{\underline{1+k}, \underline{1}}^0 a (\lambda x : C. \uparrow_l \diamond b) \\
 769 \quad &\equiv \pi_{\underline{k}, \underline{0}}^1 a (\lambda x : C. \uparrow_l \diamond b) \\
 770 \quad &\equiv \uparrow_{\underline{1+k}} (\pi_{\underline{1+k}, \underline{0}}^0 a (\lambda x : C.b)) && \text{(by } \uparrow_{\pi}^6) \\
 771 \quad &\equiv \uparrow_{\underline{n}} (\pi_{\underline{n}, \underline{0}}^0 a (\lambda x : C.b)) \\
 772 \quad &\equiv \uparrow_{\underline{0}}^n \diamond (\pi_{\underline{n}, \underline{0}}^0 a (\lambda x : C.b)) \\
 773
 \end{aligned}$$

774 \dashv $m < n$ with $m > 0$: Then we have $m = 1 + k$ for some $k \in \mathbb{N}$ and $n = m + 1 + k'$ for
775 some $k' \in \mathbb{N}$ and

$$\begin{aligned}
 776 \quad \pi_{\underline{n}, \underline{1+m}}^0 a (\lambda x : C. \uparrow_l \diamond b) &= \pi_{\underline{2+k+k'}, \underline{2+k}}^0 a (\lambda x : C. \uparrow_l \diamond b) \\
 777 \quad &\equiv \pi_{\underline{k'}, \underline{0}}^{\underline{2+k}} a (\lambda x : C. \uparrow_l \diamond b) && \text{(by } \uparrow_{\pi}^4) \\
 778 \quad &\equiv \pi_{\underline{1+k'}, \underline{0}}^{\underline{1+k}} a (\lambda x : C.b) \\
 779 \quad &\equiv \pi_{\underline{n}, \underline{m}}^0 a (\lambda x : C.b) && \blacktriangleleft \\
 780
 \end{aligned}$$

781 **D** Omitted proofs of Section 5

782 **D.1** Confluence of $\beta\mathcal{R}_1$

783 Given a rewrite system \mathcal{R} , the *orthogonal rewriting relation* $\Longrightarrow_{\beta\mathcal{R}}$ [22, 27] (also known
784 as *developments* or *multi-step reduction* [12]) is defined over metaterms by the following
785 inference rules, where we write $\theta \Longrightarrow \theta'$ as an abbreviation for $\text{dom}(\theta) = \text{dom}(\theta')$ and for all
786 $\vec{x}.t/\mathfrak{t} \in \theta$ and $\vec{x}.t'/\mathfrak{t} \in \theta'$ we have $t \Longrightarrow t'$.

$$\begin{array}{c}
 \text{VAR} \qquad \text{CONST} \qquad \text{SORT} \qquad \text{META} \\
 \frac{}{x \Longrightarrow x} \qquad \frac{}{c \Longrightarrow c} \qquad \frac{}{s \Longrightarrow s} \qquad \frac{t_i \Longrightarrow t'_i \text{ for all } i}{\mathfrak{t}\{t_1..t_k\} \Longrightarrow \mathfrak{t}\{t'_1..t'_k\}} \\
 \\
 \text{APP} \qquad \text{ABS} \qquad \text{FUN} \\
 \frac{t \Longrightarrow t' \quad u \Longrightarrow u'}{t u \Longrightarrow t' u'} \qquad \frac{A \Longrightarrow A' \quad t \Longrightarrow t'}{\lambda x : A.t \Longrightarrow \lambda x : A'.t'} \qquad \frac{A \Longrightarrow A' \quad B \Longrightarrow B'}{(x : A) \rightarrow B \Longrightarrow (x : A') \Longrightarrow B'} \\
 \\
 \text{RED}_{\mathcal{R}} \qquad \text{RED}_{\beta} \\
 \frac{\theta \Longrightarrow \theta'}{l[\theta] \Longrightarrow r[\theta']} \qquad \frac{t \Longrightarrow t' \quad u \Longrightarrow u'}{(\lambda x : A.t) u \Longrightarrow t'[u'/x]}
 \end{array}$$

787 Orthogonal rewriting satisfies the following well-known properties—see [27, Lemma 3.1.2]
788 and [27, Lemma 3.1.6] for the proofs.

789 **► Proposition 29.** *We have $\longrightarrow_{\beta\mathcal{R}_1} \subseteq \Longrightarrow_{\beta\mathcal{R}_1} \subseteq \longrightarrow_{\beta\mathcal{R}_1}^*$, hence $\longrightarrow_{\beta\mathcal{R}_1}^*$ and $\Longrightarrow_{\beta\mathcal{R}_1}^*$ are equal.*

790 **► Proposition 30.** *If $t \Longrightarrow_{\beta\mathcal{R}} t'$ and $\theta \Longrightarrow_{\beta\mathcal{R}} \theta'$ then $t[\theta] \Longrightarrow_{\beta\mathcal{R}} t'[\theta']$.*

791 In the following, recall that a rule $l \mapsto r$ overlaps $l' \mapsto r'$ when some non-metavariable
792 subterm of l unifies with l' .

793 **► Proposition 31.** *$\Longrightarrow_{\beta\mathcal{R}_1}$ satisfies the diamond property.*

794 **Proof.** Given t, u, v with $u \Leftarrow t \Rightarrow v$ we show that there is w with $u \Rightarrow w \Leftarrow v$. The
 795 proof is by induction on $t \Rightarrow u$ and $t \Rightarrow v$. The only interesting cases is when $t \Rightarrow u$
 796 (or dually, $t \Rightarrow v$) is derived with rule $\text{RED}_{\mathcal{R}}$, in which case we have $t = l[\theta]$ for some
 797 $l \mapsto r \in \mathcal{R}_1$ and $u = r[\theta']$ with $\theta \Rightarrow \theta'$. There are then three possibilities regarding $t \Rightarrow v$.

- 798 ■ If all applications of RED in $t \Rightarrow v$ occur inside the substitution θ , then because l is
 799 linear we have $v = l[\theta'']$ with $\theta \Rightarrow \theta''$. By i.h. we have $\theta' \Rightarrow \theta''' \Leftarrow \theta''$ for some θ''' ,
 800 and thus $u = r[\theta'] \Rightarrow r[\theta'''] \Leftarrow l[\theta''] = v$.
- 801 ■ If $t \Rightarrow v$ starts with an application of $\text{RED}_{\mathcal{R}}$ using the same rule as the one applied in
 802 $t \Rightarrow u$, then we have $v = r[\theta'']$ with $\theta \Rightarrow \theta''$. By i.h. we have $\theta' \Rightarrow \theta''' \Leftarrow \theta''$ for some
 803 θ''' , and thus $u = r[\theta'] \Rightarrow r[\theta'''] \Leftarrow r[\theta''] = v$.
- 804 ■ If RED is applied in $t \Rightarrow v$ with a rule $l' \mapsto r'$ overlapped by $l \mapsto r$, we consider all
 805 such possible cases (which correspond to the critical pairs in Figure 4).
 806 ■ Case $\uparrow_{\textcircled{a}}$ overlaps \uparrow_{λ} .

$$\begin{array}{ccc}
 \uparrow_l (_ _ \blacktriangleleft \lambda x : C.D) (\lambda x : A.t) u & \xrightarrow{\quad\quad\quad} & \uparrow_{l'} D'[u'/x] ((\lambda x : A'.t') u') \\
 \Downarrow & & \begin{array}{ccc} X & \xrightarrow{\quad\quad} & X' \\ \Downarrow & \text{i.h.} & \Downarrow \\ X'' & \xrightarrow{\quad\quad} & X''' \end{array} \\
 (\lambda x : A''. \uparrow_{l''} D'' t'') u'' & \xrightarrow{\quad\quad\quad} & \uparrow_{l'''} D'''[u'''/x] t'''[u'''/x]
 \end{array}$$

- 808 ■ Case \uparrow_{λ} overlaps $\uparrow_{\blacktriangleleft}$.

$$\begin{array}{ccc}
 \uparrow_l ((_ _ \blacklozenge B) _ \blacktriangleleft \lambda x : C.D) (\lambda x : A.t) & \xrightarrow{\quad\quad\quad} & \lambda x : A'. \uparrow_{l'} D' t' \\
 \Downarrow & & \begin{array}{ccc} X & \xrightarrow{\quad\quad} & X' \\ \Downarrow & \text{i.h.} & \Downarrow \\ X'' & \xrightarrow{\quad\quad} & X''' \end{array} \\
 \uparrow_{l''} (B'' (_ _ l''_0) \blacktriangleleft \lambda x : C''.D'') (\lambda x : A''.t'') & \xrightarrow{\quad\quad\quad} & \lambda x : A'''. \uparrow_{l'''} D''' t'''
 \end{array}$$

- 810 ■ Case $\uparrow_{\textcircled{a}}$ overlaps $\uparrow_{\blacktriangleleft}$.

$$\begin{array}{ccc}
 \uparrow_l ((_ _ \blacklozenge B) _ _ \blacktriangleleft \lambda x : C.D) t u & \xrightarrow{\quad\quad\quad} & \uparrow_{l'} D'[u'/x] (t' u') \\
 \Downarrow & & \begin{array}{ccc} X & \xrightarrow{\quad\quad} & X' \\ \Downarrow & \text{i.h.} & \Downarrow \\ X'' & \xrightarrow{\quad\quad} & X''' \end{array} \\
 \uparrow_{l''} (B'' (_ _ l''_0) \blacktriangleleft \lambda x : C''.D'') t'' u'' & \xrightarrow{\quad\quad\quad} & \uparrow_{l'''} D'''[u'''/x] (t''' u''')
 \end{array}$$

- 812 ■ Case $\Rightarrow_{\blacktriangleleft}$ overlaps $\uparrow_{\blacktriangleleft}$.

$$\begin{array}{ccc}
 ((\uparrow_{l_1} \blacklozenge A) _ _ \blacktriangleleft \lambda x : C.D) \Rightarrow l & \xrightarrow{\quad\quad\quad} & (x : \text{El}_{l'_0} (\uparrow_{l'_1} \blacklozenge A')) \rightarrow D' \Rightarrow l' \\
 \Downarrow & & \begin{array}{ccc} X & \xrightarrow{\quad\quad} & X' \\ \Downarrow & \text{i.h.} & \Downarrow \\ X'' & \xrightarrow{\quad\quad} & X''' \end{array} \\
 (A'' (_ _ l''_0) \blacktriangleleft \lambda x : C''.D'') \Rightarrow l'' & \xrightarrow{\quad\quad\quad} & (x : \text{El}_{(_ _ l''_0)} A''') \rightarrow D'' \Rightarrow l''
 \end{array}$$

814

23:24 Impredicativity, Cumulativity and Product Covariance in Dedukti

815 ▶ Remark 32. In a first read, one can have the impression that the above proof always applies
 816 when all critical pairs close in at most one step. This is not the case, and it crucially relies on
 817 the fact that all *orthogonal critical pairs* are simple, ensuring two facts. First, at most one
 818 rule in $t \Longrightarrow v$ can be overlapped by $l \mapsto r$ in $t \Longrightarrow u$. Second, if a rule $l \mapsto r$ in $t \Longrightarrow u$
 819 overlaps an a rule $l' \mapsto r'$ in $t \Longrightarrow v$, then no rule in $t \Longrightarrow u$ is overlapped by $l' \mapsto r'$.

820 ▶ **Corollary 6.** $\beta\mathcal{R}_1$ is confluent.

821 **E** Omitted proofs of Section 7

822 E.1 Injectivity

823 We start with the following generalization of Assaf's full reflection equations, used in the
 824 proof of the injectivity of **El** modulo lifting. From now on, let us write $(\uparrow_{-} D)^k t$ for
 825 $\uparrow_{l_1} D (\dots (\uparrow_{l_k} D t) \dots)$ where the l_1, \dots, l_k can be any terms.

826 ▶ **Lemma 33** (Generalized full reflection). *For all $k_1, k_2, n_1, n_2 \in \mathbb{N}$ we have*

$$827 \pi_{k_1+n_1, k_2+n_2}^0 ((\uparrow_{-} \blacklozenge)^{k_1} A) (\lambda x : C. (\uparrow_{-} \blacklozenge)^{k_2} B) \equiv \uparrow_{\mathfrak{A}(n_1, n_2)}^{\mathfrak{A}(n_1+k_1, n_2+k_2)} \blacklozenge (\pi_{n_1, n_2}^0 A (\lambda x : C.B))$$

828 **Proof.** By induction on $k_1 + k_2$, using Proposition 3. ◀

829 In the following, we use the greek letter ρ to refer to rewrite sequences $t \longrightarrow^* u$. Given a
 830 rewrite sequence ρ , we write $\bar{h}\rho$ for the first rewrite rule applied in the head in ρ or $\bar{h}\rho = \perp$
 831 if no step takes place at the head, and we write $\#\rho$ for the total number of rewrite steps in
 832 ρ . For instance, if ρ denotes the sequence

$$833 \text{El}_l ((\lambda x. \uparrow_{l'} \blacklozenge x) u_0) \longrightarrow \text{El}_l (\uparrow_{l'} \blacklozenge u_0) \longrightarrow \text{El}_{(P \ l)} u_0 \longrightarrow U_0$$

834 then we have $\#\rho = 3$ and $\bar{h}\rho = \text{El}_\uparrow$, which is the rule applied in the middle.

835 We can now show that the constant **El** is injective modulo the insertion of some lifts.

836 ▶ **Proposition 34** (Injectivity of **El** modulo lifting). *If $\text{El}_{l_1} A_1 \equiv \text{El}_{l_2} A_2$, where both sides are*
 837 *guarded and well typed, then there are natural numbers k_1, k_2 such that*

- 838 (1) $A_1 \equiv (\uparrow_{-} \blacklozenge)^{k_1} A_0$ and $A_2 \equiv (\uparrow_{-} \blacklozenge)^{k_2} A_0$ for some term A_0 .
 839 (2) $S^{k_1} l_0 \equiv l_1$ and $S^{k_2} l_0 \equiv l_2$ for some term l_0 .

840 **Proof.** First note that, under the hypotheses of the lemma, (1) implies (2). Indeed, by
 841 applying confluence multiple times we obtain a term A' with $A_1 \longrightarrow^* (\uparrow_{-} \blacklozenge)^{k_1} A'$ and
 842 $A_2 \longrightarrow^* (\uparrow_{-} \blacklozenge)^{k_2} A'$, and by subject reduction and inversion of typing we get $\Gamma \vdash A' : U_{l'_1}$
 843 with $S^{k_1} l'_1 \equiv l_1$, and $\Gamma \vdash A' : U_{l'_2}$ with $S^{k_2} l'_2 \equiv l_2$. Therefore, by uniqueness of typing and
 844 injectivity of **U** we have $l'_1 \equiv l'_2$, and so $l_1 \equiv S^{k_1} l'_1$ and $l_2 \equiv S^{k_2} l'_1$.

845 We now proceed to show that the hypotheses imply (1), however when applying the i.h.
 846 we also obtain (2) for free. By confluence we have $\text{El}_{l_1} A_1 \longrightarrow^* B \xleftarrow{*} \text{El}_{l_2} A_2$ for some B .
 847 Let us to refer to the reduction sequence $\text{El}_{l_1} A_1 \longrightarrow^* B$ by ρ_1 and to $\text{El}_{l_2} A_2 \longrightarrow^* B$ by ρ_2 .
 848 We show the result by induction on $\#\rho_1 + \#\rho_2$, and by case analysis on $\bar{h}\rho_1$ and $\bar{h}\rho_2$.

849 ■ Case $\bar{h}\rho_1 = \text{El}_\uparrow$. Then ρ_1 is of the form

$$850 \text{El}_{l_1} A_1 \longrightarrow^* \text{El}_{l'_1} (\uparrow_{l'_1} \blacklozenge A'_1) \longrightarrow \text{El}_{(P \ l'_1)} A'_1 \xrightarrow{\rho'_1} B$$

851 Note that we have $\#\rho'_1 + \#\rho_2 < \#\rho_1 + \#\rho_2$, therefore we can apply the i.h. to deduce
 852 that for some term A_0 and natural numbers k_1, k_2 we have $A'_1 \equiv (\uparrow_{-} \blacklozenge)^{k_1} A_0$ and
 853 $A_2 \equiv (\uparrow_{-} \blacklozenge)^{k_2} A_0$. We therefore have $A_1 \equiv (\uparrow_{-} \blacklozenge)^{(k_1+1)} A_0$ as required.

854 ■ Case $\hbar\rho_1 = \text{El}_\uparrow$. Then ρ_1 is of the form

$$855 \quad \text{El}_{l_1} A_1 \longrightarrow^* \text{El}_{l'_1} (\uparrow_{\underline{n}} A'_1) \longrightarrow \text{El}_{(l'_1 - \underline{n})} A'_1 \xrightarrow{\rho'_1}^* B$$

856 where the first argument \uparrow must be a concrete sort, because it is a reduct of a guarded
857 term. Note that we have $\#\rho'_1 + \#\rho_2 < \#\rho_1 + \#\rho_2$, therefore we can apply the i.h. to
858 deduce that for some term A_0 and natural numbers k_1, k_2 we have $A'_1 \equiv (\uparrow_{-} \blacklozenge)^{k_1} A_0$ and
859 $A_2 \equiv (\uparrow_{-} \blacklozenge)^{k_2} A_0$. Because n is concrete, we have $\uparrow_{\underline{n}} A'_1 \longrightarrow^* (\uparrow_{-} \blacklozenge)^n A'_1$. Therefore,
860 we have $A_1 \equiv (\uparrow_{-} \blacklozenge)^{k'_1} A_0$ by taking $k'_1 = k_1 + n$.

861 The cases $\hbar\rho_2 = \text{El}_\uparrow$ and $\hbar\rho_2 = \text{El}_\uparrow$ are symmetric to the above. Note that if $\hbar\rho_1$ and $\hbar\rho_2$
862 are both different from El_\uparrow and El_\uparrow , then we must have $\hbar\rho_1 = \hbar\rho_2$. Therefore, to conclude
863 the proof it suffices to consider the following three cases:

- 864 ■ Case $\hbar\rho_1 = \hbar\rho_2 = \perp$. Immediate.
865 ■ Case $\hbar\rho_1 = \hbar\rho_2 = \text{El}_u$. For $i = 1, 2$ we can decompose ρ_i as

$$866 \quad \text{El}_{l_i} A_i \longrightarrow^* \text{El}_{l'_i} u_{l'_i} \longrightarrow U_{l'_i} \longrightarrow^* B$$

867 By injectivity of U we have $l'_1 \equiv l'_2$, so by taking $A_0 = u_{l'_i}$ and $k_1 = k_2 = 0$ we conclude.

- 868 ■ Case $\hbar\rho_1 = \hbar\rho_2 = \text{El}_\pi$. For $i = 1, 2$ we can decompose ρ_i as

$$869 \quad \text{El}_{l_i} A_i \longrightarrow^* \text{El}_{l'_i} (\pi_{\frac{m_i}{n_i^a, n_i^b}} A_i^a \lambda x : C_i.A_i^b) \longrightarrow (x : \text{El}_{(\underline{m}_i + \underline{n}_i^a)} A_i^a) \rightarrow \text{El}_{(\underline{m}_i + \underline{n}_i^b)} A_i^b \xrightarrow{\rho'_i}^* B$$

870 where the first arguments of π must be concrete sorts because these are reducts of guarded
871 terms. In the following, we write δ for either a or b . Then it must be the case that B is of
872 the form $(x : B^a) \rightarrow B^b$ and that we can decompose ρ'_1 and ρ'_2 into $\rho_1^\delta, \rho_1^b, \rho_2^\delta, \rho_2^b$ given by

$$873 \quad \text{El}_{(\underline{m}_1 + \underline{n}_1^\delta)} A_1^\delta \xrightarrow{\rho_1^\delta}^* B^\delta \xleftarrow{\rho_2^\delta} \text{El}_{(\underline{m}_2 + \underline{n}_2^\delta)} A_2^\delta$$

874 We have $\#\rho_1^\delta + \#\rho_2^\delta < \#\rho_1 + \#\rho_2$, therefore by i.h. we deduce that for some terms A_0^δ, l_0^δ
875 and natural numbers k_1^δ, k_2^δ we have

- 876 (a) $A_1^\delta \equiv (\uparrow_{-} \blacklozenge)^{k_1^\delta} A_0^\delta$ and $A_2^\delta \equiv (\uparrow_{-} \blacklozenge)^{k_2^\delta} A_0^\delta$
877 (b) $\underline{m}_1 + \underline{n}_1^\delta \equiv S^{k_1^\delta} l_0^\delta$ and $\underline{m}_2 + \underline{n}_2^\delta \equiv S^{k_2^\delta} l_0^\delta$

878 Because $\underline{m}_1 + \underline{n}_1^\delta \longrightarrow^* \underline{m}_1 + \underline{n}_1^\delta$, by confluence it follows that l_0^δ also reduces to a concrete
879 sort $p^\delta \in \mathbb{N}$. We therefore have $\underline{m}_1 + \underline{n}_1^\delta = k_1^\delta + p^\delta$ and $\underline{m}_2 + \underline{n}_2^\delta = k_2^\delta + p^\delta$. Together with
880 the equations from (a), this allows us to show the following for $i = 1, 2$.

$$881 \quad A_i \equiv \pi_{\frac{m_i}{n_i^a, n_i^b}} A_i^a \lambda x : C_i.A_i^b \equiv \pi_{\frac{0}{m_i + n_i^a, m_i + n_i^b}} A_i^a \lambda x : C_i.A_i^b$$

$$882 \quad \equiv \pi_{\frac{0}{k_i^a + p^a, k_i^b + p^b}} ((\uparrow_{-} \blacklozenge)^{k_i^a} A_0^a) (\lambda x : C_i. (\uparrow_{-} \blacklozenge)^{k_i^b} A_0^b)$$

$$883 \quad \equiv \uparrow_{\mathfrak{R}(p^a + k_i^a, p^b + k_i^b)} (\pi_{p^a, p^b}^0 A_0^a (\lambda x : C_i.A_0^b))$$

$$884 \quad \equiv \uparrow_{\mathfrak{R}(p^a, p^b)}$$

885 where the last equation follows from Lemma 33. It suffices now to show that $C_1 \equiv C_2$.
886 To see this, note that by typing constraints we must have $C_i \equiv \text{El}_{\underline{m}_i + \underline{n}_i^a} A_i^a$ and thus

$$887 \quad C_i \equiv \text{El}_{\underline{k}_i^a + p^a} ((\uparrow_{-} \blacklozenge)^{k_i^a} A_0^a) \equiv \text{El}_{\underline{p}^a} A_0^a$$

888 where the right-hand side does not depend on i . ◀

23:26 Impredicativity, Cumulativity and Product Covariance in Dedukti

889 The injectivity of El modulo lifting is then used to establish the injectivity of \Rightarrow .

890 ▶ **Proposition 35** (Injectivity of \Rightarrow). *If $D_1 \Rightarrow l_1 \equiv D_2 \Rightarrow l_2$ and both sides are well typed*
891 *and guarded, then $D_1 \equiv D_2$ and $l_1 \equiv l_2$.*

892 **Proof.** By confluence there is some term B such that $D_1 \Rightarrow l_1 \longrightarrow^* B \ast \longleftarrow D_2 \Rightarrow l_2$. Let
893 us refer to the reduction $D_1 \Rightarrow l_1 \longrightarrow^* B$ by ρ_1 and to the reduction $D_2 \Rightarrow l_2 \longrightarrow^* B$ by ρ_2 .
894 We show the result by induction on $\#\rho_1$. We proceed with a case analysis on $\mathfrak{h}\rho_1$, which by
895 inspection must be equal to $\mathfrak{h}\rho_2$.

896 ■ Case $\mathfrak{h}\rho_1 = \mathfrak{h}\rho_2 = \perp$. It follows that $B = D_0 \Rightarrow l_0$ for some D_0, l_0 with $D_1 \longrightarrow^* D_0 \ast \longleftarrow$
897 D_2 and $l_1 \longrightarrow^* l_0 \ast \longleftarrow D_2$.

898 ■ Case $\mathfrak{h}\rho_1 = \mathfrak{h}\rho_2 = \Rightarrow \blacklozenge$. For $i = 1, 2$, we can decompose ρ_i as

$$899 \quad D_i \Rightarrow l_i \longrightarrow^* \blacklozenge \Rightarrow l'_i \longrightarrow \text{U}_{l'_i} \longrightarrow^* B$$

900 So we have $D_1 \equiv \blacklozenge \equiv D_2$, and from $\text{U}_{l'_1} \equiv \text{U}_{l'_2}$ we deduce $l'_1 \equiv l'_2$ and thus $l_1 \equiv l'_1 \equiv l'_2 \equiv l_2$.

901 ■ Case $\mathfrak{h}\rho_1 = \mathfrak{h}\rho_2 = \Rightarrow \blacktriangleleft$. For $i = 1, 2$ we can decompose ρ_i as

$$902 \quad D_i \Rightarrow l_i \longrightarrow^* (A_i \text{ l}'_i \blacktriangleleft \lambda x : C_i.D'_i) \Rightarrow l'_i \longrightarrow (x : \text{El}_{l'_i} A_i) \rightarrow D'_i \Rightarrow l'_i \longrightarrow^* B$$

903 In the last reduction sequence there can be no other steps in the head, so we must have
904 B of the form $(x : P) \rightarrow Q$ with $\text{El}_{l'_1} A_1 \longrightarrow^* P \ast \longleftarrow \text{El}_{l'_2} A_2$ and

$$905 \quad D'_1 \Rightarrow l'_1 \xrightarrow{\rho'_1} Q \ast \longleftarrow D'_2 \Rightarrow l'_2$$

906 where $\#\rho'_1 < \#\rho_1$. Therefore, by i.h. we deduce $D'_1 \equiv D'_2$ and $l'_1 \equiv l'_2$. Moreover,
907 by inversion of typing in $A_i \text{ l}'_i \blacktriangleleft \lambda x : C_i.D'_i$ we get $C_i \equiv \text{El}_{l'_i} A_i$ and thus $C_1 \equiv C_2$.
908 Finally, by applying Proposition 34 with $\text{El}_{l'_1} A_1 \equiv \text{El}_{l'_2} A_2$ we get $A_1 \equiv (\uparrow _ \blacklozenge)^{k_1} A_0$
909 and $A_2 \equiv (\uparrow _ \blacklozenge)^{k_2} A_0$ and $l'_1 \equiv \text{S}^{k_1} l_0$ and $l'_2 \equiv \text{S}^{k_2} l_0$ for some terms A_0, l_0 and natural
910 numbers k_1, k_2 . Therefore, we conclude

$$\begin{aligned} 911 \quad D_1 \equiv A_1 \text{ l}'_1 \blacktriangleleft (\lambda x : C_1.D'_1) &\equiv ((\uparrow _ \blacklozenge)^{k_1} A_0) (\text{S}^{k_1} l_0) \blacktriangleleft (\lambda x : C_2.D'_2) \\ 912 &\equiv A_0 l_0 \blacktriangleleft (\lambda x : C_2.D'_2) \\ 913 &\equiv ((\uparrow _ \blacklozenge)^{k_2} A_0) (\text{S}^{k_2} l_0) \blacktriangleleft (\lambda x : C_2.D'_2) \\ 914 &\equiv A_2 \text{ l}'_2 \blacktriangleleft (\lambda x : C_2.D'_2) \equiv D_2 \end{aligned} \quad \blacktriangleleft$$

916 E.2 Coherence

917 We first need the following technical lemma, allowing to decompose a telescope D when
918 $D \Rightarrow l$ is convertible to a function type.

919 ▶ **Lemma 36** (Telescope decomposition). *If $D \Rightarrow l \equiv (x : P) \rightarrow Q$ then $D \longrightarrow^* A \text{ l}' \blacktriangleleft \lambda x : C.D'$*
920 *for some A, l', C, D' with $P \equiv \text{El}_{l'} A$ and $Q \equiv D' \Rightarrow l$.*

921 **Proof.** By confluence, we have $D \Rightarrow l \longrightarrow^* B \ast \longleftarrow (x : P) \rightarrow Q$. We must have B of the
922 form $(x : P') \rightarrow Q'$ with $P' \equiv P$ and $Q' \equiv Q$, and we can decompose $D \Rightarrow l \longrightarrow^* B$ as

$$923 \quad D \Rightarrow l \longrightarrow^* (A \text{ l}' \blacktriangleleft \lambda x : C.D') \Rightarrow l'' \longrightarrow (x : \text{El}_{l'} A) \rightarrow D' \Rightarrow l'' \longrightarrow^* (x : P') \rightarrow Q'$$

924 We thus have $D \longrightarrow^* A \text{ l}' \blacktriangleleft \lambda x : C.D'$ and $\text{El}_{l'} A \equiv P' \equiv P$ and $D' \Rightarrow l \equiv Q' \equiv Q$. \blacktriangleleft

925 ▶ **Theorem 18** (Coherence). *Let $t_1, t_2 \in \Lambda_o$ with $\Gamma \vdash t_1 : A_1$ and $\Gamma \vdash t_2 : A_2$. If $|t_1| = |t_2|$*
 926 *then at least one of the following holds:*

- 927 (1) $t_1 \equiv t_2$
 928 (2) $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D t_2 : D \Rightarrow \underline{m}$ and $t_1 \equiv \uparrow_{\underline{n}}^{\underline{m}} D t_2$ for some D guarded
 929 (3) $\Gamma \vdash \uparrow_{\underline{n}}^{\underline{m}} D t_1 : D \Rightarrow \underline{m}$ and $t_2 \equiv \uparrow_{\underline{n}}^{\underline{m}} D t_1$ for some D guarded

930 **Proof.** The proof is by induction on t_1 and t_2 , following the definition of $|-|$.

931 ■ Case $t_1 = \uparrow_{\underline{n}} D u$. By inversion of typing, uniqueness of type and injectivity of function
 932 types, we have $\Gamma \vdash D : \text{Tele}$ and $\Gamma \vdash u : D \Rightarrow \underline{n}$. By i.h. on u and t_2 , we have three cases
 933 to consider.

- 934 (a) $u \equiv t_2$. By confluence, u and t_2 have a common reduct w . Using subject reduction we
 935 know w has both types $D \Rightarrow \underline{n}$ and A_2 so by uniqueness of type, we know $D \Rightarrow \underline{n} \equiv A_2$
 936 so we can conclude that $\Gamma \vdash t_2 : D \Rightarrow \underline{n}$ and thus that $\Gamma \vdash \uparrow_{\underline{n}} D t_2 : D \Rightarrow (\text{S } \underline{n})$.
 937 Knowing that $t_1 \equiv \uparrow_{\underline{n}} D t_2$ by congruence, we conclude.
 938 (b) $\Gamma \vdash \uparrow_{\underline{n}'}^{m'} D' t_2 : D' \Rightarrow \underline{m}'$ and $u \equiv \uparrow_{\underline{n}'}^{m'} D' t_2$. Similarly to above, we can show
 939 $D \Rightarrow \underline{n} \equiv D' \Rightarrow \underline{m}'$ by confluence, subject reduction and uniqueness of type. By
 940 injectivity of \Rightarrow (Proposition 35) we get $D \equiv D'$ and $\underline{n} \equiv \underline{m}'$ which means $n = m'$
 941 given that they are concrete. So $t_1 = \uparrow_{\underline{n}} D u \equiv \uparrow_{\underline{n}} D (\uparrow_{\underline{n}'}^{\underline{n}} D t_2) = \uparrow_{\underline{n}'}^{1+n} D t_2$ by folding
 942 notations. Finally, we have $\Gamma \vdash t_2 : D' \Rightarrow \underline{n}'$, so by conversion we get $\Gamma \vdash t_2 : D \Rightarrow \underline{n}'$
 943 and thus $\Gamma \vdash \uparrow_{\underline{n}'}^{1+n} D t_2 : D \Rightarrow \underline{1+n}$.
 944 (c) $\Gamma \vdash \uparrow_{\underline{n}'}^{m'} D' u : D' \Rightarrow \underline{m}'$ and $t_2 \equiv \uparrow_{\underline{n}'}^{m'} D' u$. This gives us in particular that
 945 $\Gamma \vdash u : D' \Rightarrow \underline{n}'$ so by uniqueness of type we get $D \Rightarrow \underline{n} \equiv D' \Rightarrow \underline{n}'$ and thus $D \equiv D'$
 946 and $n = n'$. If $m' = n$ then we have $t_2 \equiv u$ so we proceed as in case (a), otherwise
 947 $m' \geq 1+n$ so we can conclude with $t_2 \equiv \uparrow_{\underline{n}}^{m'} D u = \uparrow_{\underline{1+n}}^{m'} D (\uparrow_{\underline{n}} D u) = \uparrow_{\underline{1+n}}^{m'} D t_1$
 948 and $\Gamma \vdash \uparrow_{\underline{1+n}}^{m'} D t_1 : D \Rightarrow \underline{m}'$.

949 The case $t_2 = \uparrow_{\underline{n}} D v$ is symmetric to the one above, therefore, in the following, we consider
 950 t_1 and t_2 not headed by \uparrow , in which case the definition of $|-|$ impose that they have the
 951 same head structure. Moreover, when applying the induction hypothesis, the proofs for the
 952 cases (b) and (c) are almost always symmetric, so we only give the proofs for case (c) when
 953 they are not symmetric.

- 954 ■ Case $t_1 = x = t_2$. Trivial.
 955 ■ Case $t_1 = \mathbf{u}_{\underline{n}} = t_2$. Trivial.
 956 ■ Case $t_1 = \pi_{\underline{n}_1, m_1}^0 A_1 (\lambda x : C_1. B_1)$, $t_2 = \pi_{\underline{n}_2, m_2}^0 A_2 (\lambda x : C_2. B_2)$. By inversion we know
 957 that A_1 and A_2 are well typed, of type $\mathbf{U}_{\underline{n}_1}$ and $\mathbf{U}_{\underline{n}_2}$ respectively. We also know that
 958 B_1 and B_2 are well typed—of type $\mathbf{U}_{\underline{m}_1}$ and $\mathbf{U}_{\underline{m}_2}$ respectively—but in *a priori* different
 959 contexts: $\Gamma, x : \mathbf{El}_{\underline{n}_1} A_1$ and $\Gamma, x : \mathbf{El}_{\underline{n}_2} A_2$ (we exploited here inversion and injectivity to
 960 conclude that $C_i \equiv \mathbf{El}_{\underline{n}_i} A_i$). We thus first apply induction hypothesis on A_1 and A_2 :
 961 (a) $A_1 \equiv A_2$. We can thus deduce that $\mathbf{U}_{\underline{n}_1} \equiv \mathbf{U}_{\underline{n}_2}$ and thus that $n_1 = n_2$, which also
 962 implies $C_1 \equiv \mathbf{El}_{\underline{n}_1} A_1 \equiv \mathbf{El}_{\underline{n}_2} A_2 \equiv C_2$. Using context conversion, we type both B_1
 963 and B_2 in context $\Gamma, x : \mathbf{El}_{\underline{n}_1} A_1$ and thus apply induction hypothesis on them:
 964 (a) $B_1 \equiv B_2$. We deduce $m_1 = m_2$ and we can thus conclude $\pi_{\underline{n}_1, m_1}^0 A_1 (\lambda x : C_1. B_1) \equiv$
 965 $\pi_{\underline{n}_2, m_2}^0 A_2 (\lambda x : C_2. B_2)$.

966 (b) $\Gamma, x : \text{El}_{\underline{n}_1} A_1 \vdash \uparrow_{\underline{q}'}^{p'} D' B_2 : D' \Rightarrow \underline{p}'$ and $B_1 \equiv \uparrow_{\underline{q}'}^{p'} D' B_2$. Since $\Gamma, x : \text{El}_{\underline{n}_1} A_1 \vdash B_2 :$
 967 $\underline{U}_{\underline{m}_2}$ we get $D' \Rightarrow \underline{q}' \equiv \underline{U}_{\underline{m}_2}$ which by confluence means that $D' \equiv \blacklozenge$ and $\underline{q}' = \underline{m}_2$.
 968 We thus have $\Gamma, x : \text{El}_{\underline{n}_1} A_1 \vdash \uparrow_{\underline{q}'}^{p'} D' B_2 : \underline{U}_{\underline{p}'}$ and $\Gamma, x : \text{El}_{\underline{n}_1} A_1 \vdash B_1 : \underline{U}_{\underline{m}_1}$ so
 969 $\underline{p}' = \underline{m}_1$. Therefore, we get $B_1 \equiv \uparrow_{\underline{m}_2}^{\underline{m}_1} \blacklozenge B_2$. Together with $C_1 \equiv C_2$ and $A_1 \equiv A_2$
 970 and Lemma 33, we can show $t_1 \equiv \uparrow_{\underline{\mathfrak{R}(n_1, m_2)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_2$. Finally, it is easy to see that
 971 $\Gamma \vdash \uparrow_{\underline{\mathfrak{R}(n_1, m_2)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_2 : \blacklozenge \Rightarrow \underline{\mathfrak{R}(n_1, m_1)}$.

972 (b) $\Gamma \vdash \uparrow_{\underline{q}}^p D A_2 : D \Rightarrow \underline{p}$ and $A_1 \equiv \uparrow_{\underline{q}}^p D A_2$. Since $\Gamma \vdash A_1 : \underline{U}_{\underline{n}_1}$ we deduce $\underline{U}_{\underline{n}_1} \equiv D \Rightarrow \underline{p}$
 973 which means that $\underline{D} \equiv \blacklozenge$ and $\underline{p} = \underline{n}_1$. Since $\Gamma \vdash A_2 : \underline{U}_{\underline{n}_2}$ we also get that $\underline{q} = \underline{n}_2$.
 974 Hence we have $A_1 \equiv \uparrow_{\underline{n}_2}^{\underline{n}_1} \blacklozenge A_2$ and thus $\text{El}_{\underline{n}_1} A_1 \equiv \text{El}_{\underline{n}_1} (\uparrow_{\underline{n}_2}^{\underline{n}_1} \blacklozenge A_2) \equiv \text{El}_{\underline{n}_2} A_2$. We
 975 can thus apply our induction hypothesis on B_1 and B_2 :

976 (a) $B_1 \equiv B_2$. We obtain $\underline{m}_1 = \underline{m}_2$, thus by combining $C_1 \equiv C_2$ and $A_1 \equiv \uparrow_{\underline{n}_2}^{\underline{n}_1} \blacklozenge A_2$ and
 977 Lemma 33, we can show $t_1 \equiv \uparrow_{\underline{\mathfrak{R}(n_2, m_1)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_2$, and moreover it is easy to see that
 978 $\Gamma \vdash \uparrow_{\underline{\mathfrak{R}(n_2, m_1)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_2 : \blacklozenge \Rightarrow \underline{\mathfrak{R}(n_1, m_1)}$.

979 (b) $\Gamma \vdash \uparrow_{\underline{q}'}^{p'} D' B_2 : D' \Rightarrow \underline{p}'$ and $B_1 \equiv \uparrow_{\underline{q}'}^{p'} D' B_2$. As before, we obtain $B_1 \equiv \uparrow_{\underline{m}_2}^{\underline{m}_1} \blacklozenge B_2$,
 980 so by combining $C_1 \equiv C_2$ and $A_1 \equiv \uparrow_{\underline{n}_1}^{\underline{n}_2} \blacklozenge A_2$ and Lemma 33, we can show
 981 $t_1 \equiv \uparrow_{\underline{\mathfrak{R}(n_2, m_2)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_2$, and we indeed have $\Gamma \vdash \uparrow_{\underline{\mathfrak{R}(n_2, m_2)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_2 : \blacklozenge \Rightarrow \underline{\mathfrak{R}(n_1, m_1)}$.

982 (c) $\Gamma \vdash \uparrow_{\underline{q}'}^{p'} D' B_1 : D' \Rightarrow \underline{p}'$ and $B_2 \equiv \uparrow_{\underline{q}'}^{p'} D' B_1$. As before, we obtain $B_2 \equiv \uparrow_{\underline{m}_1}^{\underline{m}_2} \blacklozenge B_1$,
 983 so by combining $C_1 \equiv C_2$ and $A_1 \equiv \uparrow_{\underline{n}_2}^{\underline{n}_1} \blacklozenge A_2$ and Lemma 33, we can show $t_1 \equiv$
 984 $\uparrow_{\underline{\mathfrak{R}(n_2, m_1)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge (\pi_{\underline{n}_2, \underline{m}_1}^0 A_2 (\lambda x : C_2. B_1))$ and $t_2 \equiv \uparrow_{\underline{\mathfrak{R}(n_2, m_1)}}^{\underline{\mathfrak{R}(n_2, m_2)}} \blacklozenge (\pi_{\underline{n}_2, \underline{m}_1}^0 A_2 (\lambda x : C_2. B_1))$.
 985 If $\underline{\mathfrak{R}(n_2, m_2)} = \underline{\mathfrak{R}(n_1, m_1)}$ then it follows that $t_1 \equiv t_2$. Otherwise, suppose wlog
 986 that $\underline{\mathfrak{R}(n_2, m_2)} > \underline{\mathfrak{R}(n_1, m_1)}$, the other case being symmetric. Then we conclude
 987 $t_2 \equiv \uparrow_{\underline{\mathfrak{R}(n_2, m_2)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_1$ and it is easy to see that $\Gamma \vdash \uparrow_{\underline{\mathfrak{R}(n_2, m_2)}}^{\underline{\mathfrak{R}(n_1, m_1)}} \blacklozenge t_1 : \blacklozenge \Rightarrow \underline{\mathfrak{R}(n_1, m_1)}$.

988 ■ Case $t_1 = u_1 v_1$ and $t_2 = u_2 v_2$. By inversion we have $\Gamma \vdash u_i : (x : A_i) \rightarrow B_i$ and
 989 $\Gamma \vdash v_i : A_i$. We first apply induction hypothesis on u_1 and u_2 :

990 (a) $u_1 \equiv u_2$. We thus get $A_1 \equiv A_2$ and $B_1 \equiv B_2$. Looking at the induction hypothesis on
 991 v_1 and v_2 , in all cases we must have $v_1 \equiv v_2$. Indeed, if we are in cases (2) or (3) then
 992 we get $A_1 \equiv D \Rightarrow \underline{p}$ and $A_2 \equiv D \Rightarrow \underline{q}$, but together with $A_1 \equiv A_2$ this implies $\underline{p} = \underline{q}$,
 993 meaning that no lifts are inserted between v_1 and v_2 . We thus conclude that $t_1 \equiv t_2$.

994 (b) $\Gamma \vdash \uparrow_{\underline{n}}^m D u_2 : D \Rightarrow \underline{m}$ and $u_1 \equiv \uparrow_{\underline{n}}^m D u_2$. Now, $(x : A_1) \rightarrow B_1 \equiv D \Rightarrow \underline{m}$, so
 995 by Lemma 36 we have $D \rightarrow^* a \blacktriangleleft \lambda x : C. D'$ with $\text{El}_l a \equiv A_1$ and $B_1 \equiv D' \Rightarrow \underline{m}$.
 996 Moreover, we also get that $\text{El}_l a \equiv A_2$ and $B_2 \equiv D' \Rightarrow \underline{n}$. We are again in a situation
 997 where v_1 and v_2 share a type, so by the same arguments as in case (a) the i.h. gives
 998 $v_1 \equiv v_2$. Therefore, we have

$$999 \quad t_1 = u_1 v_1 \equiv (\uparrow_{\underline{n}}^m (a \blacktriangleleft \lambda x : C. D') u_2) v_2 \equiv \uparrow_{\underline{n}}^m D'[v_2/x] (u_2 v_2) = \uparrow_{\underline{n}}^m D'[v_2/x] t_2$$

1001 For typing, we have $\Gamma \vdash t_2 : B_2[v_2/x]$ so by conversion we have $\Gamma \vdash t_2 : D'[v_2/x] \Rightarrow \underline{n}$
 1002 and thus $\Gamma \vdash \uparrow_{\underline{n}}^m D'[v_2/x] t_2 : D'[v_2/x] \Rightarrow \underline{m}$.

1003 ■ Case $t_1 = \lambda x : \text{El}_{\underline{n}_1} A_1. u_1$ and $t_2 = \lambda x : \text{El}_{\underline{n}_2} A_2. u_2$. By inversion we have $\Gamma \vdash A_i : \underline{U}_{\underline{n}_i}$
 1004 and $\Gamma, x : \text{El}_{\underline{n}_i} A_i \vdash u_i : B_i$. We first show $\text{El}_{\underline{n}_1} A_1 \equiv \text{El}_{\underline{n}_2} A_2$ by applying induction
 1005 hypothesis on A_1 and A_2 :

- 1006 (a) $A_1 \equiv A_2$. Immediate.
 1007 (b) $\Gamma \vdash \uparrow_{\underline{q}}^p D A_2 : D \Rightarrow \underline{p}$ and $A_1 \equiv \uparrow_{\underline{q}}^p D A_2$. Since $\Gamma \vdash A_1 : \underline{U}_{n_1}$ we get $\underline{U}_{n_1} \equiv$
 1008 $D \Rightarrow \underline{p}$ which entails $D \equiv \blacklozenge$ and $p = n_1$. Similarly, $q = n_2$. Therefore, $\underline{\text{El}}_{\underline{n_1}} A_1 \equiv$
 1009 $\underline{\text{El}}_{\underline{n_1}} (\uparrow_{\underline{n_2}}^{\underline{n_1}} \blacklozenge A_2) \equiv \underline{\text{El}}_{\underline{n_2}} A_2$ which is the wanted result.

1010 We now have $\Gamma, x : \underline{\text{El}}_{\underline{n_1}} A_1 \vdash u_i : B_i$ and can thus apply the i.h. on u_1 and u_2 :

- 1011 (a) $u_1 \equiv u_2$. We conclude $t_1 \equiv t_2$.
 1012 (b) $\Gamma, x : \underline{\text{El}}_{\underline{n_1}} A_1 \vdash \uparrow_{\underline{q}}^p D u_2 : D \Rightarrow \underline{p}$ and $u_1 \equiv \uparrow_{\underline{q}}^p D u_2$. Let us define $D' := A_2 \underline{n_2} \blacktriangleleft (\lambda x :$
 1013 $\underline{\text{El}}_{\underline{n_2}} A_2.D)$. We have

$$\begin{aligned}
 1014 \quad t_1 &= \lambda x : \underline{\text{El}}_{\underline{n_1}} A_1.u_1 \equiv \lambda x : \underline{\text{El}}_{\underline{n_2}} A_2.\uparrow_{\underline{q}}^p D u_2 \\
 1015 &\equiv \uparrow_{\underline{q}}^p (A_2 \underline{n_2} \blacktriangleleft (\lambda x : \underline{\text{El}}_{\underline{n_2}} A_2.D)) \lambda x : \underline{\text{El}}_{\underline{n_2}} A_2.u_2 \\
 1016 &= \uparrow_{\underline{q}}^p D' t_2
 \end{aligned}$$

1018 As for the typing, we have $\Gamma, x : \underline{\text{El}}_{\underline{n_2}} A_2 \vdash u_2 : D \Rightarrow \underline{q}$ so $\Gamma \vdash t_2 : (x : \underline{\text{El}}_{\underline{n_2}} A_2) \rightarrow$
 1019 $D \Rightarrow \underline{q}$ so by conversion $\Gamma \vdash t_2 : D' \Rightarrow \underline{q}$ hence we conclude $\Gamma \vdash \uparrow_{\underline{q}}^p D' t_2 : D' \Rightarrow \underline{p}$. \blacktriangleleft

1020 E.3 Soundness

1021 \blacktriangleright **Lemma 21** (Computing the El of a translation). *Let $A \in \Lambda_o$ with $\text{El}_l A$ well typed.*

- 1022 1. *If $|A| = n$ then $\text{El}_l A \rightarrow^* \underline{U}_n$.*
 1023 2. *If $|A| = \Pi x : A_1.A_2$ then $\text{El}_l A \rightarrow^* (x : \underline{\text{El}}_{\underline{n_1}} A'_1) \rightarrow \underline{\text{El}}_{\underline{n_2}} A'_2$ with $|A'_i| = A_i$.*

1024 **Proof.** For the first part, note that by definition of $|-|$, if $|A| = n$ then A is of the form
 1025 $\uparrow_{\underline{n_1}} D_1 (\dots (\uparrow_{\underline{n_k}} D_k u_n) \dots)$ for some k . By typing constraints we can then deduce $D_i \equiv \blacklozenge$ for all
 1026 i , which by confluence gives $D_i \rightarrow^* \blacklozenge$. We conclude $\text{El}_l A \rightarrow^* \text{El}_l ((\uparrow_{\underline{}} \blacklozenge)^k u_n) \rightarrow^* \underline{U}_n$.

1027 For the second part, by definition of $|-|$, if $|A| = \Pi x : A_1.A_2$ then we must have A of the
 1028 form $\uparrow_{\underline{n_1}} D_1 (\dots (\uparrow_{\underline{n_k}} D_k (\pi_{\underline{n_1}, \underline{n_2}}^0 A'_1 \lambda x : P.A'_2)) \dots)$ with $|A'_i| = A_i$. By inversion of typing,
 1029 we must have $D_i \equiv \blacklozenge$ for all i , which by confluence gives $D_i \rightarrow^* \blacklozenge$. We then conclude
 1030 $\text{El}_l A \rightarrow^* \text{El}_l ((\uparrow_{\underline{}} \blacklozenge)^k (\pi_{\underline{n_1}, \underline{n_2}}^0 A'_1 \lambda x : P.A'_2)) \rightarrow^* (x : \underline{\text{El}}_{\underline{n_1}} A'_1) \rightarrow \underline{\text{El}}_{\underline{n_2}} A'_2$. \blacktriangleleft

1031 \blacktriangleright **Lemma 22** (Telescope translation). *Let $A_1, A_2 \in \Lambda_o$ with $\Gamma \vdash A_i : \underline{U}_{n_i}$. If $|A_i| = \Delta \Rightarrow m_i$
 1032 for some $m_1 \leq m_2$, then we have $\underline{\text{El}}_{\underline{n_i}} A_i \equiv D \Rightarrow \underline{m_i}$ for some guarded D with $\Gamma \vdash D : \text{Tele}$.*

1033 **Proof.** We prove this by induction on Δ .

- 1034 \blacksquare $\Delta = \cdot$. In this case $|A_i| = m_i$, so by Lemma 21 we get $\underline{\text{El}}_{\underline{n_i}} A_i \equiv \underline{U}_{\underline{m_i}} \equiv \blacklozenge \Rightarrow \underline{m_i}$.
 1035 \blacksquare $\Delta = x : B, \Delta'$. We have $|A_i| = \Pi x : B.\Delta' \Rightarrow m_i$, so by Lemma 21 we get $\underline{\text{El}}_{\underline{n_i}} A_i \rightarrow^* (x :$
 1036 $\underline{\text{El}}_{\underline{p_i}} B_i) \rightarrow \underline{\text{El}}_{\underline{q_i}} A'_i$ with $|B_i| = B$ and $|A'_i| = \Delta' \Rightarrow m_i$. By inversion of typing we have
 1037 $\Gamma \vdash B_i : \underline{U}_{\underline{p_i}}$ and because $|B_1| = |B_2|$ then Proposition 20 gives $\underline{\text{El}}_{\underline{p_1}} B_1 \equiv \underline{\text{El}}_{\underline{p_2}} B_2$. By
 1038 inversion of typing once more and conversion in context, we get $\Gamma, x : \underline{\text{El}}_{\underline{p_1}} B_1 \vdash A'_i : \underline{U}_{\underline{q_i}}$.
 1039 Now we can apply the i.h. to obtain a guarded D satisfying $\Gamma, x : \underline{\text{El}}_{\underline{p_1}} B_1 \vdash D : \text{Tele}$
 1040 and $\underline{\text{El}}_{\underline{q_i}} A'_i \equiv D \Rightarrow \underline{m_i}$. By taking $D' := B_1 \underline{p_1} \blacktriangleleft \lambda x : \underline{\text{El}}_{\underline{p_1}} B_1.D$ we can now show
 1041 $\Gamma \vdash D' : \text{Tele}$ and $\underline{\text{El}}_{\underline{n_i}} A_i \equiv (x : \underline{\text{El}}_{\underline{p_1}} B_1) \rightarrow D \Rightarrow \underline{m_i} \equiv D' \Rightarrow \underline{m_i}$ as required. \blacktriangleleft

1042 \blacktriangleright **Theorem 23** (Soundness). *If $\Gamma \vdash_{\text{cc}} t : A$ then we have $\Gamma' \vdash t' : \underline{\text{El}}_{\underline{n}} A'$ for some $\Gamma' \in \text{Ctx}_o$
 1043 and $t', A' \in \Lambda_o$ and $n \in \mathbb{N}$ with $\|\Gamma'\| = \Gamma$ and $|t'| = t$ and $|A'| = A$.*

1044 **Proof.** We instead show the following two points, which together imply the theorem.

- 1045 \blacksquare If $\Gamma \vdash_{\text{cc}}$ then $\Gamma' \vdash$ for some $\Gamma' \in \text{Ctx}_o$ with $\|\Gamma'\| = \Gamma$.

23:30 Impredicativity, Cumulativity and Product Covariance in Dedukti

1046 ■ If $\Gamma \vdash_{\text{cc}} t : A$ and $\Gamma' \vdash$ for some $\Gamma' \in \text{Ctx}_o$ with $\|\Gamma'\| = \Gamma$ then $\Gamma' \vdash t' : \text{El}_{\underline{n}} A'$ for some
 1047 $n \in \mathbb{N}$ and $A', t' \in \Lambda_o$ with $|A'| = A$ and $|t'| = t$.

1048 We prove them by induction on the derivation of $\Gamma \vdash_{\text{cc}}$ or $\Gamma \vdash_{\text{cc}} t : A$.

1049 ■ Case

$$1050 \quad \frac{\text{EMPTYCTX}}{\cdot \vdash_{\text{cc}}}$$

1051 The empty context in the target works.

1052 ■ Case

$$1053 \quad \frac{\text{EXTCTX} \quad \Gamma \vdash_{\text{cc}} \quad \Gamma \vdash_{\text{cc}} A : n}{\Gamma, x : A \vdash_{\text{cc}}}$$

1054 By induction hypothesis we get Γ' with $|\Gamma'| = \Gamma$ and $\Gamma' \vdash$. Then by i.h. again we have
 1055 $\Gamma' \vdash A' : \text{El}_{\underline{m}} B$ with $|A'| = A$ and $|B| = n$, so by Lemma 21 we get $\Gamma' \vdash A' : \text{U}_{\underline{n}}$. Thus
 1056 $\Gamma' \vdash \text{El}_{\underline{n}} A' : \mathbf{Type}$ and we can conclude $\Gamma', x : \text{El}_{\underline{n}} A' \vdash$.

1057 ■ Case

$$1058 \quad (x : A) \in \Gamma \quad \frac{\text{VAR} \quad \Gamma \vdash_{\text{cc}}}{\Gamma \vdash_{\text{cc}} x : A}$$

1059 Assuming $\Gamma' \vdash$ and $\|\Gamma'\| = \Gamma$, we know that $(x : A) \in \|\Gamma'\|$ meaning there exists some A'
 1060 and n such that $(x : \text{El}_{\underline{n}} A') \in \Gamma'$ and $|A'| = A$. We thus conclude with $\Gamma' \vdash x : \text{El}_{\underline{n}} A'$.

1061 ■ Case

$$1062 \quad \frac{\text{SORT} \quad \Gamma \vdash_{\text{cc}}}{\Gamma \vdash_{\text{cc}} n : \mathfrak{A}(n)}$$

1063 Follows because we have $\Gamma' \vdash \underline{u}_n : \text{El}_{\mathfrak{A}^2(n)} \underline{u}_{\mathfrak{A}(n)}$ with $|\underline{u}_n| = n$ and $|\underline{u}_{\mathfrak{A}(n)}| = \mathfrak{A}(n)$.

1064 ■ Case

$$1065 \quad \frac{\text{PI} \quad \Gamma \vdash_{\text{cc}} A : n \quad \Gamma, x : A \vdash_{\text{cc}} B : m}{\Gamma \vdash_{\text{cc}} \Pi x : A. B : \mathfrak{R}(n, m)}$$

1066 By i.h. and Lemma 21 we have $\Gamma' \vdash A' : \text{U}_{\underline{n}}$ and $|A'| = A$. Therefore we have
 1067 $\Gamma', x : \text{El}_{\underline{n}} A' \vdash$, so by i.h. and Lemma 21 again we obtain $\Gamma', x : \text{El}_{\underline{n}} A' \vdash B' : \text{U}_{\underline{m}}$ with
 1068 $|B'| = B$. We then derive $\Gamma' \vdash \pi_{\underline{n}, \underline{m}}^0 A' (\lambda x : \text{El}_{\underline{n}} A'. B') : \text{El}_{\mathfrak{A}(\mathfrak{R}(n, m))} \underline{u}_{\mathfrak{R}(n, m)}$ to conclude.

1069 ■ Case

$$1070 \quad \frac{\text{LAM} \quad \Gamma \vdash_{\text{cc}} A : n \quad \Gamma, x : A \vdash_{\text{cc}} t : B}{\Gamma \vdash_{\text{cc}} \lambda x : A. t : \Pi x : A. B}$$

1071 By i.h. and Lemma 21 we have $\Gamma' \vdash A' : \text{U}_{\underline{n}}$ and $|A'| = A$. Therefore we have
 1072 $\Gamma', x : \text{El}_{\underline{n}} A' \vdash$, so by i.h. we get $\Gamma', x : \text{El}_{\underline{n}} A' \vdash t' : \text{El}_{\underline{m}} B'$ for some m and with $|t'| = t$
 1073 and $|B'| = B$. By inversion, we then deduce $\Gamma', x : \text{El}_{\underline{n}} A' \vdash B' : \text{U}_{\underline{m}}$. We can now
 1074 show $\Gamma' \vdash \lambda x : \text{El}_{\underline{n}} A'. t' : (x : \text{El}_{\underline{n}} A') \rightarrow \text{El}_{\underline{m}} B'$ and because its type is convertible to
 1075 $\text{El}_{\mathfrak{R}(n, m)} (\pi_{\underline{n}, \underline{m}}^0 A' (\lambda x : \text{El}_{\underline{n}} A'. B'))$, which is well typed, we conclude by applying the
 1076 conversion rule.

1077 ■ Case

$$1078 \quad \frac{\text{APP} \quad \Gamma \vdash_{\text{cc}} t : \Pi x : A.B \quad \Gamma \vdash_{\text{cc}} u : A}{\Gamma \vdash_{\text{cc}} t u : B[u/x]}$$

1079 By induction hypothesis we have $\Gamma' \vdash t' : \text{El}_p C'$ and $\Gamma' \vdash u' : \text{El}_q A'$ for some t', C', u', A'
 1080 with $|t'| = t$ and $|C'| = \Pi x : A.B$ and $|u'| = u$ and $|A'| = A$. By Lemma 21 we get
 1081 $\Gamma' \vdash t' : (x : \text{El}_n A'') \rightarrow \text{El}_m B'$ for some A'', B' with $|A''| = A$ and $|B'| = B$, and
 1082 inversion of typing gives $\Gamma' \vdash A'' : \text{U}_n$ and $\Gamma', x : \text{El}_n A'' \vdash B' : \text{U}_m$. We also have
 1083 $\Gamma' \vdash A' : \text{U}_q$, so because $|A'| = |A''|$ we can apply Proposition 20 to get $\text{El}_n A'' \equiv \text{El}_q A'$.
 1084 So we have $\Gamma' \vdash u' : \text{El}_n A''$, and therefore $\Gamma' \vdash t' u' : (\text{El}_m B')[u'/x]$. Since m is concrete,
 1085 this type is in fact $\text{El}_m B'[u'/x]$, and Proposition 17 ensures that $|B'[u'/x]| = B[u/x]$.

1086 ■ Case

$$1087 \quad A \subseteq B \quad \frac{\text{CONV} \quad \Gamma \vdash_{\text{cc}} t : A \quad \Gamma \vdash_{\text{cc}} B : n}{\Gamma \vdash_{\text{cc}} t : B}$$

1088 By induction hypothesis we have $\Gamma' \vdash t' : \text{El}_m A'$ and $\Gamma' \vdash B' : \text{U}_n$ with $|t'| = t$, $|A'| = A$
 1089 and $|B'| = B$ (using Lemma 21 for the second derivation). By inversion we obtain
 1090 $\Gamma' \vdash A' : \text{U}_m$. We now use Lemma 2 to split $A \subseteq B$ into two cases:

- 1091 ■ $A \equiv B$. We have $|A'| \equiv |B'|$ so by Proposition 20 we conclude $\text{El}_m A' \equiv \text{El}_n B'$, and
 1092 thus $\Gamma' \vdash t' : \text{El}_n B'$.
- 1093 ■ $A \xrightarrow{*} \Delta \Rightarrow p$ and $B \xrightarrow{*} \Delta \Rightarrow q$ with $p \leq q$. We apply Lemma 19 on A' to get some
 1094 A'' such that $|A''| = \Delta \Rightarrow p$ and $A' \xrightarrow{*} A''$. Similarly, we get B'' with $|B''| = \Delta \Rightarrow q$
 1095 and $B' \xrightarrow{*} B''$. We can then apply Lemma 22 to obtain a guarded term D such that
 1096 $\Gamma' \vdash D : \text{Tele}$ and $\text{El}_m A'' \equiv D \Rightarrow p$ and $\text{El}_n B'' \equiv D \Rightarrow q$. We can now conclude with
 1097 $\Gamma' \vdash \uparrow_{\frac{q}{p}} D t : \text{El}_n B'$. ◀

1098 **F** Omitted proofs of Section 8

1099 ▶ **Theorem 25** (Conservativity for object terms). *Let $\Gamma \in \text{Ctx}_o$ and $A \in \Lambda_o$ with $\|\Gamma\| \vdash_{\text{cc}} |A| : n$
 1100 for some n . If $\Gamma \vdash t : \text{El}_n A$ with t an object term, then we have $\|\Gamma\| \vdash_{\text{cc}} |t| : |A|$.*

1101 **Proof.** We instead show the following claim.

1102 ▷ **Claim 37.** Let $\Gamma \vdash t : A$ with $\Gamma \in \text{Ctx}_o$ and $\|\Gamma\| \vdash_{\text{cc}}$. If t is an object term, then there
 1103 exists $A' \in \Lambda_o^\bullet$ with $A \equiv A'$ and $\|\Gamma\| \vdash_{\text{cc}} |t| : |A'|^\bullet$.

1104 First note that this implies the statement of the theorem. Indeed, by the claim we have
 1105 $\|\Gamma\| \vdash_{\text{cc}} |t| : |B|^\bullet$ for some $B \in \Lambda_o^\bullet$ with $B \equiv \text{El}_n A$. Therefore $|B|^\bullet \equiv |A|^\bullet = |A|$, so we
 1106 conclude $\|\Gamma\| \vdash_{\text{cc}} |t| : |A|$ by the conversion rule.

1107 We proceed with the proof of the claim, by induction on t , following the definition of Λ_o .

- 1108 ■ Case $t = x$. By inversion we have $x : \text{El}_n B \in \Gamma$ with $A \equiv \text{El}_n B$. Therefore we have
 1109 $x : |B| \in \|\Gamma\|$, so by the variable rule we get $\|\Gamma\| \vdash_{\text{cc}} x : |B|$ and so $\|\Gamma\| \vdash_{\text{cc}} x : |\text{El}_n B|^\bullet$.
- 1110 ■ Case $t = \underline{u}_m$. Then by inversion we have $A \equiv \text{U}_{\underline{\alpha}(m)}$, and we can easily show $\|\Gamma\| \vdash_{\text{cc}} m :
 1111 |\text{U}_{\underline{\alpha}(m)}|^\bullet$.

23:32 Impredicativity, Cumulativity and Product Covariance in Dedukti

- 1112 ■ Case $t = \lambda x : \mathbf{El}_n A_1.u$. By inversion we have $\Gamma \vdash A_1 : \mathbf{U}_n$ and $\Gamma, x : \mathbf{El}_n A_1 \vdash u : A_2$
1113 for some A_2 with $A \equiv (x : \mathbf{El}_n A_1) \rightarrow A_2$. By i.h. we thus have $\|\Gamma\| \vdash_{\text{cc}} |A_1| : |B_1|^\bullet$ with
1114 $B_1 \equiv \mathbf{U}_n$. Therefore, we have $|B_1|^\bullet \equiv n$, so by conversion we can derive $\|\Gamma\| \vdash_{\text{cc}} |A_1| : n$,
1115 and so $\|\Gamma\|, x : |A_1| \vdash_{\text{cc}}$. By i.h. once more, we have $\|\Gamma\|, x : |A_1| \vdash_{\text{cc}} |u| : |B_2|^\bullet$ for some
1116 B_2 with $B_2 \equiv A_2$. We can thus derive $\|\Gamma\| \vdash_{\text{cc}} \lambda x : |A_1|. |u| : |(x : \mathbf{El}_n A_1) \rightarrow B_2|^\bullet$ and
1117 $A \equiv (x : \mathbf{El}_n A_1) \rightarrow B_2$.
- 1118 ■ Case $t = u v$. By inversion of typing we have $\Gamma \vdash u : (x : A_1) \rightarrow A_2$ and $\Gamma \vdash v : A_1$
1119 and $A \equiv A_2[v/x]$. By i.h. we thus have $\|\Gamma\| \vdash_{\text{cc}} |u| : |B|^\bullet$ with $B \equiv (x : A_1) \rightarrow A_2$,
1120 and $\|\Gamma\| \vdash_{\text{cc}} |v| : |C|^\bullet$ with $C \equiv A_1$. Using confluence multiple times, it follows that
1121 $B \longrightarrow^* (x : A'_1) \rightarrow A'_2$ and $C \longrightarrow^* A'_1$ for some $A'_1 \equiv A_1$ and $A'_2 \equiv A_2$. Therefore
1122 $\|\Gamma\| \vdash_{\text{cc}} |u| : \Pi x : |A'_1|^\bullet. |A'_2|^\bullet$ and $\|\Gamma\| \vdash_{\text{cc}} |v| : |A'_1|^\bullet$, allowing us to deduce $\|\Gamma\| \vdash_{\text{cc}} |v| :$
1123 $|A'_2[v/x]|^\bullet$ and such that $A \equiv A_2[v/x] \equiv A'_2[v/x]$.
- 1124 ■ Case $t = \uparrow_{\underline{m}} D u$. By inversion we have $\Gamma \vdash u : D \Rightarrow \underline{m}$ and $A \equiv D \Rightarrow \underline{m+1}$. By i.h. we
1125 get $\|\Gamma\| \vdash_{\text{cc}} |u| : |B|^\bullet$ with $B \equiv D \Rightarrow \underline{m}$. By confluence we have $B \longrightarrow^* P^* \longleftarrow D \Rightarrow \underline{m}$,
1126 and because B is in the domain of $|\cdot|^\bullet$ then so is P , and because it is also a reduct of
1127 $D \Rightarrow \underline{m}$ it follows that it must be of the form

$$(x_1 : A_1) \rightarrow \dots \rightarrow (x_k : A_k) \rightarrow \mathbf{U}_{\underline{m}}$$

- 1128
- 1129 Therefore we have $|B|^\bullet \longrightarrow^* \Pi x_1 : |A_1|^\bullet \dots x_k : |A_k|^\bullet. m$ and thus $\|\Gamma\| \vdash_{\text{cc}} |u| : \Pi x_1 :$
1130 $|A_1|^\bullet \dots x_k : |A_k|^\bullet. m$. Because this type is well typed, it is easy to see that $\Pi x_1 : |A_1|^\bullet \dots x_k :$
1131 $|A_k|^\bullet. m+1$ also is, and thus by rule Cumul we get $\|\Gamma\| \vdash_{\text{cc}} |u| : \Pi x_1 : |A_1|^\bullet \dots x_k : |A_k|^\bullet. m+1$.
1132 Now we conclude by noticing that if $D \Rightarrow \underline{m} \longrightarrow^* (x_1 : A_1) \rightarrow \dots \rightarrow (x_k : A_k) \rightarrow \mathbf{U}_{\underline{m}}$,
1133 then it must be the case that $D \Rightarrow \underline{m+1} \longrightarrow^* (x_1 : A_1) \rightarrow \dots \rightarrow (x_k : A_k) \rightarrow \mathbf{U}_{\underline{m+1}}$.
- 1134 ■ Case $t = \pi_{\underline{p}, \underline{q}}^0 A_1 (\lambda x : C.A_2)$. By inversion we have $\Gamma \vdash A_1 : \mathbf{U}_{\underline{p}}$ and $\Gamma, x : \mathbf{El}_{\underline{p}} A_1 \vdash$
1135 $A_2 : \mathbf{U}_{\underline{q}}$ with $A \equiv \mathbf{U}_{\mathfrak{R}(\underline{p}, \underline{q})}$. So by i.h. we have $\|\Gamma\| \vdash_{\text{cc}} |A_1| : |A'_1|^\bullet$ with $A'_1 \equiv \mathbf{U}_{\underline{p}}$, so
1136 $|A'_1|^\bullet \equiv p$ and by the conversion rule we get $\|\Gamma\| \vdash_{\text{cc}} |A_1| : p$ and so $\|\Gamma\|, x : |A_1| \vdash_{\text{cc}}$.
1137 By the i.h. once more, we also have $\|\Gamma\|, x : |A_1| \vdash_{\text{cc}} |A_2| : |A'_2|^\bullet$ with $A'_2 \equiv \mathbf{U}_{\underline{q}}$, thus
1138 $|A'_2|^\bullet \equiv q$ and by the conversion rule we have $\|\Gamma\|, x : |A_1| \vdash_{\text{cc}} |A_2| : q$. Therefore, we can
1139 derive $\|\Gamma\| \vdash \Pi x : |A_1|. |A_2| : |\mathbf{U}_{\mathfrak{R}(\underline{p}, \underline{q})}|^\bullet$ and we indeed have $A \equiv \mathbf{U}_{\mathfrak{R}(\underline{p}, \underline{q})}$. ◀