Modal functional interpretations

(variant with extended⋆ Appendix)

Mircea-Dan Hernest[1,⋆⋆] and Trifon Trifonov[2,⋆⋆⋆]

[1] The “Simion Stoilow” Institute of Mathematics of the Romanian Academy
danhernest@gmail.com
[2] Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”
triffon@fmi.uni-sofia.bg

Abstract. We extend our light Dialectica interpretation [10] to usual and light modal formulas and prove it sound for pseudo-modal arithmetics based on Gödel's T and classical S_4. The range of this *light modal Dialectica* interpretation is the usual (non-modal) classical Arithmetic in all finite types. We also illustrate the use of the new tools for optimized program synthesis with new examples.

This recent work comes in addition to the program extraction technology outlined in our previous paper [10] by adding a useful device for combining the effect of previous optimizations by semi- and non-computational quantifiers in a compact one-step `content eraser`, namely the modal operator $\Box$ (and its weak co-modality $\diamond^c \equiv \neg\Box\neg$). Beside the seemingly cosmetic improvement, we bring the following new result: *while the modal propositional axioms of system S_4 are realizable, the defining axiom of S_5 is generally not realizable under (light) modal Dialectica.*

The use and interpretation of modal operators in this paper were inspired by work of Oliva (partly joint with the first author, see [9]) at the linear logic sublevel, see [14,15]. It is no coincidence that, at formulas level, our interpretation of $\Box A$ is syntactically the same as Oliva's modified realizability interpretation of $!A$ in intuitionistic linear logic. However, a bureaucratic detour would be needed in order to simulate $\Box A$ in terms of $!A$, which seems less suitable for an efficient computer implementation.

The second author independently noticed the possibility of using the same supra-linear modal operators for light program extraction in [18], see also [19]. However, the initiative of studying the full employment of $\Box$ for more efficient program synthesis in the formal context of a classical first-order modal logic (in the sense of Schütte, [16]) belongs to the first author. As we will see, for our extractive purposes it is useful to depart from Schütte's original semantics for quantified modal logic. E.g., the propositional fragments of our first-order modal systems are no longer modal, but purely boolean, as

⋆ In the present paper we give a more detailed treatment of the induction for naturals and we correct the *typo* in the definition of the weak compatibility rule `CMP`: on page 1382 of [10], it is s instead of x and t instead of y; we also give the treatment of `CMP` under Dialectica light.

⋆⋆ The first author gratefully acknowledges financial support by the Romanian National Agency for Scientific Research grant PN-II-RU-TE-2011-3-0122.

⋆⋆⋆ The second author gratefully acknowledges the financial support of the Bulgarian National Science Fund within project DTK 02/69-2009.

$\Box p \equiv p \equiv \Diamond^c p$ for propositional atoms p. We thus design *pseudo* (i.e., non-standard) modal arithmetics for program extraction, with relative soundness syntactically given via our (light) modal functional interpretation by the target system, namely classical predicate Arithmetic with higher-type functionals, in a Natural Deduction presentation.

We stress the fact that we are only concerned with fragments of Arithmetics without undecidable predicates: *all propositional atoms of our systems are a priori decidable.*

For an easier presentation we will give up the 'pseudo' prefix. Throughout the paper, our modal Arithmetics are pseudo-modal. Note that soundness of Schütte's predicate modal logics (e.g., $S_4^\star$) is proved non-constructively, using models, see [16].

1 Arithmetical systems for Modal Dialectica extraction

We build upon functional arithmetical systems NA and (the light annotated) NA_l from [10]. While *verifying system* NA basically is the Arithmetic Z of Berger, Buchholz and Schwichtenberg [4] in a slightly different presentation which is more suitable for light functional synthesis and features full classical logic (without strong existence) and full extensionality[3], its light counterpart NA_l is only partly classical. Moreover, the *input system* NA_l is weakly extensional and its contraction (and hence also induction) rule is restricted for soundness of the (light) functional interpretation of NA_l into NA. In computing terms, the program synthesis algorithm provided by the light Dialectica (of [10], as inherited from the one[4] of [7]) terminates without error only modulo the above-mentioned restrictions on Extensionality and Contraction[5].

For (light) modal functional synthesis we will use the same verifying system NA. The simpler input system NA^m is obtained by adding $\Box$ to a restricted variant of NA. This modal Arithmetic will be proved sound via the *modal Dialectica interpretation*. The fully-fledged input system NA_l^m adds to NA^m all light universal quantifiers and is a modal extension of NA_l; its soundness will be given by the light modal Dialectica interpretation. We will not detail here the arithmetics NA and NA_l, but rather refer the reader to [10]. We mostly enumerate the new items that are added in order to get NA^m and respectively NA_l^m. *(Systems* NA *and* NA_l *are retaken in the Appendix section 5.)*

---

[3] As inherited from system Z, our NA is mostly a Natural Deduction presentation of the so-called 'negative arithmetic' from [20], basically a double-negation, Gödel-Gentzen embedding of classical into Heyting Arithmetic HA^ω.

[4] The restriction on extensionality is at its turn inherited from the pure Gödel's functional interpretation [1,6], whereas the restriction on contraction was first added by Hernest, as it was imposed by the necessity of decidability of the translation of light contraction formulas.

[5] These restrictions are more relaxed than those from the first author's PhD thesis and weaker than Gödel's restriction on extensionality, Kreisel's avoiding of contraction in his Modified Realizability [12] and Girard's total elimination of contraction in his original Linear Logic [5].

The sets of finite types T, terms $\mathcal{T}$ (of Gödel's T), formulas $\mathcal{F}$ (of NA) and, with the addition of $\Box$, formulas $\mathcal{F}^m$ of NA^m and $\mathcal{F}^m_l$ of NA^m_l are defined as follows:

$$
\begin{array}{rrcl}
T & \rho, \sigma & ::= & \iota \mid o \mid (\rho\sigma) \\
\mathcal{T} & s, t & ::= & x^\rho \mid \mathtt{T}^o \mid \mathtt{F}^o \mid \mathtt{0}^\iota \mid \mathtt{S}^{\iota\iota} \mid \mathtt{If}^{o\rho\rho\rho} \mid \mathtt{R}^{\iota\rho(\iota\rho\rho)\rho} \mid (\lambda x^\rho .\, t^\sigma)^{\rho\sigma} \mid (t^{\rho\sigma} s^\rho)^\sigma \\
\mathcal{F} & A, B & ::= & \mathtt{at}(t^o) \mid A \to B \mid A \wedge B \mid \forall x^\rho A \mid\ \widetilde{\exists} x^\rho A \ :\equiv\ \neg\forall x^\rho \neg A \\
\mathcal{F}^m & A, B & ::= & \mathtt{at}(t^o) \mid A \to B \mid A \wedge B \mid \forall x^\rho A \mid\ \Box A \mid \Diamond^c A \ :\equiv\ \neg\Box\neg A \\
\mathcal{F}^m_l & A, B & ::= & \mathtt{at}(t^o) \mid A \to B \mid A \wedge B \mid \forall x^\rho A \mid \Box A \mid\ \forall_{\{\emptyset,+,-,\pm\}} x^\rho A
\end{array}
$$

Recall that we employ just two basic types: integers ι and booleans o, and use $\rho\sigma\tau$ for $(\rho(\sigma\tau))$. Building blocks for terms are the usual constructors for booleans $(\mathtt{T}, \mathtt{F})$ and integers $(\mathtt{0}, \mathtt{S})$, case distinction $\mathtt{If}$ and Gödel recursion $\mathtt{R}$.

The operator $\mathsf{FV}(\cdot)$ returns the set of free variables of its argument $t \in \mathcal{T}$ or $A \in \mathcal{F}$. Atomic formulas are decidable by definition, as they are identified with boolean terms. In particular, we have decidable falsity $\bot :\equiv \mathtt{at}(\mathtt{F})$ and truth $\top :\equiv \mathtt{at}(\mathtt{T})$. As usual, we abbreviate $A \to \bot$ by $\neg A$.

For the necessity operator $\Box$ we have the following *enhanced* introduction rule, which applies to many more premise sequents than usual (as the context Γ may be inhabited, see also Remark 4 in Section 2 for an extended motivation):

$\Box^i : \dfrac{\Gamma \vdash A}{\Gamma \vdash \Box A}$, where Γ is restricted depending on the translation of the (sub)proof of the premise sequent, in ways that will be described below for each of the two proof translations: modal and light modal.

The following axioms of modal propositional logic S_4 are part of NA^m and NA^m_l:

$\mathtt{AxT} :\ \Box A \to A$ $\qquad$ $\mathtt{AxT^c} :\ A \to \Diamond^c A$

$\mathtt{Ax4} :\ \Box A \to \Box\Box A$ $\qquad$ $\mathtt{Ax4^c} :\ \Diamond^c\Diamond^c A \to \Diamond^c A$

$\mathtt{AxK} :\ [\Box(A \to B) \wedge \Box A] \to \Box B$

In fact only $\mathtt{AxT}$ is needed as axiom of our non-standard modal systems. Of course, $\mathtt{AxT^c}$ and $\mathtt{Ax4^c}$ were syntactically deducible from $\mathtt{AxT}$ and respectively $\mathtt{Ax4}$ already in the propositional modal system S_4, only using minimal logic (the proof of $\mathtt{Ax4^c}$ also uses $\mathtt{AxK}$ and the empty-context $\Box^i$). It turns out that also $\mathtt{Ax4}$ and $\mathtt{AxK}$ are easily deducible in $\mathsf{NA}^m/\mathsf{NA}^m_l$ just from $\mathtt{AxT}$ (and only using minimal logic), given our very liberal necessity introduction rule, see Definition 3 below. Note that Stability $\neg\neg B \to B$ needs to be restricted already for NA^m, due to the necessary restriction on Contraction, see Remark 3 further below, Remark 5 in Section 2 and Section 3.1 of [10].

We denote by $A \to_k B :\equiv \Box A \to B$ the so-called 'Kreisel implication', since its translation by modal Dialectica coincides with its Modified Realizability interpretation.

Definition 1 (modal Dialectica interpretation). The interpretation does not change atomic[6] formulas, i.e., $|\mathtt{at}(t^o)| :\equiv \mathtt{at}(t^o)$. Assuming $|A|^{\boldsymbol{x}}_{\boldsymbol{y}}$ and $|B|^{\boldsymbol{u}}_{\boldsymbol{v}}$ are already defined,

$$|A \wedge B|^{\boldsymbol{x},\boldsymbol{u}}_{\boldsymbol{y},\boldsymbol{v}} :\equiv |A|^{\boldsymbol{x}}_{\boldsymbol{y}} \wedge |B|^{\boldsymbol{u}}_{\boldsymbol{v}} \qquad |\forall z\, A(z)|^{\boldsymbol{f}}_{z,\boldsymbol{y}} :\equiv |A(z)|^{\boldsymbol{f}z}_{\boldsymbol{y}}$$

$$|A \to B|^{\boldsymbol{f},\boldsymbol{g}}_{\boldsymbol{x},\boldsymbol{v}} :\equiv |A|^{\boldsymbol{x}}_{\boldsymbol{f}\boldsymbol{x}\boldsymbol{v}} \to |B|^{\boldsymbol{g}\boldsymbol{x}}_{\boldsymbol{v}} \qquad |\Box A|^{\boldsymbol{x}} :\equiv \forall \boldsymbol{y} |A|^{\boldsymbol{x}}_{\boldsymbol{y}}$$

As an immediate consequence,

$$|\Diamond^c A \equiv (\neg\Box\neg A)|_{\boldsymbol{f}} \equiv \widetilde{\exists}\boldsymbol{x}|A|^{\boldsymbol{x}}_{\boldsymbol{f}\boldsymbol{x}}$$

$$|A \to_k B \equiv (\Box A \to B)|^{\boldsymbol{g}}_{\boldsymbol{x},\boldsymbol{v}} \equiv \forall \boldsymbol{y}|A|^{\boldsymbol{x}}_{\boldsymbol{y}} \to |B|^{\boldsymbol{g}\boldsymbol{x}}_{\boldsymbol{v}}$$

$$|\widetilde{\exists} z\, A(z) \equiv (\neg\forall z \neg A(z))|^{Z,\boldsymbol{f}}_{\boldsymbol{g}} \equiv \neg\neg|A(Z\boldsymbol{g})|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}(Z\boldsymbol{g})(\boldsymbol{f}\boldsymbol{g})}$$

Definition 2 (light modal Dialectica interpretation). The following are added to the above (the deduced translation of $\widetilde{\exists}_{\emptyset} z$ is outlined below for use at the end of Section 2):

$$|\forall_+ z\, A(z)|^{\boldsymbol{f}}_{\boldsymbol{y}} :\equiv \forall z |A(z)|^{\boldsymbol{f}z}_{\boldsymbol{y}} \qquad |\forall_- z\, A(z)|^{\boldsymbol{x}}_{z,\boldsymbol{y}} :\equiv |A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}}$$

$$|\forall_{\emptyset} z\, A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}} :\equiv \forall z |A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}} \qquad |\widetilde{\exists}_{\emptyset} z\, B(z)|^{\boldsymbol{f}}_{\boldsymbol{g}} \equiv \widetilde{\exists} z |B(z)|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}(\boldsymbol{f}\boldsymbol{g})}$$

Remark 1. The light modal translation of formulas only adds $|\Box A|^{\boldsymbol{x}} :\equiv \forall \boldsymbol{y}|A|^{\boldsymbol{x}}_{\boldsymbol{y}}$ to our light functional translation from [10].

The definition of *computation relevance* of (light) modal formulas A is basically the same as for non-modal formulas, relative to the enhanced syntactic context. Namely, A is *realization relevant* also under (light) modal Dialectica if the tuple of witness variables $\boldsymbol{x}$ of its translation $|A|^{\boldsymbol{x}}_{\boldsymbol{y}}$ is not empty and similarly A is *refutation relevant* if the tuple of challenge variables $\boldsymbol{y}$ is not empty. See Remark 1 in Section 3 of [10]. Correspondingly, A is *realization irrelevant* if it is not realization relevant (i.e., $\boldsymbol{x}$ is an empty tuple), and A is *refutation irrelevant* if it is not refutation relevant (i.e., $\boldsymbol{y}$ is an empty tuple), see also the more technical Definition 1 in Section 2 of [10].

Definition 3 (Necessity Introduction). The restriction on $\Box^i$ depends on programs synthesized from the proof of the premise A of this rule, unless all formulas in the context Γ are refutation irrelevant or A is refutation irrelevant, see the paragraph following Theorem 1 in Section 2 below. *Thus input proofs are inductively defined together with their extracted programs (and their corresponding output proofs).*

Remark 2 (restriction violation for $\Box^i$). In an automated interactive search for modal input proofs of a given specification, we can temporarily allow $\Box^i$ and postpone the validity check for when the proof of its premise is fully constructed. This approach would be similar to the 'nc-violations' check in the actual MinLog system, see [17], and to the so-called 'computationally correct proofs' from [19].

[6] Any decidable formula can (and should) be given via its associated boolean term, e.g., one should rather use $\mathtt{at}(\mathtt{Odd}(x))$ instead of the more verbose $\forall y(2y \neq x)$, which is refutation relevant in a somewhat artificial and probably unintended way.

For efficiency reasons, we recommend the use of modal operators whenever possible instead of the above partly (or non) computational quantifiers $\forall_+$, $\forall_-$, $\forall_\emptyset$ and $\widetilde{\exists}_\emptyset$. Thus it makes sense to study the (pure) modal Dialectica in itself, as the use of such light quantifiers may not be necessary in many cases of interest. It should be much easier to construct a purely modal (i.e., without light quantifiers) input proof, also for a (semi) automated proof-search algorithm. Nevertheless, it is the light variant of modal Dialectica which provides the larger range of possibilities, particularly for situations where the simpler, 'heavier' modal Dialectica does not suffice.

Remark 3 **(Contraction restriction).** We upgrade the ★ restriction from [10] on the *computationally relevant contractions* (those on refutation relevant open assumptions A), such that the interpretation $|A|$ must be decidable (rather than strictly quantifier-free). In the new modal context one needs to take into account also the translation of the necessity operator, as this introduces new quantifiers. These may alter the decidability of the translated formula (relative to the corresponding non-modal formula obtained by wiping out all instances of $\Box$). E.g., let $T(x,y,z)$ be a decidable predicate s.t. $H(x,y) :\equiv \widetilde{\exists} z T(x,y,z)$ is not decidable (take Kleene's T predicate which is expressible in Peano Arithmetic, hence also in NA, so that H expresses the Halting Problem "program with code x halts on input y"). Then $P(x) :\equiv \forall y \forall z \neg T(x,y,z)$ can be a contraction formula, whereas $P^{\Box}(x) :\equiv \forall y \Box \forall z \neg T(x,y,z)$ cannot, as its translation is $\forall z \neg T(x,y,z)$, an undecidable formula, since $\mathsf{NA} \vdash |P^{\Box}(x)|_y \leftrightarrow \neg H(x,y)$.

On the other hand, both $\forall z(3z \neq x) \wedge \forall y(2y \neq x)$ and $\forall z(3z \neq x) \wedge \Box \forall y(2y \neq x)$ can be contraction formulas, since $\forall y(2y \neq x)$ is decidable.

2 Modal and light modal functional interpretations

The following metatheorem gives the general pattern in which soundness theorems for Dialectica-based interpretations can be expressed, in a Natural Deduction setting.

Theorem 1 (general soundness for Dialectica interpretations; [ISys, VSys]**).** *Let $A_0, A_1, \ldots, A_n$ be a sequence of formulas of* ISys *with $\boldsymbol{w}$ all their free variables. If the sequent $a_1 : A_1, \ldots, a_n : A_n \vdash_l A_0$ is provable in* ISys, *then terms $\boldsymbol{t}_0, \ldots, \boldsymbol{t}_n$ can be automatically synthesized from its formal proof, such that the translated sequent $a_1 : |A_1|^{\boldsymbol{x}_1}_{\boldsymbol{t}_1}, \ldots, a_n : |A_n|^{\boldsymbol{x}_n}_{\boldsymbol{t}_n} \vdash |A_0|^{\boldsymbol{t}_0}_{\boldsymbol{x}_0}$ is provable in* VSys, *where the following* free variable condition (c) *holds:* $\mathsf{FV}(\boldsymbol{t}_i) \subseteq \{\boldsymbol{w}, \boldsymbol{x}_0, \ldots, \boldsymbol{x}_n\}$ *and* $\boldsymbol{x}_0 \notin \mathsf{FV}(\boldsymbol{t}_0)$. *Here $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n$ are tuples of fresh variables, s.t. equal avars share a common such tuple.*

In [10] the above was thoroughly proved for $\mathrm{ISys} \equiv \mathsf{NA}_l$ and $\mathrm{VSys} \equiv \mathsf{NA}$. Below we prove that (meta)Theorem 1 remains valid also for the pairs $[\mathsf{NA}^m, \mathsf{NA}]$ (modal Dialectica) and $[\mathsf{NA}^m_l, \mathsf{NA}]$ (light modal Dialectica), which share the same $\mathrm{VSys} \equiv \mathsf{NA}$.

We can now complete the definition of $\Box^i$: *the restriction is that $\boldsymbol{x}_0 \notin \cup_{i=1}^n \mathsf{FV}(\boldsymbol{t}_i)$ in the translated premise sequent $a_1 : |A_1|^{\boldsymbol{x}_1}_{\boldsymbol{t}_1}, \ldots, a_n : |A_n|^{\boldsymbol{x}_n}_{\boldsymbol{t}_n} \vdash |A_0|^{\boldsymbol{t}_0}_{\boldsymbol{x}_0}$*. This ensures that the introduction rule $\forall^i$ can be applied for variables $\boldsymbol{x}_0$ and thus the conclusion sequent $a_1 : A_1, \ldots, a_n : A_n \vdash_l \Box A_0$ is witnessed by the same realizers as the premise.

Lemma 1 (interpretation of S_4 modal axioms). *Axioms* `AxT`, `AxT`$^{\mathrm{c}}$, `Ax4`, `Ax4`$^{\mathrm{c}}$ *and* `AxK` *are realizable in* NA *under the (light) modal Dialectica translation.*

Proof: The translation of AxT is $|\Box A \to A|^{\boldsymbol{g}}_{\boldsymbol{x},\boldsymbol{y}} \equiv \forall \boldsymbol{v}|A|^{\boldsymbol{x}}_{\boldsymbol{v}} \to |A|^{\boldsymbol{gx}}_{\boldsymbol{y}}$ and we can take $\boldsymbol{g}$ to be the identity $\lambda \boldsymbol{x}.\, \boldsymbol{x}$. Similarly, the translation of $\mathtt{AxT}^{\mathrm{c}}$ is $|A \to \Diamond^c A|^{\boldsymbol{f}}_{\boldsymbol{x},\boldsymbol{y}} \equiv |A|^{\boldsymbol{x}}_{\boldsymbol{fxy}} \to \widetilde{\exists} \boldsymbol{u}|A|^{\boldsymbol{u}}_{\boldsymbol{y}}$ and we can take $\boldsymbol{f}$ to be the projection $\lambda \boldsymbol{xy}.\, \boldsymbol{y}$. For Ax4 and $\mathtt{Ax4}^{\mathrm{c}}$ it is immediate that $|\Box A| \equiv |\Box\Box A|$ and also $|\Diamond^c A| \equiv |\Diamond^c \Diamond^c A|$, thus the realizer is again the identity in both cases. In the translation of AxK below, we take $\boldsymbol{U} :\equiv \lambda \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{x}.\, \boldsymbol{gx}$, which can easily be proved to be a realizer.

$$
\begin{aligned}
|\mathtt{AxK}| \quad &\equiv \quad [\, \forall \boldsymbol{x}, \boldsymbol{v}(|A|^{\boldsymbol{x}}_{\boldsymbol{fxv}} \to |B|^{\boldsymbol{gx}}_{\boldsymbol{v}}) \;\wedge\; \forall \boldsymbol{y}|A|^{\boldsymbol{x'}}_{\boldsymbol{y}} \,]^{\boldsymbol{f},\boldsymbol{g},\boldsymbol{x'}} \;\to\; \forall \boldsymbol{v'}|B|^{\boldsymbol{u}}_{\boldsymbol{v'}} \quad \equiv \\
&\equiv \quad [\, \forall \boldsymbol{x}, \boldsymbol{v}(|A|^{\boldsymbol{x}}_{\boldsymbol{fxv}} \to |B|^{\boldsymbol{gx}}_{\boldsymbol{v}}) \;\wedge\; \forall \boldsymbol{y}|A|^{\boldsymbol{x'}}_{\boldsymbol{y}} \;\to\; \forall \boldsymbol{v'}|B|^{\boldsymbol{U}(\boldsymbol{f},\boldsymbol{g},\boldsymbol{x'})}_{\boldsymbol{v'}} \,]^{\boldsymbol{U}}_{\boldsymbol{f},\boldsymbol{g},\boldsymbol{x'}}
\end{aligned}
$$

Given the above Lemma and comments, we have completely established the following:

Theorem 2 (soundness of modal Dialectica). *Theorem 1* $[\ \mathsf{NA}^m,\ \mathsf{NA}\ ]$.

Theorem 3 (soundness of light modal Dialectica). *Theorem 1* $[\ \mathsf{NA}^m_l,\ \mathsf{NA}\ ]$.

The next result pictures the limits of our modal extension of Dialectica interpretation.

Theorem 4 (unrealizability of S_5 defining axiom). *Axiom* Ax5 : $\Diamond^c A \to \Box \Diamond^c A$ *is generally not realizable under the (light) modal Dialectica translation.*

Proof: The translation of Ax5 is a formula of shape $B(\boldsymbol{x}) \to \forall \boldsymbol{y} B(\boldsymbol{y})$ which only holds true when $\boldsymbol{x}$ is the empty tuple, special case when Ax5 requires no realizer at all.

Notice that $\Diamond^c \widetilde{\exists} \boldsymbol{x} A$ is akin to Berger's uniform existence $\{\exists \boldsymbol{x}\} A$ from [2], where one does not care about the witness for $\exists \boldsymbol{x}$ (which is actually deleted from the extraction). We can thus see $\Diamond^c$ as an extension of Berger's tool to more general formulas than just existential ones. On the other hand there are situations when $\Box$ and $\Diamond^c$ are too general tools and separate annotations for each quantifier are a better answer for the problem at hand. In some of these cases it may still be possible to use the modal operators if one changes the input specification and its proof.

Remark 4 (Necessity Introduction revisited). The usual restriction on the introduction rule for the necessity operator $\Box^i$ is that $\Gamma \equiv \emptyset$. In the natural deduction presentation of modal logic, $\Box^i$ cannot be unrestricted or $A \to \Box A$ becomes a theorem, thus all occurrences of $\Box$ becoming redundant. Our restriction on $\Box^i$ is strictly weaker, as, e.g., allows any context Γ whose formulas are all refutation irrelevant and any context at all if the conclusion is refutation irrelevant. Thus, $A \to \Box A$ not only is possible in our pseudo-modal systems, it even defines a very interesting class of formulas, see below.

Definition 4 (necessary formulas). *Formulas A s.t.* $\vdash A \to \Box A$ *in* NA^m *or* NA^m_l.

Also due to AxT, it follows that $\vdash A \leftrightarrow \Box A$ for any necessary formula, thus placing $\Box$ in front of such A would be logically redundant. We say that an occurrence of $\Box$ is *meaningful* (i.e., non-redundant) in front of any formula that is not necessary.

Note that all refutation irrelevant formulas are necessary formulas. It is easy to see that some of the refutation relevant formulas are necessary, e.g., $\forall \boldsymbol{x} \bot$ and $\forall \boldsymbol{x} \top$ (in fact any A s.t. $\vdash A$ or $\vdash \neg A$ in NA^m or NA^m_l). However, even if such formulas syntactically do require challengers, these functionals turn out to be redundant and can be soundly discarded by a $\Box$, without the need to change any other component of the input proof. In fact, *a formula A is necessary iff it can be proved equivalent (in NA^m or NA^m_l) to a refutation irrelevant formula B*. Indeed, for a necessary A take $B :\equiv \Box A$. For the converse we can use the long implication $A \to B \to \Box B \to \Box A$, where for the last implication a contextless $\Box^i$ together with $\mathtt{AxK}$ was used.

Therefore, the 'necessary' class captures those formulas whose negative computational content can always be erased regardless of the context in which they are used. On the other hand, there are cases when $\Box$ can soundly be applied to a non-necessary formula, leading to cleaner and more efficient extracted programs (see Section 3 below).

Remark 5 (modal vs. pseudo-modal). It would appear that our input Arithmetic NA^m is able to prove new modal theorems and even sentences that are invalid in Schütte's semantics. On the other hand, our $\bigstar$ restriction on contraction is not present in the usual first-order modal logic systems, thus some of the classical modal theorems will no longer be theorems of NA^m. Therefore, *we say that our input systems are 'pseudo-modal' rather than modal*. See [13] for extensive comments on the design of formalisms for predicate modal logic, particularly on the yet-unsatisfactory definition of necessity introduction in Natural Deduction systems. Contraction restriction notwithstanding, we give the optimal restriction for $\Box^i$ in view of automated program synthesis. However, this does not solve the issue for general, fully-fledged first-order modal logics.

2.1 Modal induction rule

As first argued in [9], induction (for natural numbers, but more generally also for lists, as naturals ι are a particular case of inductively defined lists) should rather be treated in a Modified Realizability style whenever possible under Dialectica extraction. In our non-standard modal context we can introduce the following *modal induction* rule of systems NA^m and NA^m_l, which is defined with a Kreisel implication at the step:

$$\frac{\Gamma \;\vdash\; \Box A(\mathtt{0}) \qquad \Box\Delta \;\vdash\; \Box A(n) \;\to\; A(\mathtt{S}n)}{\Gamma\,,\, \Box\Delta \;\vdash\; \Box A(n)}\;\mathtt{Ind}^{\mathtt{m}}_{\iota}$$

This is an upgrade of the similar rule from [9] (given at the linear logic sublevel, see also [15]), as it allows for non-empty contexts. While the base context Γ is unrestricted, the step context $\Box\Delta$ is made entirely of refutation irrelevant assumptions of shape $\Box D$. Thus the step context restriction as for $\mathtt{Ind}^{\iota}_{l}$ (see Appendix) is bluntly satisfied, since this only concerned refutation relevant assumptions (whose translations in NA had to be quantifier-free, as their decidability was needed for case distinction in their corresponding challenge realizers). Note that if D already is refutation irrelevant, placing $\Box$ in front of D is somewhat redundant. We could refine $\mathtt{Ind}^{\mathtt{m}}_{\iota}$ by splitting the step context into Δ' which consists of refutation irrelevant assumptions not of shape $\Box D$ and $\Delta'' \equiv \Box\Delta$. Nonetheless such Δ' were made of necessary formulas (cf. Definition 4).

The treatment of $\mathtt{Ind}^{\mathtt{m}}_{\iota}$ under (light) modal Dialectica is much easier than the one of $\mathtt{Ind}^{\iota}_{l}$. In fact $\mathtt{Ind}^{\mathtt{m}}_{\iota}$ is a good simplification of $\mathtt{Ind}^{\iota}_{l}$ for situations when the whole context is made entirely of refutation irrelevant assumptions but $A(n)$ is a refutation relevant formula. The challenger for $A(n)$ in the step conclusion would be unneededly produced during the treatment of such $\mathtt{Ind}_l^\iota$, as it becomes no part of any of the witnesses for the conclusion sequent. Placing $\Box$ in front of the negatively positioned $A(n)$ thus ensures a minimal optimization brought by $\mathtt{Ind}_\iota^\mathtt{m}$, in this particular case simply by elimination of redundancy: the conclusion witnessing terms are the same as for $\mathtt{Ind}_l^\iota$.

A more serious optimization concerns the challengers of $|C|$ for refutation relevant assumptions C from the Γ context. These are simply preserved by $\mathtt{Ind}_\iota^\mathtt{m}$, while under $\mathtt{Ind}_l^\iota$ they had to include the challengers for the step $A(n)$. If $A(n)$ were refutation irrelevant, it would still make sense to use $\mathtt{Ind}_\iota^\mathtt{m}$ instead of $\mathtt{Ind}_l^\iota$, if one is not interested in the challengers for the refutation relevant assumptions from the step context. While for such particular $\mathtt{Ind}_l^\iota$ we already have the preservation of challengers for refutation relevant assumptions strictly from Γ, still challengers for the refutation relevant step assumptions are more complex in the conclusion sequent (they include a meaningful Gödel recursion, even though here a challenger for the step negative $A(n)$ is no longer comprised since it does not exist). Thus $\mathtt{Ind}_\iota^\mathtt{m}$ can bring an improvement over $\mathtt{Ind}_l^\iota$ by wiping out the step challengers altogether, should these not be needed in the global construction of the topmost realizers for the goal specification.

It turns out that $\mathtt{Ind}_\iota^\mathtt{m}$ strictly optimizes $\mathtt{Ind}_l^\iota$ in many (if not most) situations. Yet $\mathtt{Ind}_l^\iota$ will have to be used also in our non-standard modal context, practically whenever $\mathtt{Ind}_\iota^\mathtt{m}$ simply cannot be applied for the goal at hand.

3 Examples

The weak extensionality of modal input systems NA^m and NA_l^m can better be expressed by means of the following *modal compatibility axiom* (the usual compatibility axiom, but with the outward implication changed to a Kreisel implication)

$$\mathtt{CmpAx}^\mathtt{m} : \ \Box(\boldsymbol{x} =_\rho \boldsymbol{y}) \ \rightarrow \ A(\boldsymbol{x}) \rightarrow A(\boldsymbol{y})$$

By straightforward calculations, it is easy to see that $\mathtt{CmpAx}^\mathtt{m}$ is realizable under (light) modal Dialectica by simple projection functionals, with the verification in the fully extensional NA given by the corresponding compatibility axiom $\mathtt{CmpAx}$, see [10].

In [9] the following class of examples was considered: theorems of the form

$$\forall \boldsymbol{x} A \rightarrow \forall \boldsymbol{y} B \rightarrow \forall \boldsymbol{z} C \tag{1}$$

possibly with parameters, where the negative information on $\boldsymbol{x}$ is irrelevant, while the one on $\boldsymbol{y}$ is of our interest. Then it must be possible to adapt the proof of (1) to a proof in NA^m or NA_l^m of $(\Box\forall \boldsymbol{x} A) \rightarrow \forall \boldsymbol{y} B \rightarrow \forall \boldsymbol{z} C$. As noticed by Oliva in [15], the Fibonacci example first treated with Dialectica in [8] falls into this category.

Oliva also suggested an interesting example, which motivated the definition of our positively computational quantifier $\forall_+$ (see [10]): "Any infinite set P of natural numbers contains numbers which are arbitrarily apart". The claim can be formalized as follows:

$$\forall x \tilde{\exists} y (y > x \wedge P(y)) \rightarrow \forall d \tilde{\exists} n_1, n_2 \big(n_2 > n_1 + d \wedge P(n_1) \wedge P(n_2)\big) \tag{2}$$

This statement can be proved only via a contraction on the premise, and as a result x is refuted by a term involving case distinction on $|P|$. However, if only the witnesses of n_1 and n_2 are needed, then the redundant challenge for x can be discarded by using a $\Box$ in front of the premise, effectively applying a Kreisel implication. This example is of the form (1) and can be treated both with the hybrid Dialectica from [9] and with the extended light Dialectica interpretation from [10].

The example can be extended so that the premise becomes more involved [19]:

$$\forall m(\widetilde{\exists} n\, Q(n,m) \to \widetilde{\exists} n_1\, Q(n_1, \mathtt{S}m)) \to \widetilde{\exists} n_0\, Q(n_0, \mathtt{0}) \to \widetilde{\exists} n_2\, Q(n_2, \mathtt{SS0}) \tag{3}$$

Again, a contraction must be used, and two semi-computational quantifiers need to be applied to erase the negative computational content:

$$\forall_+ m(\widetilde{\exists}_+ n\, Q(n,m) \to \widetilde{\exists} n_1\, Q(n_1, \mathtt{S}m)) \to \widetilde{\exists} n_0\, Q(n_0, \mathtt{0}) \to \widetilde{\exists} n_2\, Q(n_2, \mathtt{SS0}) \tag{4}$$

However, this solution is not desirable, as the light annotations would only apply to a special class of binary relations Q for which the witness n_1 for $Q(n_1, \mathtt{S}m)$ does not depend computationally on the witness n for $Q(n,m)$ for any m, hence reducing the generality of the claim. One of the solutions would be to extend the light annotations to implications as in [19], however a much simpler and more elegant approach would be to use a Kreisel implication. The negative content of the premise will be fully erased and the positive one will be fully preserved, achieving a Modified Realizability effect.

We will consider another relevant case study, known as the "integer root example", which was suggested by Berger and Schwichtenberg in [3]: "every unbounded integer function has an integer root function". The example can be formalized as follows:

$$\forall x \widetilde{\exists} y(f(y) > x) \to \forall m\Big(f(\mathtt{0}) \le m \to \widetilde{\exists} n\big(f(n) \le m < f(\mathtt{S}n)\big)\Big) \tag{5}$$

The claim can be proved by contradiction using induction on the formula $f(n) \le m$. However, in addition to computing the integer root, the (heavy) Dialectica also extracts a complicated recursive counterexample for x, with a case distinction on each step [19]. This term challenges the outermost premise, which forms the refutation relevant induction context shared by the base and the step formulas. The undesired negative content can be erased by a Kreisel implication, which converts the context to a necessary one, allowing the application of the modal induction rule. As a result, only the integer root is extracted, and additional artifacts are omitted. Note that, in contrast to the previous two examples, this proof is classical, so Modified Realizability is not applicable in this case. However, using $\forall_+ x$ would still achieve the same cleaning effect [19].

4 Conclusions and future work

Modal Dialectica provides the means of using both Modified Realizability and Gödel's Dialectica at the same time for more efficient program extraction. This was already the case for the hybrid Dialectica of [9], but here we eliminate the detour to the linear logic sublevel. Disregarding the light quantifiers, (pure) modal Dialectica represents (directly at the supra-linear logic level) a good combination of the original proof interpretations, with the possibility of carrying out both in a sound way on certain input proofs. All one needs is that some implications of the input proof can be seen as Kreisel implications.

A natural continuation of the work reported in this paper concerns the addition to our input systems of strong (intuitionistic) elements. Besides the strong $\exists$ and its light associated $\exists_\emptyset$ (originally from [7] where it was denoted $\overline{\exists}$, see also [19]), *strong possibility* $\Diamond$ also needs to be considered as the intuitionistic dual of necessity $\Box$.

The following clauses would then be added to Definition 1 for getting the *strong modal Dialectica* interpretation $|\exists z\, A(z)|^{z,\boldsymbol{f}}_{\boldsymbol{y}} :\equiv |A(z)|^{\boldsymbol{f}}_{\boldsymbol{y}}$ and $|\Diamond A|_{\boldsymbol{y}} :\equiv \exists \boldsymbol{x} |A|^{\boldsymbol{x}}_{\boldsymbol{y}}$, and further $|\exists_\emptyset z\, A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}} :\equiv \exists z |A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}}$ to Definition 2 in order to obtain the *strong light modal Dialectica* interpretation.

Intuitionistic (light) modal arithmetical systems will first be considered at input for 'strong' program synthesis. Then their enhanced classical counterparts will be interpreted, modulo some negative translation. Such systems will soundly extend NA^m with $\Diamond$ and $\exists$, and NA^m_l also with $\exists_\emptyset$. Nevertheless certain restrictions may need to be applied on NA^m and/or NA^m_l before attempting such extensions with intuitionistic elements.

Acknowledgements: Our first reading of predicate modal logic was [16], a rare small and complete presentation of the topic, recommended by Prof. Schwichtenberg, to whom we are grateful. Thanks to Diana Ratiu for providing us a copy of this book. Many thanks to Paulo Oliva for valuable comments on an early draft of this paper.

References

1. J. Avigad and S. Feferman. Gödel's functional ("Dialectica") interpretation. In S. R. Buss, editor, *Handbook of proof theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 337–405. North Holland, Amsterdam, 1998.
2. U. Berger. Program extraction from normalization proofs. In M. Bezem, editor, *Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*, pages 91–106. Springer Verlag, 1993.
3. U. Berger and H. Schwichtenberg. Program extraction from classical proofs. In D. Leivant, editor, *Logic and Computational Complexity Workshop (LCC'94)*, volume 960 of *Lecture Notes in Computer Science*, pages 77–97. Springer, Berlin, 1995.
4. U. Berger, H. Schwichtenberg, and W. Buchholz. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002.
5. J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.
6. K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.
7. M.-D. Hernest. *Optimized programs from (non-constructive) proofs by the light (monotone) Dialectica interpretation*. PhD Thesis, École Polytechnique and Universität München, 2006. `http://www.cs.au.dk/~danher/teza/thesfull.pdf`.
8. M.-D. Hernest. Light Dialectica program extraction from a classical Fibonacci proof. *Electronic Notes in Theoretical Computer Science*, 171(3):43–53, 2007.
9. M.-D. Hernest and P. Oliva. Hybrid functional interpretations. In A. Beckmann, C. Dimitracopoulos, and B. Löwe, editors, *Logic and Theory of Algorithms: 4th "Computability in Europe" conference, Athens, Greece*, volume 5028 of *Lecture Notes in Computer Science*, pages 251–260. Springer Verlag, 2008.
10. M.-D. Hernest and T. Trifonov. Light dialectica revisited. *Annals of Pure and Applied Logic*, 161:1379–1389, 2010. `http://dx.doi.org/10.1016/j.apal.2010.04.008`.
11. U. Kohlenbach. A note on Spector's quantifier-free rule of extensionality. *Archive for Mathematical Logic*, 40:89–92, 2001.

12. G. Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North Holland, 1959.
13. S. Negri and J. von Plato. Proof analysis. Manuscript, 2009. 243 pp.
14. P. Oliva. Modified realizability interpretation of classical linear logic. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science*. IEEE Press, 2007.
15. P. Oliva. Hybrid functional interpretations of linear and intuitionistic logic. *Journal of Logic and Computation*, 2010. `http://dx.doi.org/10.1093/logcom/exq007`.
16. K. Schütte. *Vollständige System modaler und institutionstischer Logik*. Springer, 1968.
17. H. Schwichtenberg. Minlog system documentation. `http://www.math.lmu.de/~logik/minlog/documentation.php`, December 2011.
18. T. Trifonov. Dialectica interpretation with fine computational control. In K. Ambos-Spies, B. Löwe, and W. Merkle, editors, *Mathematical Theory and Computational Practice: 5th "Computability in Europe" conference, Heidelberg, Germany*, volume 5635 of *Lecture Notes in Computer Science*, pages 467–477. Springer Verlag, 2009.
19. T. Trifonov. *Analysis of methods for extraction of programs from non-constructive proofs*. PhD thesis, University of Munich, Germany, 2011.
20. A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, 1973.

5 Appendix

We use a special Natural Deduction (abbreviated "ND") presentation of our systems, where proofs are represented as sequents $\Gamma \vdash B$, meaning that formula B is the root of the ND tree whose leaves Γ are typed assumption variables (abbreviated "avars") $a\!:\!A$. Here formula A is the type of the avar a, and Γ is a multiset (since there may be more leaves labeled with the same $a\!:\!A$).

5.1 The verifying system NA

The logical rules of system NA are presented in Table 2, with the usual restriction on universal quantifier introduction $\forall^i$ that

$$z \quad \notin \quad \mathsf{FV}(\Gamma) :\equiv \bigcup_{a:A\in\Gamma} \mathsf{FV}(A)$$

At $\to^i$, $[a : A]$ denotes the multisubset of all occurrences of $a : A$ in the multiset of assumptions of the premise sequent of $\to^i$. Thus $a\!:\!A \notin \Gamma$, hence $a : A$ is no longer an assumption in the conclusion sequent of $\to^i$. In the ND tree, this means that all the leaves labeled $a : A$ are inactivated (or "discharged" as one usually says in Natural Deduction terminology).

Whereas in NA alone we could have safely let all contractions be handled implicitly at $\to^i$, in relationship with the architecture of input system NA_l (see Section 5.2) we are compelled to introduce for NA the *contraction anti-rule* C in association with $\mathcal{C}_l$ of NA_l, see Table 4. We refer to contraction as "anti-rule", rather than "rule" because, despite the sequent-like representation of our calculi, in fact our formalisms are ND and in the ND directed tree the representation of explicit contractions is by convergent arrows that go in the direction which is reverse to the direction of all the other rules.

$$\mathtt{TAx}: \ \vdash \mathtt{at}(\mathtt{T}) \qquad\qquad \mathtt{CmpAx}: \ \vdash x =_\rho y \ \to \ A(x) \to A(y)$$

Table 1. Basic axioms, with CmpAx replaced by CMP rule in NA_l

$$a\!:\!A \vdash A \quad (\mathrm{id}) \qquad \frac{\Gamma, [a\!:\!A] \vdash B}{\Gamma \vdash A \to B} \to^i \qquad \frac{\Gamma \vdash A \quad \Delta \vdash A \to B}{\Gamma, \Delta \vdash B} \to^e \qquad \frac{\Gamma \vdash A}{\Gamma \vdash \forall z A} \forall^i$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge^e_l \qquad \frac{\Delta \vdash A \wedge B}{\Delta \vdash B} \wedge^e_r \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \wedge^i \qquad \frac{\Gamma \vdash \forall z A}{\Gamma \vdash A[t/z]} \forall^e$$

Table 2. Logical rules, with $z \notin \mathsf{FV}(\Gamma)$ at $\forall^i$ and certain explicit contractions at $\to^e$ and $\wedge^i$

$$\frac{\Gamma \vdash_l A}{\Gamma \vdash_l \forall_\pm z A} \forall^i_\pm \qquad \frac{\Gamma \vdash_l A}{\Gamma \vdash_l \forall_+ z A} \forall^i_+ \qquad \frac{\Gamma \vdash_l A}{\Gamma \vdash_l \forall_- z A} \forall^i_- \qquad \frac{\Gamma \vdash_l A}{\Gamma \vdash_l \forall_\emptyset z A} \forall^i_\emptyset$$

$$\frac{\Gamma \vdash_l \forall_\pm z A}{\Gamma \vdash_l A[t/z]} \forall^e_\pm \qquad \frac{\Gamma \vdash_l \forall_+ z A}{\Gamma \vdash_l A[t/z]} \forall^e_+ \qquad \frac{\Gamma \vdash_l \forall_- z A}{\Gamma \vdash_l A[t/z]} \forall^e_- \qquad \frac{\Gamma \vdash_l \forall_\emptyset z A}{\Gamma \vdash_l A[t/z]} \forall^e_\emptyset$$

Table 3. Additional rules for NA_l, with extra restrictions on $\forall^i_+$, $\forall^i_-$ and $\forall^i_\emptyset$

$$\frac{\Delta, a\!:\!A, a\!:\!A \vdash B}{\Delta, a\!:\!A \vdash B} \ \mathsf{C} \qquad\qquad \frac{\Delta, a\!:\!A, a\!:\!A \vdash_l B}{\Delta, a\!:\!A \vdash_l B} \ \mathcal{C}_l$$

Table 4. Contraction anti-rules C for NA and (restricted) $\mathcal{C}_l$ for NA_l

$$\frac{\Gamma \vdash A(\mathtt{T}) \quad \Delta \vdash A(\mathtt{F})}{\Gamma, \Delta \vdash A(b)} \ \mathtt{Ind}_o \qquad\qquad \frac{\Gamma \vdash A(\mathtt{0}) \quad \Delta \vdash A(n) \to A(\mathtt{S}n)}{\Gamma, \Delta \vdash A(n)} \ \mathtt{Ind}_\iota$$

Table 5. Induction rules, with $\Gamma \uplus \Delta$ instead of 'Γ, Δ' and Δ restricted at $\mathtt{Ind}_\iota$ of NA_l

We find it convenient to introduce induction for booleans and naturals as the rules presented in Table 5. Here we assume that the induction variables b^o and respectively n^ι do not occur freely in Γ, nor Δ, and that they do occur in the formula A.

Computation in NA is expressed via the usual β-reduction rule $(\lambda x.t)s \hookrightarrow t[x \mapsto s]$, plus rewrite rules defining the computational meaning of If and R :

$$\begin{array}{lllllll} \mathtt{If}\,\mathtt{T}\,s\,t & \hookrightarrow & s & \qquad & \mathtt{R}\,0\,s\,t & \hookrightarrow & s \\ \mathtt{If}\,\mathtt{F}\,s\,t & \hookrightarrow & t & \qquad & \mathtt{R}\,(\mathtt{S}n)\,s\,t & \hookrightarrow & t\,n\,(\mathtt{R}\,n\,s\,t) \end{array}$$

Since this typed term system is confluent and strongly normalizing (cf. [17]), we are free not to fix a particular evaluation strategy. For simplicity, we assume that all terms occurring in proofs are automatically in normal form. In fact, normalization is necessary only when matching terms in formulas. We only avoid introducing equality axioms AxEQL as in [7] and skip the corresponding easy applications of CmpAx. When building proofs, some computation is thus carried out implicitly, behind the scene.

Using recursion at higher types we can define any provably total function of ground arithmetic, including decidable predicates such as equality $\mathtt{Eq}_o$ for booleans and $\mathtt{Eq}_\iota$ for natural numbers:

$$\begin{array}{rcl} \mathtt{Eq}_o^{ooo} & :\equiv & \lambda x.\mathtt{If}\,x\,(\lambda y.y)\,(\lambda y.\mathtt{If}\,y\,\mathtt{F}\,\mathtt{T}) \\ \mathtt{Eq}_\iota^{\iota\iota o} & :\equiv & \lambda x.\mathtt{R}\,x\,\big(\lambda y.\mathtt{R}\,y\,\mathtt{T}\,(\lambda n, q^o.\mathtt{F})\big)\,\big(\lambda m, p^{\iota o}, y.\mathtt{R}\,y\,\mathtt{F}\,(\lambda n, q^o.p\,n)\big) \end{array}$$

The $\mathtt{at}(\cdot)$ construction allows us to view boolean programs as decidable predicates. Given $\mathtt{Ind}_o$, its logical meaning is settled by the truth axiom TAx, see Table 1. In this way we can define predicate equality at base types as $s =_\sigma t :\equiv \mathtt{at}(\mathtt{Eq}\,s\,t)$ for $\sigma \in \{o, \iota\}$ and further at higher types extensionally as usual $s =_{\rho\tau} t :\equiv \forall x^\rho(sx =_\tau tx)$. It is straightforward to prove by induction on ρ that $=_\rho$ is reflexive, symmetric and transitive at any type ρ.

To complete our system, we must include in NA also the compatibility (i.e., extensionality) axiom CmpAx, see Table 1. Note that ex falso quodlibet (EFQ) $\bot \to A$ and stability (Stab) $\neg\neg A \to A$ are fully provable in NA (cf. [17], by induction on A, using TAx and $\mathtt{Ind}_o$).

5.2 The input system NA_l

Light formulas $\mathcal{F}_l$ are built over usual formulas $\mathcal{F}$ of NA by adding the three light universal quantifiers: the non-computational $\forall_\emptyset$ and the two semi-computational $\forall_+$ and $\forall_-$. In order to stress the distinction of NA_l from NA it is convenient to rename NA's $\forall$ to $\forall_\pm$ in NA_l (which marks the whole computational content, both positive and negative)

$$\mathcal{F}_l \qquad A, B \qquad ::= \quad \mathtt{at}(t^o) \mid A \to B \mid A \wedge B \mid \forall_\diamond\, x^\rho A \quad \text{ for } \diamond \in \{\emptyset, +, -, \pm\}$$

Thus, system NA_l refines the clone of NA (also with CMP for CmpAx and $\mathcal{C}_l$ for C) with introduction and elimination rules for the light quantifiers (see Table 3). These are copies of the clone rules $\forall^e_\pm$ and $\forall^i_\pm$, but with the usual restriction ($\pm$) on $\forall^i_\pm$ that $z \notin \mathsf{FV}(\Gamma)$ enhanced with the following conditions referring to the LD-interpretation

(+) at the $\forall^i_+$ rule, z may be used computationally only positively, i.e., z must not be free in the *challengers* of the LD-translation of Γ.

(−) at the $\forall^i_-$ rule, z may be used computationally only negatively, i.e., z must not be free in the *witnesses* of the LD-translation of A.

(∅) at $\forall^i_\emptyset$, z may not be used computationally at all, i.e., both (+) and (−).

Notice that the restrictions (+), (−) and (∅) assume knowledge of the LD-interpretation of whole proofs, in their full depth, thus forcing the definition of NA_l proofs to go inductively in parallel with the LD-extraction of part of their computational content (namely free variables of the extracted terms). We simultaneously define the classes of *realization irrelevant* $A_\oplus$ and *refutation irrelevant* $A_\ominus$ formulas as follows:

$$A_\oplus, B_\oplus \quad ::= \quad \mathtt{at}(t) \mid A_\oplus \wedge B_\oplus \mid A_\ominus \to B_\oplus \mid \forall_\diamond x\, A_\oplus \quad \text{for } \diamond \in \{\emptyset, +, -, \pm\}$$
$$A_\ominus, B_\ominus \quad ::= \quad \mathtt{at}(t) \mid A_\ominus \wedge B_\ominus \mid A_\oplus \to B_\ominus \mid \forall_\diamond x\, A_\ominus \quad \text{for } \diamond \in \{\emptyset, +\}$$

One necessary change when adopting principles from NA is to replace `CmpAx` with a weak compatibility rule. This is because Dialectica is unable to interpret full extensionality (cf. [11,20]). We here employ an upgraded variant of the `CMP` rule from [7]:

$$\frac{\Gamma_\ominus \vdash_l s =_\rho t}{\Gamma_\ominus \vdash_l B(s) \to B(t)}\ \mathtt{CMP}_\rho$$

where all formulas in $\Gamma_\ominus$ are refutation irrelevant.

The computationally irrelevant contractions in NA_l can safely be handled implicitly at $\to^i$. The situation is different for those contractions whose formula is refutation relevant (i.e., the computationally relevant contractions), as we want to automatically ensure that their translation is decidable (instead of leaving the task of decidability check to the user). We achieve this by including in NA_l the *contraction anti-rule* $\mathcal{C}_l$ (see Table 4) for all formulas A that are refutation relevant and (★) do not contain any $\forall_+$, nor $\forall_\emptyset$. This triggers the addition to NA of an explicit (unrestricted) contraction anti-rule C which is needed in the construction of the verifying proof (it only applies to quantifier-free formulas $|A|$). The restriction ★ ensures that all contraction formulas that require at least one challenger term for their LD-interpretation will have quantifier-free (hence decidable) LD-translations. Their decidability is necessary for attaining soundness. Being a purely syntactical criterion, ★ does not admit formulas whose LD-translations contain quantifiers, but could nevertheless be decidable, e.g., $\mathtt{Odd}(x) \equiv \forall y(2y \neq x)$.

Moreover, in order to avoid having any computationally relevant contractions implicit in $\to^i$, *we constrain the deduction rules of* NA_l *to disallow multiple occurrences of refutation relevant assumptions in any of the premise sequents.* Thus, whenever a double occurrence of a refutation relevant assumption is created in a conclusion sequent by one of the binary rules of NA_l, such sequent cannot be directly a premise for the application of an(other) NA_l rule: the anti-rule $\mathcal{C}_l$ must be applied first, in order to eliminate the critical double. If ★ is not satisfied and yet $a : A$ is a refutation relevant assumption occurring at least twice in some conclusion sequent, this is a dead end: such sequent can only be the root of the NA_l proof-tree.

While $\mathtt{EFQ} : \bot \to A$ remains fully provable also in NA_l (for all formulas $A \in \mathcal{F}_l$) the situation changes for $\mathtt{Stab} : \neg\neg A \to A$ in the case of many formulas A that feature light quantifiers in certain places. As noted in [7], the usual proof in NA of $\mathtt{Stab}$ (constructed by induction on A) makes unavoidably use of contractions over $\neg\neg(B \wedge C)$ for subformulas $(B \wedge C)$ of A, and these are subject to the ★ restriction for refutation relevant $B \wedge C$. Even when such $B \wedge C$ obey ★, they may lead to the failure of restrictions $(+)$, $(-)$ or $(\emptyset)$. On the other hand $\mathtt{Stab}$ is provable in NA_l for $A \in \mathcal{F}$ or A conjunction-free.

5.3 The light Dialectica interpretation (LD-interpretation)

With each formula A of NA_l we associate its LD-translation: a not necessarily quantifier-free formula $|A|^{\boldsymbol{x}}_{\boldsymbol{y}}$ of NA where $\boldsymbol{x}, \boldsymbol{y}$ are tuples of fresh variables, not appearing in A. The variables $\boldsymbol{x}$ in the superscript are called the *witness variables*, while the subscript variables $\boldsymbol{y}$ are called the *challenge variables*. Terms $\boldsymbol{t}$ substituting witness variables (like $|A|^{\boldsymbol{t}}_{\boldsymbol{y}}$) are called *realizing terms* or "witnesses" and terms $\boldsymbol{s}$ substituting challenge variables (like $|A|^{\boldsymbol{x}}_{\boldsymbol{s}}$) are called *refuting terms* or "challengers". Intuitively, the LD-interpretation of A can be viewed as a game in which first Eloise ($\exists$) and then Abelard ($\forall$) make one move each by playing type-corresponding objects $\boldsymbol{t}$ and $\boldsymbol{s}$ for the tuples $\boldsymbol{x}$ and respectively $\boldsymbol{y}$. Formula $|A|^{\boldsymbol{x}}_{\boldsymbol{y}}$ specifies the "adjudication relation", here not necessarily decidable: Eloise wins iff $\mathsf{NA} \vdash |A|^{\boldsymbol{t}}_{\boldsymbol{s}}$. In our light context as well, Eloise has a winning move whenever A is provable in NA_l : the LD-interpretation will explicitly provide it from the input NA_l proof of A as a tuple of witnesses $\boldsymbol{t}$ (s.t. $\mathsf{FV}(\boldsymbol{t}) \subseteq \mathsf{FV}(A)$) together with the *verifying proof* in NA of $\forall \boldsymbol{y} |A|^{\boldsymbol{t}}_{\boldsymbol{y}}$ (Eloise wins by $\boldsymbol{t}$ regardless of the instances $\boldsymbol{s}$ for Abelard's $\boldsymbol{y}$).

Definition 5 (LD-translation of formulas). The interpretation does not change atomic formulas, i.e., $|\mathtt{at}(t^o)| :\equiv \mathtt{at}(t^o)$. Assuming $|A|^{\boldsymbol{x}}_{\boldsymbol{y}}$ and $|B|^{\boldsymbol{u}}_{\boldsymbol{v}}$ are already defined,

$$
\begin{aligned}
|A \wedge B|^{\boldsymbol{x},\boldsymbol{u}}_{\boldsymbol{y},\boldsymbol{v}} &:\equiv |A|^{\boldsymbol{x}}_{\boldsymbol{y}} \wedge |B|^{\boldsymbol{u}}_{\boldsymbol{v}} \\
|A \to B|^{\boldsymbol{f},\boldsymbol{g}}_{\boldsymbol{x},\boldsymbol{v}} &:\equiv |A|^{\boldsymbol{x}}_{\boldsymbol{f}\boldsymbol{x}\boldsymbol{v}} \to |B|^{\boldsymbol{g}\boldsymbol{x}}_{\boldsymbol{v}} \quad .
\end{aligned}
$$

The interpretation of the four universal quantifiers is (upon renaming, we assume that quantified variables occur uniquely in a formula):

$$
\begin{aligned}
|\forall_{\pm} z\, A(z)|^{\boldsymbol{f}}_{z,\boldsymbol{y}} &:\equiv |A(z)|^{\boldsymbol{f}z}_{\boldsymbol{y}} & \qquad |\forall_{+} z\, A(z)|^{\boldsymbol{f}}_{\boldsymbol{y}} &:\equiv \forall z\, |A(z)|^{\boldsymbol{f}z}_{\boldsymbol{y}} \\
|\forall_{-} z\, A(z)|^{\boldsymbol{x}}_{z,\boldsymbol{y}} &:\equiv |A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}} & \qquad |\forall_{\emptyset} z\, A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}} &:\equiv \forall z\, |A(z)|^{\boldsymbol{x}}_{\boldsymbol{y}}
\end{aligned}
$$

Since $|\bot| \equiv \bot$, we get

$$
|\neg A|^{\boldsymbol{f}}_{\boldsymbol{x}} \equiv \neg |A|^{\boldsymbol{x}}_{\boldsymbol{f}\boldsymbol{x}} \qquad\qquad |\neg\neg A|^{\boldsymbol{f}}_{\boldsymbol{g}} \equiv \neg\neg |A|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}(\boldsymbol{f}\boldsymbol{g})}
$$

It is straightforward to compute that

$$
\begin{aligned}
|\widetilde{\exists}_{\pm} z\, A(z)|^{Z,\boldsymbol{f}}_{\boldsymbol{g}} &:\equiv \neg\neg |A(Z\boldsymbol{g})|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}(Z\boldsymbol{g})(\boldsymbol{f}\boldsymbol{g})} & \qquad |\widetilde{\exists}_{+} z\, A(z)|^{\boldsymbol{f}}_{\boldsymbol{g}} &:\equiv \widetilde{\exists} z\, |A(z)|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}z(\boldsymbol{f}\boldsymbol{g})} \\
|\widetilde{\exists}_{-} z\, A(z)|^{Z,\boldsymbol{f}}_{\boldsymbol{g}} &:\equiv \neg\neg |A(Z\boldsymbol{g})|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}(\boldsymbol{f}\boldsymbol{g})} & \qquad |\widetilde{\exists}_{\emptyset} z\, A(z)|^{\boldsymbol{f}}_{\boldsymbol{g}} &:\equiv \widetilde{\exists} z\, |A(z)|^{\boldsymbol{f}\boldsymbol{g}}_{\boldsymbol{g}(\boldsymbol{f}\boldsymbol{g})}
\end{aligned}
$$

The length and types of the witnessing and challenging tuples are uniquely determined.

5.4 Light Dialectica treatment of extensionality

We here give the LD-interpretation of the weak compatibility rule

$$\frac{\varGamma_{\ominus} \vdash_l s =_\rho r}{\varGamma_{\ominus} \vdash_l B(s) \to B(r)} \; \mathtt{CMP}_\rho$$

where all formulas in $\varGamma_{\ominus}$ are refutation irrelevant, i.e., the negative position in their LD-translation is empty. By definition of equality at higher types, $s =_\rho r$ is $\forall \boldsymbol{z}.\, s\boldsymbol{z} = r\boldsymbol{z}$, a purely universal formula. We are given that

$$a_1 : |A_1|^{\boldsymbol{x}_1}_{\boldsymbol{t}_1}, \ldots, a_n : |A_n|^{\boldsymbol{x}_n}_{\boldsymbol{t}_n} \quad \vdash \quad |A_0|^{\boldsymbol{t}_0}_{\boldsymbol{x}_0}$$

where $|\varGamma_{\ominus}| \equiv \{a_1, \ldots, a_n\}$, $\boldsymbol{t}_0 \equiv \boldsymbol{t}_1 \equiv \ldots \boldsymbol{t}_n \equiv \sqcup$ (empty tuple), A_0 is $s =_\rho r$ and $\boldsymbol{x}_0$ corresponds to $\boldsymbol{z}$, thus the above is more conveniently rewritten as

$$a_1 : |A_1|^{\boldsymbol{x}_1}, \ldots, a_n : |A_n|^{\boldsymbol{x}_n} \quad \vdash \quad s\boldsymbol{x}_0 = r\boldsymbol{x}_0$$

To this we can apply the generalization rule, as $\boldsymbol{x}_0$ are not free in the translated context $|\varGamma_{\ominus}|$; indeed, $\boldsymbol{x}_0$ are fresh variables and they could have appeared free only via terms $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n$, were these not empty tuples (hence the need for restricting the original context). We thus obtain $|\varGamma_{\ominus}| \vdash s = r$ and further apply the extensionality Axiom to get $|\varGamma_{\ominus}| \vdash |B|(s) \to |B|(r)$. Note that the Axiom is required here, as $|\varGamma_{\ominus}|$ may contain general formulas. With $g :\equiv \lambda \boldsymbol{u}.\, \boldsymbol{u}$ and $f :\equiv \lambda \boldsymbol{u}, \boldsymbol{v}.\, \boldsymbol{v}$ we have thus constructed a verifying proof

$$a_1 : |A_1|^{\boldsymbol{x}_1}, \ldots, a_n : |A_n|^{\boldsymbol{x}_n} \quad \vdash \quad |B(s)|^{\boldsymbol{u}}_{f\boldsymbol{u}\boldsymbol{v}} \to |B(r)|^{g\boldsymbol{u}}_{\boldsymbol{v}} \;\; [\equiv\; |B(s) \to B(r)|^{f,g}_{\boldsymbol{u},\boldsymbol{v}}\,]$$

The new realizing terms f, g are closed, hence the free variable condition trivially holds.

5.5 Light Dialectica treatment of induction for naturals

Since the induction rule (for naturals) corresponds to a virtually unbounded number of contractions of each assumption from the step context $\varDelta$ (cf. [7], see Table 5), its clone in the system NA_l is subject to a restriction like the one of $\mathcal{C}_l$. Namely, we need to require that *all refutation relevant avars in $\varDelta$ satisfy* $\bigstar$. Moreover, since the contractions on $a \in \varGamma \cap \varDelta$ will be handled differently than for simple binary rules like $\to^e$ or $\wedge^i$, it is more convenient to require that naturals induction in NA_l implicitly contracts all its refutation relevant assumptions (instead of using the explicit $\mathcal{C}_l$) . We will use the notation $\varGamma \uplus \varDelta$ for a special multiset union in which refutation relevant assumptions appear only once, even if they appear in both $\varGamma$ and $\varDelta$. Thus the $\mathtt{Ind}^{\iota}_l$ rule of NA_l is finally obtained by replacing '$\varGamma, \varDelta$' with '$\varGamma \uplus \varDelta$' in the conclusion sequent of $\mathtt{Ind}_\iota$.

$$\frac{\varGamma \vdash_l A(0) \qquad \varDelta \vdash_l A(n) \to A(\mathtt{S}n)}{\varGamma \uplus \varDelta \vdash_l A(n)} \; \mathtt{Ind}^{\iota}_l$$

We are given

$$|\Gamma|^{\boldsymbol{u}}_{\boldsymbol{\gamma}[\boldsymbol{y}]} \quad \vdash \quad |A(0)|^{\boldsymbol{r}}_{\boldsymbol{y}} \tag{6}$$

and

$$|\Delta|^{\boldsymbol{z}}_{\boldsymbol{\delta}[\boldsymbol{x};\boldsymbol{v}]} \quad \vdash \quad |A(n)|^{\boldsymbol{x}}_{\boldsymbol{t}\boldsymbol{x}\boldsymbol{v}} \;\to\; |A(\mathtt{S}n)|^{\boldsymbol{s}\boldsymbol{x}}_{\boldsymbol{v}} \tag{7}$$

We show that

$$\forall \boldsymbol{v} \left(|\Gamma \uplus \Delta|^{\boldsymbol{u}\uplus\boldsymbol{z}}_{\boldsymbol{\zeta}[n]\boldsymbol{v}} \;\to\; |A(n)|^{\boldsymbol{t'}[n]}_{\boldsymbol{v}} \right) \tag{8}$$

is a theorem of NA, where

$$t'[n] \quad :\equiv \quad \mathtt{R}\, n\, r\, (\lambda n.s) \tag{9}$$

for every corresponding pair $\langle r \in \boldsymbol{r} / s \in \boldsymbol{s} \rangle$ and $\boldsymbol{\zeta}[n]$ will be constructed as functional terms depending on $\boldsymbol{v}$. We here intentionally use the same variable n that occurs freely in s and t. Implicitly, just t' denotes $t'[n]$. Also $\boldsymbol{\zeta}$ will be constructed as the collection of all $\boldsymbol{\zeta'}$ (corresponding to $\Gamma \setminus \Delta$) and $\boldsymbol{\zeta''}$ (corresponding to Δ).

Let $b{:}B$ be a refutation relevant avar in $\Gamma \uplus \Delta$. Let $\boldsymbol{\gamma'} \in \boldsymbol{\gamma}$ and/or $\boldsymbol{\delta'} \in \boldsymbol{\delta}$ be the challengers for b in Γ and/or Δ. If b appears only in Γ (hence not in Δ) we define

$$\boldsymbol{\zeta'}[n] \quad :\equiv \quad \mathtt{R}\, n\, (\lambda \boldsymbol{v}.\boldsymbol{\gamma'}[\boldsymbol{v}])\, \big(\lambda n, p, \boldsymbol{v}.p(\boldsymbol{t}\,\boldsymbol{t'}\boldsymbol{v})\big) \tag{10}$$

If b appears in Δ, then the decidability of $|B|$ is needed at each recursive step to equalize the terms $p(\boldsymbol{t}\,\boldsymbol{t'}\boldsymbol{v})$ obtained by the recursive call with the corresponding terms $\boldsymbol{\delta'}$. Thus the right stop point of the backwards construction is provided. In fact an implicit contraction over b happens at each inductive step and $\bigstar$ guarantees that $|B|$ is decidable. For $b \in \Gamma \cap \Delta$ let

$$\boldsymbol{\zeta''}[n] \quad :\equiv \quad \mathtt{R}\, n\, (\lambda \boldsymbol{v}.\boldsymbol{\gamma'}[\boldsymbol{v}]) \Big(\lambda n, p, \boldsymbol{v}.\mathtt{If}(|B|^{\boldsymbol{z'}}_{\boldsymbol{\delta'}[\boldsymbol{t'};\boldsymbol{v}]})\, \big(p(\boldsymbol{t}\,\boldsymbol{t'}\boldsymbol{v})\big)\, \boldsymbol{\delta'}[\boldsymbol{t'};\boldsymbol{v}]\Big) \tag{11}$$

and for $b \in \Delta \setminus \Gamma$ we define its $\boldsymbol{\zeta''}[n]$ by replacing in (11) the $\boldsymbol{\gamma'}$ with canonical zeros. Here $\boldsymbol{z'}$ are the challenge variables corresponding to formula B. Notice that

$$\vdash \quad t'[\mathtt{S}n] \;=\; s t'[n] \tag{12}$$

$$\vdash \quad \boldsymbol{\zeta'}[\mathtt{S}n]\boldsymbol{v} \;=\; \boldsymbol{\zeta'}[n](\boldsymbol{t}\,\boldsymbol{t'}\boldsymbol{v}) \tag{13}$$

$$\vdash \quad \boldsymbol{\zeta''}[\mathtt{S}n]\boldsymbol{v} \;=\; \mathtt{If}\,(|B|^{\boldsymbol{z'}}_{\boldsymbol{\delta'}[\boldsymbol{t'};\boldsymbol{v}]})\, \big(\boldsymbol{\zeta''}[n](\boldsymbol{t}\,\boldsymbol{t'}\boldsymbol{v})\big)\;\; \boldsymbol{\delta'}[\boldsymbol{t'};\boldsymbol{v}] \tag{14}$$

We attempt to extend (13) to the whole $\boldsymbol{\zeta}$ by proving from (14) the following

$$|B|^{\boldsymbol{z'}}_{\boldsymbol{\zeta''}[\mathtt{S}n]\boldsymbol{v}} \quad \vdash \quad \boldsymbol{\zeta''}[\mathtt{S}n]\boldsymbol{v} \;=\; \boldsymbol{\zeta''}[n](\boldsymbol{t}\,\boldsymbol{t'}\boldsymbol{v}) \tag{15}$$

We obtain this as an immediate consequence of

$$|B|^{\boldsymbol{z'}}_{\boldsymbol{\zeta''}[\mathtt{S}n]\boldsymbol{v}} \quad \vdash \quad |B|^{\boldsymbol{z'}}_{\boldsymbol{\delta'}[\boldsymbol{t'};\boldsymbol{v}]} \tag{16}$$

Assuming $\neg|B|^{\boldsymbol{z}'}_{\boldsymbol{\delta}'[\boldsymbol{t}';\boldsymbol{v}]}$ by (14) we get $\boldsymbol{\zeta}''[\mathtt{S}n]\boldsymbol{v} = \boldsymbol{\delta}'[\boldsymbol{t}';\boldsymbol{v}]$ hence $\neg|B|^{\boldsymbol{z}'}_{\boldsymbol{\zeta}''[\mathtt{S}n]\boldsymbol{v}}$ and thus (16) follows via `Stab` (which is fully available in the verifying system).

We now prove (8) by an assumptionless induction on n. Let $\boldsymbol{\zeta}^*$ be the collection of all $\boldsymbol{\zeta}'$ and those $\boldsymbol{\zeta}''$ corresponding to $\Gamma \cap \Delta$. For $n \equiv 0$ it is sufficient that

$$|\Gamma|^{\boldsymbol{u}}_{\boldsymbol{\zeta}^*[0]\boldsymbol{v}} \quad \vdash \quad |A(0)|^{\boldsymbol{t}'[0]}_{\boldsymbol{v}}$$

which follows from (6) since by definition (9) we have $\vdash \boldsymbol{t}'[0] = \boldsymbol{r}$ and by definitions (10) and (11) we have $\vdash \boldsymbol{\zeta}^*[0] = \lambda \boldsymbol{v}.\, \boldsymbol{\gamma}[\boldsymbol{v}]$. Now given (8) we want to prove

$$|\Gamma \uplus \Delta|^{\boldsymbol{u}\uplus\boldsymbol{z}}_{\boldsymbol{\zeta}[\mathtt{S}n]\boldsymbol{v}} \quad \vdash \quad |A(\mathtt{S}n)|^{\boldsymbol{t}'[\mathtt{S}n]}_{\boldsymbol{v}} \tag{17}$$

To (8) we apply $\forall^e_{[\boldsymbol{v} \mapsto \boldsymbol{t}\,\boldsymbol{t}'\boldsymbol{v}]}$ and via easy deductions in NA we get

$$|\Gamma \uplus \Delta|^{\boldsymbol{u}\uplus\boldsymbol{z}}_{\boldsymbol{\zeta}[n](\boldsymbol{t}\,\boldsymbol{t}'\boldsymbol{v})} \quad \vdash \quad |A(n)|^{\boldsymbol{t}'[n]}_{\boldsymbol{t}\,\boldsymbol{t}'\boldsymbol{v}} \tag{18}$$

With (13) and (15) we can rewrite (18) to

$$|\Gamma \uplus \Delta|^{\boldsymbol{u}\uplus\boldsymbol{z}}_{\boldsymbol{\zeta}[\mathtt{S}n]\boldsymbol{v}} \quad \vdash \quad |A(n)|^{\boldsymbol{t}'[n]}_{\boldsymbol{t}\,\boldsymbol{t}'\boldsymbol{v}} \tag{19}$$

In (7) we substitute $\boldsymbol{x} \mapsto \boldsymbol{t}'[n]$ and get

$$|\Delta|^{\boldsymbol{z}}_{\boldsymbol{\delta}[\boldsymbol{t}';\boldsymbol{v}]} \quad \vdash \quad |A(n)|^{\boldsymbol{t}'[n]}_{\boldsymbol{t}\boldsymbol{t}'\boldsymbol{v}} \to |A(\mathtt{S}n)|^{\boldsymbol{s}\boldsymbol{t}'[n]}_{\boldsymbol{v}}$$

which gives (17) by means of easy NA deductions using (12), (16) and (19).

We have treated the most general situation, with all context sets $\Gamma \setminus \Delta$, $\Gamma \cap \Delta$ and $\Delta \setminus \Gamma$ inhabited by refutation relevant assumptions, and conclusion formula A accepting both witnesses and challengers. Many particular situations amount to easier treatments, with simpler extracted terms. These can be obtained as simplifications of the general witnesses and challengers presented above, by means of the reduction properties of the empty tuple, which was denoted ϵ in [17]. We outline below only those particular cases which are relevant in connection with the modal induction rule $\mathtt{Ind}^{\mathtt{m}}_{\iota}$.

- If $\Gamma \cup \Delta$ contains no refutation relevant assumption, but $A(n)$ is refutation relevant, then terms $\boldsymbol{t}$ are no part of the realizers for the conclusion sequent, in this case only $\boldsymbol{t}'$. Hence $\boldsymbol{t}$ would be redundantly produced and a mechanism is needed to prevent their construction. This is ensured by $\Box$ in front of the step $A(n)$ at $\mathtt{Ind}^{\mathtt{m}}_{\iota}$.
- If $A(n)$ is refutation relevant, Δ has no refutation relevant element but Γ is refutation relevant inhabited, then $\boldsymbol{\delta}$ and $\boldsymbol{\zeta}''$ are empty. Yet $\boldsymbol{\zeta}^* \equiv \boldsymbol{\zeta}'$ has to be produced as (10) and includes $\boldsymbol{t}[n]$, which is no longer the case for $\mathtt{Ind}^{\mathtt{m}}_{\iota}$.
- If $A(n)$ is refutation irrelevant then $\boldsymbol{v}$, $\boldsymbol{t}$ and $\boldsymbol{t}\,\boldsymbol{t}'\boldsymbol{v}$ are empty tuples. Thus $\boldsymbol{\zeta}' \equiv \boldsymbol{\gamma}'$ and (11) simplifies to (recall $n \notin \mathsf{FV}(\boldsymbol{\gamma}')$, $n \in \mathsf{FV}(\boldsymbol{t}')$, and possibly $n \in \mathsf{FV}(\boldsymbol{\delta}')$)

$$\boldsymbol{\zeta}''[n] \quad \equiv \quad \mathtt{R}\; n\; \boldsymbol{\gamma}' \left(\lambda n, p\,.\, \mathtt{If}\,(|B|^{\boldsymbol{z}'}_{\boldsymbol{\delta}'[\boldsymbol{t}']})\; p\; \boldsymbol{\delta}'[\boldsymbol{t}'] \right)$$

Modal induction rule - technical details

$$\frac{\Gamma \vdash \Box A(0) \qquad \Box\Delta \vdash \Box A(n) \to A(\mathtt{S}n)}{\Gamma\,,\ \Box\Delta \vdash \Box A(n)}\ \mathtt{Ind}^{\mathtt{m}}_{\iota}$$

We are given

$$|\Gamma|^{\boldsymbol{u}}_{\boldsymbol{\gamma}} \quad \vdash \quad \forall \boldsymbol{y}\, |A(0)|^{\boldsymbol{r}}_{\boldsymbol{y}} \tag{20}$$

and

$$|\Box\Delta|^{\boldsymbol{z}} \quad \vdash \quad \forall \boldsymbol{y'}\, |A(n)|^{\boldsymbol{x}}_{\boldsymbol{y'}} \to |A(\mathtt{S}n)|^{\boldsymbol{sx}}_{\boldsymbol{v}}$$

Since $\boldsymbol{v} \notin \mathsf{FV}(|\Box\Delta|^{\boldsymbol{z}})$ and $\boldsymbol{v} \notin \mathsf{FV}(\forall \boldsymbol{y'}\, |A(n)|^{\boldsymbol{x}}_{\boldsymbol{y'}})$ from the latter we easily obtain

$$|\Box\Delta|^{\boldsymbol{z}} \quad \vdash \quad \forall \boldsymbol{y'}\, |A(n)|^{\boldsymbol{x}}_{\boldsymbol{y'}} \to \forall \boldsymbol{v}\, |A(\mathtt{S}n)|^{\boldsymbol{sx}}_{\boldsymbol{v}} \tag{21}$$

With $t[n] :\equiv \mathtt{R}\, n\, r\, (\lambda n.s)$ for every corresponding pair $\langle r \in \boldsymbol{r} / s \in \boldsymbol{s} \rangle$ we show by induction on n in NA with base context $|\Gamma|^{\boldsymbol{u}}_{\boldsymbol{\gamma}}$ and step context $|\Box\Delta|^{\boldsymbol{z}}$ that

$$|\Gamma|^{\boldsymbol{u}}_{\boldsymbol{\gamma}}\,,\ |\Box\Delta|^{\boldsymbol{z}} \quad \vdash \quad \forall \boldsymbol{v}\, |A(n)|^{\boldsymbol{t}[n]}_{\boldsymbol{v}}$$

As $\boldsymbol{t}[0] \equiv \boldsymbol{r}$ the base is given by (20) and the step follows from (21) with $\boldsymbol{x} \mapsto \boldsymbol{t}[n]$ since $\boldsymbol{t}[\mathtt{S}n] \equiv \boldsymbol{s}\,\boldsymbol{t}[n]$. Thus challengers $\boldsymbol{\gamma}$ are simply preserved for $|\Gamma|$ and witnesses $\boldsymbol{t}[n]$ are easily constructed for $|\Box A(n)|$ in the conclusion sequent of $\mathtt{Ind}^{\mathtt{m}}_{\iota}$.

Remark 6. Our modal induction rule is equivalent to a special case of $\mathtt{Ind}_{\iota}$, since a $\Box$ can be placed in front of $A(\mathtt{S}n)$ from the step sequent of $\mathtt{Ind}^{\mathtt{m}}_{\iota}$. The equivalence of the two formulations for the step sequent can easily be proved using $\mathtt{AxT}$, $\mathtt{Ax4}$, $\mathtt{AxK}$ and $\Box^{i}$. Extracted terms are the same and the verifying proof only gets more direct.