

Deciding Intuitionistic Propositional Logic via Translation into Classical Logic

Daniel S. Korn¹ Christoph Kreitz²

¹ FG Intellektik, FB Informatik, TH-Darmstadt
Alexanderstraße 10, D-64238 Darmstadt
e-mail: korn@informatik.th-darmstadt.de, phone: +49-6151-16 6634

² Department of Computer Science
Cornell University, Ithaca, NY 14853-7501
e-mail: kreitz@cs.cornell.edu, phone: +1-607-255 1068

Abstract. We present a technique that efficiently translates propositional intuitionistic formulas into propositional classical formulas. This technique allows the use of arbitrary classical theorem provers for deciding the intuitionistic validity of a given propositional formula. The translation is based on the constructive description of a finite counter-model for any intuitionistic non-theorem. This enables us to replace universal quantification over all accessible worlds by a conjunction over the constructed finite set of these worlds within the encoding of a refuting Kripke-frame. This way, no additional theory handling by the theorem prover is required.

1 Introduction

Following the ideas of Ohlbach in [10] deduction systems for classical logic can be viewed as “processors” for theorem proving environments. In order to obtain a theorem prover for a given non-classical logic (which is then regarded as a “user-friendly surface language”) we simply need to write a “compiler” (i.e., translation procedure) for that particular logic into classical logic. Once we have proved soundness and completeness of this translation, we are free to use the very best of the existing classical theorem provers and let it do the work for us. A basic as well as natural approach to this end is given, for instance, by Moore [9] or van Benthem [17]. This approach essentially encodes the definition of validity w.r.t. the Kripke-semantics (see [3, 16]) of a given logic within the language of classical first-order logic. For example the modal formula $\Box a$ would be translated into $A \Rightarrow \forall v.(w_0 \mathbf{R} v \Rightarrow a(v))$, where \mathbf{R} denotes the accessibility relation between possible worlds, $a(v)$ denotes a being forced at v , w_0 denotes some arbitrary “root”-world and A is a conjunction of axiom formulas encoding the logic-specific properties of \mathbf{R} (e.g. reflexivity, transitivity, symmetry, etc.).

This technique has two major disadvantages: On the one hand the classical theorem prover has to reason about the semantics of the given logic over and over again at run-time. On the other hand decidability is lost even for the propositional fragment of the given logic.

There are several approaches to recover from the first problem. The so called “functional” translation proposed in [10], for instance, allows the use of theory

unification and thus to handle the theory knowledge about the given logic on a meta-level rather than on the object level. However, translation methods that explicitly deal with the second problem do not exist.

In this paper we present a translation procedure which includes both features: preserving decidability and liberating the theorem prover from dealing with theory knowledge. To this end, our translation mechanism uses more sophisticated knowledge about intuitionistic logic (which we will denote as “J” in the following). Our solution relies on a constructive view of the *finite-model property* for the propositional fragment of J (which we will denote as “J_p” in the following). Informally, this means that for any non-J_p-theorem we are able to construct a countermodel with a finite set of possible worlds. In the above example we would therefore become able to replace $\forall v.w_0\mathbf{R}v \Rightarrow \mathbf{a}(v)$ with $\bigwedge_{v \in R(w_0)} \mathbf{a}(v)$, where $R(w_0)$ denotes the finite set of all possible worlds in the constructed model which are accessible from w_0 .

In the following we begin by describing these latter ideas in more detail. We then present the translation itself. Finally, some remarks on the complexity of the translation as well as some considerations about related and future work will conclude our expositions.

2 Constructing finite potential countermodels

In this section we present the principal semantical ideas on which our translation mechanism is based. We assume the reader to be familiar with the basic concepts and terminology of classical and intuitionistic logic. Since, however, our definition of intuitionistic interpretation, forcing, model and validity results from a combination of the respective definitions in [3] and [16] and thus slightly differs from them we will repeat it in the following:

Definition 1 *Intuitionistic interpretation*

Let \mathbf{W} be a non-empty set, $\mathbf{R} \subseteq \mathbf{W} \times \mathbf{W}$ a transitive and reflexive relation (*accessibility*), and \mathbf{V} an *evaluation function* mapping elements of \mathbf{W} to sets of atomic formulas.

Then $\langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ is an *intuitionistic interpretation* or simply a *J_p-interpretation*. □

Given $\langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ and a $w \in \mathbf{W}$ we denote by w^* an arbitrary $v \in \mathbf{W}$ with $w\mathbf{R}v$. Thus by saying “ P holds for all w^* ” we mean “ P holds for all v with $w\mathbf{R}v$ ” and “there is a w^* such that P ” means “there is a v with $w\mathbf{R}v$ such that P ”.

Definition 2 *Intuitionistic forcing, model, validity*

Let $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ be a J_p-interpretation, $w \in \mathbf{W}$.

A formula A is called *forced* (otherwise *unforced*) at w — denoted as $w \Vdash A$ (and $w \not\Vdash A$ respectively) — iff one of the following conditions holds:

1. A is atomic, and $A \in \mathbf{V}(w^*)$ for any w^*
2. $A = A_1 \wedge A_2$, $w \Vdash A_1$, and $w \Vdash A_2$
3. $A = A_1 \vee A_2$, and $w \Vdash A_1$ or $w \Vdash A_2$

4. $A = \neg A_1$, and $w^* \not\vdash A_1$ for any w^*
5. $A = A_1 \Rightarrow A_2$, and $w^* \vdash A_2$ if $w^* \vdash A_1$ for any w^*

I_j is called an *intuitionistic model* or simply a J_p -*model* (otherwise J_p -*countermodel*) for A — denoted as $I_j \models_j A$ ($I_j \not\models_j A$ respectively) — iff $w \vdash A$ for any $w \in \mathbf{W}$. A is called *intuitionistically valid* or simply J_p -*valid* (otherwise J_p -*invalid*) — denoted as $\models_j A$ ($\not\models_j A$ respectively) — iff every J_p -interpretation is a J_p -model for it. \square

The elements of \mathbf{W} can be regarded as *knowledge stages* since they are meant to represent collections of formulas for which evidence is known at a particular stage in time. Consequently accessibility is to be seen in a temporal sense: for a given w any w^* is supposed to denote a future stage of knowledge. The following lemma which has been proved in [3] ensures that any evidence for some formula which is known at a given stage will remain known at any future stage:

Lemma 1 *Heredity of forcing*

Let $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ be a J_p -interpretation and A be a propositional formula.

Then for any $w \in \mathbf{W}$: $w \vdash A \Rightarrow w^* \vdash A$ \square

Throughout this paper we will refer to this property as the “*heredity condition*” or simply as “*heredity*”.

In terms of the above definitions the basic principle of our translation is to construct a decidable classical formula which describes the negation of a potential J_p -countermodel for the given input formula. Classical validity of this description will then ensure the impossibility of a J_p -countermodel for the given formula and hence its J_p -validity. To achieve decidability we must ensure to always construct a *finite* potential countermodel (i.e. a countermodel with finitely many knowledge stages) for the given formula. In the following we will focus on this issue.

According to the definition of J_p -models the construction of every J_p -countermodel for a given formula A must begin with a single root knowledge stage w_0 where A itself is not forced. We will then successively have to refine this countermodel depending on the principal operator of the formulas forced or unforced at some knowledge stage of the countermodel. If, for instance, $A = \mathbf{a} \vee \mathbf{b}$ then, since $w_0 \not\vdash A$, we need to add the facts $w_0 \not\vdash \mathbf{a}$ as well as $w_0 \not\vdash \mathbf{b}$.

For atomic, negated, and implicative formulas the refinement of our countermodel may include the addition of further knowledge stages. If, for instance, $A = \mathbf{a} \Rightarrow \mathbf{b}$ then according to the forcing conditions for implicative formulas there must be a w_0^* , say w_1 , with $w_1 \vdash \mathbf{a}$ but $w_1 \not\vdash \mathbf{b}$. We shall say w_1 *refutes* $\mathbf{a} \Rightarrow \mathbf{b}$. Note that we have to assume $w_1 \neq w_0$ since in general we cannot justify $w_0 \vdash \mathbf{a}$ (as opposed to $w_0 \not\vdash \mathbf{b}$ which must be the case since otherwise $w_0 \vdash \mathbf{a} \Rightarrow \mathbf{b}$). Hence, in general we must assume a J_p -countermodel for $\mathbf{a} \Rightarrow \mathbf{b}$ to have at least two knowledge stages as shown in fig. 1. If, however, we could for some reason justify $w_0 \vdash \mathbf{a}$ then w_0 would already refute $\mathbf{a} \Rightarrow \mathbf{b}$ by the reflexivity of accessibility. In this case the refinement of our countermodel would only consist of adding $w_0 \not\vdash \mathbf{b}$ to the previous properties. This observation is crucial to ensure finiteness of potential J_p -countermodels. The example in fig. 2 shows a countermodel which would have to be infinite if we did not use this observation.

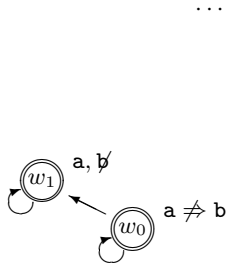


Fig. 1: countermodel for $a \Rightarrow b$

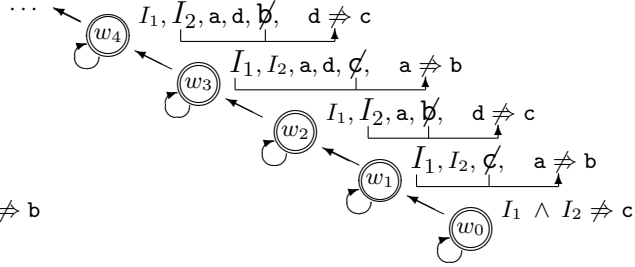


Fig. 2: infinite countermodel for

$$\underbrace{((a \Rightarrow b) \Rightarrow c)}_{I_1} \wedge \underbrace{((d \Rightarrow c) \Rightarrow b)}_{I_2} \Rightarrow c$$

As in the previous example we refute the main implication by adding a refuting knowledge stage w_1 accessible from w_0 with $w_1 \Vdash I_1 \wedge I_2$ (and thus $w_1 \Vdash I_1$ as well as $w_1 \Vdash I_2$) but $w_1 \not\Vdash c$. From $w_1 \Vdash I_1$, i.e. $w_1 \Vdash (a \Rightarrow b) \Rightarrow c$ and $w_1 \not\Vdash c$ we obtain the refinement $w_1 \not\Vdash a \Rightarrow b$ which is indicated by the arrow at w_1 in fig. 2. So we need to refine our countermodel by adding another knowledge stage w_2 refuting $a \Rightarrow b$, i.e. $w_2 \Vdash a$ but $w_2 \not\Vdash b$. However, since by the heredity condition $w_2 \Vdash I_2$, i.e. $w_2 \Vdash (d \Rightarrow c) \Rightarrow b$ we obtain $w_2 \not\Vdash d \Rightarrow c$ from $w_2 \not\Vdash b$. Therefore we must add a fourth knowledge stage w_3 refuting $d \Rightarrow c$, i.e. $w_3 \Vdash d$ but $w_3 \not\Vdash c$. Again $w_3 \Vdash I_1$ by heredity so $w_3 \not\Vdash a \Rightarrow b$ for the same reason as in w_1 . Hence we need another stage w_4 which again refutes $a \Rightarrow b$. Proceeding accordingly we continuously have to add knowledge stages that alternately refute the two implications thus yielding an infinite countermodel.

If, however, we take into account that $w_3 \Vdash a$ (which follows from $w_2 \Vdash a$ by the heredity condition) then we only have to refine our countermodel by adding the assumption $w_3 \not\Vdash b$ which must hold since otherwise $w_3 \Vdash a \Rightarrow b$. Hence we do not need to add w_4 to our countermodel anymore. The same reflection applies to $d \Rightarrow c$. Thus the construction of our countermodel actually terminates at w_3 . From these reflections we obtain, that we do not need to consider further accessible knowledge stages for refuting a given implication I whenever the current knowledge stage is accessible from one that already refutes I . Thus, our construction technique of adding knowledge stages in this manner while stepping backwards through implications will always lead to a finite countermodel.

For negated formulas we can achieve an even better behavior. Constructing a countermodel for such a formula, say $\neg A$, amounts to finding a knowledge stage w_0 where $\neg A$ is not forced. According to the intuitionistic semantics of the negation this means that A must be forced at least at one knowledge stage w_1 accessible from w_0 . As in the implicative case we must assume $w_1 \neq w_0$ unless we can somehow justify $w_0 \Vdash A$. In particular, if we again continue our construction process until we reach some w_1^* with $w_1^* \not\Vdash \neg A$ then we can conclude $w_1^* \Vdash A$ by heredity and thus we do not need to add any further knowledge stage refuting $\neg A$. Hence, stepping backwards through negations in this manner will also lead to finite countermodels. As a matter of fact, once we have refuted a

negated formula we do not need to add *any* further knowledge stage to our countermodel at all. To see why, we introduce the notion of *F-maximality*:

Definition 3 *Maximal knowledge stage*

Let $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ be an intuitionistic interpretation and F be a propositional formula. A knowledge stage $w \in \mathbf{W}$ is said to be *F-maximal* iff for any subformula A of F the following proposition is true:

$$(\exists w^*. w^* \Vdash A) \Rightarrow w \Vdash A \quad \square$$

In other words, w is maximal w.r.t. F if for every subformula A of F either evidence for A is known at w or it is impossible to construct evidence for A at all. The following theorem shows that for any propositional formula F and any knowledge stage an accessible F -maximal knowledge stage must always exist.

Theorem 1 *Existence of maximal knowledge stage*

Let $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ be a J_p -interpretation, $w \in \mathbf{W}$ and F be a propositional formula. Then there exists an F -maximal w^* .

Proof: Let $\Phi = \{A_1, \dots, A_n\}$ be the (finite) set of all subformulas of F . If there is no w^* forcing an A_i which is unforced at w then w is F -maximal. Otherwise consider some w^* where A_i is forced. Then A_i is also forced at any w^{**} by lemma 1 so we only need to look for some $A_j \in \Phi \setminus \{A_i\}$ to become forced at some w^{**} . If there is no such w^{**} then w^* is F -maximal. Otherwise we proceed in the same manner for w^{**} and $\Phi \setminus \{A_i, A_j\}$. Obviously this process terminates after at most k steps at some F -maximal $w^{(*k)}$ for $k \in \{0, \dots, n\}$. By transitivity of \mathbf{R} , $w^{(*k)}$ is also a w^* . \square

The chief feature of such an F -maximal knowledge stage is that classical and intuitionistic negation become semantically equivalent for *all* subformulas of F , since every subformula which is not forced at this stage will neither become forced at any accessible knowledge stage. Hence, $\neg A$ is forced at such a stage if and only if A is not, for any subformula A of F . So if F is the input formula of our translation and we have added a knowledge stage w_1 to our countermodel in order to refute some negated subformula $\neg A$ of F at some knowledge stage w_0 then we know by theorem 1 that there is an F -maximal w_1^* with $w_1^* \Vdash A$ by heredity. Hence we may refine our countermodel by adding w_1^* instead of w_1 . But then we do not need to add any further knowledge stage to our countermodel anymore since we can be sure that every subformula of F which we might assume to become forced at some knowledge stage accessible from w_1^* is already forced at w_1^* . Thus our technique for constructing countermodels will only add properties to the existing knowledge stages without adding further stages as soon as we have refuted a negated formula.

Eventually, refuting an atomic formula \mathbf{a} at some knowledge stage w_0 amounts to finding an accessible knowledge stage w_1 with $w_1 \Vdash \mathbf{a}$. But then also $w_0 \Vdash \mathbf{a}$ by heredity. Thus it suffices to refine our countermodel by adding $w_0 \Vdash \mathbf{a}$ so the construction technique will not have to add further stages in this case either.

These insights provide a basic scheme for the construction of a potential J_p -countermodel for a given formula F :

1. start with a knowledge stage w_0 where F is not forced.
2. for any negated or implicative subformula F' which is assumed to be unforced at some knowledge stage w_i and where w_i is neither F -maximal nor accessible from some w_j already refuting F' add an accessible stage w_{i+1} with the necessary properties to refute F' and all properties following from the heredity condition. If F' is negated then assume w_{i+1} to be F -maximal.
3. for any other subformula which is possibly not forced at some w_i or any subformula that is possibly forced at w_i add the appropriate forcing conditions for its immediate subformulas to w_i (and possibly all accessible stages) according to the definition of intuitionistic forcing.
4. repeat the two previous steps until no new knowledge stages can be added to the current countermodel and no new properties can be added to any of the knowledge stages in the current countermodel.

In the following sections we will elaborate the technical details for this.

3 The Translation

In this section we describe how we use the construction technique presented in the previous section in order to generate a classical translation $\Psi_{\mathbf{S}}(F)$ for any given formula F such that $\Psi_{\mathbf{S}}(F)$ is classically valid iff F is J_p -valid. Essentially this is achieved by generating a suitable representation for the negation of a possible J_p -countermodel within classical logic. We use unary function symbols to denote mappings from a given knowledge stage to an accessible one (cf. [10]). As shown above the existence of accessible knowledge stages within a potential J_p -countermodel depends only on negated and implicative subformulas not being forced somewhere. We will thus associate our function symbols with such subformulas. To describe this concept more precisely we introduce some formalism:

Definition 4 *Tree path, indexed subformula*

Let F be the formula under consideration and p denote a word from $\{1, 2\}^*$. Then F_p is the subformula of F that will be reached by traversing the formation tree of F according to p where a “1” means “go left” and a “2” means “go right”. A negation is defined to have a left subformula only. We shall call such a word p a *tree path* and F_p an *indexed subformula*.

By $\text{op}(F_p)$ we denote the *principal operator* of F_p and by $\text{pol}(F_p)$ we mean the *polarity* of F_p where F itself has polarity 0 and the polarity flips between 0 and 1 whenever going to the left subformula of an implication or a negation. By $\text{atomic}(F_p)$ we denote that F_p is atomic. Given a tree path p such that F_p does not denote a subformula of F (e.g. $(\neg A)_2$) then we assume $\text{op}(F_p) = \text{pol}(F_p)$ to be \emptyset and $\text{atomic}(F_p)$ to be false. \square

Fig. 3 shows the formula of fig. 2 where all connectives are marked with the tree path of the respective subformula. In this figure we have, e.g., $\text{op}(F_1) = “\wedge”$, $\text{pol}(F_{11}) = 1$ but $\text{pol}(F_{111}) = 0$ as well as $\text{atomic}(F_2)$.

The function symbols which represent transitions between knowledge stages of a potential countermodel will come in the form “ w_p ”, where p denotes the tree

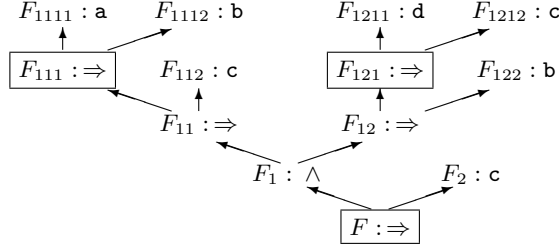


Fig. 3: formula tree for $((a \Rightarrow b) \Rightarrow c) \wedge ((d \Rightarrow c) \Rightarrow b) \Rightarrow c$

path of the subformula F_p which w_p is meant to be associated with. Thus, we will have to deal with the set $\mathbb{W}(F) = \{w_p | F_p \text{ is special}\}$, where we call a subformula F_p *special*, iff $\text{op}(F_p) \in \{\Rightarrow, \neg\}$ and $\text{pol}(F_p) = 0$. In the above example we obtain $\mathbb{W}(F) = \{w, w_{111}, w_{121}\}$. The respective subformulas appear boxed in fig. 3.

Recall that the elements w_p of $\mathbb{W}(F)$ denote functions which map a given knowledge stage w to an accessible one w^* refuting F_p . For fig. 2 and 3 we have w_{111} mapping w_1 to w_2 or w_{121} mapping w_2 to w_3 . Thus the set knowledge stages within a potential J_p -countermodel for the given formula F will be denoted by appropriate concatenations of the symbols in $\mathbb{W}(F)$ applied to some constant w_0 representing the root knowledge stage where F itself is not forced. In fig. 2, for instance, w_0 would denote w_0 and w would denote a function mapping w_0 to w_1 (thus denoted by $w(w_0)$). From there, we can possibly advance to $w_{111}(w(w_0))$ denoting w_2 or to $w_{121}(w(w_0))$, denoting another knowledge stage which could exist by refuting implication F_{121} . Eventually from w_2 , i.e. $w_{111}(w(w_0))$ we could go to $w_{121}(w_{111}(w(w_0)))$ denoting w_3 . Conversely, from $w_{121}(w(w_0))$ we can go to $w_{111}(w_{121}(w(w_0)))$. Any other possible application of function symbols to one of these six terms would denote a multiple refutation of the same implication which is unnecessary according to our reflections in the previous section. Hence these six terms constitute the desired set of all necessary knowledge stages in a possible countermodel for F , which we call the *accessibility set* for w_0 (denoted as $R_F(w_0)$). Fig. 4 illustrates the construction process we have just described.

In the following w_0 and any application of a concatenation of function symbols from $\mathbb{W}(F)$ to w_0 will be called *knowledge stage terms* or simply *k-terms*.

The example in fig. 4 yields the basic principle of constructing the accessibility set for a given k-term t : first of all, set $R_F(t) = \{t\}$. Then for any w_p occurring in t , look for a special subformula $F_{p'}$ of F_p such that $w_{p'}$ does not already occur in t and add the term $w_{p'}(t)$ to $R_F(t)$. This latter step will be called an *accessibility extension* of t' (at p over F). Eventually, $R_F(t)$ is defined as the fixpoint of

Fig. 4: construction of accessibility set

term	extension		
	via w	via w_{111}	via w_{121}
w_0	$w(w_0)$	—	—
$w(w_0)$	—	$w_{111}(w(w_0))$	$w_{121}(w(w_0))$
$w_{111}(w(w_0))$	—	—	$w_{121}(w_{111}(w(w_0)))$
$w_{121}(w(w_0))$	—	$w_{111}(w_{121}(w(w_0)))$	—
$w_{111}(w_{121}(w(w_0)))$	—	—	—
$w_{121}(w_{111}(w(w_0)))$	—	—	—

recursively constructing every accessibility extension for all its terms. However, terms of the form $\mathfrak{w}_p(t)$ with $\text{op}(F_p) = \text{“}\neg\text{”}$ do not have to be extended, since they denote F -maximal knowledge stages. In the following we shall call these k -terms F -maximal, as well. The formal definition of \mathbf{R}_F reads as follows:

Definition 5 *Accessibility extension, accessibility set*

Let F be an arbitrary propositional formula, p be a tree path for F and t be a knowledge stage term, where $\mathfrak{w}et$ denotes that \mathfrak{w} occurs somewhere within t .

The *accessibility extension* for t at p over F (denoted as $\mathbf{E}(F_p, t)$) is defined inductively as follows:

1. $\mathbf{E}(F_p, t) = \{\mathfrak{w}_p(t)\}$, iff F_p is special and $\mathfrak{w}_p \not\# t$
2. $\mathbf{E}(F_p, t) = \mathbf{E}(F_{p1}, t) \cup \mathbf{E}(F_{p2}, t)$, iff either $\text{op}(F_p) \in \{\neg, \Rightarrow\}$ and $\text{pol}(F_p) = 1$ or $\text{op}(F_p) \in \{\wedge, \vee\}$
3. $\mathbf{E}(F_p, t) = \emptyset$, otherwise

The *accessibility set* for t over F (denoted as $\mathbf{R}_F(t)$) is defined as the smallest set satisfying

1. $t \in \mathbf{R}_F(t)$
2. if t is a constant symbol then $\mathbf{E}(F, t) \subseteq \mathbf{R}_F(t)$
3. if $\mathfrak{w}_p(t') \in \mathbf{R}_F(t)$ and $\text{op}(F_p) = \text{“}\Rightarrow\text{”}$ then for any $\mathfrak{w}_q \varepsilon \mathfrak{w}_p(t')$:
 $\mathbf{E}(F_{q1}, \mathfrak{w}_p(t')) \subseteq \mathbf{R}_F(t)$ and $\mathbf{E}(F_{q2}, \mathfrak{w}_p(t')) \subseteq \mathbf{R}_F(t)$ □

Knowing how to construct the accessibility set we can now solve our initial problem, i.e. to translate a given propositional formula F into another propositional formula $\Psi_{\mathbf{S}}(F)$ which will be classically valid if and only if F is J_p -valid. The essential part of this translation is given by a morphism $\Psi_{\mathbf{F}}$ which encodes within classical logic whether a subformula F_p is forced at a knowledge stage denoted by a k -term t . The result of $\Psi_{\mathbf{F}}(t, F_p)$ will thus be a classical formula describing the negation of a partial J_p -countermodel for F_p if $\text{pol}(F_p) = 0$ or a partial J_p -model for F_p if $\text{pol}(F_p) = 1$ starting from the knowledge stage denoted by t .

In the example of fig. 2 and fig. 3 we have $F_{11} = (\mathbf{a} \Rightarrow \mathbf{b}) \Rightarrow \mathbf{c}$ (where $\text{pol}(F_{11}) = 1$) and hence translate it into $\Psi_{\mathbf{F}}(t, F_{11}) = \bigwedge_{t' \in \mathbf{R}_F(t)} (\Psi_{\mathbf{F}}(t', F_{111}) \Rightarrow \Psi_{\mathbf{F}}(t', F_{112}))$, meaning that if F_{11} is forced at t then F_{112} is forced at every t' accessible from t where F_{111} is forced. Conversely, $F_{111} = \mathbf{a} \Rightarrow \mathbf{b}$ has polarity 0 and hence $\Psi_{\mathbf{F}}(t', F_{111})$ means there is some knowledge stage $\mathfrak{w}_{111}(t')$ accessible from t' refuting F_{111} . So basically we should expect $\Psi_{\mathbf{F}}(t', F_{111})$ to reduce to $\Psi_{\mathbf{F}}(\mathfrak{w}_{111}(t'), \mathbf{a}) \Rightarrow \Psi_{\mathbf{F}}(\mathfrak{w}_{111}(t'), \mathbf{b})$. But this is only the case if $\mathfrak{w}_{111} \not\# t'$ (e.g. $t' = \mathfrak{w}(\mathfrak{w}_0)$), since otherwise t' denotes a knowledge stage accessible from one which already refutes F_{111} and hence by the heredity condition \mathbf{a} is already forced at t' (e.g. $t' = \mathfrak{w}_{121}(\mathfrak{w}_{111}(\mathfrak{w}(\mathfrak{w}_0)))$). In this case the translation would evaluate only to $\Psi_{\mathbf{F}}(t', \mathbf{b})$ thus adding the fact that \mathbf{b} is not forced at t' .

Likewise, for an F -maximal k -term t the translation will switch to a mode where no extensions of t are generated anymore. This translation will come in the form $\Psi_{\mathbf{F}\text{-max}}(t, F_p)$, essentially encoding that F_p is classically satisfied by the set of all subformulas of F that are forced at the knowledge stage denoted by t .

$$\begin{array}{ll}
\Psi_{\mathbf{S}}(F) & \Psi_{\mathbf{S}}(G) \\
= \Psi_{\mathbf{F}}(\mathbf{w}_0, F) & = \Psi_{\mathbf{F}}(\mathbf{w}_0, G) \\
= \Psi_{\mathbf{F}}(\mathbf{w}_0, F_1) \vee \Psi_{\mathbf{F}}(\mathbf{w}_0, F_2) & = \neg\Psi_{\mathbf{F}}(\mathbf{w}(\mathbf{w}_0), G_1) \\
= \mathbf{a}(\mathbf{w}_0) \vee \neg\Psi_{\mathbf{F}}(\mathbf{w}_2(\mathbf{w}_0), F_{21}) & = \neg\Psi_{\mathbf{F}\text{-max}}(\mathbf{w}(\mathbf{w}_0), G_1) \\
= \mathbf{a}(\mathbf{w}_0) \vee \neg\Psi_{\mathbf{F}\text{-max}}(\mathbf{w}_2(\mathbf{w}_0), F_{21}) & = \neg\neg\Psi_{\mathbf{F}\text{-max}}(\mathbf{w}(\mathbf{w}_0), G_{11}) \\
= \mathbf{a}(\mathbf{w}_0) \vee \neg\mathbf{a}(\mathbf{w}_2(\mathbf{w}_0)) & = \neg\neg(\Psi_{\mathbf{F}\text{-max}}(\mathbf{w}(\mathbf{w}_0), G_{111}) \vee \Psi_{\mathbf{F}\text{-max}}(\mathbf{w}(\mathbf{w}_0), G_{112})) \\
& = \neg\neg(\mathbf{a}(\mathbf{w}(\mathbf{w}_0)) \vee \neg\Psi_{\mathbf{F}\text{-max}}(\mathbf{w}(\mathbf{w}_0), G_{1121})) \\
& = \neg\neg(\mathbf{a}(\mathbf{w}(\mathbf{w}_0)) \vee \neg\mathbf{a}(\mathbf{w}(\mathbf{w}_0)))
\end{array}$$

Fig. 5: translation examples

Definition 6 *Formula morphism*

Let F be an arbitrary propositional formula, p be a tree path for F and t be an arbitrary knowledge stage term. Then the *formula morphisms* $\Psi_{\mathbf{F}}(t, F_p)$ and $\Psi_{\mathbf{F}\text{-max}}(t, F_p)$ are defined as follows:

$$\begin{aligned}
\Psi_{\mathbf{F}}(t, F_p) = & \Psi_{\mathbf{F}\text{-max}}(t, F_p), \text{ iff } t \text{ is } F\text{-maximal. Otherwise:} \\
& F_p(t), \text{ iff } \text{atomic}(F_p) \text{ and } \text{pol}(F_p) = 0 \\
& \bigwedge_{t' \in \mathbf{R}_F(t)} F_p(t'), \text{ iff } \text{atomic}(F_p) \text{ and } \text{pol}(F_p) = 1 \\
& \Psi_{\mathbf{F}}(t, F_{p1}) \text{op}(F_p) \Psi_{\mathbf{F}}(t, F_{p2}), \text{ iff } \text{op}(F_p) \in \{\vee, \wedge\} \\
\Psi_{\mathbf{F}}(\mathbf{w}_p(t), F_{p1}) \Rightarrow \Psi_{\mathbf{F}}(\mathbf{w}_p(t), F_{p2}), & \text{ iff } \text{op}(F_p) = "\Rightarrow", \text{pol}(F_p) = 0, \mathbf{w}_p \neq t \\
\Psi_{\mathbf{F}}(t, F_{p2}), & \text{ iff } \text{op}(F_p) = "\Rightarrow", \text{pol}(F_p) = 0, \mathbf{w}_p \varepsilon t \\
\bigwedge_{t' \in \mathbf{R}_F(t)} (\Psi_{\mathbf{F}}(t', F_{p1}) \Rightarrow \Psi_{\mathbf{F}}(t', F_{p2})), & \text{ iff } \text{op}(F_p) = "\Rightarrow" \text{ and } \text{pol}(F_p) = 1 \\
\neg\Psi_{\mathbf{F}}(\mathbf{w}_p(t), F_{p1}), & \text{ iff } \text{op}(F_p) = "\neg", \text{pol}(F_p) = 0 \\
\bigwedge_{t' \in \mathbf{R}_F(t)} (\neg\Psi_{\mathbf{F}}(t', F_{p1})), & \text{ iff } \text{op}(F_p) = "\neg" \text{ and } \text{pol}(F_p) = 1 \\
\Psi_{\mathbf{F}\text{-max}}(t, F_p) = & \\
& F_p(t), \text{ iff } \text{atomic}(F_p) \\
\Psi_{\mathbf{F}\text{-max}}(t, F_{p1}) \text{op}(F_p) \Psi_{\mathbf{F}\text{-max}}(t, F_{p2}), & \text{ iff } \text{op}(F_p) \in \{\vee, \wedge, \Rightarrow\} \\
\neg\Psi_{\mathbf{F}\text{-max}}(t, F_{p1}), & \text{ iff } \text{op}(F_p) = "\neg" \quad \square
\end{aligned}$$

The initial call of $\Psi_{\mathbf{F}}$ is defined through the following *specification morphism*:

Definition 7 *Specification morphism*

Let F be an arbitrary propositional formula and \mathbf{w}_0 be a constant symbol. Then the *specification morphism* $\Psi_{\mathbf{S}}(F)$ is defined as $\Psi_{\mathbf{F}}(\mathbf{w}_0, F)$. \square

Fig. 5 demonstrates the translation process for the two formulas $F = \mathbf{a} \vee \neg\mathbf{a}$ and $G = \neg\neg(\mathbf{a} \vee \neg\mathbf{a})$. Obviously $\Psi_{\mathbf{S}}(F)$ is classically invalid whereas $\Psi_{\mathbf{S}}(G)$ is classically valid. This amounts to the well-known J_p -invalidity of F and the J_p -validity of G respectively.

For $\Psi_{\mathbf{S}}$ the following theorem holds trivially:

Theorem 2 *Decidability of codomain of $\Psi_{\mathbf{S}}$*

If F is a propositional formula of then $\Psi_{\mathbf{S}}(F)$ is classically decidable.

Proof: Obviously every predicate symbol in $\Psi_{\mathbf{S}}(F)$ is ground-instantiated and no quantifiers occur in $\Psi_{\mathbf{S}}(F)$. Therefore $\Psi_{\mathbf{S}}(F)$ is a propositional formula and hence classically decidable. \square

In order to complete our logic morphism we define bidirectional *interpretation morphisms* $\Psi_{\mathbf{J}}, \Psi_{\mathbf{J}}^{-1}$ which map a \mathbf{J}_p -countermodel for a given formula to a classical countermodel for its translation and vice versa. Above we have described the intended meaning of the syntactical elements of the target language. The interpretation morphism amounts to formalizing these notions. We start with the unary function symbols \mathfrak{w}_p which are supposed to denote mappings from a given knowledge stage w to some w^* which refutes the associated subformula F_p . The following definition first introduces the notion of *knowledge extension* of a knowledge stage w indexed by a tree path p which is the set of all w^* refuting F_p . The respective function symbol \mathfrak{w}_p will be interpreted as an *extension successor function* which maps w to an arbitrary element of its knowledge extension indexed by p if it is non-empty or to an arbitrary F -maximal w^* otherwise:

Definition 8 *Indexed knowledge extension, extension successor*

Let $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ be a \mathbf{J}_p -interpretation $w \in \mathbf{W}$ be an arbitrary knowledge stage, F be an arbitrary propositional formula and p be a tree path for F .

The *knowledge extension* of w (over F indexed by p), denoted as $\mathbf{EX}(F_p, w)$, is defined as follows:

$$\mathbf{EX}(F_p, w) = \begin{cases} \{w^* | w^* \Vdash F_{p1} \text{ and } w^* \not\Vdash F_{p2}\}, & \text{if } \mathbf{op}(F_p) = \text{“}\Rightarrow\text{”} \\ \{w^* | w^* \Vdash F_{p1} \text{ and } w^* \text{ is } F\text{-maximal}\}, & \text{if } \mathbf{op}(F_p) = \text{“}\neg\text{”} \\ \emptyset, & \text{otherwise} \end{cases}$$

The *extension successor function* $\mathbf{succ}(F_p) : \mathbf{W} \longrightarrow \mathbf{W}$ over F indexed by p is defined as follows:

$$\mathbf{succ}(F_p, w) = \begin{cases} \text{some arbitrary } w' \in \mathbf{EX}(F_p, w), & \text{if } \mathbf{EX}(F_p, w) \neq \emptyset \\ \text{some arbitrary } F\text{-maximal } w^* & \text{otherwise} \end{cases} \quad \square$$

For the formula in fig. 2 and 3 we have, e.g., $\mathbf{EX}(F_{121}, w_2) = \{w_3\}$ and thus $\mathbf{succ}(F_{121}, w_2) = w_3$.

Within $\Psi_{\mathbf{J}}$ we interpret any function symbol \mathfrak{w}_p by $\mathbf{succ}(F_p)$. Hence, the universe \mathbf{U}_c of our classical interpretation $I_c = \langle \iota_c, \mathbf{U}_c \rangle$ will be the set of knowledge stages \mathbf{W} of the given \mathbf{J}_p -interpretation. The unary predicate symbols of our classical language then denote the set of all $w \in \mathbf{W}$ where the respective 0-ary predicate symbol of the intuitionistic language is forced.

Conversely, given a classical interpretation for the translated formula we will construct the set \mathbf{W} by a proper choice from the elements of \mathbf{U}_c . More precisely, \mathbf{W} will contain $\iota_c(\mathfrak{w}_0)$ as well as $\iota_c(t)$ for any k-term t in $\mathbf{R}_F(\mathfrak{w}_0)$ such that any subterm $\mathfrak{w}_p(s)$ of t actually refutes the subformula F_p , i.e. $I_c \not\models_c \Psi_{\mathbf{F}}(s, F_p)$. By this restriction we are enabled to conclude that the left hand subformula of the subformula associated with some function symbol in a given knowledge stage term t is forced at $\iota_c(t)$. For the formula of fig. 2, e.g., having $w_3 = \iota_c(t)$ with $t = \mathfrak{w}_{121}(\mathfrak{w}_{111}(\mathfrak{w}(\mathfrak{w}_0))) \in \mathbf{W}$ we want to determine from $\mathfrak{w}_{111}t$ that $F_{1111} = \mathbf{a}$ is forced at w_3 , so we may drop any consideration of further accessible knowledge stages refuting $F_{111} = \mathbf{a} \Rightarrow \mathbf{b}$ again.

Finally, we want any atomic subformula F_p to be forced at $\iota_c(t)$ if $I_c \models_c \Psi_{\mathbf{F}}(t, F_p)$. But in order to preserve the heredity condition we must restrict ourselves to po-

larity 1 subformulas, since only then we know the conjunction over all accessible knowledge stages to be satisfied by I_c . However, if we deal with an F -maximal k -term then we know that the respective knowledge stage has no successor other than itself. Hence, the above restriction will become obsolete in this case:

Definition 9 *Interpretation morphism*

Let F be an arbitrary propositional formula, $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ be a J_p -interpretation and $w_0 \in \mathbf{W}$ be an arbitrary knowledge stage.

Then the *interpretation morphism*, $\Psi_{\mathbf{J}}(I_j, F, w_0) = \langle \iota_c, \mathbf{U}_c \rangle$ is defined as follows:

1. $\mathbf{U}_c = \mathbf{W}$
2. $\iota_c(\mathbf{w}_p) = \text{succ}(F_p)$ for any $\mathbf{w}_p \in \mathbf{W}(F)$
3. $\iota_c(\mathbf{f}) = w_0$ for arbitrary $\mathbf{f} \notin \mathbf{W}(F)$
4. $\iota_c(\mathbf{p}) = \perp$ for every 0-ary predicate symbol \mathbf{p}
5. $\iota_c(\mathbf{p}) = \{w \in \mathbf{W} \mid w \Vdash \mathbf{p}\}$ for every unary predicate symbol \mathbf{p}
6. $\iota_c(\mathbf{p}) = \emptyset$ for every other predicate symbol \mathbf{p} .

Conversely, given a classical interpretation $I_c = \langle \iota_c, \mathbf{U}_c \rangle$ then the *reverse interpretation morphism* $\Psi_{\mathbf{J}}^{-1}(I_c, F) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ is defined as follows:

1. \mathbf{W} is the smallest set such that
 - (a) $\iota_c(\mathbf{w}_0) \in \mathbf{W}$
 - (b) If $\iota_c(t) \in \mathbf{W}$ then also $\iota_c(\mathbf{w}_p(t)) \in \mathbf{W}$, provided t is not F -maximal, $I_c \not\models_c \Psi_{\mathbf{F}}(t, F_p)$ and $\mathbf{w}_p(t) \in \mathbf{E}(F_{q1}, t) \cup \mathbf{E}(F_{q2}, t)$ for some $\mathbf{w}_q \varepsilon t$
2. $\mathbf{R} = \{(\iota_c(t), \iota_c(t')) \mid \iota_c(t), \iota_c(t') \in \mathbf{W} \text{ and } t' \in \mathbf{R}_F(t)\}$
3. $\mathbf{p} \in \mathbf{V}(w)$, iff there is a tree path p and a term t such that $\mathbf{p} = F_p$, $\text{atomic}(F_p)$, $w = \iota_c(t) \in \mathbf{W}$ and either
 - (a) t is F -maximal and $I_c \models_c \Psi_{\mathbf{F}\text{-max}}(t, F_p)$, or
 - (b) t is not F -maximal, $\text{pol}(F_p) = 1$ and $I_c \models_c \Psi_{\mathbf{F}}(t, F_p)$

The interpretation of terms and formulas is then defined for both directions of the interpretation morphism in the usual way. \square

As an example consider a classical countermodel $I_c = \langle \iota_c, \mathbf{U}_c \rangle$ for $\Psi_{\mathbf{S}}(F)$ as in fig. 5, i.e. $\mathbf{U}_c = \{w_0, w_1\}$, $\iota_c(\mathbf{w}_0) = w_0$, $\iota_c(\mathbf{w}_2(\mathbf{w}_0)) = w_1$ and $\iota_c(\mathbf{a}) = \{w_1\}$. Then $\Psi_{\mathbf{J}}^{-1}(I_c, F) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ with $\mathbf{W} = \{w_0, w_1\}$, $\mathbf{R} = \{(w_0, w_0), (w_0, w_1), (w_1, w_1)\}$ and $\mathbf{V}(w_1) = \{\mathbf{a}\}$ whereas $\mathbf{V}(w_0) = \emptyset$

In [7] we have proved $\Psi_{\mathbf{S}}$ to be sound and complete in the sense that there is a classical countermodel for $\Psi_{\mathbf{S}}(F)$ whenever there is a J_p -countermodel for F (soundness) as well as vice versa (completeness). The existence of these countermodels is ensured via $\Psi_{\mathbf{J}}, \Psi_{\mathbf{J}}^{-1}$ which map a given countermodel from source to target logic and vice versa respectively. In the following we give a short survey of these theorems.

Theorem 3 *Soundness of the translation*

Let F be a propositional formula and $I_j = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$ a J_p -interpretation such that $I_j \not\models_j F$, i.e. $w_0 \not\models F$ for some $w_0 \in \mathbf{W}$. Then $\Psi_{\mathbf{J}}(I_j, F, w_0) \not\models_c \Psi_{\mathbf{S}}(F)$.

Proof: the proof is split into two lemmata. The first one proves $w \Vdash F_p \Leftrightarrow I_c[t \setminus w] \models_c \Psi_{\mathbf{F}\text{-max}}(t, F_p)$ for a given F -maximal knowledge stage $w \in \mathbf{W}$ an arbitrary F -maximal knowledge stage term t and any subformula F_p of F , where $I_c = \Psi_{\mathbf{J}}(I_j, F, w_0) = \langle \iota_c, \mathbf{U}_c \rangle$. This is carried out by a straightforward induction over the degree of F_p (i.e. the number of logical connectives in F_p).

Based on this lemma, the second one proves the following for any knowledge stage $w \in \mathbf{W}$, any k-term t and any subformula F_p of F :

1. if $\text{pol}(F_p) = 0$ and $w \not\Vdash F_p$ then $I_c[t \setminus w] \not\models_c \Psi_{\mathbf{F}}(t, F_p)$
2. if $\text{pol}(F_p) = 1$ and $w \Vdash F_p$ then $I_c[t \setminus w] \models_c \Psi_{\mathbf{F}}(t, F_p)$

This is also carried out by simultaneous induction over the degree of F_p . Since by definition $\iota_c(w_0) = w_0$ and $\text{pol}(F) = 0$, we then have $I_c \not\models_c \Psi_{\mathbf{F}}(w_0, F) = \Psi_{\mathbf{S}}(F)$. \square

Theorem 4 *Completeness of the translation*

Let F be a propositional formula and $I_c = \langle \iota_c, \mathbf{U}_c \rangle$ a classical interpretation such that $I_c \not\models_c \Psi_{\mathbf{S}}(F)$. Then $\Psi_{\mathbf{J}}^{-1}(I_c, F) \not\models_j F$.

Proof: the proof is again split into two lemmata. The first one proves $I_c \models_c \Psi_{\mathbf{F}\text{-max}}(t, F_p) \Leftrightarrow \iota_c(t) \Vdash F_p$ for any F -maximal k-term t with $\iota_c(t) \in \mathbf{W}$ and any subformula F_p of F , where $I_j = \Psi_{\mathbf{J}}^{-1}(I_c, F) = \langle \mathbf{W}, \mathbf{R}, \mathbf{V} \rangle$. This is again carried out by a straightforward induction over the degree of F_p .

For the second lemma we need the definition of *accessibility-set compatibility*:

A knowledge stage term t is accessibility-set compatible to a tree path p w.r.t. a given formula F (denoted as $\text{comp}_F(p, t)$) if t is F -maximal or if $\mathbf{E}(F_p, t) \subseteq \mathbf{R}_F(t)$. This ensures $\iota_c(t') \in \mathbf{W}$ for extensions t' of t which may occur within $\Psi_{\mathbf{F}}(t, F_p)$.

Based on this definition, the previous lemma, and some lemmata which ensure the necessary properties for \mathbf{W} and \mathbf{R} , the second lemma proves the following for any k-term t such that $\iota_c(t) \in \mathbf{W}$ and any subformula F_p of F with $\text{comp}_F(p, t)$:

1. if $\text{pol}(F_p) = 0$ and $I_c \not\models_c \Psi_{\mathbf{F}}(t, F_p)$ then $\iota_c(t) \not\Vdash F_p$
2. if $\text{pol}(F_p) = 1$ and $I_c \models_c \Psi_{\mathbf{F}}(t, F_p)$ then $\iota_c(t) \Vdash F_p$

This is also carried out by simultaneous induction over the degree of F_p . The only non-trivial case is for $\text{op}(F_p) = "\Rightarrow"$, $\text{pol}(F_p) = 0$ and $w_p \varepsilon t$. Then the restrictive definition of \mathbf{W} allows to deduce $I_c \models_c \Psi_{\mathbf{F}}(t, F_{p1})$ and hence by induction hypothesis $\iota_c(t) \Vdash F_{p1}$ from $w_p \varepsilon t$ which otherwise does not follow from $I_c \not\models_c \Psi_{\mathbf{F}}(t, F_p) = \Psi_{\mathbf{F}}(t, F_{p2})$.

Finally, knowing by definition that $\iota_c(w_0) \in \mathbf{W}$, $\text{pol}(F) = 0$ and $\text{comp}_F(\emptyset, w_0)$, we obtain $\iota_c(w_0) \not\Vdash F$ and hence $I_j \not\models_j F$. \square

4 Complexity of the morphism

As mentioned above the set of knowledge stages in a potential \mathbf{J}_p -countermodel for a given formula only depends on its special subformulas. Therefore we consider the complexity of a countermodel compared to the number of these special subformulas. By means of the notation introduced in the previous section we will have to consider the cardinality of $\mathbf{R}_F(w_0)$ for the root knowledge stage w_0 compared to the one of $\mathbf{W}(F)$ for a given formula F . The following theorem deals with the worst-case complexity:

No.	Res.	Tran.	DP	Tot.	ft	No.	Res.	Tran.	DP	Tot.	ft
1	inv.	100	17	117	0	12	inv.	684	483	1167	0
2	inv.	17	0	17	10	13	val.	50	0	50	0
3	val.	16	0	16	0	14	inv.	67	16	83	0
4	inv.	50	17	67	0	15	inv.	34	16	50	0
5	inv.	33	17	50	0	16	inv.	0	0	0	0
6	inv.	0	0	0	0	17	inv.	84	116	200	0
7	inv.	16	0	16	0	71a	val.	17	0	17	0
8	inv.	0	0	0	0	71b	inv.	83	34	117	0
9	val.	17	0	17	0	72a	val.	116	134	250	450
10	val.	33	67	100	0	72b	val.	250	583	833	12020
11	val.	16	0	16	0	74	val.	116	17	133	3540

No. =
Pelletier's problem no.
Res. =
validity result
Tran. =
translation runtime
DP =
Davis-Putnam runtime
Tot. =
Tran.+DP
ft =
ft runtime

Fig. 6: Runtime comparison for propositional Pelletier problems

Theorem 5 *Worst-case complexity of countermodels*

Let $n = |\mathbf{w}(F)|$ for a propositional formula F . Then $|\mathbf{R}_F(\mathbf{w}_0)| \in \mathcal{O}(n!)$. \square

The formula in fig. 2 and fig. 3 shows an example for the worst-case complexity. The two special subformulas F_{111} and F_{121} are not subformulas of each other. We shall say that they are *unordered* w.r.t. the given formula. Hence, any possible extension ordering between these two must be considered for $w_1 = \mathbf{w}(\mathbf{w}_0)$ (cf. fig. 4). In general, having n unordered implicative special subformulas which are all left hand subformulas of the same implicative special subformula F_p and not of any negated special subformula we must consider $\mathcal{O}(n!)$ different reduction orderings and hence $\mathcal{O}(n!)$ elements of $\mathbf{R}_F(\mathbf{w}_p(t))$ for a suitable k-term t .

This situation becomes less severe the more the implicative special subformulas are ordered and the more negated special subformulas occur in the given formula F . If, for instance, $F = \neg F_1$ then the cardinality of $\mathbf{R}_F(\mathbf{w}_0) = \{\mathbf{w}_0, \mathbf{w}(\mathbf{w}_0)\}$ is just 2 and hence in $\mathcal{O}(1)$ (cf. formula G in fig. 5).

Therefore the average-case cardinality of $\mathbf{R}_F(\mathbf{w}_0)$ is as difficult to predict as the average shape of a formula. Moreover the relation between $|\mathbf{R}_F(\mathbf{w}_0)|$ and the actual computation time for the translation as well as the size of its output is not a trivial one since it heavily depends on various structural features of the given formula. It is obvious from the definition of our translation, however, that in the worst-case the translation output encodes the maximum search space for a contraction-free sequent derivation. Yet our technique is *constructively adequate* in the sense that search space complexity increases along with the degree of constructivity of the possible proof. Fig. 6 shows a practical runtime comparison for the propositional problems presented in [13] between a prototypical Eclipse-PROLOG implementation of our technique together with the non-normalform Davis-Putnam prover presented in [12] and the C implementation of the ft-prover [14] (version 1.23). All runtimes were measured on a SPARC 10 in microseconds where “0” means “not measurable”. This comparison actually confirms the claim of constructive adequateness:

- We could beat ft for formulas involving negated special subformulas in the immediate vicinity of the root (72a,72b,74).
- We were beaten for the above-mentioned worst-case type formulas (12,17).
- For intermediate cases we achieved comparable results (1-11,13-16,71a,71b).

We regard these evaluations as promising inasmuch as neither our translation method nor the Davis-Putnam prover have yet been optimized or encoded in an efficient programming language (as opposed to *ft* for which both has been done).

5 Conclusion, related and further work

In this paper we have presented a procedure which translates propositional intuitionistic formulas into propositional classical ones and thus preserves decidability. The output of this procedure neither encodes theory knowledge within the object language such that prover would have to derive it at run-time, nor does it yield the use of theory unification. Rather, it incorporates insights on how to construct finite countermodels for any non- J_p -theorem. In addition to a comprehensive presentation of the principal ideas and the technical details we have briefly considered the main aspects of the complexity of our method.

There are various approaches which deal with intuitionistic theorem-proving — theoretical as well as practical ones. We have mentioned the general translation methods for modal logics which (explicitly or implicitly) cover J_p in a very unspecific manner via the embedding into $S4$ (e.g. [9, 10, 15, 17]). The matrix method presented in [18] amounts to the functional translation principle described in [10] and has been implemented for intuitionistic logic as described in [11]. All these approaches share the disadvantage of neither preserving decidability (at least not explicitly) nor of liberating the theorem prover of dealing somehow with theory knowledge at run-time.

Contraction-free tableau calculi for intuitionistic logic have been presented in [1, 5, 8], where the one in [1] is implemented as described in [4]. They explicitly preserve decidability but since they are connective-driven they cannot compete with matrix or resolution based calculi which are based on connecting complementary literals. Nevertheless, they provided us with the key ideas for our translation method. The refutation system for first-order intuitionistic logic which we have formulated in [6] is based on the ideas presented here and allows a direct comparison between our approach and the contraction-free calculi. A practically efficient implementation of the contraction-free techniques is presented in [14]. Decidability is achieved there by a kind of loop checking that resembles the ideas mentioned in section 2. However, it does not incorporate any concept comparable to our maximality principles which would allow unrestricted classical reasoning. Yet, it turns out to have an astonishing practical performance (see fig. 6) but this may — at least partly — be explained by the efficient encoding in *C*.

In [2] a technique to construct finite J_p -countermodels for non- J_p -theorems based on a refutation calculus (“*CRIP*”) has been presented. For implicative special subformulas the potential countermodels constructed by our approach are perfectly equivalent to the ones constructed by *CRIP* disregarding the provisos for rules *axiom* and (11). However, since negated subformulas are treated there only as special cases of implications we are able to construct considerably smaller countermodels for negated special subformulas.

The present stage of our translation leaves open a lot of space for optimizations. In particular, this includes further reductions of the generated accessibility sets.

Some of these optimizations are obvious but we have not mentioned them here for the sake of simplicity. Yet, our long-term goal is to extend our ideas to full first-order logic. Work on this issue is currently in progress (cf. [6]).

Acknowledgments

We would like to thank Heiko Mantel for helping us to put the text into a readable form as well as Jens Otten for providing us with a non-normalform Davis-Putnam prover which is as fast as its source code is short and cryptic.

References

1. R. DYCKHOFF: *Contraction-Free Sequent Calculi for Intuitionistic Logic*, Journal of Symbolic Logic, Vol. 57, No. 3, pp. 795–807, 1992
2. R. DYCKHOFF, L. PINTO: *Loop-free construction of counter-models for intuitionistic propositional logic*, In Behara, Fritsch, Lintz (eds.): *Symposia Gaussiana, Conf. A*, pp. 225–232, de Gruyter, 1995
3. M. FITTING: *Intuitionistic Logic Model Theory and Forcing*, Amsterdam, ¹1969
4. A. HEUERDING, E. A.: *Propositional Logics on the Computer*, Proceedings of the 4th TABLEAUX '95, LNAI 918, pp. 310–323, 1995
5. J. HUDELMAIER: *An $O(n \log n)$ -space decision procedure for intuitionistic propositional logic*, In: *Journal of Logic and Computation* 3(1)1, pp. 63–75, 1993
6. D. KORN, C. KREITZ.: *A Constructively Adequate Refutation System for Intuitionistic Logic*, Tech. Rep. AIDA-96-14, TH Darmstadt, 1996.
7. D. KORN, C. KREITZ.: *Efficiently Deciding Intuitionistic Propositional Logic via Translation into Classical Logic.*, Tech. Rep. AIDA-96-10, TH Darmstadt, 1996.
8. P. MIGLIOLI, U. MOSCATO, M. ORNAGHI: *An Improved Refutation System for Intuitionistic Predicate Logic*, Journal of Automated Reasoning 13, pp 361–373, 1994
9. ROBERT C. MOORE: *Reasoning about Knowledge and Action*, Proceedings of the IJCAI-77, pp 223–227, Stanford, California 94305, 1977
10. H. J. OHLBACH: *Semantics-Based Translation Methods for Modal Logics*, Journal of Logic and Computation, Vol. 1, no. 6, pp 691–746, 1991
11. J. OTTEN, C. KREITZ: *A Connection Based Proof Method for Intuitionistic Logic*, Proceedings of the 4th TABLEAUX '95, LNAI 918, pp. 122–137, 1995
12. J. OTTEN: *On the Advantage of a Non-Clausal Davis-Putnam Procedure*, Tech. Rep. AIDA-97-1, TH Darmstadt, 1997
13. F. J. PELLETIER: *Seventy-Five Problems for Testing Automatic Theorem Provers*, Journal of Automated Reasoning 2, pp 191–216, 1986
14. D. SAHLIN, T. FRANZÉN, S. HARIDI: *An Intuitionistic Predicate Logic Theorem Prover*, Journal of Logic and Computation, Vol. 2, no. 5, pp 619–656, 1992
15. R. B. SCHERL: *A Constraint Logic Approach to Automated Modal Deduction*, Report UIUCDCS-R-93-1803, University of Illinois at Urbana-Champaign, 1993
16. K. SCHÜTTE: *Vollständige Systeme modaler und intuitionistischer Logik*, Springer, 1968
17. J. VAN BENTHEM: *Correspondence Theory*, In: D. Gabbay and F. Guenther (eds.): *Handbook of Philosophical Logic, Vol. II*, pp. 167–247, Reidel, ¹1984
18. L. WALLEN: *Automated Proof Search in Non-Classical Logics*, MIT Press, ¹1990