Reasoning inside a formula and ontological correctness of a formal mathematical text

Andrei Paskevich[1], Konstantin Verchinine[1],
Alexander Lyaletski[2], and Anatoly Anisimov[2]

[1] Université Paris 12, IUT Sénart/Fontainebleau, 77300 Fontainebleau, France,
andrei@capet.iut-fbleau.fr verko@capet.iut-fbleau.fr
[2] Kyiv National Taras Shevchenko University, Faculty of Cybernetics,
03680 Kyiv, Ukraine,
lav@unicyb.kiev.ua ava@unicyb.kiev.ua

Abstract. Dealing with a formal mathematical text (which we regard as a structured collection of hypotheses and conclusions), we often want to perform various analysis and transformation tasks on the initial formulas, without preliminary normalization. One particular example is checking for "ontological correctness", namely, that every occurrence of a non-logical symbol stems from some definition of that symbol in the foregoing text. Generally, we wish to test whether some known fact (axiom, definition, lemma) is "applicable" at a given position inside a statement, and to actually apply it (when needed) in a logically sound way.
In this paper, we introduce the notion of a locally valid statement, a statement that can be considered true at a given position inside a first-order formula. We justify the reasoning about "innards" of a formula; in particular, we show that a locally valid equivalence is a sufficient condition for an equivalent transformation of a subformula. Using the notion of local validity, we give a formal definition of ontological correctness for a text written in a special formal language called ForTheL.

1 Introduction

In a mathematical text, be it intended for a human reader or formalized for automated processing (Mizar Mathematical Library [1] is a classical example, see also [2]), we rarely meet "absolute", unconstrained rules, definitions, statements. Usually, everything we use is supplied with certain conditions so that we have to take them into consideration. For example, we can not reduce the fraction $\frac{xy}{x}$ until we prove that x is a nonzero number.

Let us consider a formula of the form $(\cdots \forall x\,(x \in \mathbb{R}^+ \supset (\cdots \frac{xy}{x} \cdots)) \cdots)$. It seems to be evident that we can replace $\frac{xy}{x}$ with y, but could we justify that? The task itself seems to be absurd: as soon as a term depends on bound variables, we can not reason about it. In the traditional fashion, we should first split our big formula up to the quantifier that binds x, make a substitution (or skolemization), separate $x \in \mathbb{R}^+$, and only then make the simplification.

However, while the statement "x is non-zero" is surely meaningless, we can say that "x is non-zero in this occurrence of $\frac{xy}{x}$". Our intuition suggests that

along with the usual notion of validity, a certain *local validity* of a proposition can be defined with respect to some position in a formula. A statement that is generally false or just meaningless can become locally valid being considered in the corresponding context. In what follows, we call such a proposition a *local property* of the position in question.

It can be argued that there is no gain in any simplifications when a formula to be simplified lies deep inside. We would split our big formula anyway to use the properties of that fraction in a proof. However, we believe that it is useful and instructive to simplify a problem in its initial form as far as possible in order to select the most perspective direction of the proof search.

Local properties are also necessary to verify (and even to define!) what we call *ontological correctness* of a mathematical text. Informally, we consider a text ontologically correct whenever it contains no occurrence of a non-logical symbol that comes from nowhere: for every such occurrence there must be an applicable definition or some other "introductory" premise. The purpose of ontological correctness may be not immediately obvious: for example, the famous disjunction "*to be or not to be*" is perfectly valid (at least, in classical logic) even if the sense of being has never been defined. Nevertheless, ontologically correct texts are preferable in many aspects.

Ontological correctness is closely related to type correctness in typed languages (especially, in weakly typed systems such as WTT [3]). It allows to spot formalization errors which otherwise could hardly be detected. Indeed, an accidental ontological incorrectness most often implies logical incorrectness (i.e. presence of false or unprovable claims). And it is much harder to trace a failure log of a prover back to an invalid occurrence than to discover it in the first place.

Moreover, during ontological verification, we obtain information about applicability of definitions and other preliminary facts at individual positions in the text in question. As long as subsequent transformations (e.g. during the logical verification phase) preserve ontological correctness and other local properties (and that should always be the case) we can unfold definitions and apply lemmas without further checking.

This paper is devoted to formalization of ontological correctness for a particular language of formal mathematical texts, called ForTheL [4]. To this purpose, we develop a theoretical background for reasoning about local properties based on the notion of *local image*. The rest of the paper is organized as follows. In Section 2, we briefly describe the ForTheL language and provide an informal (at the moment) definition of ontological correctness of a ForTheL text. Section 3 introduces preliminary notions and notation which we use to define and investigate the notion of local image in Section 4. With the help of local images, we give a precise definition of ontological correctness in Section 5.

2 ForTheL language

Like any usual mathematical text, a ForTheL text consists of definitions, assumptions, affirmations, theorems, proofs, etc. The syntax of a ForTheL sentence

follows the rules of English grammar. Sentences are built of units: statements, predicates, notions (that denote classes of objects) and terms (that denote individual entities). Units are composed of syntactical primitives: nouns which form notions (e.g. "`subset of`") or terms ("`closure of`"), verbs and adjectives which form predicates ("`belongs to`", "`compact`"), symbolic primitives that use a concise symbolic notation for predicates and functions and allow to consider usual quantifier-free first-order formulas as ForTheL statements. Of course, just a little fragment of English is formalized in the syntax of ForTheL.

There are three kinds of sentences in the ForTheL language: assumptions, selections, and affirmations. Assumptions serve to declare variables or to provide some hypotheses for the following text. For example, the following sentences are typical assumptions: "`Let S be a finite set.`", "`Assume that m is greater than n.`". Selections state the existence of representatives of notions and can be used to declare variables, too. Here follows an example of a selection: "`Take an even prime number X.`". Finally, affirmations are simply statements: "`If p divides n - p then p divides n.`". The semantics of a sentence is determined by a series of transformations that convert a ForTheL statement to a first-order formula, so called *formula image*.

Example 1. The formula image of the statement "`all closed subsets of any compact set are compact`" is:

$$\forall\, \texttt{A}\,((\texttt{A is a set} \wedge \texttt{A is compact}) \supset \forall\, \texttt{B}\,((\texttt{B is a subset of A} \wedge \texttt{B is closed}) \supset \texttt{B is compact}))$$

ForTheL sections are: sentences, sentences with proofs, cases, and top-level sections: axioms, definitions, signature extensions, lemmas, and theorems. A top-level section is a sequence of assumptions concluded by an affirmation. Proofs attached to affirmations and selections are simply sequences of low-level sections. A case section consists of an assumption called *case hypothesis* followed by a sequence of low-level sections (the proof of a case).

Any section $\mathbb{A}$ or sequence of sections Δ has a formula image, denoted $|\mathbb{A}|$ or, respectively, $|\Delta|$. The image of a sentence with proof is the same as the image of that sentence taken without proof. The image of a case section is the implication ($H \supset$ `thesis`), where H is the formula image of the case hypothesis and `thesis` is a placeholder that will be replaced by the statement being proved during verification. The formula image of a top-level section is simply the image of the corresponding sequence of sentences.

The formula image of a sequence of sections $\mathbb{A}, \Delta$ is a conjunction $|\mathbb{A}| \wedge |\Delta|$, whenever $\mathbb{A}$ is a conclusion (affirmation, case section, lemma, theorem); or a universally quantified implication $\forall \boldsymbol{x}_{\mathbb{A}}(|\mathbb{A}| \supset |\Delta|)$, whenever $\mathbb{A}$ is a hypothesis (assumption, selection, case hypothesis, axiom, definition, signature extension). Here, $\boldsymbol{x}_{\mathbb{A}}$ denotes the set, possibly empty, of variables declared in $\mathbb{A}$ (this set also depends on the logical context of $\mathbb{A}$, since any variable which is declared above $\mathbb{A}$ in the text must not be bound in $|\mathbb{A}|$). The formula image of the empty sequence is $\top$, the truth.

Thus, a ForTheL text as a whole, being a sequence of section, can be expressed as a single logical formula. In what follows, we often use formulas, sections and sequence of sections interchangeably: whenever a section or a sequence of sections is mentioned where a formula is expected, the corresponding formula image should be considered.

In this syntax, we can express various proof schemes like proof by contradiction, by case analysis, and by general induction. The last scheme merits special consideration. Whenever an affirmation is marked to be proved by induction, the system constructs an appropriate induction hypothesis and inserts it into the statement to be verified. The induction hypothesis mentions a binary relation which is declared to be a well-founded ordering, hence, suitable for induction proofs. Note that we cannot express the very property of well-foundness in ForTheL (since it is essentially a first-order language), so that the correctness of this declaration is unverifiable and we take it for granted. After that transformation, the proof and the transformed statement can be verified in a first-order setting, without any specific means to build induction proofs.

Example 2. Consider the following ForTheL formalization of Newman's lemma about term rewriting systems. We give it in an abridged form, with some preliminary definitions and axioms omitted. The expression "`x -R> y`" means that `y` is a reduct of `x` in the rewriting system `R`; `R+` and `R*` denote, respectively, the transitive and the reflexive transitive closure of `R`.

```
Let a,b,c,d,u,v,w,x,y,z denote elements.
Let R,S,T denote rewriting systems.

Definition CRDef.   R is confluent iff
  for all a,b,c  such that a -R*> b and a -R*> c
  there exists d such that b -R*> d and c -R*> d.
Definition WCRDef.  R is locally confluent iff
  for all a,b,c  such that a -R> b  and a -R> c
  there exists d such that b -R*> d and c -R*> d.
Definition TrmDef.  R is terminating iff
  for all a,b  a -R+> b => b -<- a.
Definition NFRDef.  A normal form of x in R is
  an element y such that x -R*> y and y has no reducts in R.

Lemma TermNF.   Let R be a terminating rewriting system.
                Every element x has a normal form in R.
Proof by induction. Obvious.

Lemma Newman.
  Any locally confluent terminating rewriting system is confluent.
Proof.
  Let R be locally confluent and terminating.
  Let us demonstrate by induction that
        for all a,b,c  such that a -R*> b and a -R*> c
        there exists d such that b -R*> d and c -R*> d.
```

```
    Assume that a -R+> b and a -R+> c.
    Take u such that a -R> u and u -R*> b.
    Take v such that a -R> v and v -R*> c.
    Take w such that u -R*> w and v -R*> w.
    Take a normal form d of w in R.

    Let us show that b -R*> d.
      Take x such that b -R*> x and d -R*> x.
    end.
    Let us show that c -R*> d.
      Take y such that c -R*> y and d -R*> y.
    end.
  end.
qed.
```

Our formalization is simplified in that the notion "`element`" takes no arguments, i.e. we consider rewriting systems to be defined on a common (implicit) universum. Also, in our current implementation of ForTheL, one can not declare a given binary relation to be well-founded, and therefore a rewriting system is defined to be terminating iff its inverted transitive closure is a subset of *the* well-founded relation "`-<-`" (Definition `TrmDef`). The induction hypothesis (namely, that any reduct of `a` is confluent) is used to verify the selections "`Take x`..." and "`Take y`..." at the end of the proof. Note that we do not consider cases where `b` or `c`, or both are equal to `a` — these cases are trivial enough so that the system can handle them without our assistance.

The ForTheL proof of Newman's lemma, while being quite terse and close to a hand-written argument, is formal and has been automatically verified by the SAD proof assistant, using SPASS 2.2, E 0.99, and Vampire 7.45 as background provers. We refer the reader to the papers [4, 5] and to the website `http://ea.unicyb.kiev.ua` for a description of SAD and further examples.

We call a ForTheL text *ontologically correct* whenever: (a) every non-logical symbol (constant, function, notion or relation) in the text is either a signature symbol or is introduced by a definition; and (b) in every occurrence of a non-logical symbol, the arguments, if any, satisfy the guards of the corresponding definition or signature extension. Since ForTheL is a one-sorted and untyped language, these guards can be arbitrary logical formulas. Therefore, the latter condition cannot be checked by purely syntactical means nor by type inference of any kind. Instead, it requires proving statements about terms inside complex formulas, possibly, under quantifiers. The following sections provide a theoretical basis for such reasoning.

3 Preliminary notions

We consider a one-sorted first-order language with equality ($\approx$), the standard propositional connectives $\neg$, $\wedge$, $\vee$, $\supset$, $\equiv$, the quantifier symbols $\forall$ and $\exists$, and the boolean constant $\top$, denoting truth. The respective order of subformulas is

significant for some of our definitions, therefore we consider $F \wedge G$ and $G \wedge F$ as different formulas (the same is true for $\vee$, $\equiv$, and $\approx$). We write the negated equality $\neg(s_1 \approx s_2)$ as $s_1 \not\approx s_2$ and the negated truth $\neg\top$ as $\bot$.

We suppose that the sets of free and bound variables in any term or formula are always disjoint. Also, a quantifier on a variable may never appear in the scope of another quantifier on the same variable.

We consider substitutions as functions which map variables to terms. For any substitution ϕ, if $x\phi$ is different from x, we call the term $x\phi$ a *substitute* of x in ϕ. A substitution is *finite* whenever the set of its substitutes is finite. We write finite substitutions as sequences of the form $[t_1/x_1, \ldots, t_n/x_n]$.

Position is a word in the alphabet $\{0, 1, \ldots\}$. In what follows, positions are denoted by Greek letters τ, μ and ν; the letter ϵ denotes the null position (the empty word). Positions point to particular subformulas and subterms in a formula or term.

The *set of positions* in an atomic formula or a term E, denoted $\Pi(E)$, is defined as follows (below $i.\Pi$ stands for $\{\, i.\tau \mid \tau \in \Pi \,\}$):

$$\Pi(P(s_0, \ldots, s_n)) = \{\epsilon\} \cup \bigcup i.\Pi(s_i) \qquad \Pi(s \approx t) = \{\epsilon\} \cup 0.\Pi(s) \cup 1.\Pi(t)$$
$$\Pi(f(s_0, \ldots, s_n)) = \{\epsilon\} \cup \bigcup i.\Pi(s_i) \qquad \Pi(\top) = \{\epsilon\}$$

The *set of positions* in a formula H, denoted $\Pi(H)$, is the disjoint union

$$\Pi(F) = \Pi^+(F) \cup \Pi^-(F) \cup \Pi^\circ(F)$$

of the *set of positive positions* $\Pi^+(H)$, the *set of negative positions* $\Pi^-(H)$, and the *set of neutral positions* $\Pi^\circ(H)$ (in what follows, A stands for an atomic formula):

$$\begin{aligned}
\Pi^+(F \equiv G) &= \{\epsilon\} & \Pi^+(\forall x\, F) &= \{\epsilon\} \cup 0.\Pi^+(F)\\
\Pi^+(F \supset G) &= \{\epsilon\} \cup 0.\Pi^-(F) \cup 1.\Pi^+(G) & \Pi^+(\exists x\, F) &= \{\epsilon\} \cup 0.\Pi^+(F)\\
\Pi^+(F \vee G) &= \{\epsilon\} \cup 0.\Pi^+(F) \cup 1.\Pi^+(G) & \Pi^+(\neg F) &= \{\epsilon\} \cup 0.\Pi^-(F)\\
\Pi^+(F \wedge G) &= \{\epsilon\} \cup 0.\Pi^+(F) \cup 1.\Pi^+(G) & \Pi^+(A) &= \Pi(A)
\end{aligned}$$

$$\begin{aligned}
\Pi^-(F \equiv G) &= \varnothing & \Pi^-(\forall x\, F) &= 0.\Pi^-(F)\\
\Pi^-(F \supset G) &= 0.\Pi^+(F) \cup 1.\Pi^-(G) & \Pi^-(\exists x\, F) &= 0.\Pi^-(F)\\
\Pi^-(F \vee G) &= 0.\Pi^-(F) \cup 1.\Pi^-(G) & \Pi^-(\neg F) &= 0.\Pi^+(F)\\
\Pi^-(F \wedge G) &= 0.\Pi^-(F) \cup 1.\Pi^-(G) & \Pi^-(A) &= \varnothing
\end{aligned}$$

$$\begin{aligned}
\Pi^\circ(F \equiv G) &= 0.\Pi(F) \cup 1.\Pi(G) & \Pi^\circ(\forall x\, F) &= 0.\Pi^\circ(F)\\
\Pi^\circ(F \supset G) &= 0.\Pi^\circ(F) \cup 1.\Pi^\circ(G) & \Pi^\circ(\exists x\, F) &= 0.\Pi^\circ(F)\\
\Pi^\circ(F \wedge G) &= 0.\Pi^\circ(F) \cup 1.\Pi^\circ(G) & \Pi^\circ(\neg F) &= 0.\Pi^\circ(F)\\
\Pi^\circ(F \vee G) &= 0.\Pi^\circ(F) \cup 1.\Pi^\circ(G) & \Pi^\circ(A) &= \varnothing
\end{aligned}$$

For the sake of consistency, we set $\Pi^+(t) = \Pi(t)$ and $\Pi^-(t) = \Pi^\circ(t) = \varnothing$ for any term t.

Among positions, we distinguish those of formulas ($\Pi_{\mathbf{F}}$), those of atomic formulas ($\Pi_{\mathbf{A}}$), and those of terms ($\Pi_{\mathbf{t}}$). Obviously, $\Pi(F) = \Pi_{\mathbf{t}}(F) \cup \Pi_{\mathbf{F}}(F)$, $\Pi_{\mathbf{A}}(t) = \Pi_{\mathbf{F}}(t) = \varnothing$, $\Pi_{\mathbf{A}}(F) \subseteq \Pi_{\mathbf{F}}(F)$, $\Pi(t) = \Pi_{\mathbf{t}}(t)$. We split the sets $\Pi_{\mathbf{t}}$, $\Pi_{\mathbf{A}}$, and $\Pi_{\mathbf{F}}$ into positive, negative, and neutral parts, too.

Given a formula H and a position $\pi \in \Pi(H)$, the position $\widehat{\pi}$ is the maximal prefix of π in $\Pi_{\mathbf{F}}(H)$. In what follows, we will often use this conversion to extend notions and operations defined on positions from $\Pi_{\mathbf{F}}$ to the whole Π.

A formula or a term E along with a position $\tau \in \Pi(E)$ defines an *occurrence*.

Let us say that π is a *textual predecessor* of τ whenever $\pi = \omega.i.\mu$ and $\tau = \omega.j.\eta$ and $i < j$. Such positions will be called *adjacent*. If $\mu = \epsilon$, we will say that π is a *logical predecessor* of τ. By default, "predecessor" means "logical predecessor".

Given a formula or a term E and a position τ in $\Pi(E)$, we will denote by $E|_\tau$ the subformula or subterm occurring in that position. In what follows, $(*\,F)$ stands for $(\neg F)$, $(\forall x\, F)$, or $(\exists x\, F)$; and $(F\ *\ G)$ stands for $(F \equiv G)$, $(F \supset G)$, $(F \wedge G)$, or $(F \vee G)$:

$$\begin{aligned} E|_\epsilon &= E & (*\,F)|_{0.\tau} &= F|_\tau \\ (F * G)|_{0.\tau} &= F|_\tau & (F * G)|_{1.\tau} &= G|_\tau \\ P(s_0, \ldots, s_n)|_{i.\tau} &= s_i|_\tau & (s \approx t)|_{0.\tau} &= s|_\tau \\ f(s_0, \ldots, s_n)|_{i.\tau} &= s_i|_\tau & (s \approx t)|_{1.\tau} &= t|_\tau \end{aligned}$$

Given a formula or a term E, a position τ in $\Pi(E)$, and a formula or a term e, we will denote by $E[e]_\tau$ the result of replacement of $E|_\tau$ with e:

$$\begin{aligned} E[e]_\epsilon &= e & (*\,F)[e]_{0.\tau} &= *\,F[e]_\tau \\ (F * G)[e]_{0.\tau} &= F[e]_\tau * G & (F * G)[e]_{1.\tau} &= F * G[e]_\tau \\ P(s_0, \ldots, s_n)[e]_{i.\tau} &= P(s_0, \ldots, s_i[p]_\tau, \ldots, s_n) & (s \approx t)[e]_{0.\tau} &= s[e]_\tau \approx t \\ f(s_0, \ldots, s_n)[e]_{i.\tau} &= f(s_0, \ldots, s_i[p]_\tau, \ldots, s_n) & (s \approx t)[e]_{1.\tau} &= s \approx t[e]_\tau \end{aligned}$$

The expression e must be a term if $\tau \in \Pi_{\mathbf{t}}(E)$, and a formula otherwise. Free variables of e may become bound in $F[e]_\tau$.

4 Local validity and local properties

Given a formula F, a position $\pi \in \Pi_{\mathbf{F}}(F)$, and a formula U, we define the *local image* of U w.r.t. F and π, denoted $\langle U \rangle_\pi^F$, as follows:

$$\begin{aligned} \langle U \rangle_{0.\pi}^{F \equiv G} &= \langle U \rangle_\pi^F & \langle U \rangle_{1.\pi}^{F \equiv G} &= \langle U \rangle_\pi^G & \langle U \rangle_{0.\pi}^{\forall x F} &= \forall x\, \langle U \rangle_\pi^F \\ \langle U \rangle_{0.\pi}^{F \supset G} &= G \vee \langle U \rangle_\pi^F & \langle U \rangle_{1.\pi}^{F \supset G} &= F \supset \langle U \rangle_\pi^G & \langle U \rangle_{0.\pi}^{\exists x F} &= \forall x\, \langle U \rangle_\pi^F \\ \langle U \rangle_{0.\pi}^{F \wedge G} &= G \supset \langle U \rangle_\pi^F & \langle U \rangle_{1.\pi}^{F \wedge G} &= F \supset \langle U \rangle_\pi^G & \langle U \rangle_{0.\pi}^{\neg F} &= \langle U \rangle_\pi^F \\ \langle U \rangle_{0.\pi}^{F \vee G} &= G \vee \langle U \rangle_\pi^F & \langle U \rangle_{1.\pi}^{F \vee G} &= F \vee \langle U \rangle_\pi^G & \langle U \rangle_\varepsilon^F &= U \end{aligned}$$

Roughly, the formula $\langle U\rangle^F_\pi$ says "U is true at the position π in F". Note that this formula does not depend on the subformula $F|_\pi$. For a position $\pi \in \Pi_{\mathbf{t}}(F)$, we define $\langle U\rangle^F_\pi$ to be $\langle U\rangle^F_{\widehat{\pi}}$, where $\widehat{\pi}$ is the longest prefix of π in $\Pi_{\mathbf{F}}(F)$.

Example 3. Let F be the formula

$$\forall x\,(x \in \mathbb{N} \supset \forall n\,(n \in \mathbb{N} \supset (x \approx \mathbf{fib}(n) \equiv \\ \equiv ((n \leq 1 \wedge x \approx 1) \vee x \approx (\mathbf{fib}(n-1) + \mathbf{fib}(n-2))))))$$

This formula represents a recursive definition. We want to verify that the arguments $(n-1)$ and $(n-2)$ satisfy the guards of the definition and are strictly less than n.

Consider the second argument. Let π denote its position, 0.1.0.1.1.1.1.1.0. We want to prove $\langle (n-2) \in \mathbb{N} \wedge (n-2) < n\rangle^F_\pi$. The latter formula is equal to

$$\forall x\,(x \in \mathbb{N} \supset \forall n\,(n \in \mathbb{N} \supset ((n \leq 1 \wedge x \approx 1) \vee ((n-2) \in \mathbb{N} \wedge (n-2) < n))))$$

But this formula is false given $n = x = 0$. And that reveals an error in our definition: $x = 0$ makes false the left side of the disjunction $F|_{0.1.0.1.1}$, so we have to consider the right side with $n = 0$ in order to evaluate the truth value of the whole disjunction. Now it is easy to build a good definition F' of **fib**:

$$\forall x\,(x \in \mathbb{N} \supset \forall n\,(n \in \mathbb{N} \supset (x \approx \mathbf{fib}(n) \equiv \\ \equiv ((n \leq 1 \wedge x \approx 1) \vee (n \geq 2 \wedge x \approx (\mathbf{fib}(n-1) + \mathbf{fib}(n-2)))))))$$

Obviously, the local image $\langle (n-2) \in \mathbb{N} \wedge (n-2) < n\rangle^{F'}_{0.1.0.1.1.1.1.1.1.0}$ is a valid formula:

$$\forall x\,(x \in \mathbb{N} \supset \forall n\,(n \in \mathbb{N} \supset \\ \supset ((n \leq 1 \wedge x \approx 1) \vee (n \geq 2 \supset ((n-2) \in \mathbb{N} \wedge (n-2) < n)))))$$

Lemma 1. *For any F, $\pi \in \Pi(F)$, and a formula U, $\forall U \vdash \langle U\rangle^F_\pi$.*

Proof. Here, $\forall U$ denotes the universal closure of U. The formula $\langle U\rangle^F_\pi$ is equivalent to a universally quantified disjunction and U is a positive component of this disjunction. □

Lemma 2. (local modus ponens) $\vdash \langle U \supset V\rangle^F_\pi \supset (\langle U\rangle^F_\pi \supset \langle V\rangle^F_\pi)$

Lemma 2 can be proved by a simple induction on the length of π.

The lemmas above show that we can consistently reason about local properties. They are powerful enough to prove some interesting corollaries.

Corollary 1. $\vdash \langle U \equiv V\rangle^F_\pi \supset (\langle U\rangle^F_\pi \equiv \langle V\rangle^F_\pi)$

Proof. By Lemma 1 we have $\vdash \langle (U \equiv V) \supset (U \supset V)\rangle^F_\pi$. Hence by Lemma 2, $\vdash \langle (U \equiv V)\rangle^F_\pi \supset (\langle U \supset V\rangle^F_\pi)$. Again by local *modus ponens*, $\vdash \langle (U \equiv V)\rangle^F_\pi \supset (\langle U\rangle^F_\pi \supset \langle V\rangle^F_\pi)$. In the same way, $\vdash \langle (U \equiv V)\rangle^F_\pi \supset (\langle V\rangle^F_\pi \supset \langle U\rangle^F_\pi)$. □

Corollary 2. $\vdash \langle U \wedge V\rangle_\pi^F \equiv (\langle U\rangle_\pi^F \wedge \langle V\rangle_\pi^F)$

Proof. In order to prove the necessity, we take the propositional tautologies $(U \wedge V) \supset U$ and $(U \wedge V) \supset V$. In order to prove the sufficiency, we take the propositional tautology $U \supset (V \supset (U \wedge V))$. Then we "immerse" a chosen tautology inside the formula F by Lemma 1 and apply local *modus ponens*. □

Corollary 3. *For any quantifier-free context C,*

$$\vdash \big(\langle U_1 \equiv V_1\rangle_\pi^F \wedge \cdots \wedge \langle U_n \equiv V_n\rangle_\pi^F \wedge \langle t_1 \approx s_1\rangle_\pi^F \wedge \cdots \wedge \langle t_m \approx s_m\rangle_\pi^F\big) \supset \\ \supset \langle C[U_1,\ldots,U_n,t_1,\ldots,t_m] \equiv C[V_1,\ldots,V_n,s_1,\ldots,s_m]\rangle_\pi^F$$

The term "context" stands here for a formula with "holes", in which formulas or terms can be inserted, completing the context up to a well-formed formula. The corollary can be proved similarly to previous statements.

The key property of local images is given by the following theorem.

Theorem 1. *For any formulas F, U, V*

$$\begin{aligned} \pi \in \Pi_{\mathbf{F}}(F) \quad &\Rightarrow \quad \vdash \langle U \equiv V\rangle_\pi^F \supset (F[U]_\pi \equiv F[V]_\pi) \\ \pi \in \Pi^+_{\mathbf{F}}(F) \quad &\Rightarrow \quad \vdash \langle U \supset V\rangle_\pi^F \supset (F[U]_\pi \supset F[V]_\pi) \\ \pi \in \Pi^-_{\mathbf{F}}(F) \quad &\Rightarrow \quad \vdash \langle V \supset U\rangle_\pi^F \supset (F[U]_\pi \supset F[V]_\pi) \end{aligned}$$

This theorem is proved by induction on the length of π. The proof is quite straightforward and we omit it because of lack of space.

By Theorem 1, we can safely replace subformulas not only by equivalent formulas but by locally equivalent ones as well. Note that the inverse of the theorem holds in the propositional logic: $\vdash_0 \langle U \equiv V\rangle_\pi^F \equiv (F[U]_\pi \equiv F[V]_\pi)$. Local equivalence is there a criterion of substitutional equivalence. It is not the case for the first-order logic, where $(\exists x\, x \approx 0)$ is equivalent to $(\exists x\, x \approx 1)$.

Remark 1. In what follows, we often apply Theorem 1 and related results to positions from $\Pi_{\mathbf{t}}$, having in mind the position of the enclosing atomic formula. Note that any statement which is locally true in a term position is also locally true in the position of the enclosing atomic formula, since the local images are the same.

Corollary 4. *For any formula F, a position $\pi \in \Pi_{\mathbf{t}}(F)$, and terms s and t,*

$$\vdash \langle s \approx t\rangle_\pi^F \supset (F[s]_\pi \equiv F[t]_\pi)$$

Follows from Theorem 1 and Corollary 3.

Corollary 5. *For any formula F, a position $\pi \in \Pi_{\mathbf{F}}(F)$, and formulas U, V*

$$\begin{aligned} \vdash \langle U\rangle_\pi^F \supset (F[V]_\pi \equiv F[U \wedge V]_\pi) \qquad & \vdash \langle U\rangle_\pi^F \supset (F[V]_\pi \equiv F[U \supset V]_\pi) \\ \vdash \langle V \supset U\rangle_\pi^F \supset (F[V]_\pi \equiv F[U \wedge V]_\pi) \quad & \vdash \langle U \supset V\rangle_\pi^F \supset (F[V]_\pi \equiv F[U \vee V]_\pi) \end{aligned}$$

Consider a closed formula H of the form $\forall \boldsymbol{x}\,(C \supset (A \equiv D))$, where A is an atomic formula. Consider a formula F and a position $\pi \in \mathit{\Pi}_{\mathbf{A}}(F)$ such that $F|_\pi = A\sigma$ for some substitution σ. If we can prove $\langle C\sigma \rangle^F_\pi$, then we have $\langle A\sigma \equiv D\sigma \rangle^F_\pi$ by Lemma 1 and Corollary 2 (provided that H is among the premises). Then we can replace $A\sigma$ with $D\sigma$ by Theorem 1 (we generalize this technique in the following section). Returning to Example 3, we can guarantee that such an expansion is always possible (since $\langle n-1 \in \mathbb{N} \ \wedge \ n-2 \in \mathbb{N} \rangle^F_\pi$ holds) and is never infinite (since $\langle n-1 < n \ \wedge \ n-2 < n \rangle^F_\pi$ holds).

However, the notion of a local image introduced above has a disadvantage: it is not invariant w.r.t. transformations at adjacent positions.

Example 4. Since $\langle A \rangle^{A \wedge A}_0$ is valid, $(A \wedge A)$ is equivalent to $(\top \wedge A)$ by Theorem 1. But $\langle A \rangle^{A \wedge A}_1$ is also valid, whereas $\langle A \rangle^{\top \wedge A}_1$ is not.

Generally, we can build a formula F whose two subformulas U and V assure certain local properties for each other. Using these properties, we replace U with a locally equivalent formula U'. But thus we can lose the local properties of V.

This does not play an important role when we consider one-time transformations, e.g. simplifications. Indeed, one should check that simplification is possible just before doing it. But there are also certain local properties that we would prefer keep intact during the entire proof.

For example, we can verify the ontological correctness of a given occurrence of a function symbol in the initial task and it is quite desirable to preserve further this correctness in order to expand the definition of that symbol at any moment, without extra verifications.

To that aim, we slightly change the definition of a local image in such a way that only the formulas at *precedent* positions get into the context. Psychologically, this is natural, since assertions of that kind (type declarations, limits, etc) are usually written before "significant" formulas.

The *directed local image* of a formula U w.r.t. a formula F and a position $\pi \in \mathit{\Pi}_{\mathbf{F}}(F)$, denoted $\langle\!|U|\!\rangle^F_\pi$, is defined as follows:

$$\begin{array}{lll}
\langle\!|U|\!\rangle^{F \equiv G}_{0.\pi} = \langle\!|U|\!\rangle^F_\pi & \langle\!|U|\!\rangle^{F \equiv G}_{1.\pi} = \langle\!|U|\!\rangle^G_\pi & \langle\!|U|\!\rangle^{\forall x F}_{0.\pi} = \forall x\, \langle\!|U|\!\rangle^F_\pi \\
\langle\!|U|\!\rangle^{F \supset G}_{0.\pi} = \langle\!|U|\!\rangle^F_\pi & \langle\!|U|\!\rangle^{F \supset G}_{1.\pi} = F \supset \langle\!|U|\!\rangle^G_\pi & \langle\!|U|\!\rangle^{\exists x F}_{0.\pi} = \forall x\, \langle\!|U|\!\rangle^F_\pi \\
\langle\!|U|\!\rangle^{F \wedge G}_{0.\pi} = \langle\!|U|\!\rangle^F_\pi & \langle\!|U|\!\rangle^{F \wedge G}_{1.\pi} = F \supset \langle\!|U|\!\rangle^G_\pi & \langle\!|U|\!\rangle^{\neg F}_{0.\pi} = \langle\!|U|\!\rangle^F_\pi \\
\langle\!|U|\!\rangle^{F \vee G}_{0.\pi} = \langle\!|U|\!\rangle^F_\pi & \langle\!|U|\!\rangle^{F \vee G}_{1.\pi} = F \vee \langle\!|U|\!\rangle^G_\pi & \langle\!|U|\!\rangle^F_\varepsilon = U
\end{array}$$

For a position $\pi \in \mathit{\Pi}_{\mathbf{t}}(F)$, we define $\langle\!|U|\!\rangle^F_\pi$ to be $\langle\!|U|\!\rangle^F_{\widehat{\pi}}$, where $\widehat{\pi}$ is the longest prefix of π in $\mathit{\Pi}_{\mathbf{F}}(F)$.

First, note that all statements proved so far about "indirected" images hold for directed ones, too. In some sense, directed image is just a reduction, with some conditions and alternatives eliminated. This is illustrated by the following trivial lemma.

Lemma 3. $\quad \vdash \langle\!|U|\!\rangle^F_\pi \supset \langle U \rangle^F_\pi$

Theorem 2. *For any formula F and two adjacent $\pi, \tau \in \Pi_{\mathbf{F}}(F)$,*

$$\vdash \langle\!|U \equiv V|\!\rangle_{\pi}^{F} \supset \left(\langle\!|W|\!\rangle_{\tau}^{F[U]_{\pi}} \equiv \langle\!|W|\!\rangle_{\tau}^{F[V]_{\pi}}\right)$$

Proof. We proceed by induction on the length of π. It is easy to see that, if τ textually precedes π, then the formulas $\langle\!|W|\!\rangle_{\tau}^{F[U]_{\pi}}$ and $\langle\!|W|\!\rangle_{\tau}^{F[V]_{\pi}}$ are identical. So we can suppose that π textually precedes τ, that is, there exist ω, μ, and η such that $\pi = \omega.0.\mu$ and $\tau = \omega.1.\eta$. It is easy to see that we can reduce our problem to

$$\vdash \langle\!|U \equiv V|\!\rangle_{0.\mu}^{G*H} \supset \left(\langle\!|W|\!\rangle_{1.\eta}^{(G*H)[U]_{0.\mu}} \equiv \langle\!|W|\!\rangle_{1.\eta}^{(G*H)[V]_{0.\mu}}\right)$$

where $(G * H) = F|_{\omega}$. The latter is equivalent to

$$\vdash \langle\!|U \equiv V|\!\rangle_{\mu}^{G} \supset \left(\langle\!|W|\!\rangle_{1.\eta}^{G[U]_{\mu}*H} \equiv \langle\!|W|\!\rangle_{1.\eta}^{G[V]_{\mu}*H}\right)$$

and then to

$$\vdash \langle\!|U \equiv V|\!\rangle_{\mu}^{G} \supset \left((G[U]_{\mu} \star \langle\!|W|\!\rangle_{\eta}^{H}) \equiv (G[V]_{\mu} \star \langle\!|W|\!\rangle_{\eta}^{H})\right)$$

where $\star$ is either $\supset$ or $\vee$, in dependence of $*$. By Lemma 3 and Theorem 1, $\langle\!|U \equiv V|\!\rangle_{\mu}^{G}$ implies $(G[U]_{\mu} \equiv G[V]_{\mu})$, hence the claim is proved. □

Corollary 6. *For any formula F and two adjacent $\pi, \tau \in \Pi_{\mathbf{t}}(F)$,*

$$\vdash \langle\!|s \approx t|\!\rangle_{\pi}^{F} \supset \left(\langle\!|W|\!\rangle_{\tau}^{F[s]_{\pi}} \equiv \langle\!|W|\!\rangle_{\tau}^{F[t]_{\pi}}\right)$$

Finally, we introduce the notion of *local substitution*. Let H be a formula such that no quantifier occurs in H in the scope of another quantifier over the same variable. Given a position $\pi \in \Pi_{\mathbf{F}}(H)$, the result of local substitution $H[\sigma]_{\pi}$ is defined as follows:

$$\begin{aligned}
F[\sigma]_{\varepsilon} &= F & (F * G)[\sigma]_{0.\tau} &= F[\sigma]_{\tau} * G \\
(\neg F)[\sigma]_{0.\tau} &= \neg F[\sigma]_{\tau} & (F * G)[\sigma]_{1.\tau} &= F * G[\sigma]_{\tau} \\
(\forall x\, F)[\sigma]_{0.\tau} &= (F[x/x\sigma])[\sigma]_{\tau} & (\forall y\, F)[\sigma]_{0.\tau} &= \forall y\, F[\sigma]_{\tau} \\
(\exists x\, F)[\sigma]_{0.\tau} &= (F[x/x\sigma])[\sigma]_{\tau} & (\exists y\, F)[\sigma]_{0.\tau} &= \exists y\, F[\sigma]_{\tau}
\end{aligned}$$

where $x\sigma \neq x$ and $y\sigma = y$ in the last four equations, i.e. we eliminate the quantifiers over the instantiated variables. Here and below, we will assume that $x\sigma$ is free for x in F and further, σ does not instantiate any variable that occurs in one of the substitutes of σ.

When applied without restrictions, local substitutions may produce illegal instances (e.g. when variables of opposite polarities are instantiated). Also, local substitutions do not preserve local properties in adjacent positions. Consider the formula $F = \forall x\, P(x) \wedge A$ and the substitution $\sigma = [s/x]$ to be applied in F at $\pi = 1.0$, so that $F[\sigma]_{\pi} = (P(s) \wedge A)$. The atom A has the local property $\forall x\, P(x)$ in F but loses this property in $F[\sigma]_{\pi}$ — something we would like to avoid.

Therefore, we introduce a more fine-grained operation. As before, let H be a formula such that no quantifier occurs in H in the scope of another quantifier over the same variable, and π be a position in $\Pi_{\mathbf{F}}(H)$.

$$
\begin{array}{rlrl}
(F \supset G)[\sigma]^+_{0.\tau} &= F[\sigma]^-_\tau \supset \bot & (F \supset G)[\sigma]^+_{1.\tau} &= F \supset G[\sigma]^+_\tau \\
(F \vee G)[\sigma]^+_{0.\tau} &= F[\sigma]^+_\tau \vee \bot & (F \vee G)[\sigma]^+_{1.\tau} &= F \vee G[\sigma]^+_\tau \\
(F \wedge G)[\sigma]^+_{0.\tau} &= F[\sigma]^+_\tau \wedge G & (F \wedge G)[\sigma]^+_{1.\tau} &= F \wedge G[\sigma]^+_\tau \\
(\exists x\, F)[\sigma]^+_{0.\tau} &= (F[x/x\sigma])[\sigma]^+_\tau & (F \equiv G)[\sigma]^+_\tau &= F \equiv G \\
(\exists y\, F)[\sigma]^+_{0.\tau} &= \exists y\, F[\sigma]^+_\tau & (\neg F)[\sigma]^+_{0.\tau} &= \neg F[\sigma]^-_\tau \\
(\forall z\, F)[\sigma]^+_{0.\tau} &= \forall z\, F[\sigma]^+_\tau & F[\sigma]^+_\varepsilon &= F
\end{array}
$$

$$
\begin{array}{rlrl}
(F \supset G)[\sigma]^-_{0.\tau} &= F[\sigma]^+_\tau \supset G & (F \supset G)[\sigma]^-_{1.\tau} &= F \supset G[\sigma]^-_\tau \\
(F \vee G)[\sigma]^-_{0.\tau} &= F[\sigma]^-_\tau \vee G & (F \vee G)[\sigma]^-_{1.\tau} &= F \vee G[\sigma]^-_\tau \\
(F \wedge G)[\sigma]^-_{0.\tau} &= F[\sigma]^-_\tau \wedge \top & (F \wedge G)[\sigma]^-_{1.\tau} &= F \wedge G[\sigma]^-_\tau \\
(\forall x\, F)[\sigma]^-_{0.\tau} &= (F[x/x\sigma])[\sigma]^-_\tau & (F \equiv G)[\sigma]^-_\tau &= F \equiv G \\
(\forall y\, F)[\sigma]^-_{0.\tau} &= \forall y\, F[\sigma]^-_\tau & (\neg F)[\sigma]^-_{0.\tau} &= \neg F[\sigma]^+_\tau \\
(\exists z\, F)[\sigma]^-_{0.\tau} &= \exists z\, F[\sigma]^-_\tau & F[\sigma]^-_\varepsilon &= F
\end{array}
$$

where $x\sigma \neq x$ and $y\sigma = y$. For a position $\pi \in \Pi_{\mathbf{t}}(H)$, we define $H[\sigma]^+_\pi = H[\sigma]^+_{\widehat{\pi}}$ and $H[\sigma]^-_\pi = H[\sigma]^-_{\widehat{\pi}}$, where $\widehat{\pi}$ is the longest prefix of π in $\Pi_{\mathbf{F}}(H)$.

These operations keep track of polarity of an occurrence in question and do not instantiate inappropriate variables. Also they eliminate subformulas in certain adjacent positions — exactly those ones which may lose their local properties after instantiation.

Lemma 4. *Let H be a formula such that no quantifier occurs in H in the scope of another quantifier over the same variable. Let π be a position in $\Pi(H)$ and σ, a substitution. Then we have:*

$$\vdash H[\sigma]^+_\pi \supset H \qquad\qquad \vdash H \supset H[\sigma]^-_\pi$$

Theorem 3. *Let H be a formula such that no quantifier occurs in H in the scope of another quantifier over the same variable. Let π be a position in $\Pi(H)$ and σ, a substitution. For any polarity $s \in \{+,-\}$ and any position $\tau \in \Pi_{\mathbf{A}}(H[\sigma]^s_\pi)$, either $(H[\sigma]^s_\pi)|_\tau = \top$ or there exists a position $\tau' \in \Pi_{\mathbf{A}}(H)$ such that the following holds:*

Let μ be the longest common prefix of π and τ'. Let σ' be a substitution such that for any varaible x, if a quantifier over x is eliminated in $H[\sigma]^s_\mu$, then $x\sigma' = x\sigma$, otherwise $x\sigma' = x$. Then $(H[\sigma]^s_\pi)|_\tau = (H|_{\tau'})\sigma'$ and

$$\vdash \langle\!|U|\!\rangle^H_{\tau'} \supset \langle\!|U\sigma'|\!\rangle^{H[\sigma]^s_\pi}_\tau$$

Proof. We can suppose without loss of generality that $\pi \in \Pi_{\mathbf{F}}(H)$ (otherwise $\widehat{\pi}$ should be taken instead of π). We will prove this lemma by induction on

the length of π. In the base case ($\pi = \epsilon$), we take $\tau' = \tau$ and $\sigma' = \iota$, the trivial substitution. Thus the claim is obviously true. Otherwise we consider three typical cases.

Case $H = (F \supset G)$, $\pi = 0.\pi_0$, $s = -$, $H[\sigma]^s_\pi = F[\sigma]^+_{\pi_0} \supset G$, $\tau = 1.\tau_0$. We take $\tau' = \tau$ and $\sigma' = \iota$. Obviously, $(H[\sigma]^-_\pi)|_\tau = G|_{\tau_0} = (H|_{\tau'})\sigma'$. Furthermore, $\langle\!|U|\!\rangle^H_{\tau'} = F \supset \langle\!|U|\!\rangle^G_{\tau_0}$ and $\langle\!|U\sigma'|\!\rangle^{H[\sigma]^s_\pi}_\tau = F[\sigma]^+_{\pi_0} \supset \langle\!|U|\!\rangle^G_{\tau_0}$. By Lemma 4, $\vdash F[\sigma]^+_{\pi_0} \supset F$, and the claim holds. Note that we could not make the final step in the case $s = +$, and therefore we had to define $H[\sigma]^+_\pi = F[\sigma]^-_{\pi_0} \supset \bot$.

Case $H = (F \supset G)$, $\pi = 1.\pi_0$, $s = +$, $H[\sigma]^s_\pi = F \supset G[\sigma]^+_{\pi_0}$, $\tau = 1.\tau_0$. By the induction hypothesis, there exist $\tau'_0 \in \Pi_{\mathbf{A}}(G)$ and a substitution σ' such that $(G[\sigma]^+_{\pi_0})|_{\tau_0} = (G|_{\tau'_0})\sigma'$ and $\vdash \langle\!|U|\!\rangle^G_{\tau'_0} \supset \langle\!|U\sigma'|\!\rangle^{G[\sigma]^+_{\pi_0}}_{\tau_0}$. Then we take $\tau' = 1.\tau'_0$ and obtain $(H[\sigma]^+_\pi)|_\tau = (H|_{\tau'})\sigma'$. Moreover, $\langle\!|U|\!\rangle^H_{\tau'}$ (equal to $F \supset \langle\!|U|\!\rangle^G_{\tau'_0}$) implies $\langle\!|U\sigma'|\!\rangle^{H[\sigma]^+_\pi}_\tau$ (equal to $F \supset \langle\!|U|\!\rangle^{G[\sigma]^+_{\pi_0}}_{\tau_0}$).

Case $H = (\forall x\, F)$, $\pi = 0.\pi_0$, $s = -$, $H[\sigma]^s_\pi = (F[x/x\sigma])[\sigma]^-_{\pi_0}$, $\tau = \tau_0$. Let F' stand for $F[x/x\sigma]$. By the induction hypothesis, there exist some $\tau'_0 \in \Pi_{\mathbf{A}}(F')$ and a substitution σ'_0 such that $(F'[\sigma]^-_{\pi_0})|_{\tau_0} = (F'|_{\tau'_0})\sigma'_0$ and for any V, $\vdash \langle\!|V|\!\rangle^{F'}_{\tau'_0} \supset \langle\!|V\sigma'_0|\!\rangle^{F'[\sigma]^-_{\pi_0}}_{\tau_0}$. Then we take $\tau' = 0.\tau'_0$ and $\sigma' = \sigma'_0 \circ [x/x\sigma]$ (recall that σ'_0 does not instantiate variables from $x\sigma$). We obtain $(H[\sigma]^-_\pi)|_\tau = (F'[\sigma]^-_{\pi_0})|_{\tau_0} = (F'|_{\tau'_0})\sigma'_0 = (F|_{\tau'_0})\sigma' = (H|_{\tau'})\sigma'$. Further, the local image $\langle\!|U|\!\rangle^H_{\tau'}$ (equal to $\forall x\, \langle\!|U|\!\rangle^F_{\tau'_0}$) implies $(\langle\!|U|\!\rangle^F_{\tau'_0})[x/x\sigma]$. The latter formula is equal to $\langle\!|U[x/x\sigma]|\!\rangle^{F'}_{\tau'_0}$ and thus implies $\langle\!|(U[x/x\sigma])\sigma'_0|\!\rangle^{F'[\sigma]^-_{\pi_0}}_{\tau_0}$, that is, $\langle\!|U\sigma'|\!\rangle^{H[\sigma]^-_\pi}_\tau$. □

Informally, Theorem 3 says that any atom in H that "survives" instantiation (i.e. is not replaced with a boolean constant) preserves its local properties, which are instantiated together with the atom.

5 Applying local properties

Let us consider a formula of the form $H[F]_\pi$ such that no quantifier occurs in it in the scope of another quantifier over the same variable. Let σ be a substitution. By Theorem 3, there exist a formula H', a position π', and a substitution σ' such that $(H[F]_\pi)[\sigma]^-_\pi \equiv H'[F\sigma']_{\pi'}$ and every local property of F in H is preserved (modulo instantiation) in H'. (While π is not a position of atom in $H[F]_\pi$, we can take an atom $P(\boldsymbol{x})$, where P is a new predicate symbol and $\boldsymbol{x}$ are the free variables of F, and prove $(H[P(\boldsymbol{x})]_\pi)[\sigma]^-_\pi \equiv H'[P(\boldsymbol{x})\sigma']_{\pi'}$. Note that $P(\boldsymbol{x})$ cannot turn into a boolean constant in $(H[P(\boldsymbol{x})]_\pi)[\sigma]^-_\pi$. Then we have $\forall \boldsymbol{x}\,(P(\boldsymbol{x}) \equiv F) \vdash (H[F]_\pi)[\sigma]^-_\pi = H'[F\sigma']_{\pi'}$, by Lemma 1 and Theorem 1. Since P is a new symbol, the premise $\forall \boldsymbol{x}\,(P(\boldsymbol{x}) \equiv F)$ can be discarded.) By Lemma 4, $H[F]_\pi$ implies $H'[F\sigma']_{\pi'}$.

We can prove that $H'[F\sigma']_{\pi'}$ implies $\exists \boldsymbol{x}'\,(F\sigma') \vee H'[\bot]_{\pi'}$, where $\boldsymbol{x}'$ are the free variables of $F\sigma'$. Indeed, $H'[F\sigma']_{\pi'}$ implies $\exists \boldsymbol{x}'\,(F\sigma') \vee H'[F\sigma']_{\pi'}$, which is equivalent to $\forall \boldsymbol{x}'\,(\neg F\sigma') \supset H'[F\sigma']_{\pi'}$, which is equivalent to $\forall \boldsymbol{x}'\,(\neg F\sigma') \supset H'[\bot]_{\pi'}$ by Theorem 1. Therefore, $H[F]_\pi$ implies $\neg H'[\bot]_{\pi'} \supset \exists \boldsymbol{x}'\,(F\sigma')$.

This provides us with a handy tool to test applicability of definitions in a ForTheL text. Consider a section $\mathbb{A}$ and suppose that Γ is the set of sections which logically precede $\mathbb{A}$ in the text. Let G be the formula image of $\mathbb{A}$. Let $P(\boldsymbol{s})$ occur in G in a position μ. Now, suppose that $\mathbb{D} \in \Gamma$ is a definition for the predicate symbol P. Quite naturally, the formula image of $\mathbb{D}$ is of the form $\forall \boldsymbol{x}_1(H_1 \supset \ldots \forall \boldsymbol{x}_k(H_k \supset (P(\boldsymbol{x}_{1,\ldots,k}) \equiv D)) \ldots)$. By previous, it suffices to prove $\Gamma \vdash \langle\!|H_1\sigma \supset \ldots H_k\sigma \supset \bot|\!\rangle_\mu^G$, where σ is the substitution $[\boldsymbol{x}_{1,\ldots,k}/\boldsymbol{s}]$, to obtain $\Gamma \vdash \langle\!|P(\boldsymbol{s}) \equiv D\sigma|\!\rangle_\mu^G$. Then G is equivalent to $G[D\sigma]_\mu$, that is, we can *apply* the definition $\mathbb{D}$ to $P(\boldsymbol{s})$. Moreover, all the local properties of terms and subformulas of D in $\mathbb{D}$, instantiated with σ, hold in $D\sigma$ in $G[D\sigma]_\mu$.

In a similar fashion, we define applicability for other forms of ForTheL definitions and signature extensions. Note that the substitution σ and the position of the local instantiation in $|\mathbb{D}|$ are unambiguously determined by the form of $\mathbb{D}$. Using the method described above, we can test any logical predecessor of $\mathbb{A}$ for applicability at a given position in $|\mathbb{A}|$, but then we have to choose an appropriate local instantiation ourselves.

Now, a section $\mathbb{A}$ is *ontologically correct* in view of Γ if and only if every occurrence of a non-logical symbol in $|\mathbb{A}|$ either has an applicable definition or signature extension in Γ or is the principal occurrence in a definition or signature extension $\mathbb{A}$ (which means that $\mathbb{A}$ introduces that very symbol).

A ForTheL text is *ontologically correct* whenever each section in it is ontologically correct in view of its logical predecessors.

6 Conclusion

We have introduced the notion of a locally valid statement for the classical first-order logic and showed how it can be used to reason about the interiors of a formula. In particular, we proved that a locally true equivalence is a sufficient condition for an equivalent transformation of a subformula. The local validity of a statement is expressed with the help of *local images* which can be regarded as a syntactical formalization of the notion of a logical context of the statement occurrence. Since locally equivalent transformations may break local properties of other occurrences, we introduced the notion of directed local validity which is invariant w.r.t. directed locally equivalent transformations. Finally, we defined the operation of local instantiation and showed that this transformation preserves directed local properties. Using this theoretical background, we gave a clear definition of an ontologically correct ForTheL text.

The proposed approach can be regarded as a way to handle partial relations and functions in a mathematical text. Instead of introducing special individual or truth values for undefinedness (as in Kleene's strong logic [6]), ontological correctness requires every term or atom to be well-defined *a priori*, by conformance to the guards of corresponding definitions. Using directed images and deductive techniques preserving local properties, we can guarantee that the text under consideration always stays well-defined. In our opinion, this corresponds well to the usual mathematical practice.

Of course, reasoning inside a formula is not a new idea. To our knowledge, related concepts were first introduced by L.G. Monk in [7] and were further developed in [8]. P.J. Robinson and J. Staples proposed a full-fledged inference system (so called "window inference") [9] which operated on subexpressions taking the surrounding context into account. This inference system has been generalized and extended by J. Grundy [10].

A common trait of the mentioned approaches is that the local context of an occurrence is represented by a set of formulas which are regarded as local premises for the position in question. Special inference rules are then needed to handle a local context and, what is worse, some "strong" transformations, e.g. replacing $A \vee B$ with $\neg A \supset B$, are required. The notion of local image, as described in this paper, seems to be lighter and less intrusive. In particular, the results of Section 4 are valid in intuitionistic logic, while the local contexts of [7] cannot be adapted for intuitionistic logic in any obvious way.

Moreover, the definition of a local image can be easily extended to a (uni)-modal language: $\langle U \rangle_{0.\pi}^{\Box F} = \Box \langle U \rangle_{\pi}^{F}$ and $\langle U \rangle_{0.\pi}^{\Diamond F} = \Box \langle U \rangle_{\pi}^{F}$, and similarly for directed images. Then the statements of Section 4 (local instantiation aside) can be proved in the modal logic **K**, hence in any normal modal logic.

Acknowledgements. This work is supported by the INTAS project 05-1000008-8144. Some parts were done within the scope of the project M/108-2007 in the framework of the joint French-Ukrainian programme "Egide-Dnipro".

References

1. Trybulec, A., Blair, H.: Computer assisted reasoning with Mizar. In: Proc. 9th International Joint Conference on Artificial Intelligence. (1985) 26–28
2. Barendregt, H.: Towards an interactive mathematical proof language. In Kamareddine, F., ed.: Thirty Five Years of Automating Mathematics, Heriot-Watt University, Edinburgh, Scotland, Kluwer Academic Publishers (2003) 25–36
3. Kamareddine, F., Nederpelt, R.P.: A Refinement of de Bruijn's Formal Language of Mathematics. Journal of Logic, Language and Information **13**(3) (2004) 287–340
4. Lyaletski, A., Paskevich, A., Verchinine, K.: SAD as a mathematical assistant — how should we go from here to there? Journal of Applied Logic **4**(4) (2006) 560–591
5. Lyaletski, A., Paskevich, A., Verchinine, K.: Theorem proving and proof verification in the system SAD. In Asperti, A., Bancerek, G., Trybulec, A., eds.: Mathematical Knowledge Management: Third International Conference, MKM 2004. Volume 3119 of Lecture Notes in Computer Science., Springer (2004) 236–250
6. Kleene, S.C.: Introduction to Metamathematics. Van Nostrand (1952)
7. Monk, L.G.: Inference rules using local contexts. Journal of Automated Reasoning **4**(4) (1988) 445–462
8. Corella, F.: What holds in a context? Journal of Automated Reasoning **10**(2) (1993) 79–93
9. Robinson, P.J., Staples, J.: Formalising the hierarchical structure of practical mathematical reasoning. Journal of Logic and Computation **3**(1) (1993) 47–61
10. Grundy, J.: Transformational hierarchical reasoning. The Computer Journal **39**(4) (1996) 291–302