Theorem Proving with Definitions

Fausto Giunchiglia* and Toby Walsh†
Department of Artificial Intelligence
University of Edinburgh

Keywords: Definition unfolding, abstract theorem proving, planning.

Abstract

This paper analyses a technique (called **Gazing**) for unfolding definitions on the basis of a global plan built in an abstract space. Gazing's logical properties are studied inside a formal framework which relies on a more general theory of abstraction. Some experimental results confirming the theoretical ones are also presented.

1 Introduction

The use of definitions is listed by Larry Wos as the 30th of the 33 basic research questions facing automated reasoning, the solution of which would "... mark one of the more significant advances in the field of automated reasoning ..." [Wos88]. Definitions are much more than syntactic sugar; used well, they represent meaningful concepts that determine the nature of the theory. Of course, their use enlarges the search space by increasing the branching rate. However, they allow the construction of shorter and more structured proofs. Just as we wouldn't expect a mathematician to prove a difficult theorem from "first principles", we shouldn't expect our theorem provers.

Most previous attempts at using definitions in theorem proving can be criticised as *local* strategies. See, for instance, [Ern73, Pas78], or (more closely related to this work) the *peeking* heuristic used in the UT theorem prover [BT75]. Peeking unfolds a predicate occurring in the hypothesis of a sequent with its definition only if this introduces a predicate name mentioned in the conclusion. The underlying assumption is that to complete the proof we want the hypothesis and conclusion to be about the same concepts. Peeking is a local strategy since it only looks one unfolding ahead. For example, given the definitions $p \leftrightarrow q$ and $q \leftrightarrow r$, peeking fails to suggest unfolding p by its definition in the proof of $p \vdash r$.

Gazing extends peeking by producing a global plan; this method significantly out performs the above local strategies. Gazing was originally implemented inside a rational reconstruction of the UT prover [Plu87]; variants of it have also been developed for a connection method prover (MT [War87]), and a proof development system based on Martin-Löf type theory [Sim88]. The research described in this paper further extends the above work, rationally reconstructing gazing[1], making it more powerful (more theorems are proved), modifying the way functions are dealt with and, most importantly, giving it a sound logical foundation. From now on, we will refer to the rationally reconstructed gazing simply as "gazing", and to the original gazing [Plu87] as "old gazing".

The paper is structured as follows. We first define what we mean by definitions (section 2), and describe old gazing (section 3). In section 4 we present an informal account of our reconstruction of gazing. We then develop a theory of abstraction with which to formalise gazing and prove some of the properties it possesses (section 5). In section 6 we briefly discuss how gazing can be extended to take function symbols into account. Finally the results of an implementation of gazing (section 7) and our conclusions (section 8) are reported.

2 Definition of definitions

Following Suppes [Sup57], we will consider definitions consisting of an expression being defined related by

*Supported by SERC grant GR/E/4459.8. Current address is IRST, Loc. Panté di Povo, I 38100 Trento, Italy.
†Supported by a SERC studentship.

[1] That is, our reconstruction is faithful to the ideas though not the details or actual implementation of the original.

Definition 1 : $a \subseteq b \leftrightarrow \forall\, x\, .\ (\, x \in a \rightarrow x \in b\ )$

Definition 2 : $a =_{set} b \leftrightarrow \forall\, x\, .\ (\, x \in a \leftrightarrow x \in b\ )$

Definition 3 : $a \subset b \leftrightarrow a \subseteq b \wedge \neg(\ a =_{set} b)$

Definition 4 : $a \in 2^{\mathrm{b}} \leftrightarrow a \subseteq b$

Definition 5 : $a \in \emptyset \ \leftrightarrow \bot$

Figure 1: Set theory definitions from [Plu87]

an equivalence to the defining expression. Predicate definitions are of the form

$$\forall x_1, ..., x_n\ p(x_1, ..., x_n) \leftrightarrow Q(x_1, ...x_n)$$

and function definitions of the form

$$\forall x_1, ..x_n\ r(f(x_1, ..x_n), z) \leftrightarrow S(x_1, ..x_n, z)$$

where Q and S contain no new symbols, r is not a new symbol (it may be "="), $x_1, ..., x_n$ and z (when it exists) are the only free variables of Q, S, r and p, and $\vdash\ \forall x_1, ..x_n\ \exists!\ z\ S(x_1, ..x_n, z)$. If Σ' is the formal system created by adding a new definition to Σ, then the following two facts hold:

Criterion 1 (eliminability) *if α' is any well formed formula (wff from now on) of Σ' then there exists a wff α of Σ such that $\vdash_{\Sigma'} \alpha' \leftrightarrow \alpha$.*

Criterion 2 (non-creativity) *if α is any wff of Σ such that $\vdash_{\Sigma'} \alpha$ then $\vdash_{\Sigma} \alpha$.*

A consequence of this non-creativity is that definitions cannot introduce inconsistency into a theory. Examples of predicate and function definitions (which will be used throughout the paper) are given in figure 1.

3 An informal account of old gazing

The idea underlying old gazing (which generalises the peeking heuristic) is to unfold only the definitions necessary to find a common language of concepts[2] between the hypotheses and conclusion. We describe this as the process of building the *common currency*. For example, to construct a proof about set equality ("$=_{set}$") and subset ("$\subseteq$") we need to unfold their definitions into the common currency of set membership, ("$\in$"). Old gazing finds this common currency by planning ahead in a hierarchy of abstraction spaces: the predicate space and the function/polarity space.

In the predicate space, the hypotheses and conclusion are abstracted to give the set of predicate names they contain, whilst the definitions of the theory are similarly abstracted to give directed rewrite rules; the direction of the rewrite rules ensure that predicates are only unfolded in terms of more "primitive" predicates (that is, predicates defined earlier in the theory).

For example, given the predicate definitions in figure 1, and the following theorem to prove:

$$a =_{set} b\ \vdash\ a \subseteq b$$

old gazing abstracts this problem and the definitions in figure 1 to the problem[3]:

$$\{=_{set}\}\ \vdash\ \{\subseteq\}$$

and the rewrite rules:

$$\{\subseteq\} \Rightarrow \{\in\}$$
$$\{=_{set}\} \Rightarrow \{\in\}$$
$$\{\subset\} \Rightarrow \{\subseteq, =_{set}\}$$

For every predicate name in the conclusion set, we try to find a common rewriting of this and predicate name(s) in the hypotheses set. In this case, we just unfold both $=_{set}$ and $\subseteq$ in terms of $\in$.

The problem with this abstraction is that there is no guarantee that we will be able to find an abstract solution for every theorem (for example, $\vdash p \vee \neg p$ but $\nvdash \{p\}$), and that there will be an abstract solution which unfolds all the definitions necessary to complete the proof (for example, given the definitions $p \leftrightarrow (q \rightarrow q)$ and $p \leftrightarrow (r \wedge s)$, old gazing will not suggest the right unfolding of p in the proof of $r \vee s \vdash p$ because the connective structure is ignored).

Finally, in the function/polarity space, old gazing checked the plan of rewritings to see if the predicates

[2] Concept is used here to describe both predicates and functions.

[3] In this informal introduction, we use the same provability symbol $\vdash$ for the abstract and unabstract problem; later on, we will very clearly distinguish between the two uses.

predicates guarantees that they appear equivalently negated in hypothesis and conclusion. The plan is also checked to see if the predicates mention the same function symbols; preference is given to plans that keep the predicate symbols the same but, as a last resort, the predicate symbols may have to be rewritten.

4 Rationally reconstructed gazing

Our reconstruction of gazing generalises old gazing by keeping the connective structure of the formulae. The abstract space is constructed by rewriting both the theorem to prove and the definitions available with the same abstraction. The language used in the abstract space is propositional; gazing thereby shifts problem solving from an undecidable first order theory into a decidable propositional theory. The abstraction keeps the predicate symbols and the connective structure but throws away the quantifiers and the predicate arguments. For example, given the same theorem to prove:

$$\vdash a =_{set} b \rightarrow a \subseteq b$$

gazing abstracts it to the problem:

$$\vdash =_{set} \rightarrow \subseteq$$

given the abstracted predicate definitions of figure 1:

$$\begin{array}{c} \subseteq \leftrightarrow \ (\in \rightarrow \in) \\ =_{set} \leftrightarrow \ (\in \leftrightarrow \in) \\ \subset \leftrightarrow \ (\subseteq \wedge \neg =_{set}) \end{array}$$

The above theorem holds in the abstract space (if we simply just unfold $\subseteq$); however, it is only *the longer proof of the abstracted formula, involving unfolding the definition of* $\subseteq$ *and* $=_{set}$*, that guides the proof of the unabstracted formula.* The unabstracted theorem has a proof which uses the same definitions and which, though significantly more complicated, has a very similar shape (see figures 2 and 3). Indeed, the proof in figure 3 has been obtained from the proof in figure 2 just by applying quantifier inference rules which were unnecessary in the abstract space. This is a general phenomenon: the patching up which must

[4]The polarity of a formula is +(-) if it appears within the scope of an even(odd) number of negation signs. As $p \rightarrow q$ is equivalent to $\neg p \vee q$, p appears *implicitly* negated in an implication; similarly in $p \vdash q$.

$$\dfrac{\dfrac{\dfrac{=_{set} \qquad =_{set} \leftrightarrow (\in \leftrightarrow \in)}{\in \leftrightarrow \in}}{\in \rightarrow \in} \qquad \subseteq \leftrightarrow (\in \rightarrow \in)}{\subseteq}$$
$$=_{set} \rightarrow \subseteq$$

Figure 2: Proof of the abstract theorem

be performed on the abstract proof *fixes the details which have been left out of the abstract space.*

Nothing is, of course, free and the abstraction loses some information. In this case, the major consequence is that the existence of a proof in the abstract space does *not* guarantee the existence of a proof of the original goal. For instance the wff $a =_{set} c \rightarrow a \subseteq b$ has the same abstraction as the wff above, its abstraction is a theorem in the abstract space but it is not a theorem in the original space. However, as we prove in the following section, in our reconstruction of gazing, if a wff is a theorem in the original space then its abstraction is a theorem in the abstract space.

5 A formal account of gazing

In order to give a formal account of gazing we have developed a general theory of abstraction. This framework seems very powerful and has allowed us to formalise and analyse all the (informally described) work in "abstraction" of which we are aware (for example, GPS, ABSTRIPS, Plaisted's work ...); a full description of this framework is given in [GW90]. In this paper, however, only the details needed to analyse gazing are given. In particular the theorems not directly concerning gazing and its extensions will be given but not proved. The final goal of the exercise is:

- to prove that there is a plan for every theorem and that
- the plan unfolds **all** the definitions necessary to complete the proof.

These two properties were not possessed by old gazing. We begin be defining what we mean by a formal system.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a =_{set} b \ \leftrightarrow \ \forall x.x \in a \leftrightarrow x \in b}}{\forall x.x \in a \leftrightarrow x \in b}}{h \in a \leftrightarrow h \in b}}{h \in a \rightarrow h \in b}}{\forall x.x \in a \rightarrow x \in b}}{\forall x.x \in a \leftrightarrow x \in b} \qquad \dfrac{}{a \subseteq b \ \leftrightarrow \ \forall x.x \in a \rightarrow x \in b}}{a \subseteq b}}{a =_{set} b \rightarrow a \subseteq b}$$

Figure 3: Proof of the unabstracted theorem

Definition 6 (Formal system) *: A* **formal system** Σ *is an ordered pair* (Λ, Δ)*, where* Λ *is the* **Language** *and* Δ *is the* **Deductive Machinery** *of* Σ*.*

The language Λ is composed of an alphabet, the set of (well formed) terms and the set of well formed formulae (wffs from now on). The deductive machinery is composed of a set of inference rules and a, possibly empty, set of axioms Ω. Ω is a subset of the wffs of Λ. Examples of formal systems are Natural deduction systems, Sequent calculus systems, and Hilbert-style systems. The languages we consider are languages of first order logic formulated in the usual way. We give the proofs inside a natural deduction deductive machinery and use standard natural deduction conventions [5], but the results could be equally well generalised to other formal systems [GW90].

Definition 7 (Abstraction) *: If* $\Sigma_1 = (\Lambda_{\Sigma_1}, \Delta_{\Sigma_1})$ *and* $\Sigma_2 = (\Lambda_{\Sigma_2}, \Delta_{\Sigma_2})$ *are two formal systems, an* **abstraction** f*, written as* $f : \Sigma_1 \mapsto \Sigma_2$*, is an ordered pair of total functions* (f_Λ, f_Δ) *such that:*

$$f_\Lambda : \Lambda_{\Sigma_1} \mapsto \Lambda_{\Sigma_2}$$
$$f_\Delta : \Delta_{\Sigma_1} \mapsto \Delta_{\Sigma_2}.$$

When no confusion arises we drop the subfixes. Gazing can be formally defined as $f_{gaze} : \Sigma_1 \mapsto \Sigma_2$ where:

Σ_1: A first order calculus, defined as follows:

Λ_{Σ_1}: first order language;

Δ_{Σ_1}: natural deduction rules of inference plus axioms defining a theory (*eg.* set theory).

[5] The reader is referred to [Pra65] for a formal definition of the concepts assumed in this paper.

Σ_2: A formal system, defined as follows:

Λ_{Σ_2}: propositional language containing denumerably many (new) constants. As it will be seen later, $\Lambda_{\Sigma_1} \cap \Lambda_{\Sigma_2} = \emptyset$.

Δ_{Σ_2}: any complete propositional decision procedure with axioms formed by applying f_{gaze_Λ} to all the axioms of Σ_1.

f_{gaze_Δ} is not explicitly defined since we rely on the various completeness results for first order and propositional calculus. Any Δ_{Σ_2} is acceptable provided the axioms are mapped appropriately and we are guaranteed completeness.

$f_{gaze_\Lambda}(\varphi_{\Sigma_1}) \in \Lambda_{\Sigma_2}$ is defined as follows (α, β are two wffs in Λ_{Σ_1}, and P_l is a propositional constant):

Definition 8 *:*
1. $f_{gaze}(\alpha) = P_k$, where α is an atomic formula. Occurrences of atomic formulae with the same predicate symbol are rewritten to occurrences of the same propositional constant*;*
2. $f_{gaze}(\exists x.\alpha) = f_{gaze}(\alpha)$*;*
3. $f_{gaze}(\forall y.\alpha) = f_{gaze}(\alpha)$*;*
4. $f_{gaze}(\alpha \wedge \beta) = f_{gaze}(\alpha) \wedge f_{gaze}(\beta)$*;*
5. $f_{gaze}(\alpha \vee \beta) = f_{gaze}(\alpha) \vee f_{gaze}(\beta)$*;*
6. $f_{gaze}(\neg\alpha) = \neg f_{gaze}(\alpha)$*;*
7. $f_{gaze}(\alpha \rightarrow \beta) = f_{gaze}(\alpha) \rightarrow f_{gaze}(\beta)$*;*
8. $f_{gaze}(\alpha \leftrightarrow \beta) = f_{gaze}(\alpha) \leftrightarrow f_{gaze}(\beta)$.

For example, $f_{gaze}(\forall x.(x \in a \ \wedge \ a \subseteq b) \rightarrow x \in b) = ((\in \ \wedge \ \subseteq) \rightarrow \in)$

The notion of abstraction given in definition 7 is very general. Our next step is to characterise the various forms of abstractions. The main idea underlying the use of abstractions is to switch from one formal system Σ_1 to a new formal system Σ_2 which preserves certain desirable properties and is simpler to handle. The notion of simplicity depends on the application; for example, we may require that we map from a semi-decidable theory (*eg.* first order predicate logic) to a decidable theory (*eg.* propositional logic). The desirable property we consider preserving is provability; for example:

Definition 9 *: An Abstraction* $f : \Sigma_1 \mapsto \Sigma_2$ *is said to be* **truthful** *iff, for any wff* φ_{Σ_1}*, if* $\vdash_{\Sigma_1} \varphi_{\Sigma_1}$ *then* $\vdash_{\Sigma_2} f(\varphi_{\Sigma_1})$.

Many of the abstractions defined in the past are not truthful (see [GW90] for a complete discussion). Usually (for example, in old gazing) there is at least one wff which is a theorem in the original space whose abstraction is **not** a theorem in the abstract space. The authors claim that truthfulness is the one property you want your abstraction mapping to have. Note that one could also require the opposite property, namely that if a wff is a theorem in the abstract space then its unabstracted version is a theorem in the original space. The union of two requirements is very strong and amounts to require that a wff is a theorem in the original space iff its abstracted version is a theorem in the abstract space. In this case it is very difficult to satisfy the simplicity requirement; for example. it is impossible for Σ_1 to be semidecidable and Σ_2 decidable [GW90].

The first important result is that gazing is truthful.

Theorem 1 *f_{gaze} is truthful.*

Proof: By proving that, given a deduction tree Π_{Σ_1} of $\vdash_{\Sigma_1} \varphi_{\Sigma_1}$ in Σ_1, we can build a deduction tree Π_{Σ_2} of $\vdash_{\Sigma_2} f_{gaze}(\varphi_{\Sigma_1})$ in Σ_2.

The proof proceeds by induction on the weight N of Π_{Σ_1}[6].

For $N = 1$, we can have only an axiom φ_{Σ_1} and Π_{Σ_2} in this case is simply $f_{gaze}(\varphi_{\Sigma_1})$ which is an axiom of the abstract theory.

Let's suppose we have already rewritten a deduction tree Π_{Σ_1} of weight N into a new deduction tree Π_{Σ_2}. We show how we can build a new deduction tree Π'_{Σ_2} whichever inference rule is applied to obtain the deduction tree Π'_{Σ_1} of weight $(N+1)$ from Π_{Σ_1}.

In the applications of all the rules which are not quantifier rules, Π_{Σ_1} translates unmodified into Π_{Σ_2} in the sense that, for example, an $\wedge$I on α in Π_{Σ_1} gets translated into an $\wedge$I on $f_{gaze}(\alpha)$ in Π_{Σ_2}. Note that, any time a new formula β is introduced into Π_{Σ_1} (which happens with assumptions, and may happen with $\vee$I), a (not necessarily) new formula $f_{gaze}(\beta)$ is introduced into Π_{Σ_2}. We thus consider only applications of the quantifier rules. There are four cases to consider: $\forall$I, $\forall$E, $\exists$I and $\exists$E (during the proof we will write $f(\Pi_1)$ to represent the translation from Σ_1 into Σ_2 of a deduction tree Π_1 of weight less than N).

[6]The *weight* of a deduction tree is the number of formula occurrences in that tree.

$$\forall\text{I}\quad \frac{\dfrac{\Pi_1}{p(y)}}{\forall x.p(x)} \quad\Longrightarrow\quad f\left(\frac{\Pi_1}{p(y)}\right)$$

$$\forall\text{E}\quad \frac{\dfrac{\Pi_1}{\forall x.p(x)}}{p(y)} \quad\Longrightarrow\quad f\left(\frac{\Pi_1}{\forall x.p(x)}\right)$$

$$\exists\text{I}\quad \frac{\dfrac{\Pi_1}{p(y)}}{\exists x.p(x)} \quad\Longrightarrow\quad f\left(\frac{\Pi_1}{p(y)}\right)$$

$$\exists\text{E}\quad \frac{\dfrac{\Pi_1}{\exists x.p(x)} \quad \dfrac{[p(y)]}{\dfrac{\Pi_2}{Q}}}{Q} \quad\Longrightarrow\quad f\left(\begin{array}{c}\left[\frac{\Pi_1}{p(y)}\right]\\ \frac{\Pi_2}{Q}\end{array}\right)$$

□

It is interesting to make some observations about the above result.

As Π_{Σ_2} does not contain any quantifier rule applications (which must be so since in Δ_2 we only have the propositional connective inference rules), it is a smaller proof that Π_{Σ_1}. Indeed, note that all the proof transformation steps are guaranteed to reduce the size of the proof. If M is weight of Π_{Σ_1} then the weight of Π_{Σ_2} is $M - N_{\forall\text{I}} - N_{\forall\text{E}} - N_{\exists\text{I}} - 2 * N_{\exists\text{E}}$ where $N_{\forall\text{I}}$ is the number of applications of $\forall$I, etc.

Further, if we look at the proof that gazing is truthful, we notice that the abstract proof and the unabstracted proof only differ in the application of the quantifier rules. *The abstract proof therefore contains the same unfolding of definitions as needed to complete the unabstracted proof.* The plan of definitions to unfold extracted from the abstract proof is hence guaranteed to succeed in the proof of the unabstracted wff[7]. We cannot tell in the abstract space what the right sequence of unfoldings is but at least we know that at least one exists.

Note that this fact, together with the proof that gazing is truthful, confirms the informal discussion of the previous section and guarantees that we have achieved all the goals we set ourselves in the beginning

[7]Which *does not mean* that the resulting wff is a theorem of the original space.

cability of gazing in a formal system where Δ_1 and Δ_2 work by refutation (*eg.* Δ_1 and Δ_2 are resolution-based).

Definition 10 *: An abstraction $f : \Sigma_1 \mapsto \Sigma_2$ is said to be* **falseful** *iff, for any wff φ_{Σ_1} , if $\vdash_{\Sigma_1} \neg\varphi_{\Sigma_1}$ then $\vdash_{\Sigma_2} \neg f(\varphi_{\Sigma_1})$.*

In proofs by refutation, falseful abstractions play the same rôle that truthful abstractions play in proofs respecting validity. For instance, in resolution, given the wff α to prove, we negate α, add it to the set of axioms and try to prove that the resulting theory is inconsistent (by showing that $\perp$ is a valid consequence of the theory). In such proofs, the abstraction mapping must be classified by the way inconsistent theories are mapped into inconsistent theories. The following two theorem hold:

Theorem 2 *An abstraction $f : \Sigma_1 \mapsto \Sigma_2$ is falseful iff, for any wff φ_{Σ_1}, if adding φ_{Σ_1} to the axioms of Σ_1 yields an inconsistent formal system then adding $f(\varphi_{\Sigma_1})$ to the axioms of Σ_2 yields an inconsistent formal system.*

Theorem 3 *: An abstraction mapping $f : \Sigma_1 \mapsto \Sigma_2$ that preserves negation (ie. $f(\neg\varphi_{\Sigma_1}) = \neg f(\varphi_{\Sigma_1})$) is a truthful abstraction iff it is a falseful abstraction.*

Thus as a simple corollary to theorem 1, gazing is also falseful; this guarantees that there is an abstract proof for every theorem or non-theorem.

Corollary 1 *: f_{gaze} is falseful.*

6 Gazing with functions

Our reconstruction of gazing has so far ignored function symbols, concepts which can be as (or even more) important in a theory as the predicate symbols. Unfortunately, keeping function symbols in the abstract theory causes difficulties; it remains an open problem whether it is possible to do so whilst retaining both truthfulness and decidability. We have been exploring how little truthfulness we can sacrifice in order to gain decidability. We can readily improve upon old gazing's treatment of function symbols by modifying the abstraction of the last section to include both the name of function symbols and their positions. This abstraction (called "gazing with function symbols" from now on) is described in more detail in [GW89].

example	gazing with functions	gazing	old gazing	peeking
1	$\surd$	$\surd$	$\surd$	$\surd$
2	$\surd$	$\surd$	$\surd$	$\otimes$
3	$\surd$	$\surd$	$\oplus$	$\surd$
4	$\surd$	$\otimes$	$\surd$	$\otimes$
5	$\surd$	$\surd$	$\times$	$\surd$

Figure 4: Summary of results

7 Implementation and results

We have implemented a definitional theorem prover (in Prolog) to test the various abstractions and their successes in unfolding definitions. The prover consists of three parts: a (user-defined) abstraction, a planner, and a first order natural deduction theorem prover. The planner incorporates an efficient propositional decision procedure for theorem proving in the abstract theory; this is used to determine which definitions to unfold. The natural deduction theorem prover is then used to complete the proof by logical inference alone.

Two abstractions (gazing with function symbols and gazing) have been tested using the set theory defined in figure 1. Gazing out-performed old gazing and peeking on all theorems without function symbols, though old gazing obviously had the edge on theorems mentioning function symbols. Gazing with function symbols, however, out-performed all the other abstractions on all the theorems (with or without function symbols).

Figure 4 summarises some of the results. The results for old gazing and peeking come from [Plu87]. The examples chosen highlight the weaknesses and strengths of the different abstractions. In the figure, a $\surd$ indicates that the appropriate definitions are unfolded to allow the theorem to be completed by logical inference alone, a $\times$ indicates that the wrong definitions are unfolded, a $\oplus$ indicates that definitions are unfolded unnecessarily, and a $\otimes$ indicates that insufficient definitions are unfolded to complete the proof by logical inference.

Example 1: the theorem $\vdash\ a =_{set} b\ \rightarrow\ a \subseteq b$. All the various abstractions succeed in forming a plan.

Example 2: the theorem $\vdash\ a \subset b\ \rightarrow\ (x \in a \rightarrow x \in b)$. This demonstrates the weakness of peeking look-

needs to be unfolded into "$\subseteq$" and "$=_{set}$". The "$\subseteq$" then needs to be unfolded to give a common currency, "$\in$" with the conclusion.

Example 3: the theorem $\vdash\ a =_{set} \emptyset \rightarrow\ (\exists x.\ x =_{set} a)$. With this theorem, old gazing unnecessarily unfolds the definition of the empty set; gazing with function symbols overcomes this problem as we retain the connection between a predicate and a predicate with a function as argument.

Example 4: the theorem $\vdash\ \ a \subseteq \emptyset \rightarrow\ \ a =_{set} \emptyset$. Gazing and peeking both fail on this theorem because they ignore function symbols.

Example 5: the theorem $\vdash\ \ 2^a \in b\ \ \rightarrow\ \ \exists x.x \in b$. This example highlights the importance of remembering the position of function symbols in a predicate; old gazing ignored the position of function symbols and so suggested unfolding the definition of power set, "2^a" in the first argument position of "$\in$" even though the definition of power set is for the second argument position.

Gazing is, of course, not free; it would be pointless if the overhead of gazing outweighed its benefits; we therefore compared the time it took to tackle a theorem using gazing against the time taken to "blindly" unfold definitions. To give gazing a sporting chance, we used a theory in which predicates were multiply defined, so that choosing the right unfolding was important. The results are summarised in figure 5; the times given are for a Sun 3/60 running Quintus Prolog 2.2. The break-even point came after unfolding only one definition, showing how (as with peeking) even looking ahead a little can be worthwhile.

8 Final remarks and conclusions

Most theorem provers have used very *local* strategies for handling definitions. In comparison, gazing is a very powerful way of *globally* planning the unfolding of definitions. To understand gazing, we have developed a general theory of abstraction; this allows us to prove the desirable properties that an abstraction should possess. These ideas have been successfully implemented in a definitional theorem prover that determines which definitions to unfold to complete a proof by logical inference alone.

Much work still needs to be done. Two directions seem worth investigating. First of all there are still problems in how we deal with function symbols. It remains an open problem whether it is possible to abstract function symbols and retain both truthfulness and decidability. More work, very much related to the more general issue of abstract theorem proving, also needs to be done on the issue of extracting further information from the abstract proof. The goal is to define criteria and techniques which allow the structure of the proof in the abstract space to be "as close as possible" to the structure of the proof in the unabstracted space.

Acknowledgements

The research described in this paper owes a lot to the openness and sharing of ideas which exists in the mathematical reasoning group. The authors wish to thank Alan Bundy, Jane Hesketh, Alex Simpson, Alan Smaill and Andrew Stevens for the many discussions on this topic and on the work previously done inside the group. We would also like to thank Alan Bundy for reading the paper.

References

[BT75] W.W. Bledsoe and M. Tyson. The ut interactive prover. Technical report, Mathematics Department, University of Texas, 1975. ATP-17.

[Ern73] G.W. Ernst. A definition-driven theorem prover. In *Proceedings of the 3rd IJCAI*, pages 51–55. International Joint Conference on Artificial Intelligence, 1973.

[GW89] F. Giunchiglia and T. Walsh. Theorem Proving with Definitions. In *Proceedings of AISB 89*. Society for the Study of Artificial Intelligence and Simulation of Behaviour, 1989. Also available as DAI Research Paper No 429, Dept. of Artificial Intelligence, Edinburgh.

[GW90] F. Giunchiglia and T. Walsh. A Theory of Abstraction. Research paper 516, Dept. of Artificial Intelligence, University of Edinburgh, 1990. Accepted to Journal of Artificial Intelligence.

[Pas78] D. Pastre. Automatic theorem proving in set theory. *Artificial Intelligence*, 10:1–27, 1978.

[Plu87] D. Plummer. *Gazing: Controlling the Use of Rewrite Rules.* PhD thesis, Dept. of Artificial Intelligence, University of Edinburgh, 1987.

[Pra65] D. Prawitz. *Natural Deduction - A proof theoretical study.* Almquist and Wiksell, 1965.

[Sim88] A. Simpson. Grazing: A Stand Alone Tactic for Theoretical Inference. Master's thesis, Dept. of Artificial Intelligence, University of Edinburgh, 1988.

[Sup57] P. Suppes. *Introduction to Logic.* D. Van Nostrand Company, 1957.

[War87] K. Warren. Implementation of a definition expansion mechanism in a connection method theorem prover. Master's thesis, Dept. of Artificial Intelligence, University of Edinburgh, 1987.

[Wos88] L. Wos. *Automated Reasoning: 33 Basic Research Problems.* Prentice Hall, 1988.

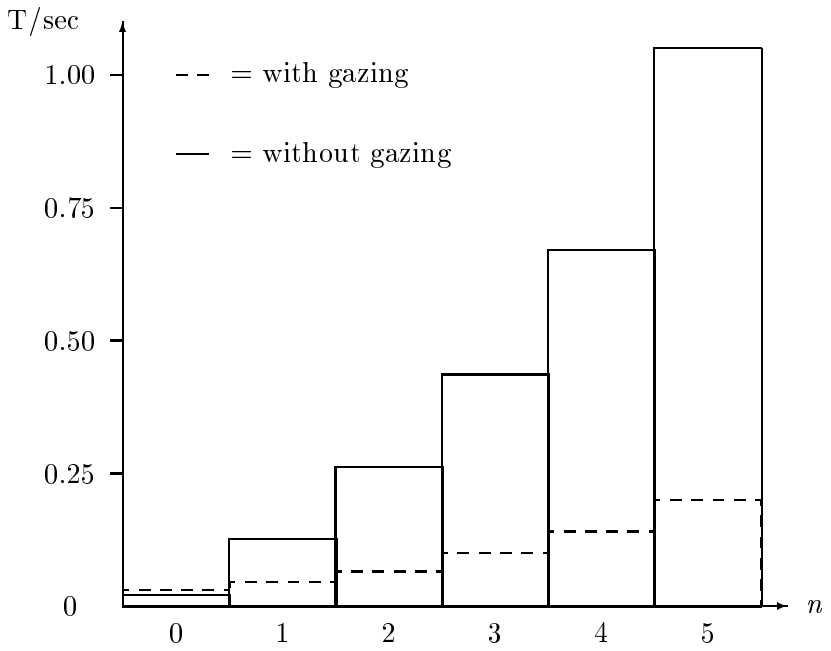

Figure 5: Time to prove theorem that requires n definitions unfolded