Proc. of the 1991 IEEE
Int. Conf. on Tools for AI
San Jose, CA-Nov. 1991

Experience-Based Deductive Learning

Joongmin Choi and Stuart C. Shapiro
Department of Computer Science
State University of New York at Buffalo
Buffalo, New York 14260

Abstract

A method of deductive learning is proposed to control deductive inference. Our goal is to improve problem solving time by experience, when that experience monotonically adds knowledge to the knowledge base. Accumulating and exploiting experience are done by the schemes of knowledge migration *and* knowledge shadowing. *Knowledge migration generates specific (migrated) rules from general (migrating) rules, and accumulates deduction experience represented by specificity relationships between migrating and migrated rules. Knowledge shadowing recognizes rule redundancies during a deduction, and prunes deduction branches activated from redundant rules. Three principles for knowledge shadowing are suggested depending on the details of deduction experience representation.*

1 Introduction

Learning in deduction became an active area recently by the emergence of several deductive learning methods such as explanation-based learning (EBL) [4] and chunking [2]. Deductive learning attempts to improve a system's performance by exploiting past problem-solving experience. However, one of major drawbacks of previous deductive learning methods was that they spent a considerable amount of time to learn new rules or apply learned rules to subsequent reasoning, for instance, to traverse an explanation structure to build a new description in EBL, and to match a number of complex chunks in chunking.

This paper proposes an experience-based deductive learning mechanism that controls deductive inference and improves the performance of a deductive reasoning system. Our system unifies learning with reasoning, i.e. the learning process is incorporated within the inference engine, and consequently a low-cost learning module is maintained that is strong enough to show significant differences in the system's behavior. We focus on deductive reasoning systems where partial results are saved during a deduction and at least some partial results are, themselves, deduction rules. In this environment, the general issue would be how maximal advantage can be taken of old partial results and how the regeneration of partial results can be avoided when solving new problems.

In this paper, the problems of accumulating past deduction experience and using it in subsequent deductions are tackled by the schemes of *knowledge migration* and *knowledge shadowing.* Knowledge migration generates specific (migrated) rules from general (migrating) rules, and accumulates deduction experience represented by specificity relationships between migrating and migrated rules. Knowledge shadowing exploits past experience to make future inferences faster. Knowledge shadowing recognizes rule redundancies during a deduction, and prunes deduction branches activated from redundant rules.

The major contribution of this work is to improve a system's performance (especially deduction speed) over time for similar problems, even though the search space (or branching factor) is increased by monotonically adding derived knowledge to the knowledge base that creates redundancies.

2 Knowledge Migration

Knowledge migration is a process of acquiring specific rules from general rules, and is defined as below.

Definition Let $\mathcal{G} = \langle \mathcal{A}, \mathcal{C} \rangle$ be a nested rule, where $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ is a set of antecedents, $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ is a set of consequents, and each $C_i (1 \leq j \leq m)$ is a rule. *Knowledge migration* is defined as a process of acquiring a set of rules $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ from $\mathcal{G}$ during a deduction when (1) $\mathcal{G}$ is involved in the deduction, and (2) there is a substitution σ, called a *migrating substitution*, such that $A_i\sigma (1 \leq i \leq n)$ is in the knowledge base, and $S_j = C_j\sigma (1 \leq j \leq m)$.

As an example, consider a knowledge base containing a general transitive rule R1 $= \forall R\ [trans(R) \rightarrow \forall x, y, z\ [R(x, y)\ \&\ R(y, z) \rightarrow R(x, z)]]$, and facts $trans(on)$, $on(a, b)$, $on(b, c)$, $on(c, d)$. A deduction of $on(a,c)$ from this knowledge base activates R1, and generates a specific rule R2 $= \forall x, y, z\ [on(x, y)\ \&\ on(y, z) \rightarrow on(x, z)]$ from R1 with a migrating substitution $\sigma = \{on/R\}$, since $trans(R)\sigma = trans(on)$ is in the knowledge base. This migration process produces deduction experience that is represented by the specificity relationship between the migrating rule and the migrated rule. The details of the representation are described in the next section.

3 Knowledge Shadowing

Knowledge shadowing is a way of exploiting deduction experience acquired in previous inferences to make subsequent similar deductions more efficient. The main job of knowledge shadowing is to recognize unnecessary and redundant rules and to block them

0-8186-2300-4/91 $01.00 © 1991 IEEE

from the inference by investigating the specificity relationships between rules that are represented in deduction experience.

As an example, consider the example shown in the previous section. After deducing *on(a,c)*, R2 and *on(a,c)* are added to the knowledge base. Now, another deduction of *on(b,d)* from the changed knowledge base proceeds in two branches since both R1 and R2 are applicable. However, by recognizing that R2 is more specific than R1, the branch from R1 can be regarded as redundant in the sense that *on(b,d)* can be derived solely by the branch from R2 with fewer steps. As a result, the branch activated from R1 is pruned.

Knowledge shadowing is made possible by three ways of representing experience, i.e., by using instance sets, by using origin sets, and by using common instances.

In the first method, an *instance set* $\mathcal{I}^R$ is maintained for each rule R to memorize migrated instances. A *migrated instance* is represented by a pair $\langle \mathbf{r}, \sigma \rangle$, where $\mathbf{r}$ is a rule migrated from R by a migrating substitution σ. For example, after the migration from R1 occurs during the derivation of *on(a,c)*, $\mathcal{I}^{R1_{cq}}$ becomes $\{\langle \mathtt{R2}, \{on/R\}\rangle\}$, where $\mathtt{R1}_{cq} = \forall x,y,z\ [R(x,y)\ \&\ R(y,z) \rightarrow R(x,z)]$ is the consequent of R1. Knowledge shadowing by using instance sets can be accomplished by the following principle.

Shadowing Principle 1 Let $R_1, R_2, \ldots, R_k$ be rules that are all applicable by a query **q** at some point during a deduction. This implies the existence of substitutions $\phi_1, \phi_2, \ldots, \phi_k$ obtained by pattern matchings between **q** and a consequent of $R_i, 1 \leq i \leq k$. Then, a deduction branch activated by a rule R_i is **shadowed** from the inference when there is an instance $\langle R_j, \sigma \rangle \in \mathcal{I}^{R_i}$ such that $1 \leq j \leq k, j \neq i$, and $\phi_i \supseteq \sigma$.

In the transitive example, the query *on(b,d)* makes both R1 and R2 applicable but shadows R1 by the principle 1, since the pattern matching between the query *on(b,d)* and *R(x,z)*, which is the consequent of $\mathtt{R1}_{cq}$, produces a unifier $\phi_1 = \{on/R,\ b/x,\ d/z\}$, and there is a migrated instance $\langle \mathtt{R2}, \{on/R\}\rangle$ in $\mathcal{I}^{R1_{cq}}$ satisfying $\phi_1 \supset \{on/R\}$.

The second method of shadowing uses an *origin set* (OS) that is associated with each proposition to keep track of and propagate propositional dependencies in an assumption based truth maintenance system SNeBR [3]. In the above example, OS of R1 is {R1} and OS of R2 is {R1, *trans(on)*} since R2 is derived from the two propositions. From the viewpoint of deductive learning, propositional dependencies represented by OS can be regarded as a type of experience. A subset-superset comparison between OSs of two rules can be used to shadow a redundant rule. For instance, R1 can be regarded as more general than R2 since OS of R1 is a proper subset of the OS of R2. A general shadowing principle using OS is described below.

Shadowing Principle 2 Let $R_1, R_2, \ldots, R_k$ be rules that are all applicable by a query **q** at some point during a deduction. Also let $O_1, O_2, \ldots, O_k$ be origin sets of each $R_i, 1 \leq i \leq k$. Then, a deduction branch activated by a rule R_i is **shadowed** from the inference when (1) there is a rule R_j $(j \neq i, 1 \leq j \leq k)$ such that $O_j \supset O_i$, and (2) the outermost quantifier variables of R_i which also appear in R_j are bound by the pattern matching of **q** and a consequent of R_i.

The third method of shadowing uses the concept of common instances between two patterns. Matching between two patterns S and T produces the source binding σ and the target binding τ that satisfies $S\sigma = T\tau$. Essentially, σ and τ are factored versions of the most general unifier (**mgu**). By using the factorized **mgu**, we now define *the most general common instance* (mgci) of two patterns S and T such that $\mathtt{mgci}_{ST} = S\sigma = T\tau$. For example, a matching between *P(x,b)* and *P(a,y)*, where x and y are variables, produces $\sigma = \{a/x\}$ and $\tau = \{b/y\}$, and the mgci of these two is *P(a,b)*. We are striving to find a way of blocking deduction chains by using the mgci of two patterns, and obtain the following principle.

Shadowing Principle 3 A deduction branch initiated by a rule R is **shadowed** when the mgci of the query **q** and a consequent of R is ground and already asserted in the knowledge base.

In the transitive rule example when R1 and R2 coexist, the query *R(a,c)*, where R is a variable, will shadow R2 since the mgci of *R(a,c)* and *on(x,z)* is *on(a,c)* which is a ground instance and already asserted.

For all the three shadowing principles, we claim that a deduction branch shadowed by any shadowing principle would never have produced any new results that cannot be produced by non-shadowed branches.

Implementation of knowledge migration and shadowing in SNePS [5] is in progress, and a preliminary result of performance improvement is shown in [1].

References

[1] Joongmin Choi and Stuart C. Shapiro. Learning in deduction by knowledge migration and shadowing. In *Proceedings of AAAI-91 Workshop on Knowledge Acquisition: From Science to Technology to Tools*, Anaheim, CA, 1991. AAAI.

[2] John E. Laird, Paul S. Rosenbloom, and Allen Newell. Towards chunking as a general learning mechanism. In *Proceedings of National Conference on Artificial Intelligence*, pages 188–192, Austin, TX, August 1984. William Kaufmann, Los Altos.

[3] Joao P. Martins and Stuart C. Shapiro. A model for belief revision. *Artificial Intelligence*, 35:25–79, 1988.

[4] Tom M. Mitchell, Richard M. Keller, and Smadar T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1:47–80, 1986.

[5] S. C. Shapiro and W. J. Rapaport. SNePS considered as a fully intensional propositional semantic network. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier*, pages 263–315. Springer–Verlag, New York, 1987.