On the Adequateness of the Connection Method

Antje Beringer and **Steffen Hölldobler**
Intellektik, Informatik, TH Darmstadt,
Alexanderstraße 10, D–6100 Darmstadt (Germany)
E-mail: {antje, steffen}@intellektik.informatik.th-darmstadt.de

Abstract

Roughly speaking, adequatness is the property of a theorem proving method to solve simpler problems faster than more difficult ones. Automated inferencing methods are often not adequate as they require thousands of steps to solve problems which humans solve effortlessly, spontaneously, and with remarkable efficiency. L. Shastri and V. Ajjanagadde — who call this gap the artificial intelligence paradox — suggest that their connectionist inference system is a first step toward bridging this gap. In this paper we show that their inference method is equivalent to reasoning by reductions in the well-known connection method. In particular, we extend a reduction technique called evaluation of isolated connections such that this technique — together with other reduction techniques — solves all problems which can be solved by Shastri and Ajjanagadde's system under the same parallel time and space requirements. Consequently, we obtain a semantics for Shastri and Ajjanagadde's logic. But, most importantly, if Shastri and Ajjanagadde's logic really captures the kind of reasoning which humans can perform efficiently, then this paper shows that a massively parallel implementation of the connection method is adequate.

Introduction

Adequateness is one of the assumptions underlying automated deduction. Following W. Bibel [1991], *there is an* adequate *general proof method that can automatically discover any proof done by humans provided the problem (including all required knowledge) is stated in appropriately formalized terms.* Adequateness is, roughly speaking, understood as the property of a theorem proving method that, *for any given knowledge base, the method solves simpler problems faster than more difficult ones.* Furthermore, *simplicity is measured under consideration of all (general) formalisms available to capture the problem* and *intrinsic in this assumption is a belief in the existence of an algorithm that is feasible (from a complexity point of view) for the set of problems humans can solve.* Later on, Bibel defines general research goals in the field of automated deduction, the first goal being the search for general and adequate proof methods.

This paper is concerned with adequate proof methods. That adequateness is one of the main problems in automated deduction has been realized by many researchers (cf. [Levesque and Brachman, 1985; Levesque, 1989]). L. Shastri and V. Ajjanagadde [1990; 1993] even call the gap between the ability of humans to draw a variety of inferences effortlessly, spontaneously, and with remarkable efficiency on the one hand, and the results about the complexity of reasoning reported by researchers in artificial intelligence on the other hand, the *artificial intelligence paradox*. But they also developed a connectionist computational model — called SAM in the sequel — which can encode a knowledge base consisting of millions of facts and rules, and performs a class of inferences with parallel time bound by the length of the shortest proof and space bound by the size of the knowledge base.[1] Moreover, SAM is consistent with recent neurophysiological findings and makes specific predictions about the nature of reflexive reasoning — ie. spontaneous reasoning as if it were a reflex [Shastri, 1990] — that are psychologically significant. Shastri and Ajjanagadde suggest that their computational model is a step towards resolving the artificial intelligence paradox.

Logically, the knowledge bases considered by Shastri and Ajjanagadde are sets of definite clauses — ie. universally closed clauses of the form $A_1 \wedge \ldots \wedge A_n \Rightarrow A$, where the A, A_i, $1 \leq i \leq n$, are atomic sentences. The knowledge bases are queried by universally closed atomic sentences. Such queries are answered positively if they are logical consequences of the knowledge base, and negatively otherwise. The query as well as the facts and rules are restricted in some particular way, which makes the system both interesting and difficult at the same time. It is difficult to give a semantics

[1] SAM shares many features with the connectionist reasoning system ROBIN developed by Lange and Dyer [1989]. They differ mainly in their technique to represent variable bindings.

for the class of problems considered in [Shastri and Ajjanagadde, 1993] and to understand the influence of the various restrictions on the massively parallel computational model. It is interesting as we would like to understand what kind of problems can be handled by a massively parallel computational model in parallel time bound by the length of the shortest proof and in space bound by the size of the knowledge base. In the following section we present the various restrictions in detail and outline the computational model underlying Shastri and Ajjanagadde's approach.

In [Hölldobler, 1993] the suggestion was made that reflexive reasoning is reasoning by reductions and, consequently, that the problems solved by SAM are simpler than the problems investigated by the artificial intelligence community. In this paper we will show that this suggestion holds by comparing SAM with the connection method and its various reduction techniques [Bibel, 1987]. An optimal parallel implementation of these reduction techniques along the lines of CHCL [Hölldobler, 1990a; Hölldobler and Kurfeß, 1992] needs the same space and answers queries at least as fast as SAM. We also demonstrate that there are reflexive reasoning tasks which can be solved in essentially one step by the parallel connection method, whereas SAM needs parallel time bound by the depth of the search space.

We present the connection method and the relevant reduction techniques. To prove our main result we extend the definition of isolated connections [Bibel, 1988] to so-called pointwise isolated connections (PICs). The evaluation of PICs is a general reduction technique applicable to unrestricted first-order formulas. With the help of this technique we can prove our main result relating SAM and the connection method.

Thus, the paper gives a semantics for the class of formulas considered in [Shastri and Ajjanagadde, 1993] and extends the definition of isolated connections such that this reduction technique solves reflexive reasoning tasks in parallel time bound by the length of the shortest proof and space bound by the size of the knowledge base. But, most importantly, if Shastri and Ajjanagadde's logic is the kind of logic needed for representing reasoning tasks which can be solved effortlessly, spontaneously, and with remarkable efficiency by humans, then this paper shows that a parallel implementation of the connection method is adequate. These and related results are discussed in the final section.

Reflexive Reasoning

Shastri and Ajjanagadde [1990; 1993] identified a class of problems computable in space bound by the size of the knowledge base and in parallel time which is at worst sublinear in — and perhaps even independent of — the size of the knowledge base. The was motivated by the observation that humans can perform a limited class of inferences extremely fast although their knowledge base is extremely large.

In this section we introduce the backward reasoning system SAM. As we are mainly interested in the logic of the system we neither give technical details nor discuss the biological plausibility of the connectionist model underlying SAM or the psychological significance of reflexive reasoning. A detailed discussion of these topics can be found in [Shastri and Ajjanagadde, 1993].

Let C be a finite set of constants. A knowledge base KB in [Shastri and Ajjanagadde, 1993] is a conjunction of rules and facts. The rules are of the form

$$\forall X_1 \ldots X_m \, [p_1(\ldots) \wedge \ldots \wedge p_n(\ldots) \Rightarrow \exists Y_1 \ldots Y_k \; p(\ldots)], \quad (1)$$

where p, p_i, $1 \leq i \leq n$, are multi-place predicate symbols, the arguments of the p_i are variables from the set $\{X_1, \ldots, X_m\}$, and the arguments of p are from $\{X_1, \ldots, X_m\} \cup \{Y_1, \ldots, Y_k\} \cup C$.[2] The facts and queries (or goals) are of the form

$$\exists Z_1 \ldots Z_l \; q(\ldots), \quad (2)$$

where q is a multi-place predicate symbol and the arguments of q are from $\{Z_1, \ldots, Z_l\} \cup C$. The rules, facts, and goals are restricted as follows.

1. There are no function symbols except constants.
2. Only universally bound variables may occur as arguments in the conditions of a rule.
3. All variables occurring in a fact or goal occur only once and are existentially bound.
4. An existentially quantified variable (occurring in the head of a rule or in a fact) is only unified with variables.
5. A variable which occurs more than once in the conditions of a rule must occur in the conclusion of the rule and must be bound when the conclusion is unified with a goal.
6. A rule is used only a fixed number of times.

From an automated deduction point of view some of these restrictions seem to be rather peculiar. But they are closely related to the mechanism used by Shastri and Ajjanagadde for representing variable bindings. The variable binding problem is one of the major problems in connectionist systems (cf. [Barnden, 1984]). Due to lack of space we cannot discuss connectionist solutions to this problem and the interested reader is referred to [Shastri and Ajjanagadde, 1993]. In this paper we want to concentrate on the semantics of the logic described above. One should observe that restrictions 4-6 cannot be checked statically, but must be checked dynamically. The final restriction is concerned with the problem of how many copies of a rule or fact are needed for a proof of a first-order formula. The dynamic creation of copies is again a major problem in connectionist systems not discussed here. Following [Shastri and Ajjanagadde, 1990], we assume wlog. that each rule may be used at most once. As it is unpredictable how many copies of a rule are needed, SAM is incomplete. A positive answer indicates that the goal G is entailed by the knowledge base KB. A negative answer indicates that either G is not entailed by KB

[2] We use uppercase letters for variables and lower case letters for constants, function and predicate symbols.

or it cannot be proven that G is entailed by KB under the given restrictions.

Showing that KB entails G is equivalent to showing that $KB \wedge \neg G$ is unsatisfiable. To determine unsatisfiability we may replace existentially bound variables occurring in $KB \wedge \neg G$ by Skolem terms and obtain a formula $KB' \wedge G'$ which is equivalent to $KB \wedge \neg G$ wrt. unsatisfiability. Let σ be the substitution $\{Y_1 \mapsto f_1(X_1, \ldots, X_m), \ldots, Y_k \mapsto f_k(X_1, \ldots, X_m)\}$ and θ be the substitution $\{Z_1 \mapsto c_1, \ldots, Z_l \mapsto c_l\}$, where the f_i, $1 \leq i \leq k$, are pairwise different Skolem functions and the c_j, $1 \leq j \leq l$, are pairwise different Skolem constants, each of which does not occur in the set C of constants. Then each rule of the form (1) in KB is replaced by the (universally closed) clause $\sigma p(\ldots) \leftarrow p_1(\ldots) \wedge \ldots \wedge p_n(\ldots)$ or, equivalently, by $\sigma p(\ldots) \vee \neg p_1(\ldots) \vee \ldots \vee \neg p_n(\ldots)$ and each fact of the form (2) in KB is replaced by the ground fact $\theta q(\ldots)$. A query of the form (2) corresponds to the (universally closed) goal clause $\neg q(\ldots)$. Altogether, the knowledge base is a set of definite clauses and a query is a goal clause as used in pure PROLOG. Observe that now condition 4 is checked by the unification computation as Skolem constants and functions cannot occur in the goal. Eg. consider the following knowledge base.

$$\begin{array}{lll} p(a,Y) \vee \neg q(Y). & p(c,a). & q(b). \\ p(b,Z) \vee \neg r(Z). & r(a). & q(c). \end{array} \quad (3)$$

If a query like $\neg p(X,a)$ is posed then SAM computes an answer in a three-step process as follows.

1. Constants occurring in the query are recursively propagated to all atoms with the query's predicate symbol; a unification computation is performed. In our example, a is propagated as second argument to $p(a,Y)$, $p(b,Z)$, and $p(c,a)$. The unification computations are succesful and yield the substitutions $\{Y \mapsto a\}$, $\{Z \mapsto a\}$, and ε (the empty substitution), resp. After an application of these substitutions a is propagated from $\neg q(a)$ and $\neg r(a)$ to $q(b)$, $q(c)$, and $r(a)$, resp. The first two unification computations yield failures, whereas the last one is successful. Resulting from this step, each leaf of the search tree is labeled with either *success* or *failure*. Figure 1 shows the example's search tree at this time.
2. The success labels at the leafs are propagated backwards to the root of the search tree. Thereby it is checked whether each condition of a rule is satisfiable, ie. is the root of a successful branch. Otherwise, all branches starting from the conditions of the rule are turned into failure branches.
3. The bindings for the variables occurring in the initial query are now obtained by propagating the variables through the search space along the successful branches. In our example we obtain the bindings $\{X \mapsto c\}$ and $\{X \mapsto b\}$.

Clearly, the first and second step are the most important as they determine the success and failure branches of the search space. The third step only collects the answer substitutions. To define a semantics for SAM

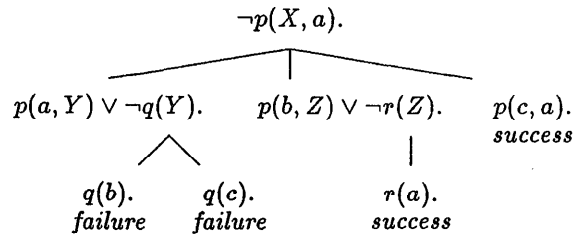


Figure 1: The example's search tree after the first step of SAM. (Substitutions are not applied.)

we will show that the computations performed by SAM are essentially reductions in a standard first-order logic calculus based on the connection method, which we present in the following section.

The Connection Method

The connection method is a formalism to compute the relationships between different statements in a first-order logic language [Bibel, 1987]. Although usually presented as an affirmative method for proving the validity of a formula in first-order logic, we present a dual version for proving the unsatisfiability of a set of Horn clauses, ie. a logic program and a single query. The connection method is based on the observation that a proof of a formula is essentially determined by a so-called spanning set of connections. A *connection* is an unordered pair of literal occurrences with the same predicate symbol but different signs. A literal L is *connected* in a set S of connections iff S contains L as an element of a connection. A set S of connections for a Horn formula of the form $KB \Rightarrow G$ consisting of a knowledge base or logic program KB and a goal G is called *spanning* iff each literal occurring in G is connected in S and, if the head of (a copy of) a clause in KB is connected in S, then each literal occurring in its body is also connected in S. A spanning set is *minimal* iff there is no spanning subset. A spanning set S of connections for $KB \Rightarrow G$ determines a proof iff there is a substitution σ such that σ simultaneously unifies each connection in S. Hence, searching a proof for $P \Rightarrow G$ amounts in generating a spanning set of connections and, then, simultaneously unifying each pair of connected literals.

In Figure 2 the connections for (3) are shown. There are two minimal spanning sets of connections which determine a proof: the sets $\{\langle \neg p(X,a), p(c,a) \rangle\}$ and $\{\langle \neg p(X,a), p(b,Z) \rangle, \langle \neg r(Z), r(a) \rangle\}$ with unifying substitutions $\{X \mapsto a\}$ and $\{X \mapsto b, Z \mapsto a\}$, resp. The other spanning sets $\{\langle \neg p(X,a), p(a,Y) \rangle, \langle \neg q(Y), q(b) \rangle\}$ and $\{\langle \neg p(X,a), p(a,Y) \rangle, \langle \neg q(Y), q(c) \rangle\}$ do not determine proofs as the variable Y cannot be bound to two different constants — viz. a, b and a, c, resp.

The problem whether a goal follows logically from a knowledge base is undecidable as one cannot determine in advance the number of needed copies of program clauses. However, if the number of copies is restricted

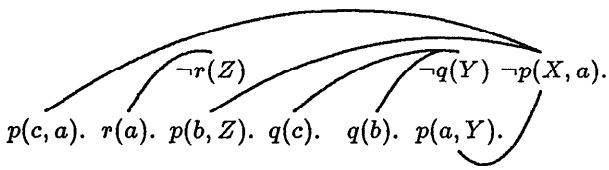


Figure 2: A simple logic formula with its connections. Each row represents either a fact, a rule, or a goal. Observe that the connection structure corresponds precisely to the search space shown in Figure 1.

as in SAM, then the problem is decidable as now the number of connections — and, hence, the number of spanning sets — is finite. Given a knowledge base and a goal, a procedure like SLD-resolution [Lloyd, 1984] may be used to find a proof if it exists. SLD-resolution is sound and complete, however, it may require exponential time (in the size of the formula) to find a proof. Hence, formulas should first be reduced as far as possible before a rule like SLD-resolution is applied.

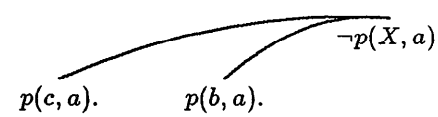
The notion of a reduction rule is not uniquely defined. But reduction rules do not change the (un-) satisfiability of a formula while decreasing some complexity measure assigned to formulas. Here we strengthen these conditions by requiring that reduction techniques are applicable in linear parallel time and linear space with respect to the knowledge base. This is in spirit of our ultimate goal to find a class of problems which can be solved extremely fast on a massively parallel machine and has reasonable space requirements. In this paper we are particularly concerned about the following reduction techniques.

- Connections between non-unifiable literals can be removed. Eg. the connections $\langle \neg q(a), q(b) \rangle$ and $\langle \neg q(a), q(c) \rangle$ are non-unifiable.
- Useless clauses may be removed. A clause is *useless* if its conclusion is not connected or its condition contains a subgoal which cannot be solved. Eg. the rule $p(a,a) \vee \neg q(a)$ is useless and, in this case, all connections with this rule can be removed.
- Isolated connections can be evaluated, ie. the connected literals can be unified. A connection $\langle L, L' \rangle$ is *isolated* iff the literals L and L' are either ground or not engaged in any other connection or one of the literals is ground and the other one is not engaged in any other connection. If an isolated connection is unifiable, then the corresponding clauses may be replaced by their resolvent; otherwise, the connection can be removed. In Figure 1, there is the single isolated connection $\langle r(a), \neg r(Z) \rangle$. The literals are unifiable with the substitution $\{Z \mapsto a\}$ and, thus, the rule $p(b,Z) \vee \neg r(Z)$ and the fact $r(a)$ may be replaced by their resolvent $p(b,a)$.

After the reduction of the isolated connection $\langle r(a), \neg r(Z) \rangle$ the formula shown in Figure 2 cannot be further reduced with the reduction techniques mentioned above. We would have to apply SLD-resolution, which may be exponential. However, as we will show in the sequel, the definition of isolated connections can be extended such that the problem shown in Figure 2 becomes solvable in linear space and linear parallel time with respect to the knowledge base by applying reduction techniques only.

The extension is based on the following observation. If an isolated connection of the form $\langle p(s_1, \ldots, s_n), \neg p(t_1, \ldots, t_n) \rangle$ is to be evaluated, then $p(s_1, \ldots, s_n)$ and $p(t_1, \ldots, t_n)$ are unified. The first step of the unification computation [Robinson, 1965] is to decompose the problem into the unification problems consisting of s_i and t_i, $0 \leq i \leq n$, and, then, to unify these (sub-)problems simultaneously. The extension consists of anticipating this step and considering *pointwise isolated connections* (PICs) between corresponding arguments of connected literals.

A connection $\langle p(s_1, \ldots, s_n), \neg p(t_1, \ldots, t_n) \rangle$ is called *isolated in its i-th argument* (or *point*) iff either the connected literals are not engaged in any other connection or s_i and t_i are ground or s_i is ground and $\neg p(t_1, \ldots, t_n)$ is not engaged in any other connection or t_i is ground and $p(s_1, \ldots, s_n)$ is not engaged in any other connection. Eg. the connection $\langle p(a,Y), \neg p(X,a) \rangle$ occurring in the example shown in Figure 2 is isolated in its second argument, but not isolated in its first argument. Evaluating the isolated point yields the substitution $\{Y \mapsto a\}$. Applying this substitution yields the connections $\langle q(b), \neg q(a) \rangle$ and $\langle q(c), \neg q(a) \rangle$, both of which are non-unifiable and can be removed. As now the subgoal $\neg q(a)$ is no longer connected, the rule $p(a,a) \vee \neg q(a)$ becomes useless and can be removed as well and we obtain the following reduced formula.



Both connections are pointwise isolated and unifiable in their second argument. Both connections define proofs with answer substitutions $\{X \mapsto c\}$ and $\{X \mapsto b\}$. One should observe that this formula corresponds precisely to the successful branches of the search space shown in Figure 1.

The following proposition is an immediate consequence of the definition of PICs.

Proposition 1 *1. A connection is isolated iff it is isolated in each point.*

2. Let F' be obtained from a formula F by evaluating PICs*. F is unsatisfiable iff F' is unsatisfiable.*

PICs were introduced as an extension of isolated connections [Bibel, 1988]. But they can also be viewed as a special case of the v-rule defined in [Munch, 1988] for the connection graph proof procedure. Whereas the

application of the (more complicated) v-rule is expensive to control, the evaluation of PICs is quite efficient.

Reflexive Reasoning is Reasoning by Reductions

The goal of this section is to elucidate the relation between SAM and the connection method. Let KB be a knowledge base, G a goal, F the formula $KB \wedge \neg G$, and F' be obtained by reducing F as far as possible. We assume that KB satisfies the conditions 1-4 defined in the second section. As conditions 5 and 6 must be tested dynamically by a meta-level controller, we assume that they are satisfied.

By definition the search space explored by SAM is defined by the connections of the given formula. As the first and second step of SAM determine the success and failure branches of the search space we have to show that the first and second step of SAM can be simulated by reductions in the connection method. In the first step the constants occurring in the intial goal are propagated through the search space. Recall that following [Shastri and Ajjanagadde, 1990] we have assumed that a rule is used only once. In the connection method this assumption translates into the condition that the conclusion of each rule is connected at most once. Hence, if a constant occurs at the i-th argument of a goal, then the connection between the goal and the conclusion of a rule is isolated in its i-th point. Recall further that facts are always ground as they are skolemized. Thus, if a constant occurs at the i-th argument of a goal, then the connection between the goal and a fact is isolated in its i-th point. Hence, the PICs can be evaluated — ie. unified — as in SAM. One should observe that condition 2 guarantees that after the evaluation of the PICs between the goal and the conclusion of a rule each constant occurring in the conditions of the rule occurs also in the goal. Using these arguments it can be shown by induction on the depth of the search space that SAM's propagation of constants occurring in the initial query corresponds to evaluations of PICs. Thereafter, all non-unifiable connections are eliminated in the connection method. Finally, if a condition of a rule is not the root of a successful branch, then the rule becomes useless and is eliminated. This takes care of the second step of SAM. Altogether we obtain the following result.

Theorem 2 *Let S be the search space after the second step of* SAM *in an attempt to show the unsatisfiability of F. The connections of F' correspond precisely to the successful branches of S.*

Hence, after reducing F we are left with all the successful branches of the search space. To show that SAM is sound, we have to prove that each minimal spanning set of F' determines a proof, ie. that for each minimal spanning set there is a substitution which simultaneously unifies each connection in the spanning set. By induction on the size of the minimal spanning set it can be shown that in the unification computation of the connected literals in a minimal spanning set each variable may be bound to at most one constant. By condition 1 no complex data structures can be built up via unification. Conditions 2, 3, and 5 ensure that there are no multiple occurrences of variables in the goal, the facts, the conclusion and the condition of a rule. This implies the following result.

Theorem 3 *Each minimal spanning set of F' determines a proof.*

The soundness of SAM is an immediate consequence of Theorems 2, 3 and Proposition 1. As the knowledge base in [Shastri and Ajjanagadde, 1993] is a logic program, we can define the usual model-theoretic, fixpoint, and operational semantics (based on SLD-resolution) as in [van Emden and Kowalski, 1976] or the S-semantics as in [Falaschi *et al.*, 1989]. Since the connection method is equivalent to SLD-resolution for Horn formulas SAM is sound with respect to these semantics. As already mentioned SAM is incomplete as conditions 5 and 6 cannot be checked in advance. One could easily change the operational and fixpoint semantics such that these conditions are obeyed. Similarly, the model-theoretic semantics can be refined by expressing these conditions as higher-order axioms which have to be satisfied. This, however, is just an exercise in defining semantics.

Discussion

We have extended the reduction technique of evaluating isolated connections. The basic idea of considering PICs is that, whenever a binding for a variable can uniquely be determined, this binding should be applied and propagated. Secondly, we have shown that reflexive reasoning as defined by Shastri and Ajjanagadde [1990; 1993] or [Lange and Dyer, 1989] is reasoning by reduction and, consequently, we have formally established the soundness of reflexive reasoning.

There exists already a connectionist model of the connection method for Horn formulas called CHCL [Hölldobler, 1990a; Hölldobler and Kurfeß, 1992]. In CHCL isolated connections are evaluated, non-unifiable connections and useless clauses are removed in parallel as soon as these reductions become applicable. CHCL can easily be extended to evaluate PICs as CHCL determines the property of being isolated with the help of Proposition 1(1) (although PICs were not introduced in [Hölldobler, 1990a; Hölldobler and Kurfeß, 1992]). Interestingly, CHCL solves some problems even faster than SAM does. For example, if the knowledge base consists of the rules

$$p_1(X_1) \vee \neg p_2(X_1), \ \ldots, \ p_{n-1}(X_{n-1}) \vee \neg p_n(X_{n-1})$$

and the fact $p_n(a)$, then the query $p_1(a)$ is solved in essentially one step by CHCL as all connections are isolated and, hence, simultaneously evaluated. SAM needs essentially n steps as the constant a occurring in the goal has to be propagated through the search

space. On the other hand, CHCL is less space efficient than SAM. CHCL does not require that formulas obey conditions 1-5 in the second section. Rather, formulas may be arbitrary Horn formulas. Hence, CHCL must solve arbitrary unification problems. This is done by a connectionist unification algorithm [Hölldobler, 1990c], which uses a quadratic number of units with respect to the size of the knowledge base. If the formulas and, consequently, unification was restricted as in SAM, then the design of CHCL could be changed such that it needs the same space and answers queries at least as fast as Shastri and Ajjanagadde's system.

The class of problems considered by Shastri and Ajjanagadde [1990; 1993] does not seem to be the largest class of problems which is computable in space bound by the size of the knowledge base and in parallel time bound by the depth of the search space. The presented reduction techniques are not restricted to Horn formulas, but may be applied to general first-order formulas. The special unification problems solved by Shastri and Ajjanagadde [1990; 1993] are not the largest class of unification problems which can be parallelized in an optimal way. Whereas unification is inherently sequential [Dwork *et al.*, 1984], matching is known to be efficiently parallelizable [Ramesh *et al.*, 1989].

The results of this paper show that SAM computes by reductions. From Shastri and Ajjanagadde's work we learn that automated theorem provers which apply these reduction techniques in parallel are adequate in the sense that they solve simpler problems faster than more difficult ones. [Shastri and Ajjanagadde, 1993] also contains some predictions on the question whether common sense reasoning problems are expressible in Shastri and Ajjanagadde's logic. It remains to be seen whether these predictions hold. In fact, if the predictions hold then the gap between the ability of humans to draw a variety of inferences as if it was a reflex and the results about the complexity of reasoning reported by researchers in artificial intelligence is not a paradox at all. If the problems which can be solved effortlessly, spontaneously, and with remarkable efficiency by humans can be expressed in Shastri and Ajjanagadde's logic, then these problems are just simpler than the problems investigated in the artificial intelligence community.

Acknowledgements

We like to thank S. Brüning for useful comments on earlier drafts of this paper. The first author is supported by the Deutsche Forschungsgemeinschaft (DFG) within project MPS under grant no. HO 1294/3-1

References

J. A. Barnden. On short term information processing in connectionist theories. *Cognition and Brain Theory*, 7:25–59, 1984.

W. Bibel. *Automated Theorem Proving.* Vieweg Verlag, Braunschweig, second edition, 1987.

W. Bibel. Advanced topics in automated deduction. In *Fundamentals of Artificial Intelligence II*, p. 41–59. Springer, LNCS *345*, 1988.

W. Bibel. Perspectives on automated deduction. In *Automated Reasoning: Essays in Honor of Woody Bledsoe*, p. 77–104. Kluwer Academic, Utrecht, 1991.

C. Dwork, P. C. Kannelakis, and J. C. Mitchell. On the sequential nature of unification. *Journal of Logic Programming*, 1:35–50, 1984.

M Falaschi, G. Levi, M. Martelli, and C. Palamidessi. Declarative modelling of the operational behavior of logic languages. *TCS*, 69(3):289–318, 1989.

S. Hölldobler and F. Kurfeß. CHCL – A connectionist inference system. In *Parallelization in Inference Systems*, p. 318 – 342. Springer, LNAI *590*, 1992.

S. Hölldobler. CHCL - A connectionist inference system for Horn logic based on the connection method and using limited resources. TR-90-042, ICSI, Berkeley, 1990.

S. Hölldobler. On high-level inferencing and the variable binding problem in connectionist networks. In *Proc. ÖGAI*, p. 180–185. Springer IFB *252*, 1990.

S. Hölldobler. A structured connectionist unification algorithm. In *Proc. AAAI*, p. 587–593, 1990.

S. Hölldobler. On the artificial intelligence paradox. Journal of Behavioral and Brain Sciences, 1993. (to appear).

T. E. Lange and M. G. Dyer. High-level inferencing in a connectionist network. *Connection Science*, 1:181 – 217, 1989.

H. J. Levesque and R. J. Brachman. A fundamental tradeoff in knowledge representation and reasoning. In *Readings in Knowledge Representation*, p. 41–70. Morgan Kaufmann, 1985.

H. J. Levesque. Logic and the complexity of reasoning. KRR-TR-89-2, Dept. of CS, Univ. of Ontario, Toronto, 1989.

J. W. Lloyd. *Foundations of Logic Programming.* Springer, 1984.

K. H. Munch. A new reduction rule for the connection graph proof procedure. *JAR*, 4:425–444, 1988.

R. Ramesh, R. M. Verma, T. Krishnaprasad, and I. V. Ramakrishnan. Term matching on parallel computers. *Journal of Logic Programming*, 6:213 – 228, 1989.

J. A. Robinson. A machine-oriented logic based on the resolution principle. *JACM*, 12:23–41, 1965.

L. Shastri and V. Ajjanagadde. An optimally efficient limited inference system. In *Proc. AAAI*, p. 563–570, 1990.

L. Shastri and V. Ajjanagadde. From associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioural and Brain Sciences*, 1993. (to appear).

L. Shastri. Connectionism and the computational effectiveness of reasoning. *Theoretical Linguistics*, 16(1):65–87, 1990.

M. H. van Emden and R. A. Kowalski. The semantics of predicate logic as a programming language. *JACM*, 23(4):733–742, 1976.