Geolog and Skolem Machines

technical notes, spring 2007

John Fisher and Marc Bezem

1 The Geolog Language

Geolog is a language for expressing first-order geometric logic in a format suitable for computations using an *abstract machine*. *Geolog* rules are used as machine instructions for an abstract machine that computes consequences for first-order geometric logic.

A *Geolog* rule has the general form

$$A_1, A_2, \ldots, A_m \Rightarrow C_1; C_2; \ldots; C_n \tag{1}$$

where the A_i are atomic expressions and each C_j is a conjunction of atomic expressions, $m, n \geq 1$. The left-hand side of a rule is called the *antecedent* of the rule (a conjunction) and the right- hand side is called the *consequent* (a disjunction). All atomic expressions can contain variables.

If $n = 1$ then there is a single consequent for the rule (1), and the rule is said to be *definite*. Otherwise the rule is a *splitting rule* that requires a case distinction (case of C_1 or case of C_2 or . . . case of C_n).

The separate cases (disjuncts) C_j must have a conjunctive form

$$B_1, B_2, \ldots, B_h \tag{2}$$

where the B_i are atomic expressions, and $h \geq 1$ varies with j. Any free variables occurring in (2) other than those which occurred free in the antecedent of the rule are taken to be existential variables and their scope is this disjunct (2).

As an example, consider the *Geolog* rule

```
s(X,Y) => e(X,Y) ; dom(Z),r(X,Z),s(Z,Y) .
```

The variables `X,Y` are universally quantified and have scope covering the entire formula, whereas `Z` is existentially quantified and has scope covering the last disjunct in the consequent of rule. A fully quantified first-order logical formula representation of this *Geolog* rule would be

$$(\forall X)(\forall Y)[s(X, Y) \rightarrow e(X, Y) \vee (\exists Z)(dom(Z) \wedge r(X, Z) \wedge s(Z, Y))]$$

Now we come to two special cases of rule forms, the *true* antecedent and the *goal* or *false* consequents. Rules of the form

$$true \Rightarrow C_1; C_2; \ldots; C_n \tag{3}$$

are called *factuals*. Here '*true*' is a special constant term denoting the empty conjunction. Factuals are used to express initial information in *Geolog* theories. Rules of the form

$$A_1, A_2, \ldots, A_m \Rightarrow goal \tag{4}$$

are called *goal* rules. Here '*goal*' is a special constant term. A goal rule expresses that its antecedent is sufficient (and relevant) for *goal*. Similarly, rules of the form

$$A_1, A_2, \ldots, A_m \Rightarrow false \tag{5}$$

are called *false* rules. Here '*false*' is a special constant term denoting the empty disjunction. A *false* rule expresses rejection of its antecedent.

The constant terms *true*, *goal* and *false* can only appear in *Geolog* rules as just described. All other predicate names, individual constants, and variable names are the responsibility of the *Geolog* programmer.

A *Geolog theory* (or *program*) is a finite set of *Geolog* rules. A theory may have any number of factuals and any number of *goal* or *false* rules.

The logical formulas characterized by *Geolog*, and the bottom-up approach to reasoning with those logical formulas, finds its earliest precursor (1920) in a particular paper by Thoralf Skolem [8].

2 Skolem Machines

Geolog theories serve as the instruction set for an abstract *Skolem machine* (SM). Skolem machines resemble multitape Turing machines and the two machine models have actually the same computational power. See the discussion in the last section.

An SM starts with one tape having *true* written on it. The basic operations of an SM use the *Geolog* rules in the instruction set to

- extend a tape (write logical terms at the end)
- create new tapes (for splitting rules)

The tapes are also called *states*. An SM with more than one tape is said to be in a *disjunctive* state, comprised of multiple separate simple states or tapes.

The basic purpose of a particular SM is to compute its instruction set and to halt when all of its tapes have '*goal*' or '*false*' written on them.

In order to motivate the general definitions for the workings of SM, let us work through a small example. To this end, consider the *Geolog* rulebase (SM instructions) in Figure 1.

```
true => domain(X), p(X).                        % #1
p(X) => q(X) ; r(X) ; domain(Y), s(X,Y).   % #2
domain(X) => u(X).                              % #3
u(X), q(X) => false.                            % #4
r(X) => goal.                                   % #5
s(X,Y) => goal.                                 % #6
```

Fig. 1. Sample instructions

The only instruction that applies to the initial tape is instruction #1. The antecedent of the rule matches `true` on the tape, so the tape can be *extended* using the consequent of the rule. In order to extend the tape using `domain(X),p(X)` an instance for the free existential variable `X` is first generated and then substituted, and the resulting terms are written on the tape, as shown in Figure 2.

```
-------------------------------------
 true domain(sk1) p(sk1)
-------------------------------------
```

Fig. 2. After applying rule #1

At this point in machine operation time either of the rules #2 or #3 can apply. The general definition of SM operation does not specify the order, but we will apply applicable rules in top-down order. So, applying instruction #2 we get tape *splitting*, as shown in Figure 3.

Each of the disjuncts in the consequent of rule #2 is used to extend the previous single tape. This requires that the previous tape be copied to two new tapes and then these tapes are extended.

Now, instruction #3 applies to all three tapes, even twice to the last tape, with total result shown in Figure 4.

Instruction #4 now adds `false` to the top tape, shown in Figure 5.

Now instruction #5 applies to the second tape, and then instruction #6 applies to the third tape, shown in Figure 6.

```
--------------------------------------
 true domain(sk1) p(sk1) q(sk1)
--------------------------------------
--------------------------------------
 true domain(sk1) p(sk1) r(sk1)
--------------------------------------
-----------------------------------------------
 true domain(sk1) p(sk1) domain(sk2) s(sk1,sk2)
-----------------------------------------------
```

Fig. 3. After applying rule #2

```
--------------------------------------
 true domain(sk1) p(sk1) q(sk1) u(sk1)
--------------------------------------
--------------------------------------
 true domain(sk1) p(sk1) r(sk1) u(sk2)
--------------------------------------
---------------------------------------------------------------
 true domain(sk1) p(sk1) domain(sk2) s(sk1,sk2) u(sk1) u(sk2)
---------------------------------------------------------------
```

Fig. 4. After applying rule #3 four times (!)

At this point the SM *halts* because each tape has either the term `goal` or the term `false` written on it.

The SM has effectively computed a proof that the *disjunction*

$$(\exists X)(u(X) \wedge q(X)) \vee (\exists X)r(X) \vee (\exists X)(\exists Y)s(X,Y)$$

is a *logical* consequence of the *Geolog* theory consisting of the first three rules in Figure 1. This is so because every tape of the halted machine either has `q(sk1),u(sk1)` written on it or has `r(sk1)` written on it or else has `s(sk1,sk2)` written on it. Note that the three disjuncts correspond to the *goal* and *false* rules in Figure 1. We will continue a discussion of this example (specifically, the role intended for the *false* rule) later in this section.

The *proof tree* displayed in Figure 7 was automatically drawn by the program whose implementation is described in [2]. The diagram displays the tapes generated by the SM in the form of a directed tree. Notice that the tree splits where the SM would have copied a tape. It is possible to describe an SM using trees rather than multiple tapes; see the next section.

```
--------------------------------------------
 true domain(sk1) p(sk1) q(sk1) u(sk1) false
--------------------------------------------
--------------------------------------
 true domain(sk1) p(sk1) r(sk1) u(sk2)
--------------------------------------
---------------------------------------------------------------
 true domain(sk1) p(sk1) domain(sk2) s(sk1,sk2) u(sk1) u(sk2)
---------------------------------------------------------------
```

Fig. 5. Goal tape, rule #4

```
--------------------------------------------
 true domain(sk1) p(sk1) q(sk1) u(sk1) false
--------------------------------------------
-------------------------------------------
 true domain(sk1) p(sk1) r(sk1) u(sk2) goal
-------------------------------------------
-------------------------------------------------------------------
 true domain(sk1) p(sk1) domain(sk2) s(sk1,sk2)  u(sk1) u(sk2) goal
-------------------------------------------------------------------
```

Fig. 6. After applying rule #5 and then #6, HALTED

Definition of Skolem machine operations

- A *Geolog* rule $ANT \Rightarrow CONS$ is *applicable* to an SM tape T, provided that it is the case that all of the terms of ANT can be simultaneously matched against ground terms (no free variables) written on T. (It may be that ANT can be matched against T in more than one way; for example, rule #3 and the third tape of Figure 3.)
- If the rule $ANT \Rightarrow CONS$ is applicable to tape T, then for some matching substitution σ apply σ to $CONS$ and then *expand* tape T using $\sigma(CONS)$.
- In order to *expand* tape T by $\sigma(CONS) = C_1; C_2; \ldots; C_k$ copy tape T making $k-1$ new tapes $T_2, T_3, \ldots, T_k$, and then *extend* T using C_1, extend T_2 using C_2, $\ldots$, and extend T_k using C_k. (No copying if $k = 1$.)
- In order to *extend* a tape T using a conjunction C, suppose that $X_1, \ldots, X_p$ are all of the free existential variables in C. Create new constants c_j, $1 \le j \le p$ and substitute c_j for X_j in C, obtaining C',

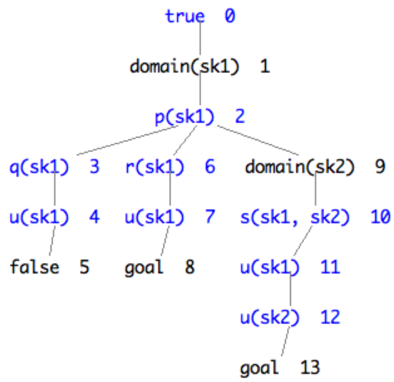

Fig. 7. Tree display

and then write each of the terms of C' on tape T. It is mandatory that the constant is new with respect to the theory and the tape. [1]

Notice that only ground terms ever appear on any SM tape. Thus the matching algorithm does not really need the full power of general term unification. Simple left-to-right term matching suffices.

Given an SM with tapes $T_1, \ldots, T_t$, $t \geq 0$, we say a particular tape T_i is *saturated* if no new instance of a rule can be applied to it.

A tape is *halted* if it is either saturated or contains *goal* or contains *false* (any of which could occur at the same time).

A tape is *halted* if it is either saturated or contains *goal* or contains *false* (any of which could occur at the same time). An SM is called *halted* if all its tapes are halted, it is *halted successfully* if it is halted with all tapes containing either *goal* or *false*. If a tape of an SM is saturated with neither *goal* nor *false* on it, then this tape actually constitutes a countermodel: all rules are satisfied, they are consistent (by absence of *false*) and yet the goal is false (by absence of *goal*).

The set of terms on any saturated tape that is not successfully halted is said to be a *counter model*.

Suppose that we write a *Geolog* theory in the form

$$T = A \cup G \cup F \tag{6}$$

where A is the *axioms*, G contains all of the affirming *goal* rules and F contains all of the rejecting *false* rules. It is intended that A contains all

[1] These witnesses are called Skolem constants by some, but we would prefer to view them as *eigenvariables* in the elimination of existential quantification.

the rules of the theory other than the *goal* rules and the *false* rules and that A, G, and F are mutually disjoint sets.

The *Geolog query* Q for a *Geolog* theory $T = A \cup G \cup F$ is the disjunctive normal form $Q = C_1; C_2; \ldots; C_k$ consisting of all of the conjunctions C_i such that either C_i appears as antecedent of one of the *goal* rules (in G) or of one of the *false* rules (in F). As before, the free variables in Q are taken to be existential variables. The scope of a variable X appearing in a particular C_i (within Q) is restricted to C_i.

We say that a *Geolog* theory T *supports* its query Q if there is a successfully halted SM such that each tape satisfies some C_i.

Theorem 1. If theory T supports its query Q then Q is a logical consequence of the axioms.

Theorem 2. Suppose that Q is the query for Geolog theory G and that Q is a logical consequence of G. Then G supports Q.

Theorem 1 is proved in the next section, as a corollary to more general considerations. Theorem 2 is proved in [4]. The references [1],[4], provide additional theoretical background.

The paper [3] describes algorithms for implementing a Geolog abstract machine, and the website [2] has examples of usage, as well as other topics.

3 Geolog Trees

The splitting of tapes during Skolem machine operations suggests that tree structures can provide an alternate description. This was depicted in Figure 7 and the concept is quite simple.

Suppose that we are given a *Geolog* theory G. There is only one *Geolog tree* with one node, and that is the tree *true*. This singular tree corresponds to the initial tape of a Skolem machine for G.

Assume that we have a correspondence between Skolem machine tape configurations and *Geolog* trees up to some number k of rule applications. If the Skolem machine would have had b tapes then the *Geolog* tree T_k has a total of b branches. Let us examine a $(k+1)$st rule application. This application would have been applied to a particular tape of the Skolem machine. For the Geolog tree, the application is at the leaf of the corresponding branch B of the tree.

Consider again the general form of a *Geolog* rule (1). When such a rule is applied to the current tape it splits into n tapes. The corresponding

Geolog tree branches instead, as visualized in the following diagram. $c_1 = b_1, b_2, \ldots, b_j$ is the extension defined using the Skolem machine operations for the leftmost branch, or the first disjunct of the consequent of the rule. The other new branches (if any) are similarly formed.

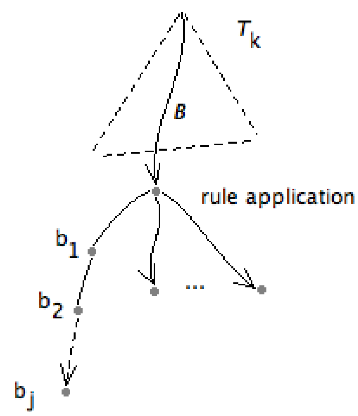

Fig. 8. Branching

By induction, any Skolem machine computation (sequence of operations) can be expressed by a corresponding *Geolog* tree.

Suppose that B is a branch (from root to leaf) in a *Geolog* tree. A *branch conjunction* is any conjunction $b_1, b_2, \ldots, b_i$ of logical terms which appear at the nodes of the tree on branch B. Suppose that $s_1, s_2, \ldots, s_r$ are the distinct eigenvariables appearing in the branch conjunction. Let $b'_1, b'_2, \ldots, b'_i$ be the branch conjunction expression with the eigenvariables replaced by distinct variables $x_1, x_2, \ldots, x_r$ and then form the logical formula $\exists x_1, x_2, \ldots, x_r(b'_1, b'_2, \ldots, b'_i)$. A *branch wff* is any such well-formed logical formula, where the ordering of the logical variables is arbitrary and the ordering of the conjuncts is also arbitrary.

A *tree wff* is any disjunction $c_1; c_2; \ldots; c_p$ such that for any branch B of the *Geolog* tree, one of the c_j is a branch wff for B.

For example, $(\exists x)p(x)$ is a tree wff for the *Geolog* tree in Figure 7, and so is $(\exists x)(q(x), u(x)); (\exists y)u(y); (\exists x, y)s(x, y)$.

Proposition. If w is a tree wff for a Geolog theory then w is a logical consequence of the axioms of the theory.

Proof. The proposition is vacuously true in the case that there are 0 rule applications to build the tree; this is just the tree *true*. Suppose that the proposition is true whenever k rule applications build the tree. Now, assume that $k+1$ rule applications built our tree. The last rule application can again be depicted as in Figure 8. Consider a tree wff w for this tree. If any branch wff of w comes from branch B in tree T_k, then w is a logical consequence of the axioms by the induction hypothesis (ignore the application of the last rule). Otherwise, the branch wffs in the expansion below T_k all come from the consequent of the rule application. Let us write the instance of the expansion rule as

$$a_1, a_2, \ldots, a_n \Rightarrow c_1; c_2; \ldots; c_m \tag{7}$$

Now, the branch wffs of w below T_k are a logical consequence of the consequent of (7). Why? And the facts $a_1, a_2, \ldots, a_n$ occur along B. Let $w = w_{old}; w_{new}$ express the tree wff w as two parts: the w_{old} part comes from tree T_k (but not along B), and the w_{new} part comes from the expansion of the tree. By induction, the tree wff $w_{old}; (a_1, a_2, \ldots, a_n)$ is a logical consequence of the axioms of the theory, since this is a tree wff for T_k before expansion. Thus $w_{old}; c_1; c_2; \ldots; c_m$ is a logical consequence of the axioms. And, therefore $w = w_{old}; w_{new}$ is a logical consequence of the axioms, as claimed. □

Notice that the set of all propositions along any branch of the tree is a model for a tree wff. Call these models *branch models* of the tree wff.

Proof of Theorem 1. Suppose that the theory supports its query Q. Then Q is a tree wff for the *Geolog* tree corresponding to the halted Skolem machine. By Proposition 1, Q is a logical consequence of the axioms of the theory. □

The reference [4] defines *complete Geolog trees* and uses them to prove Theorem 2. Roughly speaking, complete trees require that all applicable rules be used to expand trees in stages.

References

1. Marc Bezem. On the Undecidability of Coherent Logic. In Aart Middeldorp e.a., editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, LNCS 3838, pages 6–13, Springer-Verlag, Berlin, 2005.
2. *Geolog* website: `http://www.csupomona.edu/~jrfisher/www/geolog`
3. John Fisher and Marc Bezem, Skolem Machines and Geometric Logic. In C.B. Jones, Z. Liu and J. Woodcock, *Proc. ICTAC 2007 The 4th International Colloquium on Theoretical Aspects of Computing*, Macao SAR, China, September 26-28, 2007. Springer LNCS vol. 4711, pp. 201-215.

4. John Fisher and Marc Bezem, Query Completeness of Skolem Machine Computations. In J. Durand-Losé and M. Margenstern, editors, *Proc. Machines, Computations and Universality '07*, Universite d'Orleans - LIFO, Orleans, France September 10-14, 2007. Springer LNCS vol. 4664, pp. 182-192.
5. Jacques Herbrand, *Logical Writings*, edited by Warren D. Goldfarb, D. Reidel Publishing Company, Springer edition, 2006.
6. Dénes König, *Theorie der endlichen und unendlichen Graphen, Akademische Verlagsgesellschaft*, Leipzig, 1936. Translated from German by Richard McCoart, *Theory of finite and infinite graphs*, Birkhauser, 1990.
7. Marvin L. Minsky, Recursive unsolvability of Post's problem of 'tag' and other topics in theory of Turing machines, *Annals of Mathematics* **74**(3):437–455, 1961.
8. Thoralf Skolem, *Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit und Beweisbarkeit mathematischen Sätze nebst einem Theoreme über dichte Mengen*, Skrifter I **4**:1–36, Det Norske Videnskaps-Akademi, 1920. Also in Jens Erik Fenstad, editor, *Selected Works in Logic by Th. Skolem*, pp. 103–136, Universitetsforlaget, Oslo, 1970.