Take-home exam for the course in geometric logic
Tallinn, 17–23 February 2005

General instructions for using the GL prover in the directory GL

1. The command `./gl` launches SWI-Prolog with the GL prover loaded.
2. There are basically five commands that you can use after the `?-` prompt:
   - `test(<file>)` tries to prove the proposition goal from the axioms in `<file>.in`.
   - `out(<file>)` does the same as `test(<file>)` but writes to `<file>.out`.
   - `prf(<file>)` reconstructs a proof from the log and writes this to `<file>.prf`.
   - `coq(<file>)` generates a Coq script from `<file>.prf` and writes to `<file>.v`.
   - `run(<file>)` does `out`, `prf` and `coq` with `<file>` and lists axioms that are not used.
3. You can check the proofs by the command `coqc <file>.v` after the Unix prompt. This creates `<file>.vo` which could be used in later Coq developments.

Instructions for preparing input files

1. If you want to prove $p \wedge q$, the first axiom should be:

```
_ axiom <name>:(p,q => goal).
```

2. If you want to prove $p \vee q$, the first two axioms should be:

```
_ axiom <name1>: (p => goal).
_ axiom <name2>: (q => goal).
```

   Proving that a theory is inconsistent goes by proving goal WITHOUT axioms as above.

3. If you want to prove that there exists an x such that $p(x)$, the first axiom should be:

```
_ axiom <name>(X): (p(X) => goal).
```

   Note that in Prolog variables start with an upper case letter.

4. If you want to prove that for all x with $p(x)$ there exists a y such that $q(x, y)$ the first three axioms should be (NB! a must be a new constant):

```
dom(a).
_ axiom <name1>: (true => p(a)).
_ axiom <name2>(Y): (q(a,Y) => goal).
```

5. For proving an arbitrary geometric formula you must craft some combination of 1-4.
6. Recall that the search is depth-first so that a reasonable ordering is: first the Horn clauses, then the disjunctive axioms and finally axioms like seriality. This should work for most (all?) of the exam exercises.
7. The general format of an input clause is:

```
<control> axiom <name>(<variables>): (<conj. of atoms> => <geom. disj.>).
```

The `<variables>` are the universal variables of the axiom. Use Prolog's disjunction `;`. Universal variables must occur in the premiss (range restriction), if necessary use the domain predicate `dom`. For example, reflexivity should NOT be

```
_ axiom refl(X): (true => r(X,X))
```

but

```
_ axiom refl(X): (dom(X) => r(X,X)).
```

Right of `=>` the domain predicate is used to express existential quantification. Thus, e.g., for all x we have p or there exists a y such that $q(x, y)$ should be:

```
_ axiom <name>(X): (dom(X) => p;dom(Y),q(X,Y)).
```

8. Predicates you use should be declared with their arity, e.g.,

```
:- dynamic p/0,q/2.
```

The pragmatics of the GL prover

1. For small problems the following Law of Nature for ATP seems to be valid: Either the problem is solved in 100 milliseconds, or it is not solved at all.
2. Of a complete run, test and coq take each 10% of the CPU time and prf the rest.
3. Strategies can be programmed with the binary predicate `enabled(<control>,<guard>)` and the ternary predicate `next(<control>,<guard>,<new_guard>)`. This requires some skill. It may cause false negatives and even runtime errors. False positives have not happened in practice, but could happen in principle, f.e., after `assert(false)` when the search has started. False positives NEVER EVER pass the typecheck of Coq. :-)
4. Launch `grep enabled *.in` to see which examples use strategies and how. Never use a strategy when you do not understand why the GL prover doesn't find the proof.
5. Ignore warnings about redefined static procedures.

Exam exercises (deadline 24 March 2005)

1. Run `or.in`, `exist.in` and `drinker.in` and understand the `.out` files completely.
2. Prove p in the geometric theory $(\exists x\, q(x)) \vee p$, $q(x) \rightarrow p$.
3. Prove that a transitive and symmetric relation r satisfies $r(x, y) \rightarrow r(x, x)$.
4. The file `dpe.in` expresses that the diamond property is preserved under reflexive closure of the rewrite relation, that is, if r satisfies the axiom *dp_r* then *re* defined by *e_in_re*, *r_in_re*, *e_or_r* also satisfies the diamond property. Run `dpe.in` and understand the `.out` file completely. Explain why the proof is far from optimal and give a better proof by human intelligence.
5. Prove that a relation r which satisfies the diamond property is weakly serial in the sense that $r(x, y) \rightarrow (\exists z\, r(y, z))$.
6. Reduce the classical Peirce $((p \rightarrow q) \rightarrow p) \rightarrow p$ to GL and prove it.
7. Reduce to GL and prove: $(\forall x(\exists y(p(x) \vee q(y)))) \rightarrow (\exists y(\forall x(p(x) \vee q(y))))$.

8. Let lt be transitive and strongly serial in the sense that $\exists y\, r(x, y)$. Show by reduction to GL that defining $p(x) \leftrightarrow \forall y(lt(x, y) \rightarrow \neg p(y))$ leads to a contradiction if the domain is non-empty.

9. Simulate in GL a register machine for subtraction for natural numbers like it is done for addition on p. 6 of [Bezem, Completeness and undecidability of geometric logic]. Prove some instances $P_{halt}(m, n)$ of moderate size.

10. Run in GL all the examples that are explicitly mentioned in [Manthey, Bry, SATCHMO, Proc. of CADE-9, 1988] (so not those they refer to). Where possible, restore the skolemization with proper existential quantifiers.