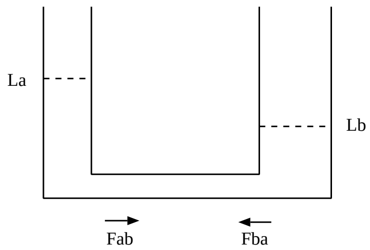
Learning qualitative models

Qualitative models are often better suited for several tasks than traditional quantitative (numeric) methods. These tasks include diagnosis, generating explanations of a system's behaviour and designing novel devices from first principles. A more fundamental problem in the theory of dynamic systems is the identification of the system itself. The basic problem is as follows. We are given examples of the behaviour of a dynamic system. We want to find a model that explains these examples. Accomplishing this, even at the qualitative level, is quite difficult. Consider the simple system with two connected containers



Even this simple system can have almost almost 195,000 possible qualitative states when using a QSIM formalism. Very few (only 32) of these are actually "legal". The problem here is to learn rules that identify legal states. These rules, along with built-in assumptions of continuity in QSIM, are sufficient to identify the dynamics of the system. The data here are as used in two different experiments. The first concerns learning rules for legal states using the ILP program *Golem* [3]. This requires determinate, ground background knowledge, and results are reported in [1]. The information used for this task is:

- The target predicate is the relation **state(La,Lb,Fab,Fba)**, where **La**, **Lb**, **Fa** and **Fb** are as shown in the figure.
- The background knowledge is a tabulation of QSIM constraint primitives.
- Consistency of infinite values in the **add** constraint can be ignored for this problem. Further, since **add** requires a very large number of ground facts, it is represented by three more efficient predicates. **norm_mag/4** normalises a qualitative value with respect to a landmark. **lookup_consist_table/3** is a lookup table for adding signs. **verify_add_deriv/3** os a lookup table for adding directions of derivatives.

- Additional background predicates provide **range** information. The QSIM constraints **m_plus/2** and **m_minus/2** were only tabulated for empty lists of corresponding values. All arguments of the background predicates are of the same type; they are compound terms of the form

$$\mathrm{Func} : \mathrm{QualValue/DirOfChange}$$

The more recent data uses the *Progol* program [4]. This does not require the background to be determinate or ground. Consequently, background knowledge consists of QSIM predicate definitions [2].

All data used in the *Progol* experiments are contained in a single compressed file with a ".pl.Z" suffix.

The data used in the original *Golem* experiments are contained in a compressed TAR file with a "tar.Z" suffix. Within this, the background knowledge files have a ".b" suffix, positive example files have a ".f" suffix, and negative example files have a ".n" suffix.

References

[1] Bratko I., Muggleton S. and Varsek A. (1992). Learning Qualitative Models of Dynamic Systems. In S. Muggleton editor. *Inductive Logic Programming*. Academic Press, London.

[2] Kuipers B. (1986). Qualitative simulation. *Artificial Intelligence*, 29:289–338.

[3] Muggleton S.H. and Feng C. (1990). Efficient induction of logic programs. *Proceedings of the First Conference on Algorithmic Learning Theory*, Tokyo.

[4] Muggleton S.H. (1995) Inverse Entailment and Progol. *New Gen. Comput.* (to appear).