HyLoRes: A hybrid logic prover based on direct resolution

Carlos Areces Juan Heguiabehere

Faculty of Science. University of Amsterdam
{carlos,juanh}@science.uva.nl

The System: HyLoRes is a direct resolution prover for hybrid logics implementing the sound and complete algorithm for satisfiability of sentences in $\mathcal{H}(@,\downarrow)$ presented in [2]. The most interesting distinguishing feature of HyLoRes is that it is not based on tableau algorithms but on (direct) resolution. HyLoRes implements a version of the "given clause" algorithm, which has become the skeleton underlying most first-order provers.

Purpose of the System: Hybrid languages are modal languages that allow direct reference to the elements of a model. The basic hybrid language ($\mathcal{H}(@)$) extends the basic modal language simply by the addition of a new set of atomic symbols called *nominals* (usually denoted as $i, j, k, \ldots$) which name particular points in the model (i.e., the interpretation of a nominal i in a model $\mathcal{M} = \langle W, R, V\rangle$ is an element $i^{\mathcal{M}} \in W$), and for each nominal i a *satisfiability operator* $@_i$. This extension increases the expressive power of the language as we can now explicitly check whether the point of evaluation w is the specific point named i in the model, $\mathcal{M}, w \Vdash i$ iff $w = i^{\mathcal{M}}$; and from any point in the model we can check whether a point named i satisfies a given formula φ, $\mathcal{M}, w \Vdash @_i\varphi$ iff $\mathcal{M}, i^{\mathcal{M}} \Vdash \varphi$.

The extended expressivity allows one to define elegant decision algorithms, where nominals and $@$ play the role of labels, or prefixes, which are usually needed during the construction of proofs in the modal setup [3, 2]. Note that they do so *inside* the object language. All these features we get with no increase in complexity: the complexity of the satisfiability problem for $\mathcal{H}(@)$ is the same as for the basic modal language, PSPACE.

When we move to very expressive hybrid languages containing binders, we obtain an impressive boost in expressivity, but usually we also move beyond the boundaries of decidability. Classical binders like $\forall$ and $\exists$ make the language as expressive as first-order logic (FOL) while the language $\mathcal{H}(@,\downarrow)$ which includes the more "modal" binder $\downarrow$ gives a logic weaker than FOL (but still undecidable) [1]. See the Hybrid Logic site at `http://www.hylo.net` for a broad on-line bibliography.

In recent years, an important number of theoretical results concerning axiomatizability, proof systems (tableaux, natural deduction, etc), interpolation, expressive power, complexity, etc. for hybrid logics has been obtained. The next natural step was to develop provers that can handle these languages.

User Groups: HyLoRes is a modal and hybrid prover, but it is not yet meant to be competitive when compared with state of the art provers for modal and description logics like DLP, FaCT, MSPASS, or RACER (see [4, 5]). On the one hand, the system is still in a preliminary stage of development (only very simple optimizations for hybrid logics have

been implemented), and on the other hand hybrid and description languages are related but different. $\mathcal{H}(@,\downarrow)$ is undecidable while the implemented description languages are mostly decidable. And even when comparing the fragment $\mathcal{H}(@)$ for which HyLoRes implements a decision algorithm, the expressive powers are incomparable ($\mathcal{H}(@)$ permits free Boolean combinations of $@$ and nominals but lacks the limited form of universal modality available in the T-Box of DL provers). A planned extension for future versions is to extend the language with the universal modality A which would give us full FOL expressivity. The language $\mathcal{H}(\mathsf{A})$ is particularly interesting as it lets us perform inference in terms of full Boolean knowledge bases of the description logic $\mathcal{ALCO}$ in HyLoRes [1].

Modal, hybrid, description and first-order logic researchers are potential users of the prover.

System Developed at: The Language and Inference Technology Group. University of Amsterdam. Amsterdam, The Netherlands. Research partly supported by the NWO project #612.069.006.

Developers: Carlos Areces and Juan Heguiabehere.

Technology Used and System Requirements: HyLoRes is implemented in Haskell, and compiled with the Glasgow Haskell Compiler (GHC) Version 5.02, generating executable code. GHC generates fairly efficient C code which is then compiled into an executable file. Thus, users need no additional software to use the prover. The HyLoRes site (`http://www.illc.uva.nl/~carlos/~HyLoRes`) provides executables for Solaris (tested under Solaris 8) and Linux (tested under Red Hat 7.0 and Mandrake 8.0). The original Haskell code is also made publicly available under the GNU General Public License (GPL). We will soon provide also the intermediate C source which could then be compiled under a wider range of platforms. In addition to HyLoRes, a graphical interface called xHyLoRes implemented in Tcl/Tk was developed. It uses HyLoRes in the background and provides full file access and editing capabilities, and a more intuitive control of the command line parameters of the prover.

Conclusions: There remain many things to try and improve in HyLoRes, but the main goal of this first prototype has been largely achieved: direct resolution can be used as an interesting, and perhaps even competitive, alternative to tableaux based methods for modal and hybrid logics.

References

[1] C. Areces. *Logic Engineering. The Case of Description and Hybrid Logics*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, The Netherlands, October 2000.

[2] C. Areces, H. de Nivelle, and M. de Rijke. Resolution in modal, description and hybrid logic. *Journal of Logic and Computation*, 11(5):717–736, 2001.

[3] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, 10(1):137–168, 2000.

[4] E. Franconi, G. De Giacomo, R. MacGregor, W. Nutt, and C. Welty, editors. *Proceedings of the 1998 International Workshop on Description Logics (DL'98)*, 1998.

[5] P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors. *Proceedings of the 1999 International Workshop on Description Logics (DL'99)*, 1999.