RACE System Description

Volker Haarslev and Ralf Möller
University of Hamburg, Computer Science Department
Vogt-Kölln-Strasse 30, 22527 Hamburg, Germany
`http://kogs-www.informatik.uni-hamburg.de/~<name>/`

Language

RACE [3] is a successor of HAM-ALC [5, 2]. Based on sound and complete algorithms RACE currently implements TBox and ABox reasoning for the description logic $\mathcal{ALCNH}_{R^+}$ [4] that supports number restrictions, role hierarchies, and transitively closed roles. Note that this DL implies general concept inclusions as a language feature. RACE accepts TBoxes and ABoxes in the KRSS syntax with appropriate extensions for GCIs.

Other Features

RACE additionally provides a Web interface based on CL-HTTP, a Common Lisp Hypermedia Server.

Implementation

RACE employs standard (exploitation of told subsumers/subsumees by marking and caching operations for computing the subsumption hierarchy, lazy unfolding) and advanced optimization techniques (semantic branching, dependency-directed backtracking, taxonomic encoding, GCI absorption) in analogy to [6] as well as deep model merging and extensive model caching. It also exploits new optimization techniques for ABox reasoning [3]. The programming language is ANSI Common Lisp.

Performance

RACE is one of the fastest systems for testing concept consistency and outperforms any known ABox reasoning system by several orders of magnitude. For details on a performance evaluation regarding ABox reasoning see [3]. The tests presented in the Appendix were performed using RACE version 1.0 on a Macintosh G3 Powerbook (266 MHz) with Macintosh Common Lisp 4.2 in a 80 MB memory partition. If a *size* attribute is specified for the problem, the runtime is given for the last size that can be solved within the specified timeout limit.

Availability

The system RACE will be presented at DL'99. It will be available from the authors' home pages.

Future Plans

We plan to extend RACE to additionally supporting ABox reasoning with qualified number restrictions and inverse roles. Future versions of RACE will also include more advanced optimization techniques for the inference algorithms dealing with number restrictions.

References

[1] E. Franconi et al., editors. *Proceedings of the International Workshop on Description Logics (DL'98), June 6-8, 1998, Trento, Italy*, June 1998.

[2] V. Haarslev and R. Möller. Applying an $\mathcal{ALC}$ ABox consistency tester to modal logic SAT problems. In Neil V. Murray, editor, *Proceedings, International Conference on Automatic Reasoning with Analytic Tableaux and Related Methods, TABLEAUX'99, Saratoga Springs, NY, USA*, number 1617 in Lecture Notes in Artificial Intelligence, pages 24–28. Springer Verlag, Berlin, June 1999.

[3] V. Haarslev and R. Möller. An empirical evaluation of optimization strategies for ABox reasoning in expressive description logics. In *Proceedings, International Workshop on Description Logics, DL'99, Linköping, Sweden*, 1999. In press.

[4] V. Haarslev and R. Möller. Expressive ABox reasoning with number restrictions, role hierarchies, and transitively closed roles. Technical report, University of Hamburg, Computer Science Department, 1999. In preparation.

[5] V. Haarslev, R. Möller, and A.-Y. Turhan. HAM-ALC. In Franconi et al. [1], pages 64–65. Benchmark results for DL'98 comparison.

[6] I. Horrocks and P. Patel-Schneider. Optimising description logic subsumption. *Journal of Logic and Computation*, 9(3), 1999. In press.

Table 1: Tableaux'98 Concept Satisfiability Tests (timeout 100s, S=Size, C=Correct)

	Incoherent			Coherent		
Test	**S**	**C**	**Time**	**S**	**C**	**Time**
k_branch	16	Y	61.35s	11	Y	65.48s
k_d4	21	Y	0.82s	21	Y	3.87s
k_dum	21	Y	0.07s	21	Y	0.07s
k_grz	21	Y	0.08s	21	Y	0.17s
k_lin	21	Y	0.02s	21	Y	0.34s
k_path	21	Y	2.07s	21	Y	2.76s
k_poly	21	Y	2.92s	21	Y	3.27s
k_t4p	21	Y	0.64s	21	Y	0.80s
k_ph	7	Y	26.02s	16	Y	73.53s

Table 3: Tableaux'98 Abox Realisation Tests (timeout 1000s, Con=Concepts, S=Size, C=Correct)

Test	Con	S	C	Time
k_branch_n	71	1	Y	0.16s
k_d4_n	48	1	Y	0.23s
k_dum_n	71	1	Y	0.08s
k_grz_n	109	1	Y	0.10s
k_lin_n	10	1	Y	0.06s
k_path_n	91	1	Y	3.21s
k_ph_n	7	1	Y	0.04s
k_poly_n	66	1	Y	1.44s
k_t4p_n	72	1	Y	2.26s

Table 4: New Abox Realisation Tests (timeout 500s, MS=Max Size, S=Size, T=Time, IC=Instance Checking)

		IC		Realisation	
Test	**MS**	**S**	**T**	**S**	**T**
k_branch_n	4	4	57s	4	355s
k_d4_n	4	4	114s	4	514s
k_dum_n	21	21	55s	19	508s
k_grz_n	10	10	4s	9	210s
k_lin_n	10	10	2s	9	497s
k_path_n	4	4	218s	3	423s
k_ph_n	7	6	6s	6	135s
k_poly_n	8	8	151s	5	416s
k_t4p_n	5	5	450s	3	228s

Table 5: Application KB Tests (timeout 10000s, Con=Concepts, C=Correct)

Test	Con	Time	C
ckb-roles	79	0.17	Y
datamont-roles	120	0.22	Y
espr-roles	142	0.24	Y
fss-roles	132	0.52	Y
wisber-roles	140	0.37	Y
galen2	3,926	27.54	Y
galen1	2,728	35.86	Y
galen	2,749	72.77	Y
espr-gcis	143	12.36	Y
wisber-gcis	140	4.41	Y
ckb-gcis	79	13.38	Y
fss-gcis	132	23.14	Y

Table 6: Synthetic KB Tests (timeout 500s, Con=Concepts, C=Correct)

Test	Con	Time	C
hc14	10	0.02	Y
hc18	18	0.06	Y
hc112	26	0.28	Y
hc24	46	0.07	Y
hc28	94	0.32	Y
hc212	142	1.58	Y
hc34	18	0.11	Y
hc36	26	0.56	Y
hc38	34	5.60	Y
hc44	7	0.03	Y
hc48	7	0.08	Y
hc412	7	0.16	Y

Table 2: Tableaux'98 KB SAT Tests (timeout 500s, S=Size, Con=Concepts, C=Correct)

	Incoherent				**Coherent**			
Test	**S**	**Con**	**C**	**Time**	**S**	**Con**	**C**	**Time**
k_branch	5	731	Y	341.54s	5	725	Y	231.42s
k_d4	21	2,121	Y	388.37s	16	1,728	Y	454.32s
k_dum	21	585	Y	3.42s	21	650	Y	9.62s
k_grz	21	1,330	Y	24.37s	21	1,349	Y	24.80s
k_lin	21	934	Y	1.88s	21	2,379	Y	181.87s
k_path	13	1,989	Y	493.13s	11	1,685	Y	474.57s
k_poly	7	500	Y	259.87s	7	534	Y	495.98s
k_t4p	21	637	Y	23.68s	21	1,192	Y	198.86s
k_ph	6	442	Y	471.88s	6	442	Y	137.16s

Table 7: Random ALCN KB Tests (timeout 500s, Con=Concepts, C=Correct)

Test	**Con**	**Time**	**C**
kris151	16	0.03	Y
kris301	31	0.06	Y
kris451	46	0.10	Y
kris601	61	0.16	Y
kris751	76	0.18	Y
kris901	91	0.27	Y
kris1051	106	0.29	Y
kris1201	121	0.33	Y
kris1351	136	0.45	Y
kris1501	151	0.52	Y
kris2001	201	0.65	Y
kris4001	401	2.44	Y
kris6001	601	4.57	Y
kris8001	801	5.61	Y
kris10001	1,001	9.27	Y
kris12001	1,201	13.81	Y
kris14001	1,401	13.22	Y
kris16001	1,601	22.14	Y
kris18001	1,801	26.03	Y
kris20001	2,001	36.21	Y
kris25001	2,501	50.16	Y
kris30001	3,001	55.58	Y
kris35001	3,501	72.84	Y
kris40001	4,001	96.45	Y
kris45001	4,501	136.26	Y
kris50001	5,001	149.66	Y

Table 8: TABLEAUX'99 tests (timeout 600s, time in seconds)

Test	**Time**
p-bound-cnf-K3-C8-V4-D2	0.01
p-bound-cnf-K3-C16-V4-D2	0.03
p-bound-cnf-K3-C32-V4-D2	0.06
p-bound-modK-K3-C8-V4-D2	0.03
p-bound-modK-K3-C16-V4-D2	0.06
p-bound-modK-K3-C32-V4-D2	0.15
p-unbound-qbf-K4-C8-V2-D3	0.35
p-unbound-qbf-K4-C16-V2-D3	0.34
p-unbound-qbf-K4-C32-V2-D3	0.39
p-unbound-qbf-modK-K4-C8-V2-D3	0.21
p-unbound-qbf-modK-K4-C16-V2-D3	0.30
p-unbound-qbf-modK-K4-C32-V2-D3	0.41
p-unbound-qbf-modS4-K4-C8-V2-D3	1.19
p-unbound-qbf-modS4-K4-C16-V2-D3	1.79
p-unbound-qbf-modS4-K4-C32-V2-D3	3.10
p-bound-cnf-modS4-K3-C8-V4-D2	117.43
p-bound-cnf-modS4-K3-C16-V4-D2	308.91
p-bound-cnf-modS4-K3-C32-V4-D2	592.83
p-persat-K4-C8-V4-D2	0.01
p-persat-K4-C16-V4-D2	0.10
p-persat-K4-C32-V4-D2	1.60
p-persat-modK-K4-C8-V4-D2	1.27
p-persat-modK-K4-C16-V4-D2	2.63
p-persat-modK-K4-C32-V4-D2	0.75
p-persat-modS4-K4-C8-V4-D2	385.41
p-persat-modS4-K4-C16-V4-D2	378.31
p-persat-modS4-K4-C32-V4-D2	56.39