aRa
An Automatic Theorem Prover for Relation Algebras

Carsten Sinz
Symbolic Computation Group, WSI for Computer Science,
Universität Tübingen, Germany
sinz@informatik.uni-tuebingen.de

June 2000

aRa is an automatic theorem prover for various relation algebras and for finite-variable first-order predicate logic without function symbols. It allows proofs in the following theories:

- Semi-associative Relation Algebras
- (Tarski's) Relation Algebras
- Representable Relation Algebras
- n-variable First-Order Predicate Logic

Special Features

Complete Relation Algebraic Input Language:

symbols	ARA notation	comment
$+$ $\vee$	`+`	absolute sum (Boolean addition, disjunction)
$\cdot$ $\wedge$	`*`	absolute product (Boolean multiplication, conjunction)
$\dagger$ $\oplus$	`#`	relative sum (Peircean addition)
$;$ $\odot$	`@`	relative product (Peircean multiplication, composition)
$^{-}$	`~`	complementation (Boolean negation)
$^{\smile}$	`^`	conversion
$\mathring{1}$ $1'$	`i`	relative unit (identity element, Peircean unit)
$\mathring{0}$ $0'$	`~i`	relative zero (diversity relation)
$1, \top$	`1`	absolute unit (truth)
$0, \bot$	`0`	absolute zero (falsity)
$\leq$	`<`	relational inclusion
$=$	`=`	relational equivalence
$\vdash$	`|-`	turnstile

Different Reduction Strategies:

1. LP (literal pair): Selects a pair of complementary literals that can be disposed of by a series of reduction steps. In order to find such pairs, an equation system is set up (similar to unification) that is solvable iff there is a reduction sequence that moves the literals (or one of their descendants) into a common disjunction. Then, RPC reductions are selected according to the equation system to make the literal pair vanish. Selection of RPC rules is divided into three phases: pre-processing, main unification process and post-processing.
2. ORP (oldest reduction possibility): While successively constructing the reduction chain, the first appearance of each reduction possibility is recorded; later-on, changes performed on this reduction possibility are tracked. In each step one of the oldest reductions is chosen.

Special Simplification Rules:

1. Give priority to shortening rules.
2. Remove quantifiers that bind no variables.
3. Minimize quantifier scopes.
4. Subgoal generation: To prove $F \wedge G$ prove first F, and then G.
5. Try to use all available variables as soon as possible.
6. Delete pure literals.

ARA is available as binary distribution for Solaris on SPARC and Windows NT on Intel processors.

`www-sr.informatik.uni-tuebingen.de/~sinz/ARA`
