CSE 3302 Notes 12: Logic Programming

(Last updated 12/3/15 1:18 PM)

References:

Gabbriell-Martini: 12

Answer Set Programming: `http://dl.acm.org.ezproxy.uta.edu/citation.cfm?doid=2043174.2043195`

THE TOP-DOWN/DEPTH-FIRST LEGACY OF PROLOG (`https://www.gprolog.org`)

J.A. Robinson (`https://en.wikipedia.org/wiki/John_Alan_Robinson`)

Resolution theorem provers (automated deduction)

First-Order Unification

(Boyer-Moore systems: `https://en.wikipedia.org/wiki/ACL2` `http://www.amazon.com/dp/0262527952`)

Robert Kowalski: Algorithm = Logic + Control (`http://dl.acm.org.ezproxy.uta.edu/citation.cfm?doid=359131.359136`)

David H. D. Warren: Successful commercial implementation (`https://en.wikipedia.org/wiki/David_H._D._Warren`)

The Book: `http://www.amazon.com/dp/3540006788/`

(An early effort at LP with constraints: `http://dl.acm.org.ezproxy.uta.edu/citation.cfm?doid=129393.129398`)

Unification - A Basic Pattern Matching/Binding Computation in Symbol Manipulation (G.-M. 12.3.)

`http://dl.acm.org.ezproxy.uta.edu/citation.cfm?doid=62029.62030`

`http://www.amazon.com/dp/0521779200/` (Term Rewriting)

Examples:

G.-M. 12.1.1 (`list_of_27`)

mergesort

(For those who take CSE 4331: `http://dl.acm.org.ezproxy.uta.edu/citation.cfm?doid=5948.5952`)

REBIRTH THROUGH ANSWER SET PROGRAMMING INTERFACES FOR SAT SOLVERS

Emphasis is on propositional logic, not first-order predicates

The Beginning: `https://en.wikipedia.org/wiki/DPLL_algorithm`

The State of the Art (for SAT): `http://www.satcompetition.org`

ASP solvers usually use *grounding* as a front-end.

General overview: `https://en.wikipedia.org/wiki/Answer_set_programming`

Significant examples: `http://potassco.sourceforge.net/clingo.html`