

Automata and Logic in Verification

CS598MP: Fall 2005

Tuesday & Thursday: 1530–1645

P. Madhusudan

University of Illinois at Urbana-Champaign

25th August 2005

What is logic?

- *(1) : a science that deals with the principles and criteria of validity of inference and demonstration : the science of the formal principles of reasoning*
- **Wikipedia:**
Formal logic is the study of logical inference whose validity derives from its explicitly formal structure.
- Key aspects:
 - Syntax determines meaning
 - Syntax determined reasoning
 - Logic as a language (naturalness in expressing thoughts)

Gödel's completeness theorem
Gödel's incompleteness theorem

- A means to specify properties of system paths
 - “All runs of the program must be such that every lock acquired is eventually released unless there is a way to exit all critical regions.”
 - “The scheduler has a chance to finish all tasks at hand if the agents co-operate; if the agents don't co-operate”
 - “No request must be left pending forever.”

- A means to specify properties of program effects
 - “If this module gets a list of numbers, it will return with a sorted list”
 - Pre-post conditions:
“If $x > 0$ and $y > 5$ and z is not a null pointer, the program will exit with $x = y$ and z assigned to the address of x ”
 - Hoare logic

- A means to abstracting programs
 - “If $x > 30$ and $y < 15$ is true, then after the statement $x := x - y$, will $x > 15$ be always true?”
 - p and q are two propositions.
 s is a statement of my program.
What combinations of p and q will assure me that after s , p is true and q is false?

- A means to reasoning about data structures
 - “The list is sorted”
 - “The list is acyclic”
 - “If I take an acyclic list and do the following operations, the list will remain acyclic”
 - “In the i 'th iteration, the first i elements of the list are sorted”.

- A means to reason about symbolic paths
 - Let's take a path in the program:
 $x := 2x + 5y;$
if $(x > y)$ then:
if $(2x < y)$ then:
*:
Is this path feasible? If it is, what values must x and y take?
 - What about program paths that handle dynamic data (say lists)? How can I give an input list structure that will lead to one of the paths being taken?

The key common aspects are

- What are the appropriate logics for phrasing properties of program traces, structures, conditions/predicates on variables?
- Which logics have a decidable satisfiability problem?
Which logics have a decidable model checking problem?
- How do I reason with program statements?
Can I infer formulas that capture effects of program statements?
- What's the complexity of reasoning with various logics?
- Assume I have 'smart' theorem provers that can tackle many theories. Can I combine these theories in a concrete/decidable fashion? (Eg. Arrays of numbers.)

- Logic on labeled word structures (FOL and MSOL)
- What logic describes regular languages?
- Büchi-Elgot-Trachtenbrot theorem
- Infinite word structures
- Automata on ω -words (properties, complementation)
- MSO, Büchi's theorem
- Linear-time temporal logic (LTL \rightarrow Automata, decidability)
- Model-checking MSO and LTL
- Determinizing Büchi automata

- Logics on trees (MSO)
- Regular trees; tree automata
- Automata and games
- Automata on infinite trees (Büchi, Rabin, parity)
- Properties (closure theorems, deciding emptiness, etc.)
- MSO, Rabin's theorem
- Linear-time temporal logic (LTL->Automata, decidability)
- Model-checking MSO and LTL

- Logics on structures; CTL, μ -calculus
- μ -calculus, parity games and MSO
- Model checking CTL/ μ -calculus

- Logics on fixed structures; decidable logics
- Logics on reals (decidability)
- Pressburger arithmetic
- Decidable logical structures: interpretability among theories
- Pushdown graphs, graphs of fixed tree-width, Courcelle's theorem
- Combining decision procedures

- Logic and computability
- Logical characterization of NP and P
- Intro to finite model theory