SAT Solvers

Ron Parr
CPS 570

Given a “good” SAT solver...

• How do you use it?
• Convert other NP-complete problems to SAT
• Use your “pretty good” SAT solver for them
• What’s the best you could hope for?

A silver bullet for SAT?

• Unless P=NP, you won’t be able to solve every problem efficiently, so
  – Either will be very slow in some cases, or
  – Will completely fail in some cases
• How do we evaluate the quality of a “pretty good” SAT solver?
  – Try on lots of instances?
  – What is a representative set of instances?

Phase Transitions

• Underlying question: What percentage of assignments are satisfying?
• Phase transition is a sharp transition from easy (lots of solutions) to hard (few or no solutions) as some function of problem description
• How does this relate to the hardness of finding a solution?
• How does this relate to the hardness of predicting the solution?
Things to consider when converting another problem to SAT

- What is the cost of doing the conversion (space/time)?
- Does the conversion preserve easy/hard instances?
- After conversion, is your fast SAT solver better than another “pretty good” method in the original problem description?

Modeling How Algorithms Perform

- This idea of modeling how algorithms perform seems neat; can we use it to our advantage?
- Choosing how to allocate computational resources is called meta-computation
- Challenges and benefits:
  - There can be a lot of overhead in doing this
  - In some cases, you can get a speedup on average
- See: http://www.cs.duke.edu/~parr/aaaifs01.ps.gz

Randomization?

- How does randomization help?
- Alternatively: How is it that a randomized algorithm can do better than a systematic algorithm?
- How do we analyze the complexity of a random algorithm?

Can people do better?

- Is there evidence that people can solve NP-hard problems efficiently?
- How does this relate to intelligence?