

CS 740 – Computational Complexity and Algorithm Analysis

Spring Quarter 2012

Slides 2

Pascal Hitzler

Kno.e.sis Center Wright State University, Dayton, OH http://www.knoesis.org/pascal/





Contents



- 1. SAT is in NP
- 2. SAT is NP-hard



SAT is in NP



$$F = \left(\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m} L_{i,j}\right)\right)$$

- Non-deterministically pick a truth assignment. Represent this in a look-up table. [linear in number of literals]
- Check if truth assignment satisfies F.
 [quadratic because of comparison of input with table entries]

• Formally, we need to do this on a TM – the encoding is a bit unwieldy, but straightforward.



Contents



- 1. SAT is in NP
- 2. SAT is NP-hard







- Give a logical formula which transforms computations of a TM M with input string u into a formula f(u) s.t.
 - u is accepted iff f(u) is satisfiable.
- + show that transformation is polynomial.

• [f(u) doesn't have to be in CNF because of Exercise 30]



Encoding



ND TM M:

- states: **q**₀,...,**q**_m
- alphabet: B=a₀,...,a_t
- accepting state: q_m
- rejecting state: q_{m-1}

(only one)

p(n) polynomial which is upper bound to number of computations

Boolean variables:

- Q_{i,k} M is in state q_i at time k
- P P_{j,k} Tape head is in position j at time k
 - **S**_{j,r,k} **Tape position j contains symbol** a_r **at time k**





	Clause	Conditions	Interpretation
i)	$\frac{\text{State}}{\overset{m}{\underset{i=0}{{\lor}}}} Q_{i,k}$	$0 \le k \le p(n)$	For each time k, M is in at least one state [p(n) clauses, m literals each]
	$\neg Q_{i,k} \lor \neg Q_{i',k}$	$0 \le i < i' \le m$ $0 \le k \le p(n)$	M is in at most one state at any time $[O(m^2) \times p(n) \text{ clauses}]$
ii)	$\frac{\text{Tape head}}{\underset{j=0}{\overset{p(n)}{\lor}}}P_{j,k}$	$0 \le k \le p(n)$	For each time k, the tape head is in at least one position [p(n) clauses, p(n) literals each]
	$\neg P_{j,k} \lor \neg P_{j',k}$	$0 \le j < j' \le p(n)$ $0 \le k \le p(n)$	and at most one position [O(p(n) ³) clauses]





	Clause	Conditions	Interpretation
iii)	$\underbrace{ \underset{r=0}{\overset{t}{}} S_{j,r,k} }$	$0 \le j \le p(n)$ $0 \le k \le p(n)$	For each time k and position j, position j contains at least one symbol [p(n) ² clauses, t literals each]
	$\neg S_{j,r.k} \lor \neg S_{j,r',k}$	$\begin{array}{l} 0 \leq j \leq p(n) \\ 0 \leq r < r' \leq t \\ 0 \leq k \leq p(n) \end{array}$	and at most one symbol $[O(t^2) \times p(n)^2 \text{ clauses}]$



SAT is NP-hard: Clauses iv & v



	Clause	Interpretation	
iv)	Initialization		
	Q _{0,0}	Begin in state 0	
	P _{0,0}	reading leftmost tape cell (position 0)	
	S _{0,0,0}	which contains a blank (symbol 0)	
	S _{1,r1,0}	The next n symbols contain the input string,	
	S _{2,r2,0}	which we'll denote a _{r1} ,a _{r2} , a _{rn}	
	S _{n,rn,0}		
	S _{n+1,0,0}	And the rest of the tape contains blanks	
	S _{p(n),0,0}	for the entire accessible portion	
v)	Final state	The computation ends in q_m – the accepting state	
	Q _{m,p(n)}		





A computation that satisfies all of these clauses still doesn't necessarily follow the rules of the machine, M.

Each state/symbol/position after time 0 must be obtained from the transition rules of M.



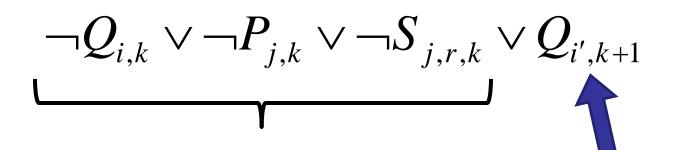


Clause	Conditions	Interpretation
vi) <u>Tape</u>		Symbols not at the position of the tape
<u>Changes</u>	$0 \le j \le p(n)$	head are unchanged
$\neg S_{j,r,k} \lor P_{j,k} \lor S_{j,r,k+1}$	$0 \le r \le t$	[p(n) ² × t clauses]
	$0 \le k \le p(n)$	



Converting rules in δ to clauses





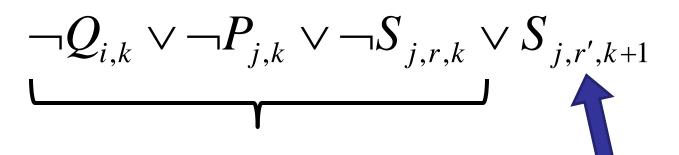
If none of these are satisfied, then we are in state q_i and position *j* scanning symbol a_r at time *k*

In that case, the next state must be $Q_{j'}$ or the clause is not satisfied.

For each $\delta(q_i, a_i) = [q_{i'}, ?, ?]$







If none of these are satisfied, then we are in state Q_i and position P_j scanning symbol S_r at time k

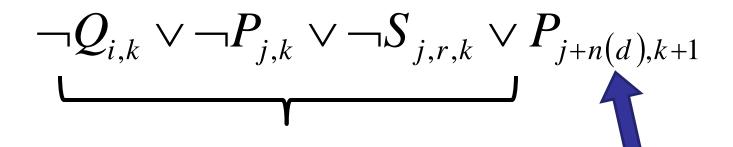
In that case, the next symbol at position j must be $S_{r'}$ or the clause is not satisfied.

For <u>each</u> $\delta(q_i, a_r) = [?, a_{r'}, ?]$



Same thing for tape head position





If none of these are satisfied, then we are in state Q_i and position P_j scanning symbol S_r at time k

In that case, the tape head will move either one position left or one position right.

Where n(L) = -1, and n(R) = +1For each $\delta(q_i, a_r) = [?, ?, L/R]$





The conjunction of these three clause types ensures that if we are in a certain state, reading a particular symbol at a particular time, we must be in the right configuration, according to δ in the following time step.

These are machine dependent.





Consistency clauses are constructed for every time, state, tape head position and tape symbol.

However, if we are scanning position 0 and attempt to move left, we go directly to the rejecting state.





We've been talking like there is only one transition for each state/symbol pair, but this is a <u>non-deterministic</u> Turing machine, right?

Let trans(i, j, r, k) be the disjunction of all the consistency clause sets for i, j, r, k. The resulting clause ensures that we are in <u>some</u> valid configuration following each transition.





Clause	Interpretation
vi) <u>Halted</u>	
$\neg Q_{i,k} \lor \neg P_{j,k} \lor \neg S_{j,r,k} \lor Q_{i,k+1}$	same state
$\neg Q_{i,k} \lor \neg P_{j,k} \lor \neg S_{j,r,k} \lor P_{j,k+1}$	same tape head position
$\neg Q_{i,k} \lor \neg P_{j,k} \lor \neg S_{j,r,k} \lor S_{j,r,k+1}$	same symbol at position r

For all appropriate *j*, *r*, *k*, and $i = q_{m-1}$, and $i = q_m$





We've defined a set of wff that are satisfiable if (and only if) some computation of ND TM M leads to an accepting final state.





Can the formula be created from any NDTM M in polynomial time?

- The values *m* and *t* are independent of the size of the input string. They <u>don't grow with *n*</u>.
- The number of clauses is polynomial in p(n).

qed

