Advanced Complexity

TD n°4

Aliaume Lopez

October 16, 2019

Exercise 1 : Circuit value

We study the following variants of Circuit-Value, all the reductions are logspace in the following exercise.

1. Show that Horn Sat is AL-easy
   — INPUT : a set ϕ of Horn clauses
   — QUESTION : is ϕ satisfiable ?
2. Show that Circuit Value is easier than Horn Sat
   — INPUT : a circuit C using $\vee, \wedge, \bar{\vee}, \bar{\wedge}$
   — QUESTION : does C evaluate to $\top$
3. Show that Monotone Circuit Value is AL-hard and easier than Circuit Value.
   — INPUT : a circuit C using $\vee, \wedge$
   — QUESTION : does C evaluate to $\top$
   Hint : build the configuration graph of the MT...
4. Show that Horn Sat is P-complete. Deduce that $\mathsf{AL} = P$.

   Hint : show that AL *is stable under logspace reductions*

Solution:

1. Let Φ be a set of Horn clauses. We build a proof system that is correct and refutationally complete for Φ.

   For every implication $x_1 \wedge \cdots \wedge x_n \implies x \in \Phi$ we add the following derivation rule. This subsumes the rules for $\bot$ and $\top$.

   $$\frac{\vdash x_1 \quad \cdots \quad \vdash x_n}{\vdash x}$$

   This system is correct Indeed, any valuation ν that satisfies Φ must agree with the derivation rules and the axioms. Therefore a proof of $\bot$ prevents any model from existing.

   This system is complete Assume that there is no derivation for $\bot$ in our system, we want to build a model of Φ. For this, just set $\nu(x) = \top$ if and only if x is derivable in our system. Because we dont derive $\bot$, every clause if Φ is validated in our model.

   Now, it is easy to see that proof searching for our very simple system is in AL and therefore testing the existence (or inexistence) of a proof of $\bot$ is in AL.

2. Circuit Value is easier than Horn-Sat. The construction is the following.

   Assume that $C = \langle V, E, L, p \rangle$ is a circuit. We wrote V for the set of nodes, E for the edges, L for the labelling function from nodes to connectors $\{\vee, \wedge, \bar{\vee}, \bar{\wedge}\}$ and p for the only terminal node.

   We build a set X of variables defined as $V \cup \bar{V}$.

   For every node $v \in V$ with ancestors $v_1, \ldots, v_n$, we build the following clauses

— When $L(v) = \wedge$

$$\bigwedge_i v_i \implies v \qquad \forall i, \bar{v_i} \implies \bar{v}$$

— When $L(v) = \vee$

$$\forall i, v_i \implies v \qquad \bigwedge_i \bar{v_i} \implies \bar{v}$$

— When $L(v) = \bar{\vee}$

$$\bigwedge_i \bar{v_i} \implies v \qquad \forall i, v_i \implies \bar{v}$$

— When $L(v) = \bar{\wedge}$

$$\forall i, \bar{v_i} \implies v \qquad \bigwedge_i v_i \implies \bar{v}$$

The last bit is adding a clause $\neg(-p)$ where p is the last node.

We have to check several little properties about our construction

This construction can be done in logspace easy

Every valuation satisfying Φ respect negation and corresponds to an evaluation of the circuit. This is done by induction on the depth of the nodes in the graph. The base case is just initial nodes, where the construction directly gives the expected property that $\nu(x) = 1 - \nu(\bar{x})$. Induction case just uses the previous the induction hypothesis along with the "two way" equations defining the values of $\bar{v}$ and v.

If the circuit evaluates to $\top$ Then we can build a valuation for Φ by copying the values of the nodes and setting $\nu(x) = 1 \ - \nu(\bar{x})$. It is routine to check that the formulas in Φ are satisfied, the only interesting case is the "last bit", working only because the value of p is $\top$ in our circuit.

Therefore Φ has a model.

If the circuit evaluates to $\bot$ Then every model of Φ must respect negation, and correspond to the evaluation of the circuit. Which is stating that a model of Φ has to set $\nu(p) = \bot$, but then $\nu(\bar{p}) = 1$ and this is absurd because $\nu(\bar{p}) = 0$ in every model of Φ (last clause added).

3. Monotone Circuit Value is AL-hard under logspace reductions.

   Let M be a machine that recognises $L \in \mathsf{AL}$. We are going to build for a word w a circuit C_w such that C_w evaluates to $\top$ if and only if $w \in L$.

   Construction Let G be the configuration graph of the machine M on word w. We implement our circuit by taking G^{op}, and setting $L(v) = \vee$ if the configuration was in an *existential state* and $L(v) = \wedge$ if it was in a *universal state*. Accepting terminal configuration is labelled $L(v) = \wedge$, non accepting terminal configuration is labelled $L(v) = \vee$. Let p be the initial state of M on input x. We have built a circuit C.

   This is a circuit We can assume that the machine M has a counter incremented at each step on another tape. It seems that the was already circuit acyclic, because we just "go back" to the initial configuration, however, inacessible configurations could do whatever they wanted... This prevents cycles in the circuit C. **To students :** actually prove this statement, it is not that easy to see that we break all cycles.

   It is logspace constructible left as an exercise

   It is a reduction this is by definition of AL (look at the definition, look at the evaluation of the circuit, now see that it is isomorphic).

4. It is easy to see that Horn Sat is in P using the standard saturation algorithm.

   The proof that it is P-complete is essentially the same as Cook. The basic steps are

   (a) Take a TM M assume that the only accepting position is with the head of the TM at start in a specific state q.

   (b) Build the "rectangle of configurations" of the machine M on input x.

   (c) For each cell of this array, build the triple (i, j, x) where (i, j, x) will state that at cell (i, j) the tape contains letter x, where letters are $\Sigma \cup \Sigma \times Q$.

   (d) Now, it is easy to see that every cell on line $i + 1$ only depends on a *uniformly fixed* number of cells in line i. This ... calls for a Horn Clause!

   (e) We add the clause $IN \implies \perp$ where IN is the content of the input tape.

   However, we only have the "positive" variables, therefore we need to add $-(i, j, x)$ stating that the cell (i, j) does *not* contain x as a letter.

   This construction can be done in logspace because polynomials are logspace constructible.

   Note that the TM accepts if and only if the formula is NOT satisfiable, which is not a problem as P is stable by complement.

   Now, because AL is stable under logspace reductions, because Horn Sat is AL-easy, Monotone Circuit Value AL-hard and Monotone Circuit Value is easier than Horn Sat, we deduce that all of the previous problems are equivalent under logspace reductions, and are AL-complete. This proves also that $\mathsf{AL} = \mathsf{P}$ because Horn Sat is P-complete.

Exercise 2 : Padding

Adapt the proof relating ATIME with DSPACE to relate ASPACE with DTIME... Hint, this uses padding.