Formal Proofs and their Lengths VI

Current exercise session is based upon the paper of P. Pudlák "Proofs as games". It can be found here: https://www.jstor.org/stable/2589349.

Exercise 1. Recall the *pigeonhole principle* which states that there is no injective mapping between $n+1$ different pigeons and n different holes. We denote this statement is PHP_n^{n+1}. Express $\neg PHP_n^{n+1}$ as a CNF of size polynomial in n.

The above CNF (or more precisely a family of CNFs) is the one for which we will derive exponential lower bounds for resolution refutations. The major part of today's session is devoted to the analysis of a certain combinatorial game. Near the end we will show the correspondence between *space complexity* of the mentioned game and size of resolution refutations of $\neg PHP_n^{n+1}$.

A game

Definition 2. We consider the following **game** played between two players Alice (*falsifier* or *spoiler* denoted simply as A) and Bob (*prover* or *delayer* denoted simply as B). A pretends there is some mapping f contradicting PHP_n^{n+1} and B tries to convict her of lying. B can query A questions of the following form "does pigeon p goes to hole h" and A must answer positively or negatively. B stores A's answers in the form $(p, h, \text{yes} / \text{no})$, but through the course of the game he might erase particular records from his list. A sees the actual list stored by B.

B **wins** iff there is a *direct contradiction* among the records stored in his list. This means that either for all holes h B's list contains records (p, h, no) for a particular pigeon, or B's list contains records (p', h, yes) and (p'', h, yes) for $p' \neq p''$, or B's list contains records (p, h', yes) and (p, h'', yes) for $h' \neq h''$, or B's list contains records (p, h, yes) and (p, h, no).

Strategy (for B) is a function from the set of all the lists of records into possible next moves which specify what to erase from the current list and what to query next.

We will consider only the *winning strategies* which we will call just strategies. We can also express any strategy as a function from the set of only the *admissible* lists, i.e. lists which can actually appear through the course of the game.

Exercise 3. Try to come up with some strategy for which the set of all lists and the set of all admissible lists do not equal.

Definition 4. Suppose a strategy is fixed. We say the the **complexity of the strategy** is the number of different records that can appear in all possible games.

Exercise 5. Recall that we identify a strategy with the function describing the next move based on the current list, and we care only about the admissible lists. Show that the strategy's complexity equals the number of *strategy's rules*, i.e. it the size of the domain of the function as above.

We emphasize that complexity manifests itself only through the course of many different games played against the strategy.

Exercise 6. Try to come up with a strategy whose complexity is exponential in n, although in any single game only poly of n different lists may appear.

Our goal is to prove that any strategy is necessarily of exponential complexity. For that we will define a *superstrategy* (for A) which would force B into using a lot of different lists. Here, superstrategy means a parameterized family of strategies of A to play against B. We will also see that this superstrategy doesn't actually use a priori knowledge of B's strategy.

Haken's superstrategy

We assume n is divisible by 4.

Definition 7. Let α be a partial one-to-one assignment of $n/4$ different pigeons to distinct holes. An A's **strategy induced by** α is defined as follows. During the course of a game A modifies α by possibly enlarging it to β. She may at some point make it smaller, but α is always kept as a part of the β. These updates are happening immediately after B asks a new question and an answer is given based upon the updated β as will be explained below.

We first explain the rules to update β. If a pigeon p is not yet in β, then β is being extended by (p, h) as soon as the number of *prohibited holes* for p given by the current list of B is $\geq n/2$. The hole h' is prohibited iff there is a record in the list of the form (p, h', no). A hole h which is being assigned to p is any hole which is neither prohibited by current list, nor is it already assigned to some different pigeon based upon the current list. If there is no such hole, then A gives up and B wins automatically (A may give some contradictory answer). A pair (p, h) is erased from β when (p, h, yes) is no longer on the B's list and the number of prohibited holes for p is $< n/2$.

After updating β, being presented with a question of the form "does the pigeon p is being mapped to h" A answers "yes" iff $(p, h) \in \beta$.

Exercise 8. * Assume that by playing against B with a strategy induced by α, A at some point forces B to store a list which contains all the records of the form (p, h, yes) for all $(p, h) \in \alpha$ (the list may contain additional records, as well). Show that the complexity of B's strategy is exponential in n. It might help to show an exponential lower bound on the number of *pairwise incompatible* αs.

Unfortunately, the above assumption does not work.

Exercise 9. * Find a particular strategy for B which does not satisfy the above assumptions.

What does hold, however, is that B must at some point store *a lot of information* regarding α. This will be enough to show exponential complexity of B's strategy, as we will later see.

Definition 10. Let us fix a list of records of B. We say that a pigeon p is **good** (w.r.t. the current list of B) iff there is a record on the list of the form (p, h, yes) or there are $\geq n/2$ prohibited holes for p.

Exercise 11. Show that a pigeon p outside of the domain of α is good iff $p \in dom(\beta) \setminus dom(\alpha)$.

We now want to prove the following lemma. The superstrategy defined above is called *Haken's supestrategy* and is denoted as HS.

Lemma 12. If A uses HS, then however B plays, at some point before the games ends, he must create a list for which there are exactly $n/4$ good pigeons.

Exercise 13 (*towards 12*)**.** Show that if A uses HS, then she looses only when she has to give up. Show that this happens only when there is a pigeon p with $\geq n/2$ prohibited holes so that all the remaining holes have already been reserved for other pigeons by the assignment β. This means the size of the domain of β is $\geq n/2$. Derive that there is at least $n/4$ good pigeons. Use this to show that at some point there were exactly $n/4$ good pigeons for B's list.

Definition 14. Let R be the set of all lists for which there are exactly $n/4$ good pigeons. We say α **is in** $r \in R$ iff r appears as B's list when A plays by the strategy induced by α. We say α **is consistent with** $r \in R$ iff there is no record in r contradicting α.

Exercise 15. Show that if α is in r, then α is compatible with r.

We say that a value is *exponentially large* (in n) iff it is greater than $2^{\epsilon n}$ for positive ϵ. Similarly, a value is *exponentially small* (in n) iff it is smaller than $2^{-\epsilon n}$.

Exercise 16. Let us pick α as above uniformly at random. Assume that for any fixed $r \in R$ the probability of α being compatible with r is exponentially small. Derive that the probability of α being in r is exponentially small, as well. Use that to show that the size of R is exponentially large. Conclude that the B's strategy has exponential complexity.

We think of the random process of selecting α as consisting of two stages. We first choose the domain of α and then randomly assign chosen pigeon to distinct holes.

Exercise 17. Let $r \in R$ be fixed and let G denote the set of all good pigeons for r. We know that $|G| = n/4$. Show that the average size of the intersection of $dom(\alpha)$ and G is $n/16$. More generally, assume we have a set X of size n and $Y \subseteq X$. Let us pick uniformly at random a subset of X of size n/k for some $k \leq n$. Then, the expected size of the intersection of Y and our random set is exactly $|Y|/k$.

Fact 18. Fix some fraction of n smaller than $n/16$, say $n/64$. Then, the probability for a size of the intersection of $dom(\alpha)$ and G to be at most $n/64$ is exponentially small.

If you want to prove the above, it is enough to calculate that the probability of the bad event is exactly

$$\binom{n+1}{n/4}^{-1}\sum_{i=0}^{n/64}\binom{n/4}{i}\binom{n+1-n/4}{n/4-i}$$

and to show that this value is exponentially small in n.

So now we may assume that the intersection of $dom(\alpha)$ and G is at least $n/64$.

Exercise 19. Fix a pigeon $p \in dom(\alpha) \cap G$. Show that the probability of $(p,h) \in \alpha$ being consistent with r is at most $1/2$. Conclude that if the holes for pigeons from $dom(\alpha)$ were all chosen randomly and independently, then the probability of α to be consistent with r would have been $\leq (1/2)^{n/64}$.

The problem is that holes are not really chose independently. One needs to ensure that if a hole has been chosen for a pigeon p, then the same hole would not be chosen again for a different pigeon. Show that the true probability of α being consistent with r is $\leq (3/4)^{n/64}$.

This concludes the whole proof of the theorem below.

Theorem 20. There exists a positive ϵ so that the complexity of any strategy of B is at least $2^{\epsilon n}$.

Proofs

It remains to establish a correspondence between resolution refutations and Bs' strategies so that the size of a refutation would correspond to the complexity of a strategy.

Fix a resolution refutation of $\neg PHP_n^{n+1}$ and consider its' DAG representation (top-to-bottom ending with the empty clause). Each node is labeled as a clause and has either zero or two ancestors. We label each edge by the literal which was used in a resolution rule to derive a lower clause. To define a strategy for B we first flip the DAG, i.e. we reverse all the edges and we relabel all the nodes by negating the clauses used in the original labeling. The flipped clauses (conjuctions of literals) are to be interpreted as B's lists where a literal $x_{p,h}$ represents the record (p, h, yes) and $\neg x_{p,h}$ represents (p, h, no).

B's strategy is as follows. At the beginning he starts with the empty list. At any point through the game, seeing a list of records ρ which corresponds to a label of some node in our graph, B asks A a question of the form "does pigeon p goes to hole h", where $x_{p,h}$ is a variable labeling edges connecting given node and its' two children. Based upon the answer he moves upwards the graph and updates his list to exactly correspond to the labeling of the node which he arrives at.

Exercise 21. Show that the above is a winning strategy for B. Show that the complexity of the strategy is exactly the size of the refutation.