Christopher Souvey
Professor Karl Lieberherr
Algorithms CS4800
27 April, 2012

Reflections on the Quantifier Game

Background

The quantifier game is not a new idea. It is a formalization of a natural process we use every day (be it academically, personally, or in the workplace) to evaluate decisions. The essence of the game is that all claims (statements proposed as truth) should include a well-defined means to dispute them. To simplify this process, the game narrows the domain to claims based on quantifiers and of the form "for all X there exists a Y such that ..." or "there exists Y such that for all X ...". A player can oppose this type of claim by instantiating the X ("for all") with a value that will cause difficulty for the proposer (who represents "there exists"). This setup creates a competition between two players, and the resulting "game" (taking turns proposing and opposing claims) allows rapid refinement and enhanced understanding of claims. In addition to opposition (or refutation), the game defines protocols for agreeing with and strengthening claims. These are fairly simple variations upon opposition, in which the agree-er or strengthen-er becomes the new proposer of the claim and the original proposer must refute it. The basic game and its variations are fairly analogous to the common mental processes to used test ideas, which include activities such as checking edge cases, imaginary worst-case scenarios, and looking for counterexamples.

Introduction

Northeastern University's Algorithms course, as taught by Professor Karl Lieberherr, makes extensive use of the quantifier game in both homeworks and lectures to serve as a baseline framework for algorithmic evaluation. Throughout the course, my homework partner and I employed two increasingly divergent forms of the quantifier game: in-person interactions for brainstorming and online collaborative playgrounds on Piazza. This paper provides a brief overview of these experiences with the quantifier game, some problems I ran into during its use in the course, and suggestions for fixing them.

The Game in Private (Making it Our Own)

Over the course of the semester, my homework partner and I refined our own variation of the quantifier game, which we used amongst ourselves to brainstorm and validate ideas before implementing them or posting them to Piazza. We quickly found that the traditional quantifier game introduces a great deal of overhead (such as the formal grammar) that lengthen interactions and slow down progress. As a result, we stripped the game down to its bare essentials, removing all the formality and jargon and replacing it with natural and fluent conversation, such as "<P1>:

I think so-and-so always works. <P2>: But what about when X is 0? <P1>: Oh yeah, that breaks it". Despite the non-technical language, this exchange maintains the spirit of the quantifier game. Another change we made was to remove the agreement protocol entirely. I found agreement failed to serve any useful purpose, as it requires the proposer to attempt to refute their own claim: if they know how to do that, they would have already done so! Instead, agreement was reached after some time passed without a successful refutation, at which point we mutually agreed we were unlikely to come up with valid opposition.

Our most important change, however, was removing the competitive element (in particular the notion that making an incorrect statement causes a loss of credibility) and making it collaborative. There were no defined "proposer" and "opposer" players: we seamlessly switched roles to suit the current context. Proposing was like a brainstorming process, in which either of us could throw out possible claims (such as algorithms). In sharp contrast to the quantifier game, which penalizes the sharing of claims that have not been fully thought through, we had a "no ideas are bad" mindset during this process. Even if we *knew* an idea was bad, we would still discuss it, since we built off each other's ideas: often a small tweak can turn a bad idea good. Refutation could be performed by either of us (including the one who made the original claim) with the other player then taking on the traditional role of the proposer. Generally this entire process took place with us both in front of a whiteboard, drawing instances (which were often graphs). In aggregate, these changes greatly increased our efficiency and made room for more creative solutions.

The Game in Public (Piazza)

Despite being a valuable tool for many problems, in my experience, the game was sometimes overused and applied beyond its appropriate use cases for homework assignments, which generally required interaction on Piazza. As a result, the assignments varied greatly in effectiveness (see Appendix A for brief notes on each). There were some general patterns, however, that should serve as effective predictors of future performance. One major problem was that solutions were often discussed (or at least very clearly hinted at) in class. This significantly decreased the amount of discourse taking place, as most students arrived at the same solution and therefore had nothing to discuss. It would better facilitate discussion if information about the solution was not provided until after the homework is due and students were actually unsure if their claims were correct or not. Another frequent issue was that many homeworks required that every team post one or more claims, which tended to result in redundancy and an overwhelming number of claims. When this happened, many claims were ignored or left incomplete, halfway through a protocol. Agreement in particular proved to be problematic and was very rarely correctly completed.

On a positive note, optimization claims (which tend to more driven by strengthening than opposing) seemed to be the most effective use of the quantifier game. In particularly, they promoted just the right balance of competition and collaboration that encouraged forward progress as teams took turns leapfrogging each other when they realized their claims were not optimal. An unfortunate subset of these, however, were algorithmic runtime claims, which I personally found to be an ineffective use of the quantifier game that often felt forced. There were two such types of claims: asymptotic and example instances. The biggest problem these both suffered from was a crippling of the most important interaction type: refutation. For asymptotic claims, refutation was really only possible if the runtime was so low that it couldn't possibly be correct (ex: searching an NxN table in N time). If the claim was reasonable, however, there was really no way to know if the players had incorrectly computed the asymptotic behavior, or even if their algorithm worked or not. And, because a player is punished for making an incorrect refutation, it would be unwise to ever challenge a runtime claim (due to lack of evidence). For instance-based claims, opposition based on runtime was not possible since the ForAll only included a single element which was used to measure the provided runtime. As a brief side note, these claims also were very difficult to compare without a consistent timing metric (too many variables for milliseconds to be useful and not everyone used a JVM-based language). Another problem that was raised by these claims was how to compare two asymptotic bounds. For example (from homework #9): Is "(n+2m)*C + m" better or worse than "nm^2 + m"? As a result, there was very little strengthening of asymptotic claims and just lots of new claims. This could be solved by defining a comparison function as part of the playground that takes 2 claims and returns whether they are equal (agree), greater/less than (strengthen), or incomparable (propose). We already made implicit assumptions about the existence such a function in a different context: answer took precedence over runtime for strengthening. Combined, all these issues raise the question of whether the quantifier game is the correct interaction type for runtime competitions, which in my opinion could be better served by a simple high-score list of runtimes for some sample instances.

Finally, there is the question of how effective the Piazza platform was at hosting the quantifier game. Piazza is a well-designed site, but we using it outside its intended use. It is designed as a question/answer system, whereas the quantifier game is more suited a forum-like environment with threads and replies. Fortunately, Piazza actually has this feature in the form of comments on questions. The next time this course is taught, students should be instructed not to use the answer field and instead use comments, which have a hierarchy and are much easier to follow.

Conclusion

Although the quantifier game is an effective teaching tool, it is not universally effective, and assignments should be either be modified to better suit its format or stop diluting the message by using it in unconventional situations.

Appendix A:

#	Assignment Summary	Connection to Quantifier Game	Effectiveness
1	IntervalScheduling WeightedIntervalScheduling BipartiteMatching IndependentSet QuantifiedBooleans QuantifiedNumerical	For each claim, how would you defend or refute them? Propose (limited number) or oppose one of these on Piazza	No interesting discourse on Piazza, but served as an introduction to the protocol
2	Highest Safe Rung	Play game with partner; turn in all exchanges and decision trees; list optimum claims; give player reputations Propose (no limit) and oppose/ agree claim on Piazza about a particular n and cost function	Solution was given away in lecture, so the exchanges with partner were uninteresting and done only to turn in per requirement. No interesting discourse on Piazza.
3	Gale Shapeley HSR Landau	Play game with partner; turn in all exchanges; describe means to arrive at claims and defense strategy. Propose (limit to 5), agree, refute, or strengthen on Piazza.	Gale Shapeley: quantifier game was helpful for arriving at solution and also played out well on Piazza. HSR: no interesting discourse
4	Graph-DIS	Play on Piazza (no requirement): propose (limit 5), agree, refute	Quantifier game was helpful for arriving at solution and also played out well on Piazza.
5	Weighted maximum subset (AtLeast, NoPairContradictions)	Turn in claims and argue why they are optimal. Play on Piazza (no requirement): propose (limit 3 & 3), agree, refute, or strengthen.	Quantifier game was helpful for arriving at solution. There was no interesting discourse on Piazza, as the professor stated when a correct claim had been made rather than letting the refutation protocol take place.
6	HSR Avatar	Avatar should propose, strengthen, and refute claims.	Nothing additional; just formalizing/turning-in code from HW2
7	Leaf Covering	Propose one claim and oppose/ agree/strength two claims of an upper bound on running	Quantifier game was useful for deriving algorithm. Quite a bit of Piazza discourse, some of it

		time (milliseconds). Multiple variables with precedence defined for strengthening.	useful; however, it was difficult to know when to disagree with a claim. Speed in milliseconds made it very depending on system and programming language.
8	Shortest and longest path	Make 2 claims and oppose 2 claims about maximum algorithm runtime (function of nodes and edges)	Extremely little discourse. Lots of repetitive claims and a very unclear strengthening precedence. We were supposed to look up an algorithm, not design our own, so the quantifier game was unhelpful.
9	Network flow	Two types of claims: bounds on algorithm speed and flow/ speed for a specific graph. Make one of each type of claim and oppose/agree with each type on Piazza.	Piazza runtime claims faced the same problem as #7 and Piazza bound claims faced the same problems as #8. Not very helpful offline since we just looked up the algorithm
10	Reduced flow	On Piazza, propose and oppose/agree with a claim included flow/speed for a specific graph	Quantifier game helpful for deriving algorithm. Piazza claims faced same problems as #7. There were a large number of claims but very little discourse.