1 Soundness of the Algorithm in [1]

Let us consider a coin toss experiment with $P(H) = p_0$ and $P(T) = 1 - p_0$, where p_0 is unknown. We want to test whether $p_0 \geq p$ or $p_0 < p$. There are two ways to set up the experiment as follows:

Type 1 Assume that $p_0 \notin [p - \delta_1, p + \delta_2]$. Let $H_0\colon p_0 < p$ and $H_a\colon p_0 \geq p$. We accept H_a if $\frac{\sum_{i \in [1,n]} z_i}{n} \geq p$ [1] and accept H_0 otherwise, where n is the sample size and z_is are our observations. Then the sample size for the experiments must be set such that both Pr[we accept H_a | H_0 is correct] $\leq \alpha$ and Pr[we accept H_0 | H_a is correct] $\leq \beta$ hold. This is how we set up the experiments in [2], and to our understanding this is how experiment is set up in [3]. Note that this experiment is not correct if $p_0 \notin [p - \delta_1, p + \delta_2]$. This is a shortcoming of this approach. We have to choose very small δ_1 and δ_2 to make sure that $p_0 \notin [p - \delta_1, p + \delta_2]$.

Type 2 Let p_0 be any value between 0 and 1; in particular, we do not assume that $p_0 \notin [p - \delta_1, p + \delta_2]$. Let $H_0\colon p_0 < p$ and $H_a\colon p_0 \geq p$. We accept H_a if $\frac{\sum_{i \in [1,n]} z_i}{n} \geq p + \delta_2$ and accept H_0 if $\frac{\sum_{i \in [1,n]} z_i}{n} \leq p - \delta_1$, where n is the sample size and z_is are our observations. We say "don't know" if $p - \delta_1 < \frac{\sum_{i \in [1,n]} z_i}{n} < p + \delta_2$. Then the sample size for the experiments must be set such that both Pr[we accept H_a | H_0 is correct] $\leq \alpha$ and Pr[we accept H_0 | H_a is correct] $\leq \beta$ hold. This removes the shortcoming of the previous approach that we have to assume that $p_0 \notin [p - \delta_1, p + \delta_2]$. However, in this case we can have "don't know" answers. Note that in this case we can choose any δ_1 and δ_2 without the fear that our experiment may go wrong. However, bigger δ_1 and δ_2 will result in more "don't know" answers.

Henceforth, we will only consider Type 2 experiments. A Type 2 experiment for the coin toss is specified by the 4-tuple $(\alpha, \beta, \delta_1, \delta_2)$; given these parameters we know how many samples we need to draw in a Type 2 experiment, and we know what decision to take once we are given a sample of sufficiently large size. Let us denote by $\text{sample}(\alpha, \beta, \delta_1, \delta_2)$ to be the (minimum) sample size required ensure that the type I and type II errors are bounded by α and β, when conducting an experiment as above. Let E be the set of all experiments of Type 2 such that $1 \geq \alpha = \beta \geq 0$, $p > \delta_1 > 0$, and $(1 - p) > \delta_2 > 0$.

Consider a sample $\hat{z} = z_1, z_2, \ldots z_n$ of n coin tosses. Define $E^{\text{yes}}(\hat{z}) \subseteq E$ to be the set of experiments (with type I and type II error bounds being equal) such that the sample $\hat{z}$ will cause such a Type 2 experiment to accept the alternate hypothesis. More formally, $e = (\alpha, \beta, \delta_1, \delta_2) \in E^{\text{yes}}(\hat{z})$, when the following holds:

[1] The threshold can of course be modified to account for the fact that the median in the binomial distribution is not the same as the mean.

1. sample$(\alpha, \beta, \delta_1, \delta_2) \leq n$, i.e, $\hat{z}$ is sufficiently large to ensure the type I and type II errors to be bounded by α and β

2. $\frac{\sum_{i=1}^{n} z_i}{n} \geq p + \delta_2$.

Similarly, we can define $E^{\text{no}}(\hat{z}) \subseteq E$ to be the set of experiments that will result in the null hypothesis being accepted. In other words, $e = (\alpha, \beta, \delta_1, \delta_2) \in E^{\text{no}}(\hat{z})$, when

1. sample$(\alpha, \beta, \delta_1, \delta_2) \leq n$

2. $\frac{\sum_{i=1}^{n} z_i}{n} < p - \delta_1$.

Consider the following algorithm A. Given a sample $\hat{z}$ (drawn by n coin tosses), the algorithm does one of two things: Finds an experiment $(\alpha, \beta, \delta_1, \delta_2) \in E^{\text{yes}}(\hat{z})$ and outputs "yes" and number α; or finds an experiment $(\alpha, \beta, \delta_1, \delta_2) \in E^{\text{no}}(\hat{z})$ and outputs "no" and number β. Let us assume A is a deterministic algorithm (once it is given the sample $\hat{z}$). Observe that

$$\begin{array}{c} Pr[A(\hat{z}) = \text{yes} \mid p_0 < p] \leq \alpha \\ Pr[A(\hat{z}) = \text{no} \mid p_0 \geq p] \leq \beta \end{array}$$

where the probability is being taken over the fact that $\hat{z}$ is a random sample of size n.

Our algorithm, presented in [1], can be viewed as doing the same thing as what algorithm A was doing in the context of coin tosses. It is trying to find an "experiment" that will allow it to consistently label the samples. In the case of model checking the "experiment" and sample have more structure. The sample is now a tree. A consistent "experiment" is a labelling of each node in the tree (along with the satisfaction or non-satisfaction of subformulas) with bounds, α, β, δ_1 and δ_2 that will correspond to right labelling, i.e., the number of samples needed to guarantee a particular error bound is less than that in the given sample from that particular state, and the "yes" and "no" labelling is consistent with what an actual run of an algorithm like [3] might require. And just like in the case of the coin toss, the label α of the root (with respect to the property being tested for) gives a bound on the p-value of such an algorithm. Once again, the algorithm is deterministic: given a sample (or tree) the algorithm will label the internal nodes with α etc. in a determined manner. The probability is taken over the fact that the sample is drawn randomly from the given system.

References

[1] K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *16th conference on Computer Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pages 202–215. Springer, July 2004.

[2] K. Sen, M. Viswanathan, and G. Agha. On statistical model checking of probabilistic systems. In *17th conference on Computer Aided Verification (CAV'05)*, LNCS (To Appear). Springer, July 2005.

[3] H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Proc. of Computer Aided Verification (CAV'02)*, volume 2404 of *LNCS*, pages 223–235, 2002.