

Concept of Statistical Model Checking

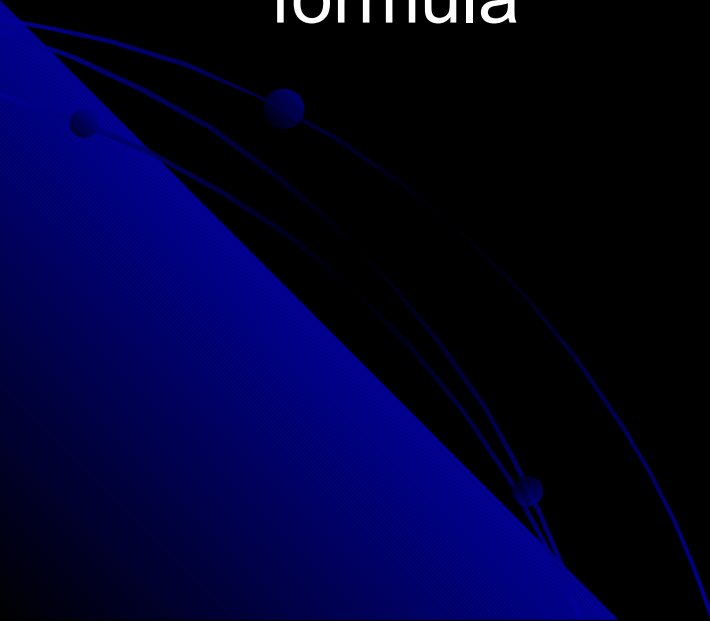
Presented By: Diana EL RABIH



Introduction

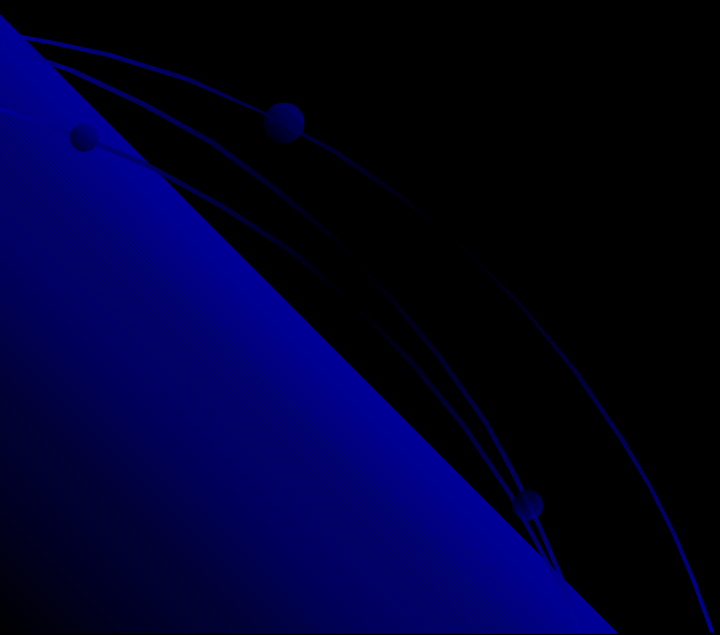
- Model checking of stochastic systems
 - Continuous-time Markov chains
 - Continuous Stochastic Logic (CSL)
 - Probabilistic time-bounded properties
- Comparison of two techniques
 - **Numerical** computation of probabilities
 - **Statistical** hypothesis testing

Probabilistic Model Checking

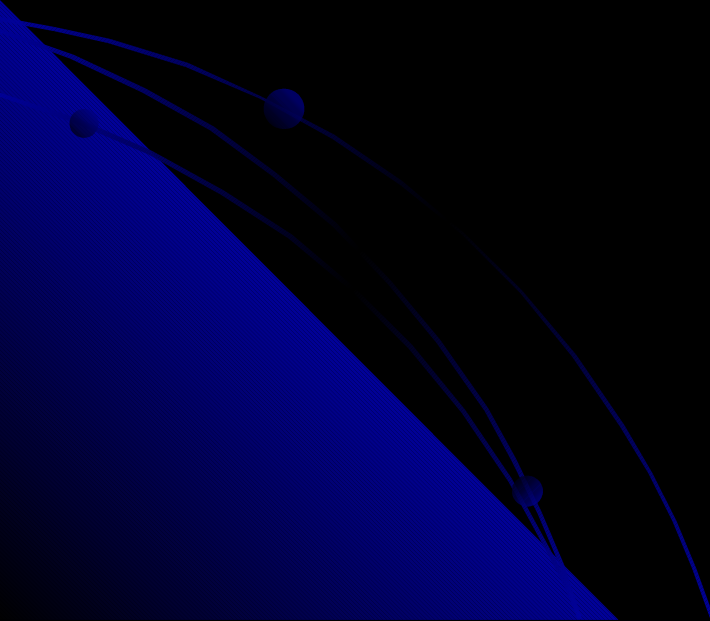
- Given a model M , a state s , and a property φ , does φ hold in s for M ?
 - Model: continuous-time Markov Chain
 - Property: Continuous Stochastic Logic (CSL) formula
- 

Continuous Stochastic Logic (CSL)

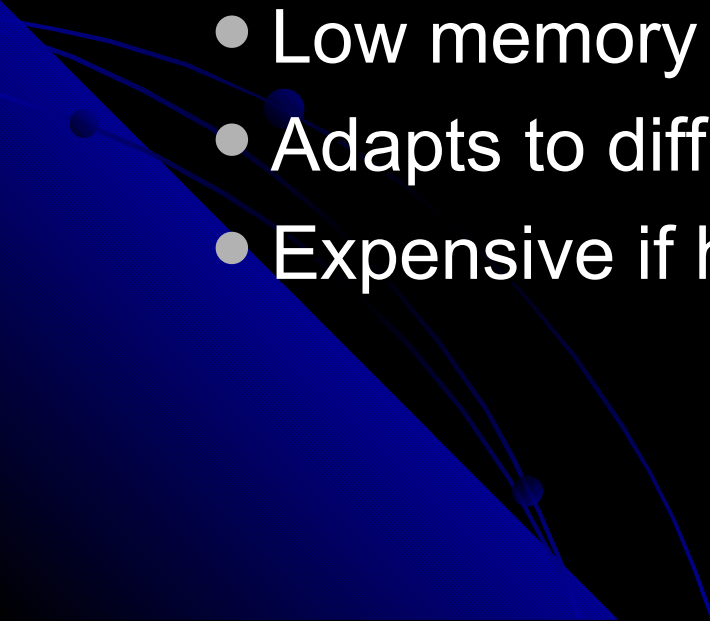
- State formulas
 - Truth value is determined in a single state
- Path formulas
 - Truth value is determined over a path



State Formulas

- Standard logic operators: $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, ...
 - Probabilistic operator: $\text{Pr}_{\geq\theta}(\rho)$
 - Holds in state s iff probability is at least θ that ρ holds over paths starting in s
- 

Numerical vs. Statistical Probabilistic Model Checking

- Numerical Method
 - Highly accurate results
 - Expensive for systems with many states
 - Statistical Method
 - Low memory requirements
 - Adapts to difficulty of problem (sequential)
 - Expensive if high accuracy is required
- 

Numerical Solution Method

- Verify $\Pr_{\geq\theta}(\varphi_1 \ U^{\leq T} \ \varphi_2)$ using transient analysis [Baier et al. 2000]
 - Make states satisfying $\neg\varphi_1 \vee \varphi_2$ absorbing
 - Compute probability p of being in a state satisfying φ_2 at time T in modified model
 - $\Pr_{\geq\theta}(\varphi_1 \ U^{\leq T} \ \varphi_2)$ holds iff $p \geq \theta$

Probability Computation

- Uniformization [Jensen 1953]
 - Transform model into discrete time Markov chain with transition matrix \mathbf{P}
 - Compute \mathbf{p} for all states as follows:

$$\sum_{k=0}^{\infty} \frac{e^{-q \cdot T} (q \cdot T)^k}{k!} (\mathbf{P}^k \cdot \mathbf{1})$$

- Truncated summation from L_ε to R_ε with truncation error ε [Fox & Glynn 1988]

Role of Truncation Error

- We know that $p \geq \tilde{p}$ and $\tilde{p} \leq p + \varepsilon$
 - If $p \geq \theta$ then $\tilde{p} \geq \theta$
 - If $\tilde{p} + \varepsilon \leq \theta$ then $p \leq \theta$
 - Otherwise, can't tell if $\Pr_{\geq \theta}(\varphi_1 U^{\leq T} \varphi_2)$ holds
- Good news: $\varepsilon = 10^{-10}$ possible without noticeable performance degradation

Complexity of Numerical Solution Method

- $O(q \cdot T)$ matrix vector multiplications
 - Rates, time bound, and number of states
- All states for same cost
 - In practice, memory and time savings for single state

$$\sum_{k=L_\varepsilon}^{R_\varepsilon} \frac{e^{-q \cdot T} (q \cdot T)^k}{k!} (\mathbf{P}^k \cdot \underline{\varphi}_2) \quad \mathbf{P}^k \cdot \underline{\varphi}_2 = \mathbf{P} \cdot (\mathbf{P}^{k-1} \cdot \underline{\varphi}_2)$$

Speedup Techniques

- Steady-state detection [Malhotra et al. 1994]
 - If $\mathbf{P}^k \approx \mathbf{P}^{k-1}$ then stop after k iterations
 - Can lead to significant savings
- Sequential stopping rule
 - Stop if $p \geq \theta$ after k iterations
 - At most $O(\sqrt{q} \cdot T)$ fewer iterations

Statistical Solution Method

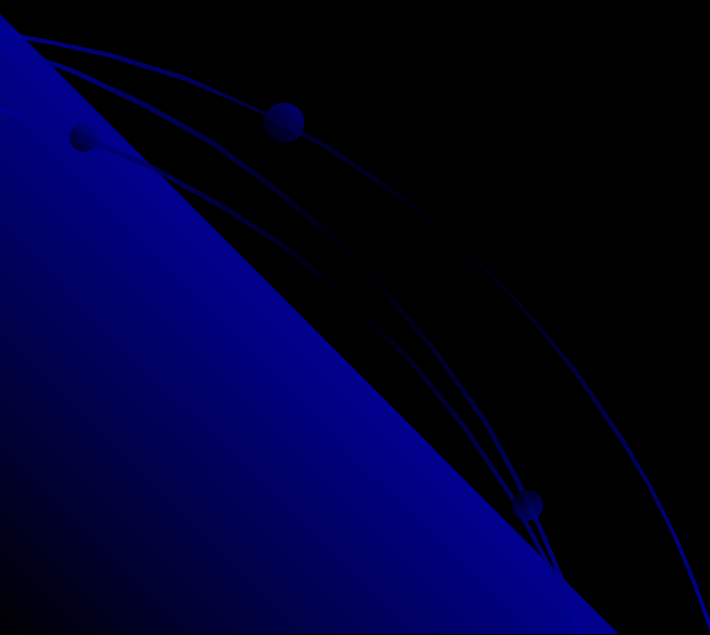
[Younes & Simmons 2002]

- Use **discrete event simulation** to generate sample paths
- Use **sequential acceptance sampling** to verify probabilistic properties
 - Hypothesis: $\Pr_{\geq \theta}(\rho)$

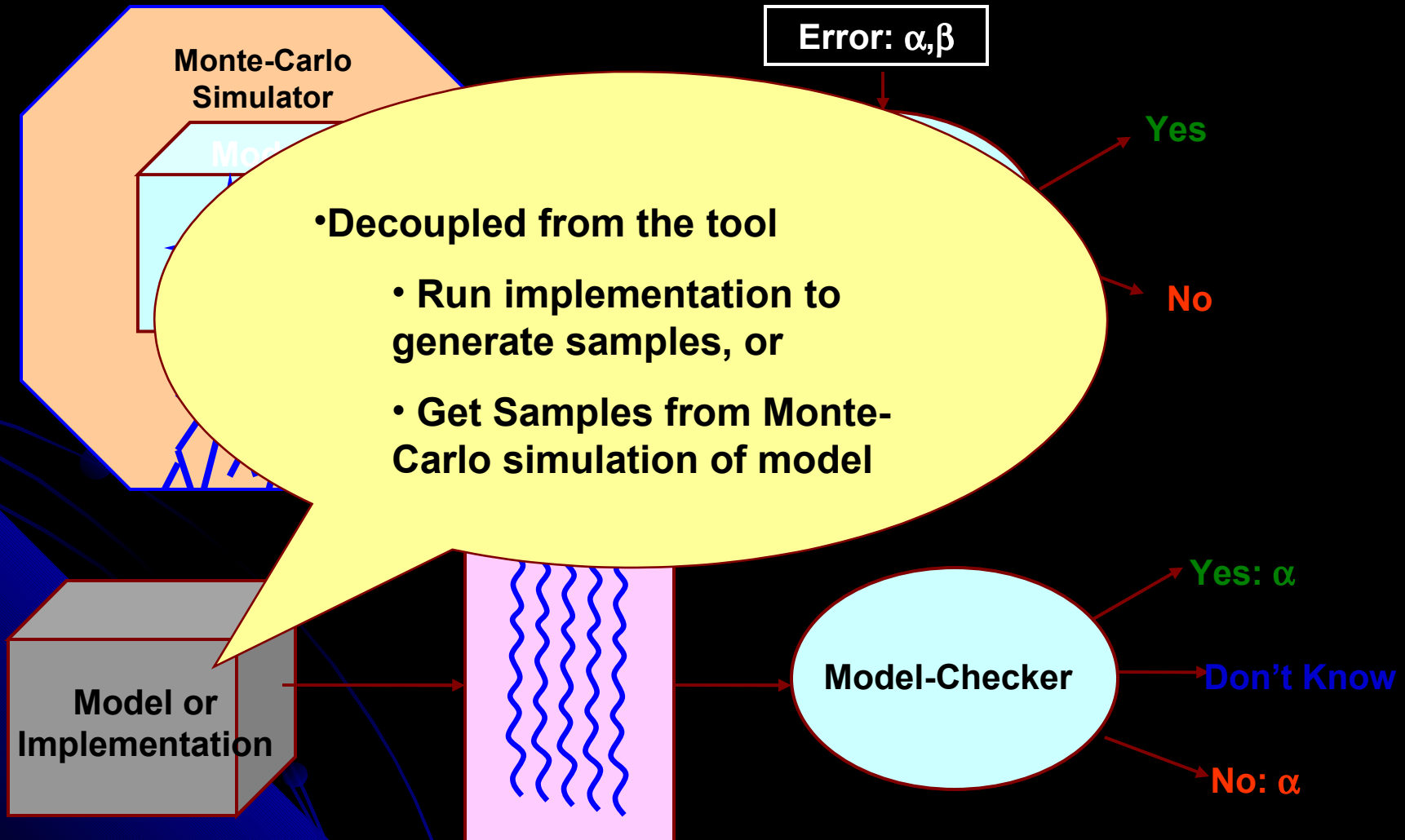
Not estimation!

Error Bounds

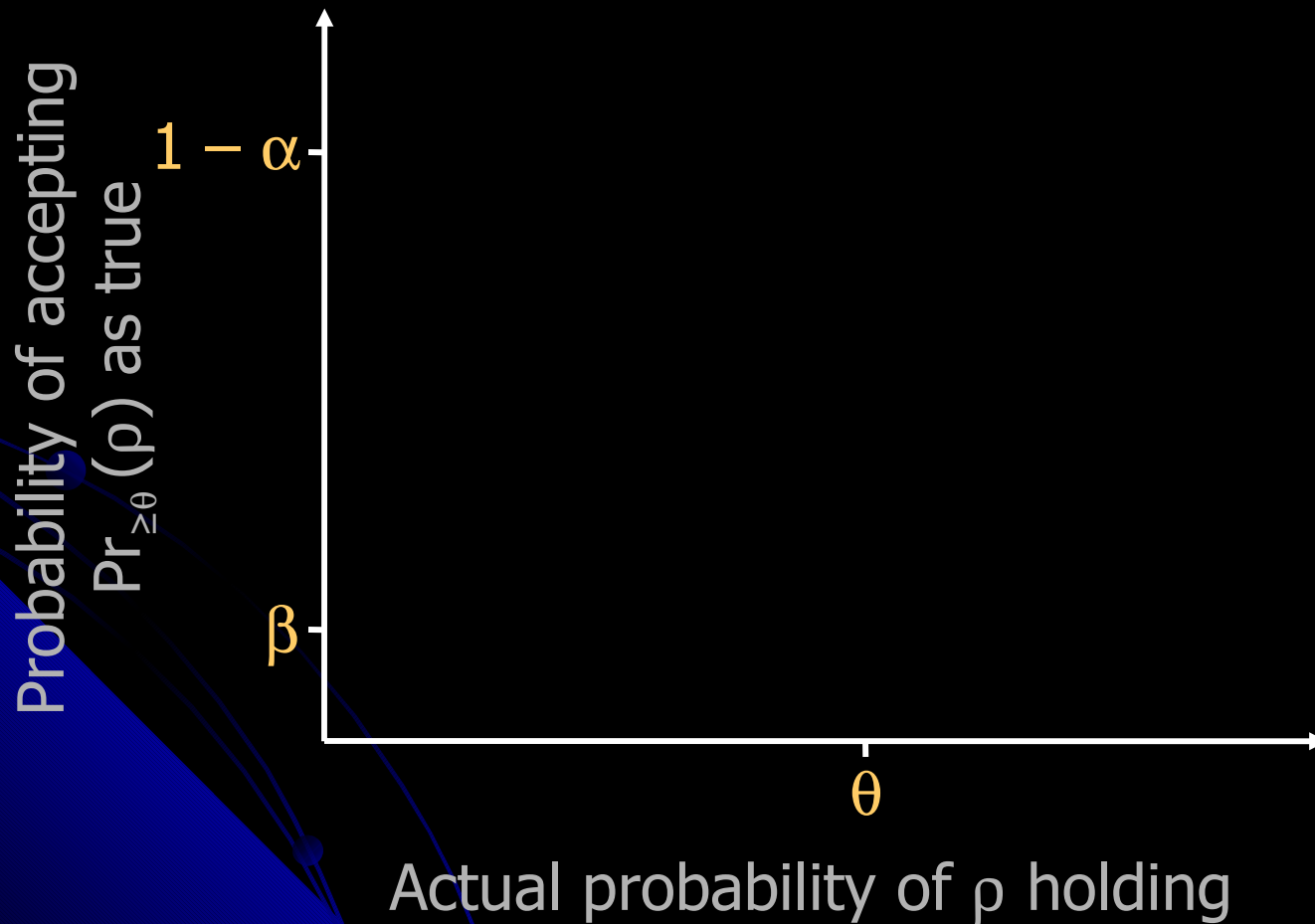
- Probability of false negative: $\leq \alpha$
 - We say that φ is false when it is true
- Probability of false positive: $\leq \beta$
 - We say that φ is true when it is false



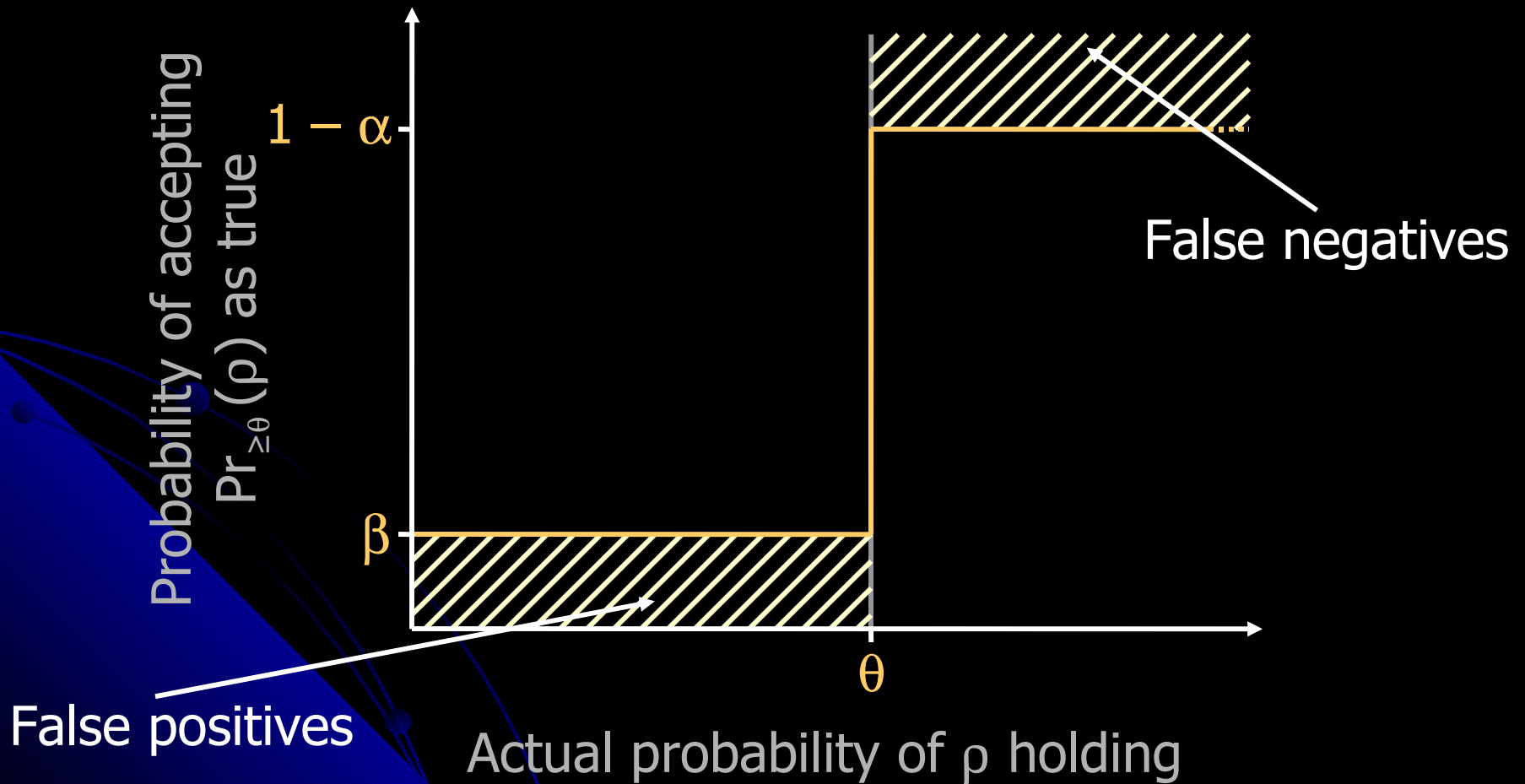
Statistical Approach



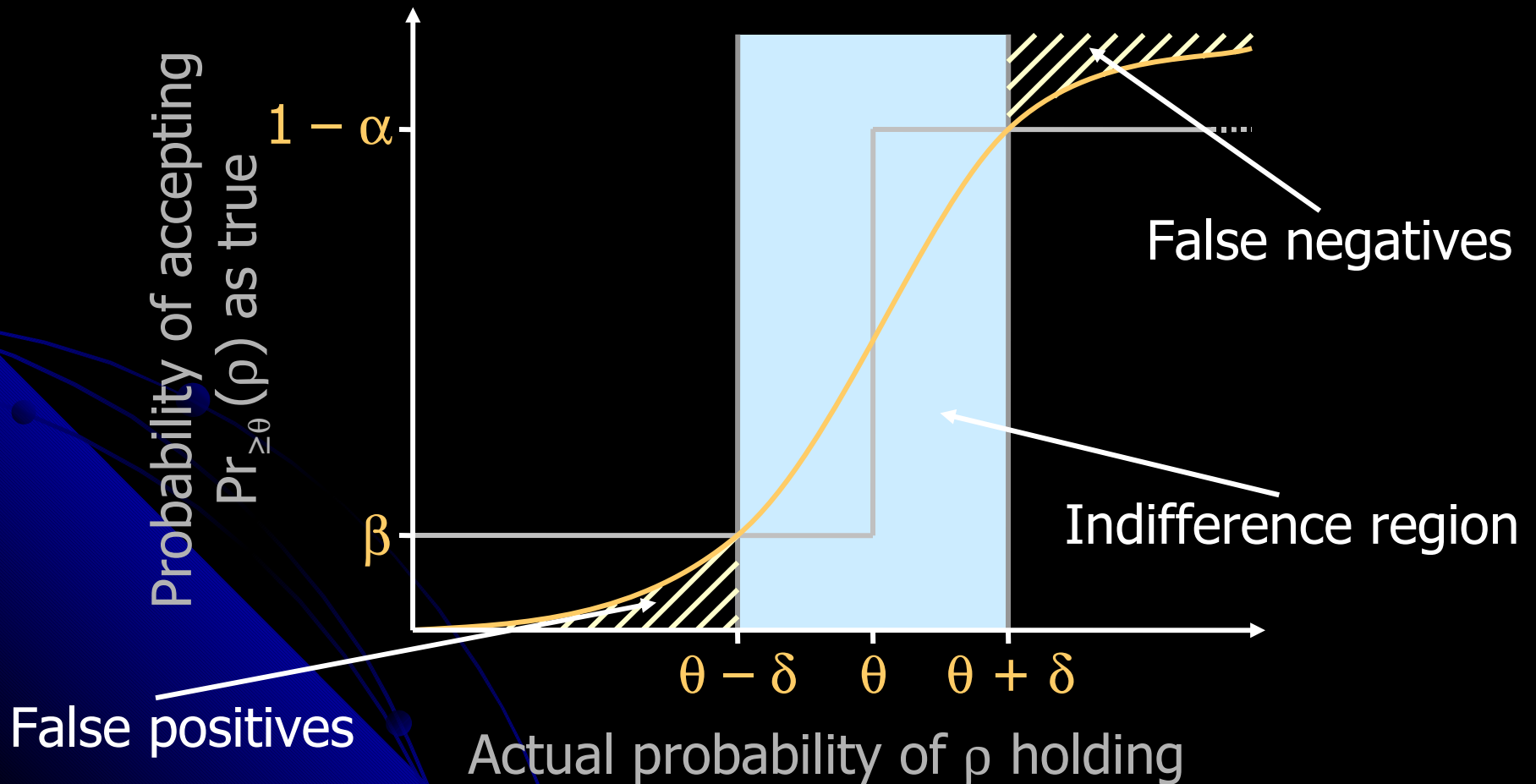
Performance of Test



Ideal Performance

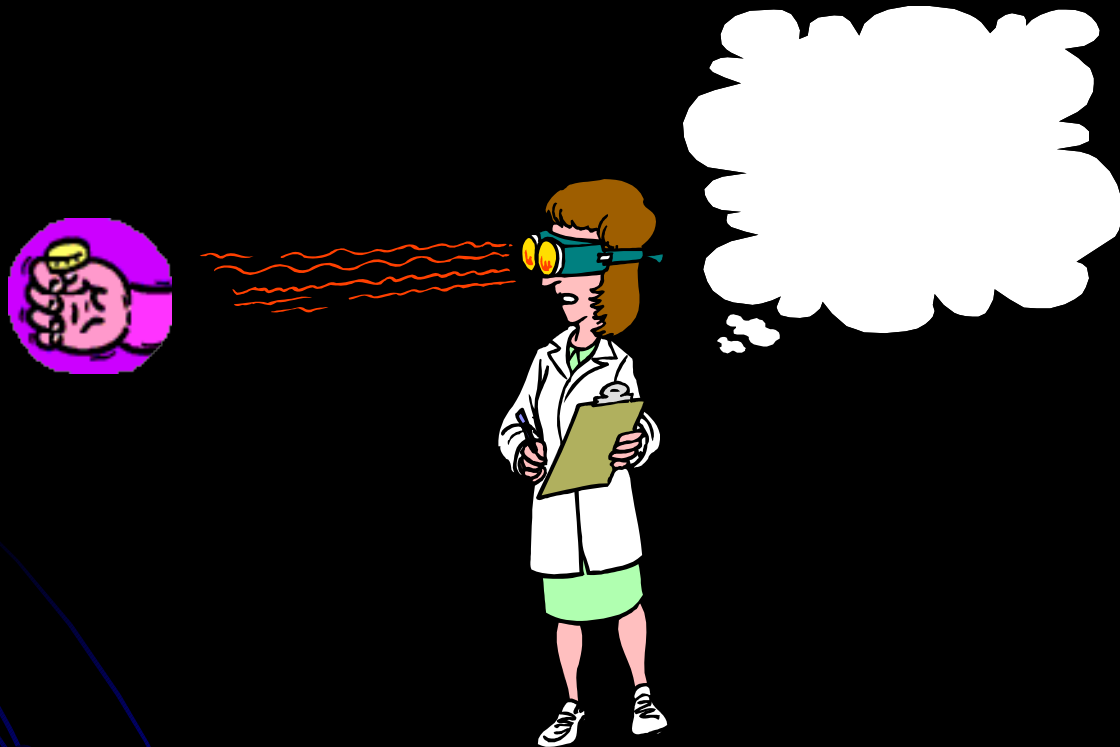


Actual Performance

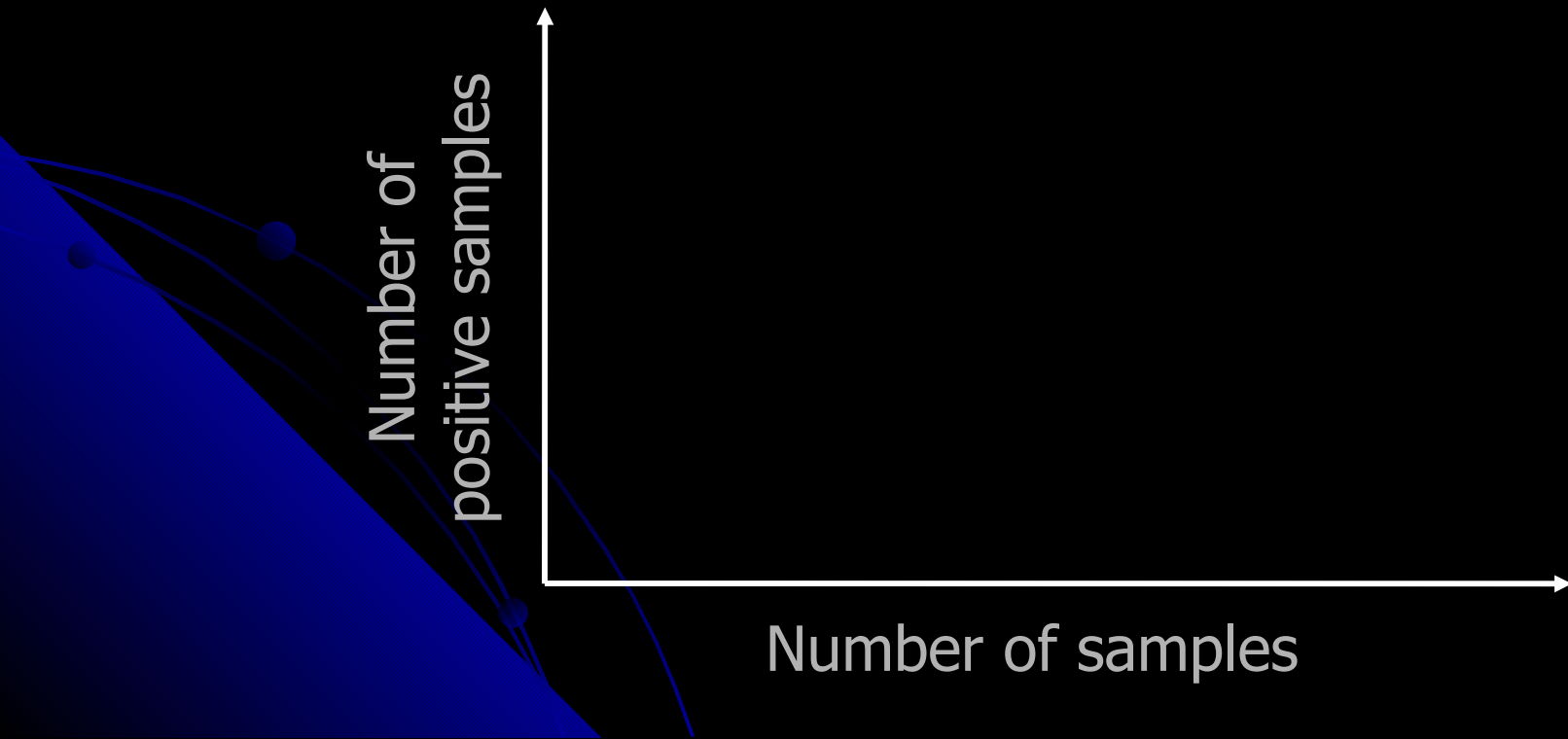


Sequential Acceptance Sampling [Wald 1945]

- Hypothesis: $\Pr_{\geq \theta}(\rho)$

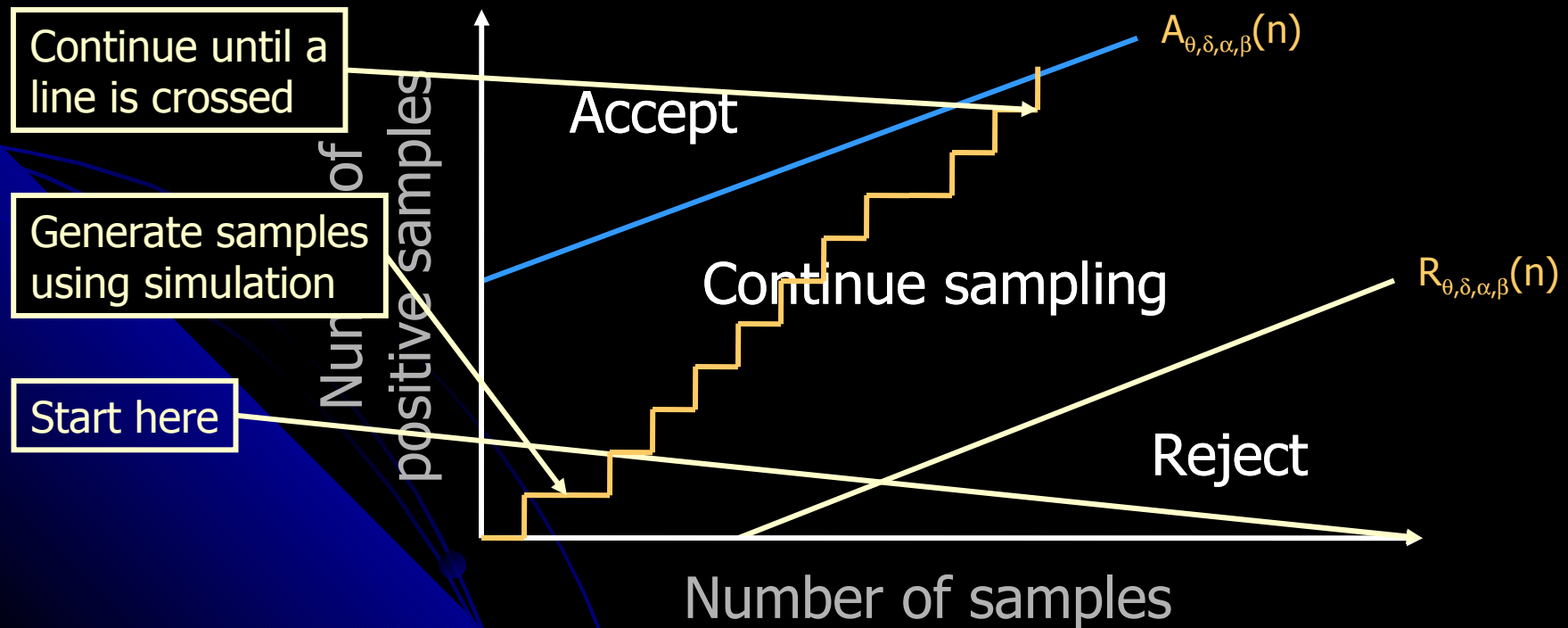


Graphical Representation of Sequential Test



Graphical Representation of Sequential Test

- We can find an **acceptance line** and a **rejection line** given θ , δ , α , and β



Verifying Probabilistic Properties

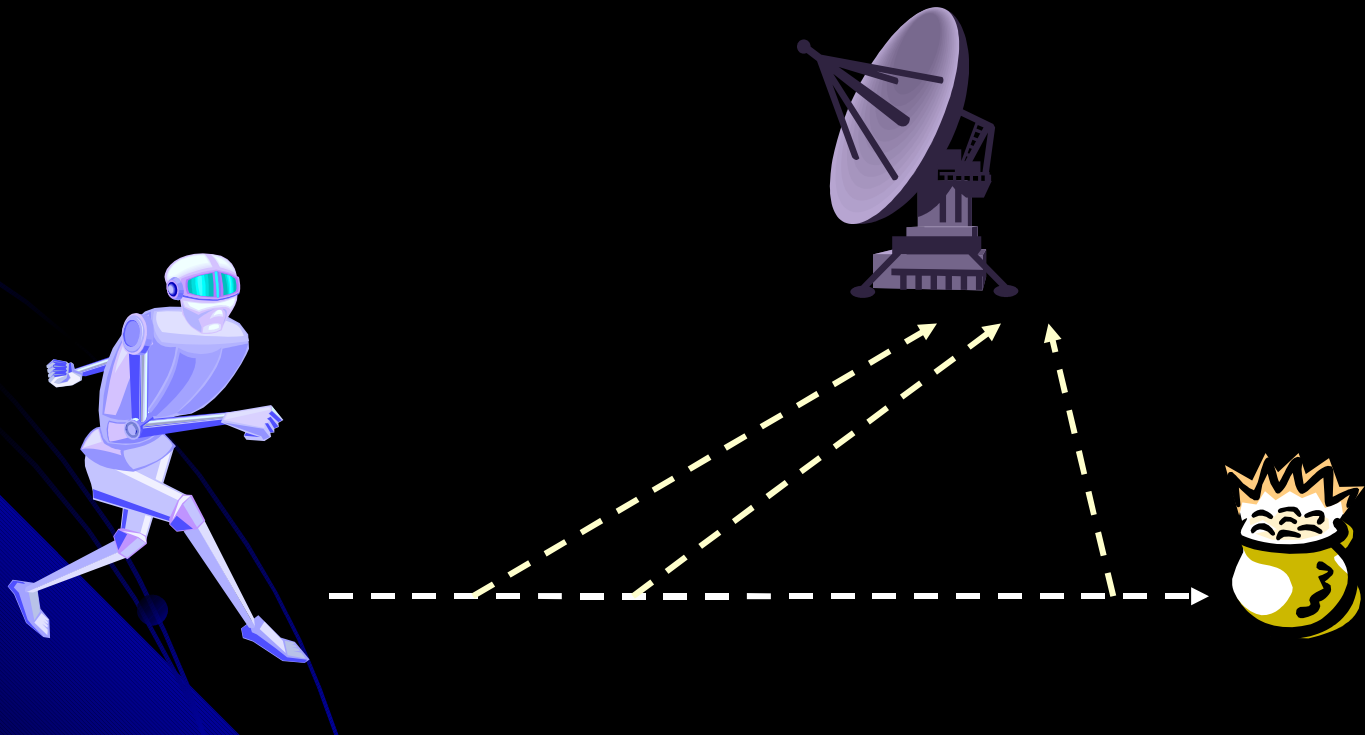
- Verify $\Pr_{\geq\theta}(\rho)$ with error bounds α and β
 - Generate sample paths using simulation
 - Verify ρ over each sample path
 - If ρ is true, then we have a positive sample
 - If ρ is false, then we have a negative sample
 - Use sequential acceptance sampling to test the hypothesis $\Pr_{\geq\theta}(\rho)$

Complexity of Statistical Solution Method

- Number of samples
 - Complex dependency on θ , δ , α , and β
- Length of sample paths
 - Expected length at most $q \cdot T$
 - Shorter paths if $\neg\varphi_1 \vee \varphi_2$ is satisfied early
- No direct dependence on size of state space

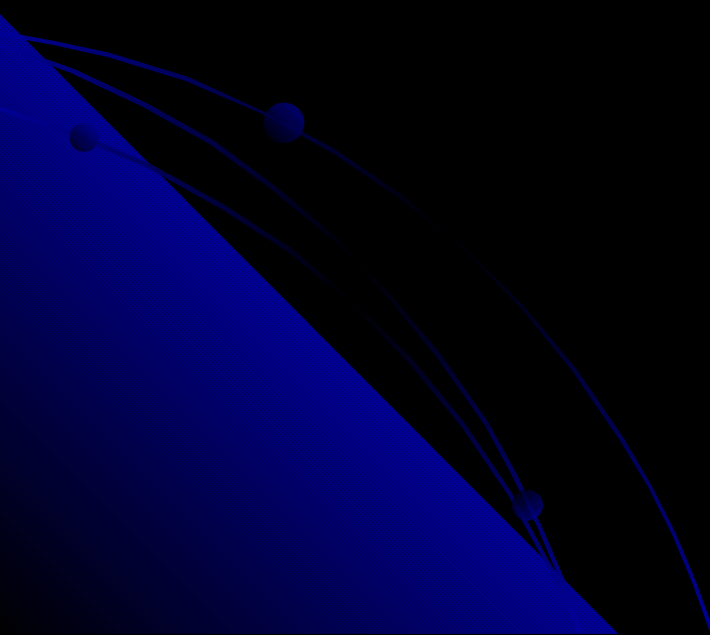
Nested Queries

- $\Pr_{\geq 0.9}(\Pr_{\geq 0.5}(\text{true } U^{\leq 5} \text{ comm.}) U^{\leq 20} \text{ gold})$
- Statistical method: hypothesis testing problem in each state along a path!



Nested Queries: Combining the Methods

- Verify inner probabilistic statement for all states using numerical method
- Verify outer probabilistic statement using statistical method



Examples (1)

- **Ymer SMC Tool**

- Ymer implements the statistical model checking techniques, based on discrete event simulation and acceptance sampling, for CSL model checking developed by Younes and Simmons [12].
- To verify a CSL path formula, Ymer uses discrete event simulation to generate sample execution paths and verifies the path formula ' over each execution path.
- The verification result over a sample execution path is the outcome of a chance experiment (Bernoulli trial), which is used as an observation for an acceptance sampling procedure. Ymer implements both sampling with a fixed number of observations and sequential acceptance sampling.
- Ymer includes support for distributed acceptance sampling, i.e. the use of multiple machines to generate observations, which can result in significant speedup as each observation can be generated independently.

Examples (2)

- **VESTA SMC Tool**

The statistical model-checking algorithm developed on this tool for stochastic models has at least three advantages over previous work.

1-The algorithm can model check CSL formulas which have unbounded untils.

2-The algorithm is inherently parallel; this parallelism is facilitated by the fact that we use simple statistical hypothesis testing rather than sequential hypothesis testing.

3-The algorithm does not suffer from the state-space explosion problem since it is not needed to store the intermediate states of an execution.

However, this algorithm also has at least two limitations.

1-The algorithm cannot guarantee the accuracy that numerical techniques achieve.

2-if we try to increase the accuracy by making the error bounds very small, the running time increases considerably. Thus this technique should be seen as an alternative to numerical techniques to be used only when it is infeasible to use numerical techniques, for example, in large-scale systems.

Summary

- Benefits of numerical method
 - All states at the price of one
 - Steady-state detection
 - High accuracy
- Benefits of statistical method
 - Easy to trade accuracy for speed
 - Scales well with size of state space
 - Parallelizable
 - Model independent

References

- Younes, H. L. S., Kwiatkowska, M., Norman, G., and Parker, D. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer*, 2005. Forthcoming.
- Younes, H. L. S. and Simmons, R. G. Probabilistic verification of discrete event systems using acceptance sampling. In *Proc. 14th International Conference on Computer Aided Verification*, volume 2404 of LNCS, pages 223–235. Springer, 2002.

Thanks For Your Attention

