통계적 모델 체킹
Statistical Model Checking

Youngjoo Kim
KAIST Provable SW Lab

Motivation

- A system has various quality requirements
  - reliability, performance, safety, resource usage, etc…
- Conventional SW model checking verifies for qualitative requirements
  - Is the system safe?
    - E.g. the temperature of the system should be less than 100 degree.
  - Does the network satisfy liveness?
    - E.g. a message should be delivered someday.
- A physical system has many quantitative requirements
  - How reliable is my car's Bluetooth network?
  - How efficient is my phone's power management?
  - How much is the minimum expected battery capacity?

➔ Probability model checking (PMC) obtains probability that a property is satisfied

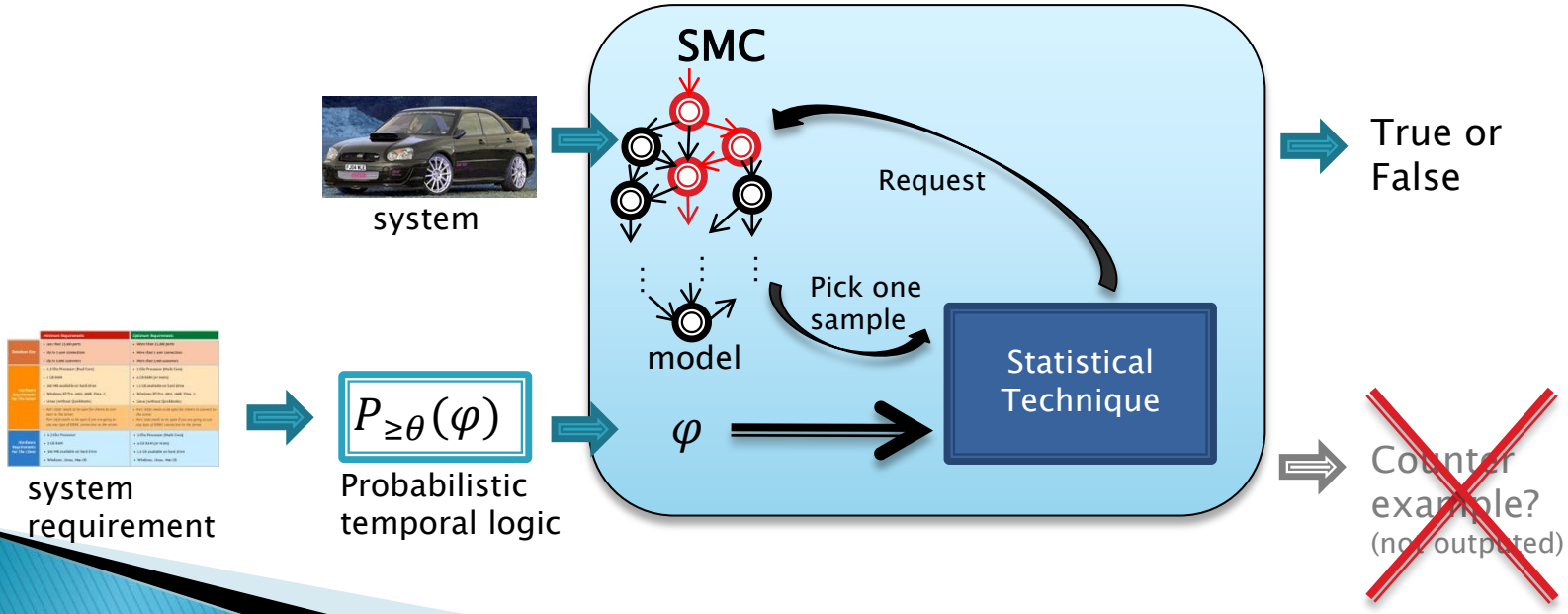
Motivation

- But, PMC has limited uses in practice.
  - State explosion problem

- Estimation can provide better practical uses
  - **Statistical Model Checking** (SMC)
    - Approximate probability of a system satisfying a property
    - Require less time than rigorous model checking
    - Has low error in computation

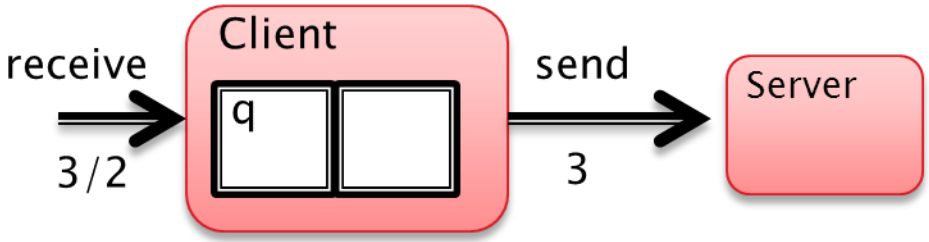
Statistical Model Checking

- Goal
  - Provide probabilistic guarantees of complex target system's correctness using a small number of simulations

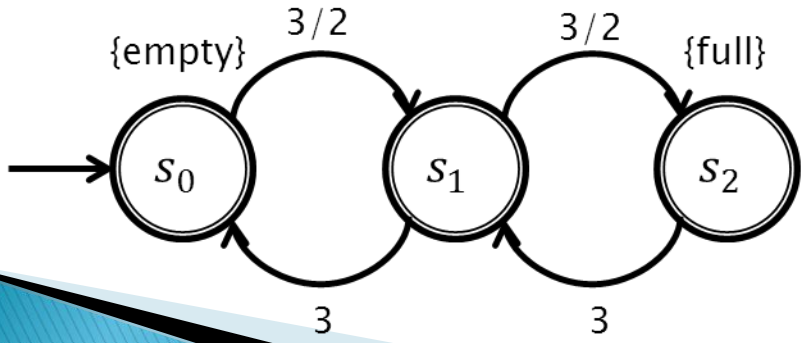


Example – Simple Sensor

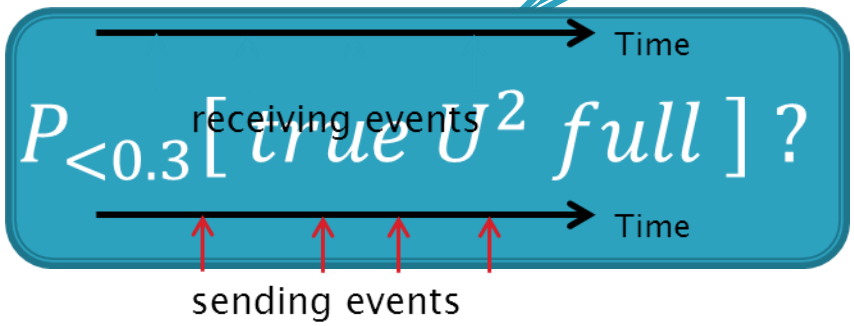
- initially the queue of client is empty
- maximum size of the queue is 2



- state space: S=$\{s_i\}_{i=0..2}$ where s_i: i jobs in queue

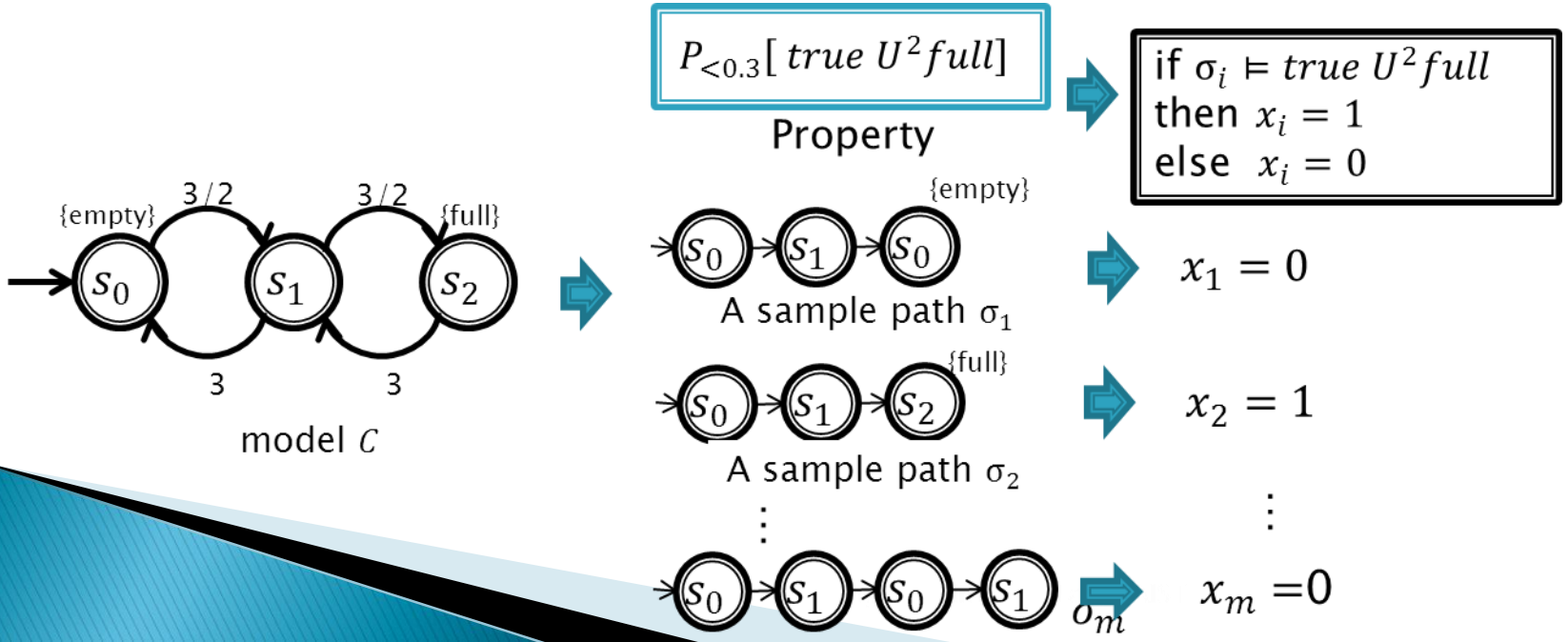


# of events ~ Poisson



Sequential Probability Ratio Test

- Observe randomly generated executions (sample)
- When do we need to stop sampling ?
  - Determine using statistical techniques
  - **Sequential Probability Ratio Test (SPRT)** [Wald, 1945]
    - Generate the smallest number of samples dynamically
    - Guarantees to bound error probability



Experiment – Simple Sensor

- Result
  - The property $P_{<0.3}[\,true\ U^2\ full]$ is rejected, i.e. the probability that the queue becomes full in 2 seconds is greater than equal to 0.3.
- Statistics

Error probability	# of samples	Verification time (sec)
10^{-1}	100.9	0.0004
10^{-2}	215.3	0.0008
10^{-4}	439.2	0.0008
10^{-8}	894.9	0.0020

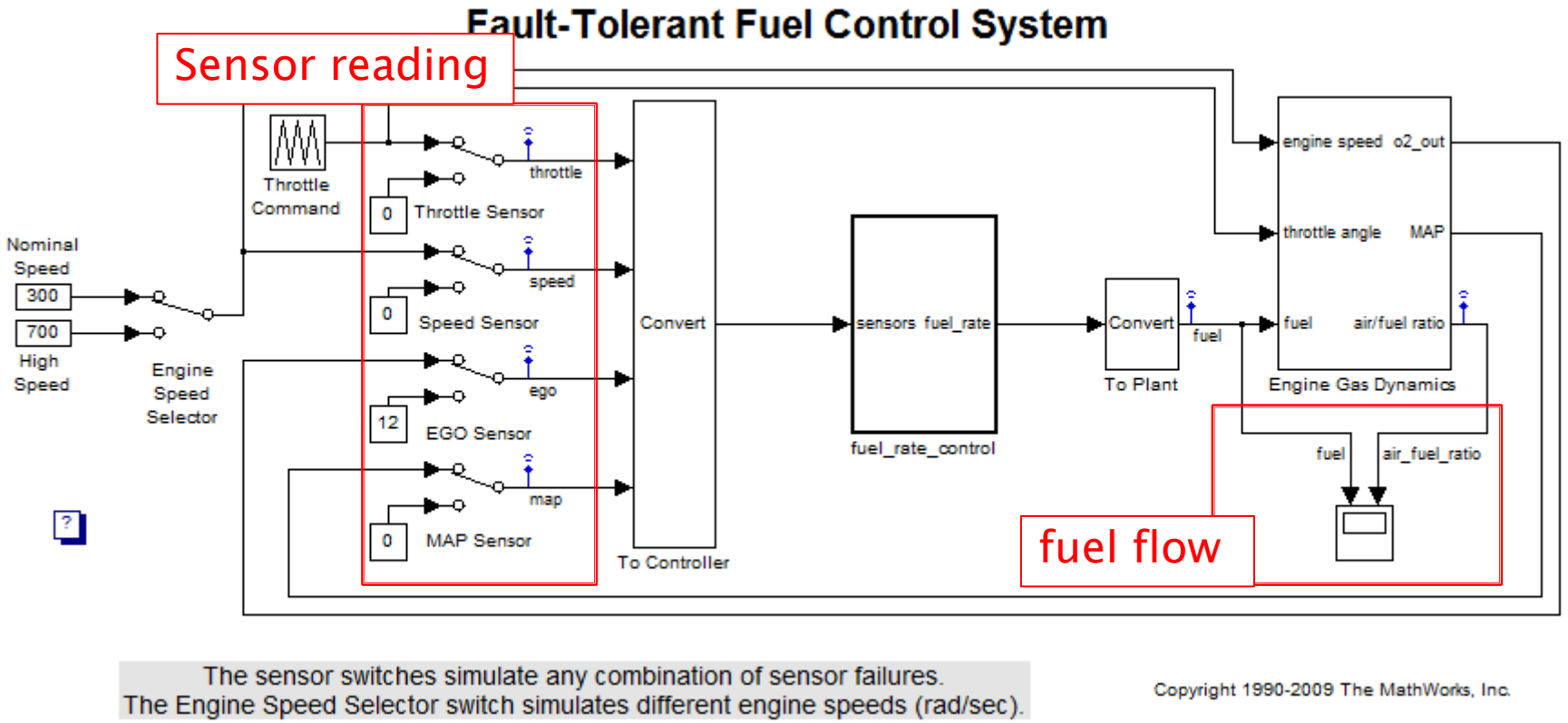
Accuracy *Increasing* *Increasing*

Real World Applications

- Control System

Property

$$P_{\geq\theta}(\neg F^{100} G^{1}(FuelFlowRate = 0))$$



Real World Applications

- Result
  - Given property $P_{\geq\theta}(\neg F^{100} G^{1}(FuelFlowRate = 0))$,

Fault rates	Error Bound	Threshold θ											
		0.7				0.9				0.99			
		n	x	isaccept	time	n	x	isaccept	time	n	x	isaccept	time
(3 7 8)	0.1	5	5	Yes	15.24	42.2	37.2	Yes	129.1	21	19	No	64.49
	0.01	26	23	Yes	79.17	3917	3476	No	11973	27	24	No	83.09
	0.001	48	42	Yes	147.8	4013	3551	No	12249	35.2	30.8	No	108.4
(10 8 9)	0.1	4	4	Yes	12.17	17	16	Yes	51.76	8	7.8	Yes	24.5
	0.01	15	14	Yes	44.42	56.6	54.2	Yes	172.4	108.6	103.8	No	332.1
	0.001	23	22	Yes	68.72	101	97.8	Yes	308	409.8	395	No	1262
(20 10 20)	0.1	4	4	Yes	12.12	7	7	Yes	21.32	9	9	Yes	27.37
	0.01	12	12	Yes	37.08	26.2	26	Yes	79.7	268.4	266.4	Yes	818.8
	0.001	16	16	Yes	48.72	44	44	Yes	134	12354	12244	Yes	43330

Conclusion

- With smaller cost, we can obtain the result of satisfiability of quantitative requirements for complex target systems without state explosion problem.

- **Statistical Model Checking** can be the standard criteria for measuring software's reliability.