Model Checking

Lecture:

Roderick Bloem, Bettina Könighofer, Stefan Pranger

Fabian Russold

Practicals:

Johannes Haring

Today

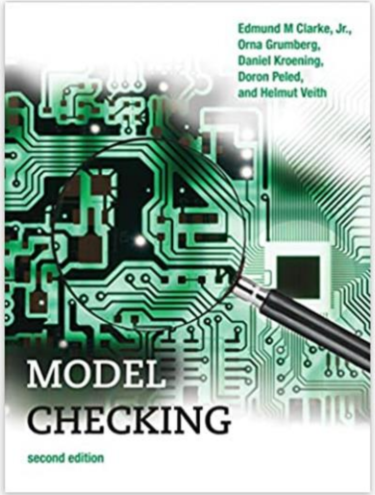
- Administrative
- Motivation
- Modeling

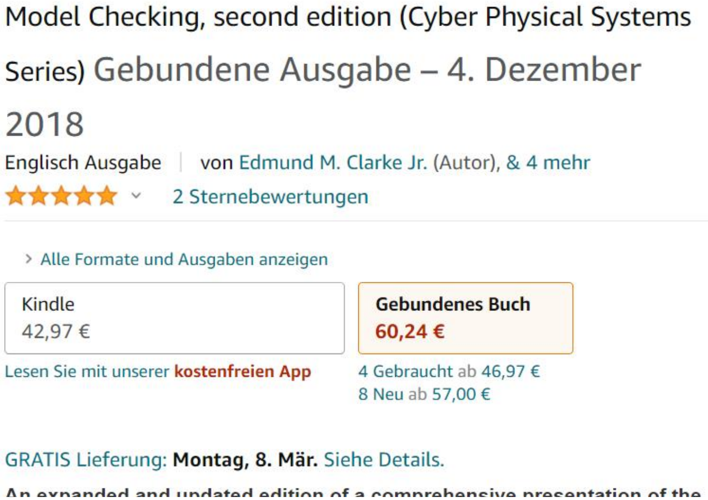
Material & Communications

Lecture: Tuesday 12-13:30, HS i11 (ICK1002H)
Practicals: Right after, only if there is something to discuss
Question Hours: Tuesdays after class.

Webpage: https://www.iaik.tugraz.at/course/model-checking-705080-sommersemester-2025/
Discord: https://discord.gg/FDcxjR728N, Channel MC (robot)
Email: bettina.koenighofer@tugraz.at, roderick.bloem@tugraz.at, stefan.pranger@tugraz.at
johannes.haring@tugraz.at, fabian.russold@student.tugraz.at
Teach Center: **https://tc.tugraz.at/main/course/view.php?id=5645**

Books




Model Checking, second edition (Cyber Physical Systems Series) Gebundene Ausgabe – 4. Dezember 2018

Englisch Ausgabe | von Edmund M. Clarke Jr. (Autor), & 4 mehr

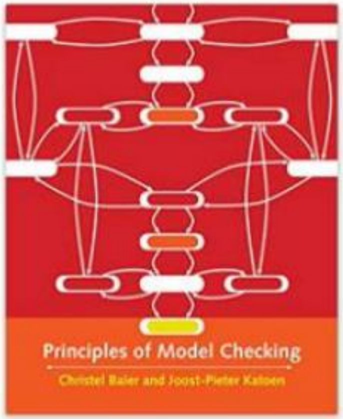
★★★★★ 2 Sternebewertungen

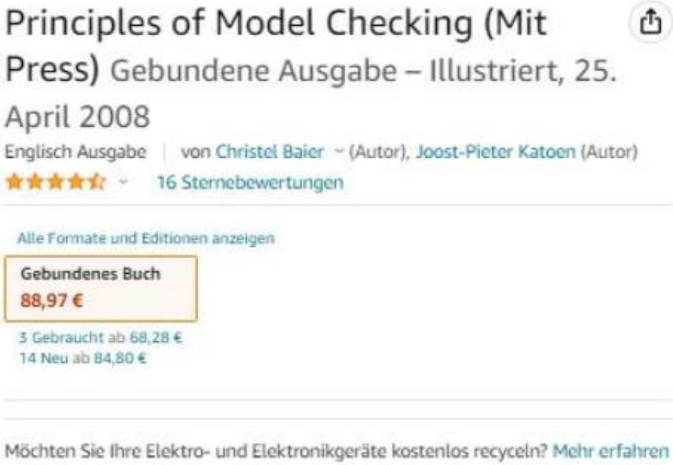
› Alle Formate und Ausgaben anzeigen

Kindle 42,97 €	Gebundenes Buch 60,24 €
Lesen Sie mit unserer kostenfreien App	4 Gebraucht ab 46,97 € 8 Neu ab 57,00 €

GRATIS Lieferung: **Montag, 8. Mär.** Siehe Details.

An expanded and updated edition of a comprehensive presentation of the


Principles of Model Checking
Christel Baier and Joost-Pieter Katoen


Principles of Model Checking (Mit Press) Gebundene Ausgabe – Illustriert, 25. April 2008

Englisch Ausgabe | von Christel Baier (Autor), Joost-Pieter Katoen (Autor)

★★★★☆ 16 Sternebewertungen

Alle Formate und Editionen anzeigen

Gebundenes Buch
88,97 €

3 Gebraucht ab 68,28 €
14 Neu ab 84,80 €

Möchten Sie Ihre Elektro- und Elektronikgeräte kostenlos recyceln? Mehr erfahren

Another good book:

- Clarke, Henzinger, Veith, Bloem, *Handbook of Model Checking,* Springer 2018

How to get a grade?

Lecture: Two options

1. Do an exam, **or**
2. Participate in class and do weekly homework. Course grade = homework grade

(Not happy with homework grade? Take exam!)

- Miss at most 2 classes

Practical:

- Individual work
- Three assignments with point distribution 30/40/30
- Final interviews

Homework

Weekly homework

- uploaded just before the lecture
- deadline = 9 am day bof the next lecture

Individually or **groups of two**.

- You can do each homework with a different group.

Submission

- In TeachCenter
- If handwritten, use **clear writing** and a good scan

Marks

- available within 1 week of submission deadline in TeachCenter
- Final mark = **average of all homework.**
- You can skip homework **at most 2** weeks. (Skipped homework = 0 points)

- Questions
  - email: Fabian.Russold@student.tugraz.at
  - I also actively answer questions in the discord channel

Lecture Schedule

Date	Topic	Lecturer
04 Mar	Intro	Roderick
11 Mar	SAT-Based Model Checking (BMC, k-induction) – Chapter 10	Roderick
18 Mar	SAT-Based Model Checking (interpolation) – Chapter 10	Roderick
25 Mar	SAT-Based Model Checking (PDR) – Chapter 10	Roderick
01 Apr	Temporal Logic – Chapter 4	Bettina
08 Apr	CTL Model Checking – Chapter 5	Bettina
29 Apr	UPPAAL - MC for Timed Properties	Florian Lorber
06 May	LTL Model Checking -Chapter 7	Bettina
13 May	LTL Model Checking - Chapter 7 + Reactive Synthesis	Bettina
20 May	Probabilistic Model Checking - Chapter 10 - PRISM & Reachability in Markov Chains	Stefan
27 May	Probabilistic Model Checking – Chapter 10 – PCTL and MDPs	Stefan
03 June	Statistical Model Checking	Bettina
17 June	Security Verification	Roderick
24 June	Reserved slot	---

Practicals

- Separate course (Übung), so please register in Teach Center
- Group Size: 1
- 3 Submissions
  - Warmup: Getting to know Z3
  - Hardware Model Checker
    - BMC
    - K-Induction
- Questions can be asked on Discord or Tuesdays after the lecture

Practicals Schedule

Date	Type	Topic	Lecturer
4 Mar	Lecture	Intro (merged with lecture)	Roderick
11 Mar	Handout	Warmup Exercise	Johannes
18 Mar	Tutorial	Introduction to Z3	Johannes
25 Mar	Handout	BMC Exercise	Johannes
01 Apr	Question Hour	Warmup Exercise	Johannes
06 Apr	**Deadline**	**Warmup Deadline**	**---**
08 Apr	Tutorial	Hardware and Verilog	Johannes
06 May	Handout	K-Induction Exercise	Johannes
13 May	Question Hour	Question Hour BMC	Johannes
25 May	**Deadline**	**BMC Deadline**	
03 Jun	Question Hour	Question Hour K-Induction	Johannes
06 Jun	**Deadline**	**K-Induction Deadline**	**---**

737 Max

“The people who wrote the code for the original MCAS system were obviously terribly far out of their league and did not know it”. (Gregory Reed Travis)
346 deaths

Deductive Verification?

```
r = false;
i = 0;
while(i != n) {

  if(a[i] == x) {

    r = true;
  } else {
  }
  i = i + 1;
}
```

Deductive Verification?

```
{false == false} ↔ {true}
r = false;
{r == ($\bigvee_{j=0}^{-1}$ a[j] == x)} ↔ {r == false}
i = 0;
{r == ($\bigvee_{j=0}^{i-1}$ a[j] == x)}
while(i != n) {
  {(r == ($\bigvee_{j=0}^{i-1}$ a[j] == x)) ∧ i != n}
  {r == ($\bigvee_{j=0}^{i-1}$ a[j] == x)}
  if(a[i] == x) {
    {(r == ($\bigvee_{j=0}^{i-1}$ a[j] == x)) ∧ a[i] == x}
    {(true == ($\bigvee_{j=0}^{i}$ a[j] == x)) ∧ a[i] == x} ↔ {true ∧ a[i] == x} ↔ {a[i] == x}
    r = true;
    {r == ($\bigvee_{j=0}^{i}$ a[j] == x)}
  } else {
   {(r == ($\bigvee_{j=0}^{i}$ a[j] == x)) ∧ a[i] != x} ↔ {(r == ($\bigvee_{j=0}^{i-1}$ a[j] == x)) ∧ a[i] != x}
  }
  {r == ($\bigvee_{j=0}^{i}$ a[j] == x)}
  i = i + 1;
  {r == ($\bigvee_{j=0}^{i-1}$ a[j] == x)}
}
{r == ($\bigvee_{j=0}^{n-1}$ a[j] == x) ∧ i == n} ↔ {r == ($\bigvee_{j=0}^{i-1}$ a[j] == x) ∧ i == n}
{r == ($\bigvee_{j=0}^{n-1}$ a[j] == x)}
```

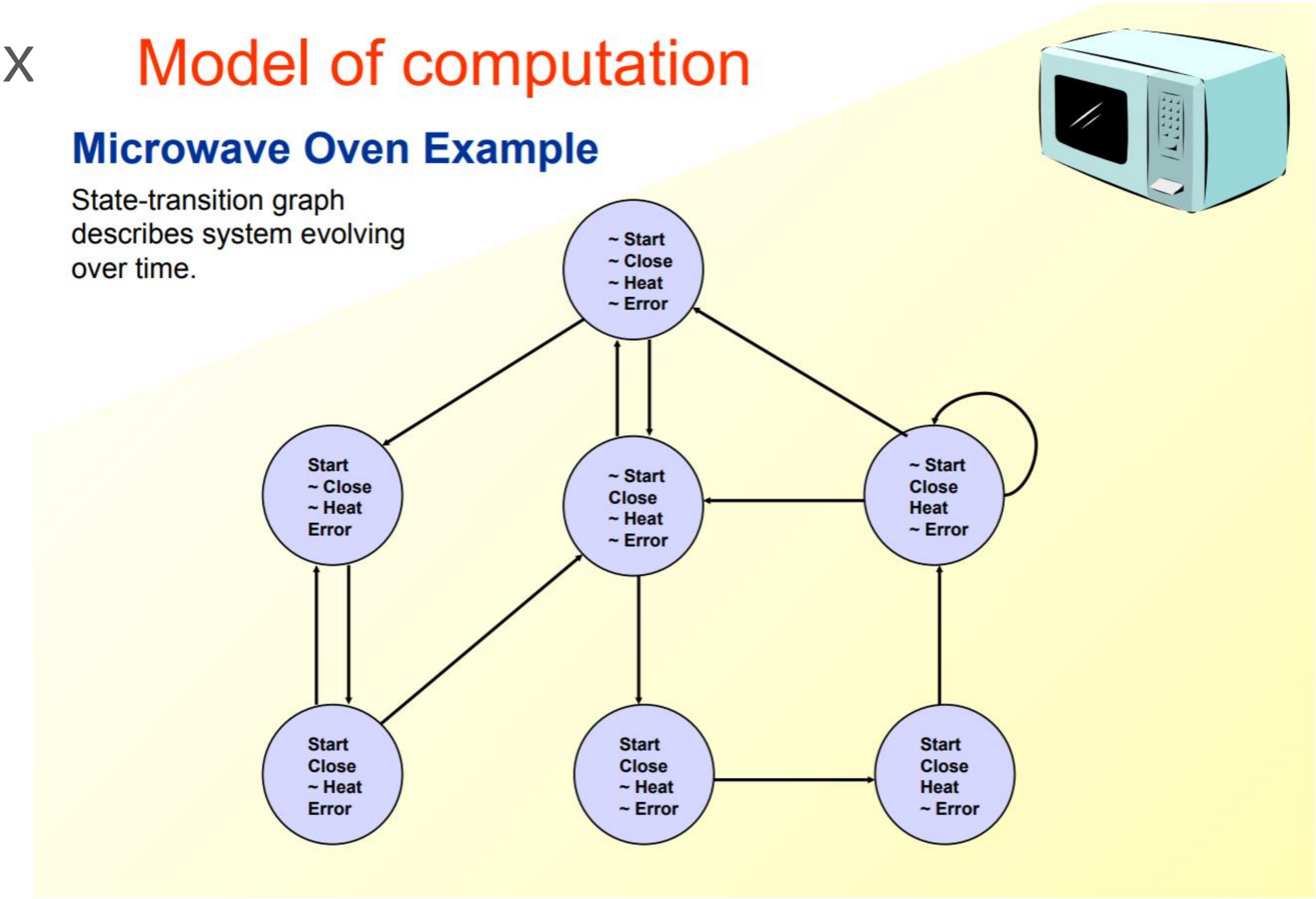
- (Manual) Proofs
- No diagnostics
- Full specifications
- Concurrency is hard

(But: things have gotten better!)

Automatic Verification!

- Program = state machine = graph
- Bug hunting = efficient graph search
- “Interesting” properties = “complicated” graph searches
  - Need language to express interesting things!
- But how to search a graph efficiently?

Ex



Model of computation

Microwave Oven Example

State-transition graph describes system evolving over time.

~ Start
~ Close
~ Heat
~ Error

Start
~ Close
~ Heat
Error

~ Start
Close
~ Heat
~ Error

~ Start
Close
Heat
~ Error

Start
Close
~ Heat
Error

Start
Close
~ Heat
~ Error

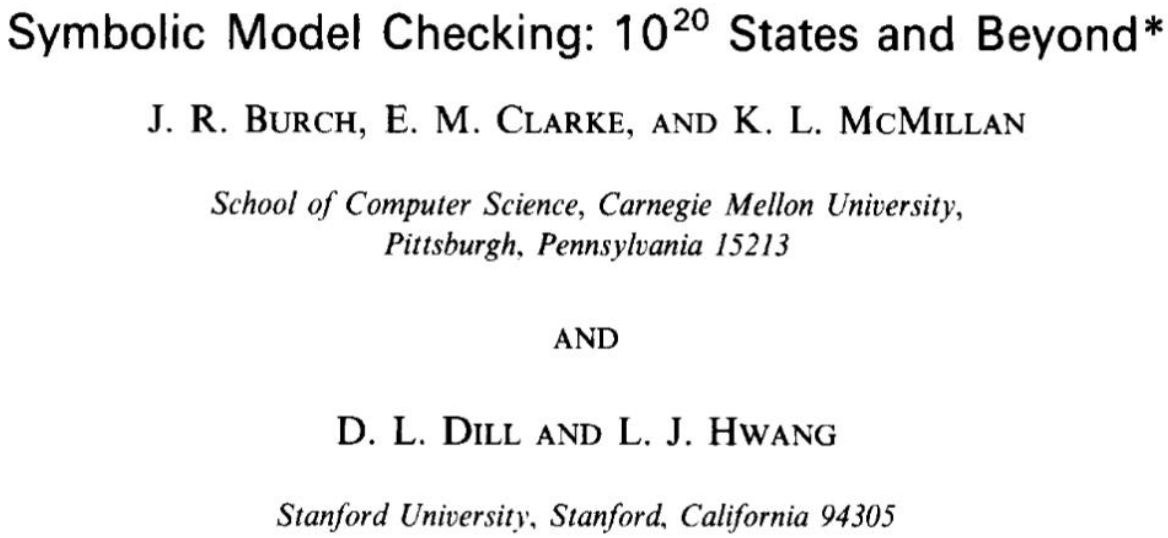
Start
Close
Heat
~ Error

What properties are interesting?

Slide by Ed Clarke

Efficiency

- 1981: EMC Model checker ~10^4 states
- 1992 BDDs:



Symbolic Model Checking: 10^{20} States and Beyond*

J. R. BURCH, E. M. CLARKE, AND K. L. MCMILLAN

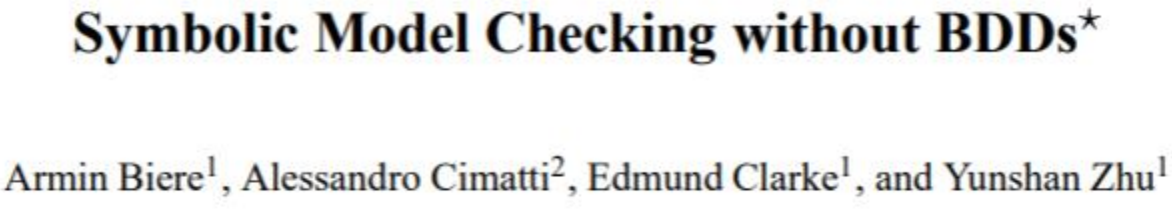
School of Computer Science, Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213

AND

D. L. DILL AND L. J. HWANG

Stanford University, Stanford, California 94305

- 1999 SAT:

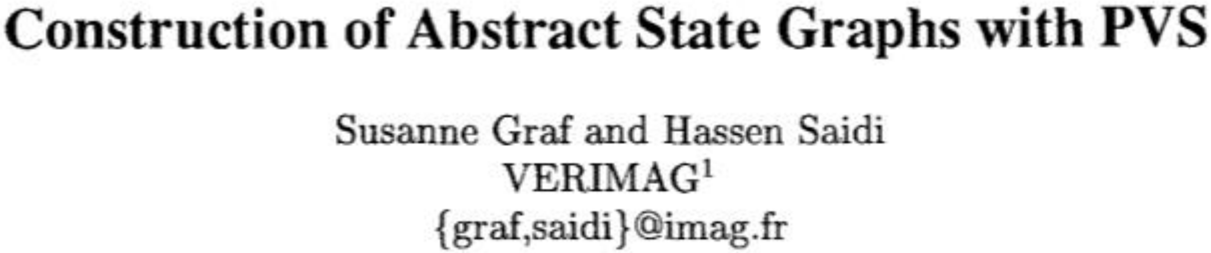


Symbolic Model Checking without BDDs*

Armin Biere[1], Alessandro Cimatti[2], Edmund Clarke[1], and Yunshan Zhu[1]

Efficiency

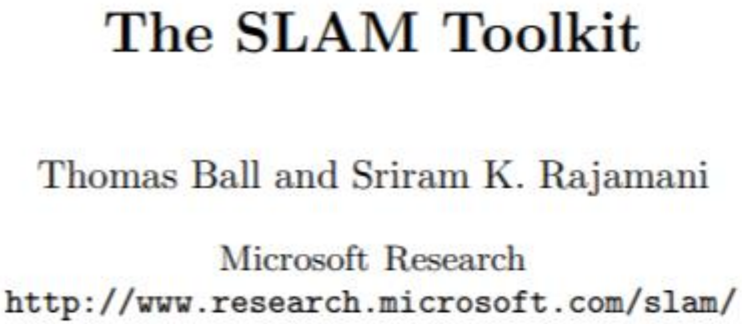
1992 Abstraction



Construction of Abstract State Graphs with PVS

Susanne Graf and Hassen Saidi
VERIMAG[1]
{graf,saidi}@imag.fr

~1995: Partial Order Reduction
~2000: Software



The SLAM Toolkit

Thomas Ball and Sriram K. Rajamani

Microsoft Research
`http://www.research.microsoft.com/slam/`

More than Microwave Ovens?

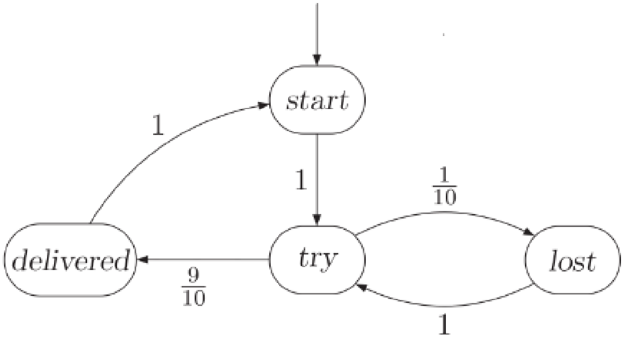
- Amazon Web Services
  - S3, DynamoDB, EBS, lock manager
  - https://assets.amazon.science/67/f9/92733d574c11ba1a11bd08bfb8ae/how-amazon-web-services-uses-formal-methods.pdf
- Facebook
  - Static Analysis https://research.facebook.com/publications/moving-fast-with-software-verification/
- Intel
  - Security https://community.cadence.com/cadence_blogs_8/b/breakfast-bytes/posts/formally-verifying-processor-security
- Microsoft
  - Device drivers
  - Smart contracts
  - Z3
- Cadence & Synopsys
  - Jasper Formal Verification, VC Formal

EDMUND M. CLARKE, E. ALLEN EMERSON, JOSEPH SIFAKIS

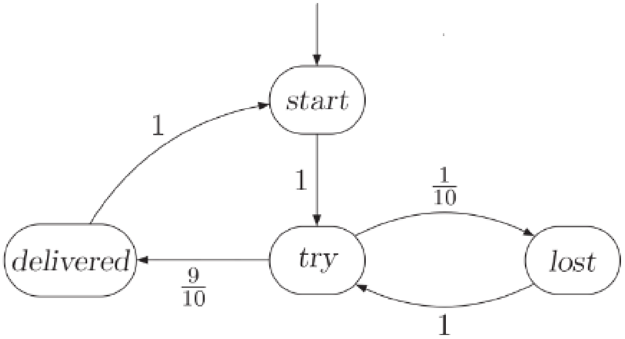
Model Checking: An Automated Quality Assurance Method

Why do we need Probabilities?

- Analysis of Reliability
  - Probability of Failure,
  - Quantify Message Loss,
  - Quantify Arrival Times, …



Why do we need Probabilities?

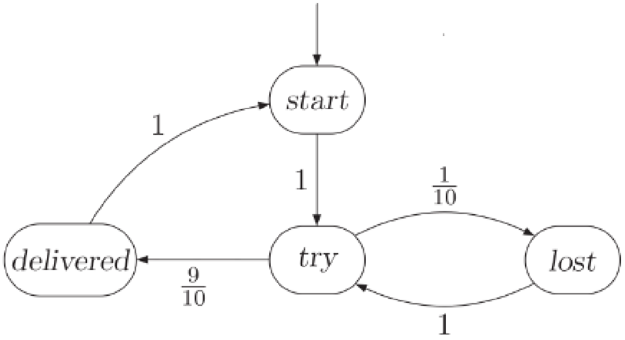


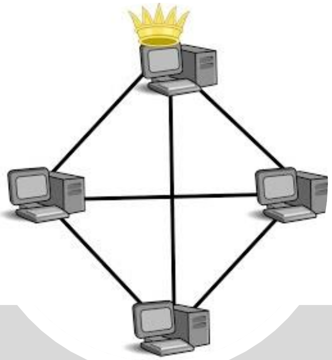
- Analysis of Reliability
  - Probability of Failure,
  - Quantify Message Loss,
  - Quantify Arrival Times, …
- Models of Safety-Critical Systems,
  - Modeling Unknowns,
  - Modeling Faults, …

Why do we need Probabilities?

- Analysis of Reliability
  - Probability of Failure,
  - Quantify Message Loss,
  - Quantify Arrival Times, …
- Models of Safety-Critical Systems,
  - Modeling Unknowns,
  - Modeling Faults, …
- Analysis of Randomized Algorithms,

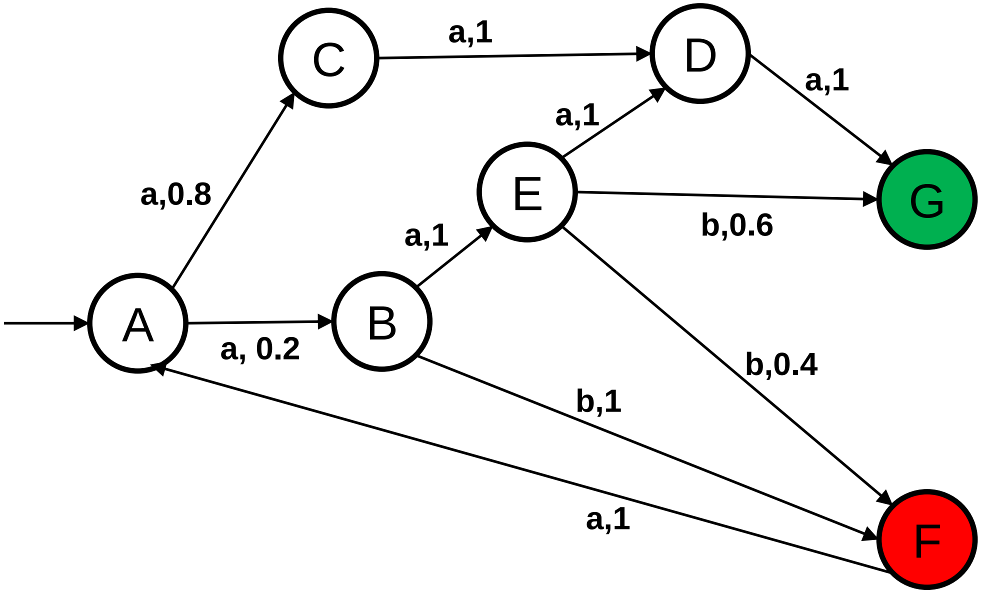
…





Probabilistic Model Checking

- Extend Models with Probabilistic Transitions
- ”Markov Models”

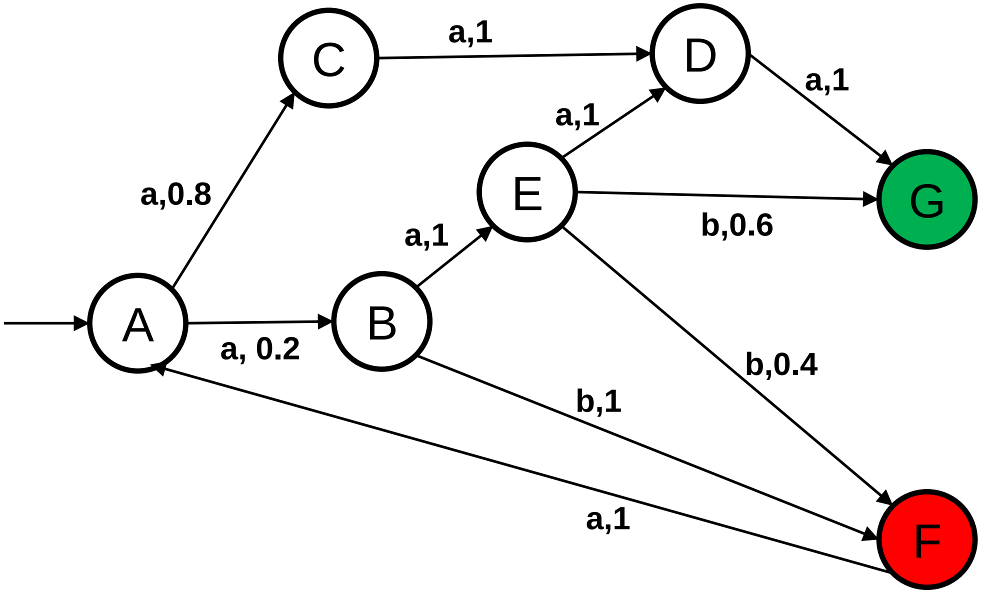


Probabilistic Model Checking

- Formalism to quantify Probabilities

$P_{\geq 0.95}\,(F \rightarrow \text{true}\ U^{<9} G)$

Is the probability of delivering a message withing 9 steps after encountering a failure greater or equal 0.95?



A Probabilistic Model

- Model containing:
  - System Dynamics
  - Controller Decisions
- $P_{<0.01}\ (\boldsymbol{F}\ dist(airplane, centerline) < 200)$?

Topics

Book

- Different Markovian Models,
  - *and* how to compute probabilities of events.
- Modelling Language,
  - *and how to use a probabilistic model checker, and*
- pMC in practice.

storm & tempest