Seminar *Trends in Computer-Aided Verification*

**Introduction**

Summer Semester 2021; April 14, 2021

Thomas Noll et al.

Software Modeling and Verification Group
RWTH Aachen University

https://moves.rwth-aachen.de/teaching/ws-20-21/propro/
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Formal Verification Methods

Formal verification methods

- Rigorous, mathematically based techniques for the specification, development and verification of software and hardware systems
- Aim at improving correctness, reliability and robustness of such systems
Formal Verification Methods

Formal verification methods

- Rigorous, mathematically based techniques for the specification, development and verification of software and hardware systems
- Aim at improving correctness, reliability and robustness of such systems

Classifications

- According to design phase
  - specification, implementation, testing, ...
- According to specification formalism
  - source code, neural networks, Bayesian networks, fault trees, ...
- According to underlying mathematical theories
  - model checking, theorem proving, static analysis, ...
### Areas Covered in this Seminar

#### Topic areas

- Robustness Analysis of Neural Networks
- Analysis of Bayesian Networks
- Synthesizing Quantitative Loop Invariants for Probabilistic Programs
- Formal Approaches to Systems Engineering
Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Goals

Aims of this seminar

- **Independent understanding** of a scientific topic
- Acquiring, reading and understanding **scientific literature**
  - given references sufficient in most cases
- **Writing of your own report** on this topic
  - far more that just a translation/rewording
  - usually an "extended subset" of original literature
    - "subset": present core ideas and omit too specific details (e.g., related work or optimisations)
    - "extended": more extensive explanations, examples, ...
    - discuss contents with supervisor!
- **Oral presentation** of your results
  - can be “proper subset” of report
  - generally: less (detailed) definitions/proofs and more examples
## Requirements on Report

### Your report

- **Independent writing of a report of 12–15 pages**
- **First milestone: detailed outline**
  - rather: overview of structure (section headers, main definitions/theorems) and initial part of main section (one page)
- **Complete** set of references to all consulted literature
- **Correct citation** of important literature
- **Plagiarism**: taking text blocks (from literature or web) without source indication causes immediate **exclusion from this seminar**
- **Font size 12pt with “standard” page layout**
  - `\LaTeX` template will be made available on seminar web page
- **Language**: German or English
- **We expect the correct usage** of spelling and grammar
  - \( \geq 10 \) errors per page \( \rightarrow \) abortion of correction
Requirements on Talk

Your talk

- Talk of **30 minutes**
- Organised as Zoom meeting
- Focus your talk on the **audience**
- **Descriptive** slides:
  - \( \leq \) 15 lines of text
  - use (base) colors in a useful manner
  - number your slides
- **Language**: German or English
- No spelling mistakes please!
- Finish in **time**. Overtime is bad
- Ask for questions
- Have **backup slides** ready for expected questions
- LaTeX/beamer template will be made available on seminar web page
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Important Dates

Deadlines

- April 18: Topic preferences due
- May 10: Detailed outline due
- June 7: Full report due
- June 28: Presentation slides due
- July 13 (?): Seminar talks

Important

Missing a deadline causes immediate exclusion from the seminar
Selecting Your Topic

Procedure

- Check out Foodle poll at https://terminplaner.dfn.de/qhUSVZHyboDWZP63
- Topics classified according to BSc/MSc level
- Please give at least three “Yes” votes ✓
- Preferably additional “Maybe” votes (✓)
- Give as comment:
  - preference of topics (if desired)
  - language of report and talk (English/German)

- **Fill form by Sunday, April 18**
- We do our best to find an adequate topic-student assignment
  - disclaimer: no guarantee for an optimal solution
- Assignment of topics and supervisors will be published on web site by mid next week

Withdrawal

- You have up to three weeks to refrain from participating in this seminar.
- Later cancellation (by you or by us) causes a not passed for this seminar and reduces your (three) possibilities by one.
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Machine Learning

Training data → Training → ML model
Machine Learning

Training data → Training → ML model

ML model → Prediction: Panda

Inference

Machine Learning
Adversarial Examples

\[
\text{Adversarial Example [Goodfellow 2015]}
\]
Adversarial Examples

Adversarial Example [Goodfellow 2015]

ML model

Prediction: Gibbon \( \frac{1}{2} \)

Adversarial Attack
Questions

How to

- find adversarial examples if they exist?
- prove that no adversarial examples exist?
- do so automatically?
- do so efficiently (avoid exponential runtime)?
1. **Efficient Formal Safety Analysis of Neural Networks** (Wang et al.) (B/M)
   - Describes a toolkit for automatic verification
   - Uses symbolic propagation (tracking of dependencies)
   - Approximates piecewise linear activation functions

2. **Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search** (Henriksen, Lomuscio) (M)
   - Describes an improved toolkit
   - Can also approximate non-linear functions (sigmoid, tanh)

3. **Star-Based Reachability Analysis of Deep Neural Networks** (Tran et al.) (M)
   - Describes an alternative approach
   - No approximation is needed (sound and complete)
   - All (not just one) adversarial examples can be found
4. **Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification** (Li et al.) (B/M)
   - Systematic investigation of symbolic domains
   - Based on Abstract Interpretation and SMT methods

5. **Improving Neural Network Verification through Spurious Region Guided Refinement** (Yang et al.) (B/M)
   - Elimination of spurious adversarial examples by linear programming techniques
   - Based on DeepPoly framework

6. **Robustness Analysis of Neural Networks via Efficient Partitioning with Applications in Control Systems** (Everett, Habibi, How) (M)
   - Application of propagation and partitioning techniques to control systems
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
1. Analysis of Bayesian Networks via Prob-Solvable Loops

Ezio Bartocci, Laura Kovács, Miroslav Stankovic: *Analysis of Bayesian Networks via Prob-Solvable Loops*. ICTAC 2020 (B/M)

- Encoding the following types of BNs as Prob-solvable loops
  - *discrete BNs*
  - *Gaussian BNs*
  - *Dynamic BNs*

- Looking into the following problems
  - *exact inference*
  - *expected number of samples*
  - *sensitivity analysis*
2. Formal Verification of Bayesian Network Classifiers

Andy Shih, Arthur Choi, Adnan Darwiche: *Formal Verification of Bayesian Network Classifiers*. PGM 2018 (B/M)

- Compiling Bayesian network classifiers into Ordered Decision Diagrams
- Verifying BN classifiers using ODDs
  - monotonicity checking
  - finding irrelevant features
  - verifying classification robustness
  - verifying If-Then rules and decision independence

![Bayesian Network Classifier Diagram](image1)

![Ordered Decision Diagram](image2)
3. On the Relative Expressiveness of Bayesian and Neural Networks


- Reviewing **class of functions** induced by neural and Bayesian networks

- Identifying the corresponding gap in **expressiveness**

- Proposing a new class of Bayesian networks, namely **Testing Bayesian Networks**

- Investigating **expressiveness of TBNs**
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Quantitative Loop Invariants

- Reasoning about loops is the **hardest task** in (probabilistic) program verification.
- “Practical” approach: capture the loop effect by an invariant\(^a\).
- **But** how to (automatically) find an appropriate loop invariant?
  1. Constraint solving-based numerical approach:
     Feng Y. *et al*.: *Finding Polynomial Loop Invariants for Probabilistic Programs*. ATVA 2017. (M)
  2. Martingale-based symbolic method:
     Barthe G. *et al*.: *Synthesizing Probabilistic Invariants via Doob’s Decomposition*. CAV 2016. (M)
  3. Moment-based approach by solving recurrences:
     Bartocci E. *et al*.: *Automatic Generation of Moment-Based Invariants for Prob-Solvable Loops*. ATVA 2019. (M)

\(^a\)A loop invariant is a property of a loop that is true before and after each iteration.
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Formal Approaches to Systems Engineering

- **Goal:** ensure Reliability, Availability, Maintainability, and Security (RAMS) of (computer) systems
- **Fault Trees** as popular modelling formalism of RAMS-domain
- Use of formal methods to analyse fault trees
- Specifically: probabilistic model checking
- Doing this efficiently is a challenge
- **Topics:**
  1. Bäckström et al.: *Effective Static and Dynamic Fault Tree Analysis*. SAFECOMP 2016 (B)
     - presents efficient static and dynamic analyses of dynamic fault trees
     - surveys various usage scenarios and semantics for attack-defence scenarios for security applications
  3. Aslanyan et al.: *Quantitative Verification and Synthesis of Attack-Defence Scenarios*. CSF 2016 (B/M)
     - translates attack defence trees to two-player games to enable their stochastic analysis
Outline

Overview

Aims of this Seminar

Important Dates

Verification of Neural Networks [Christopher Brix, Thomas Noll]

Analysis of Bayesian Networks [Bahare Salmani]

Synthesizing Quantitative Loop Invariants for Probabilistic Programs [Mingshuai Chen]

Formal Approaches to Systems Engineering [Shahid Khan]

Final Hints
Some Final Hints

Hints

- Take your time to understand your literature.
- Be proactive! Look for additional literature and information.
- Discuss the content of your report with other students.
- Be proactive! Contact your supervisor on time.
- Prepare the meeting(s) with your supervisor.
- Forget the idea that you can prepare a talk in a day or two.

We wish you success and look forward to an enjoyable and high-quality seminar!