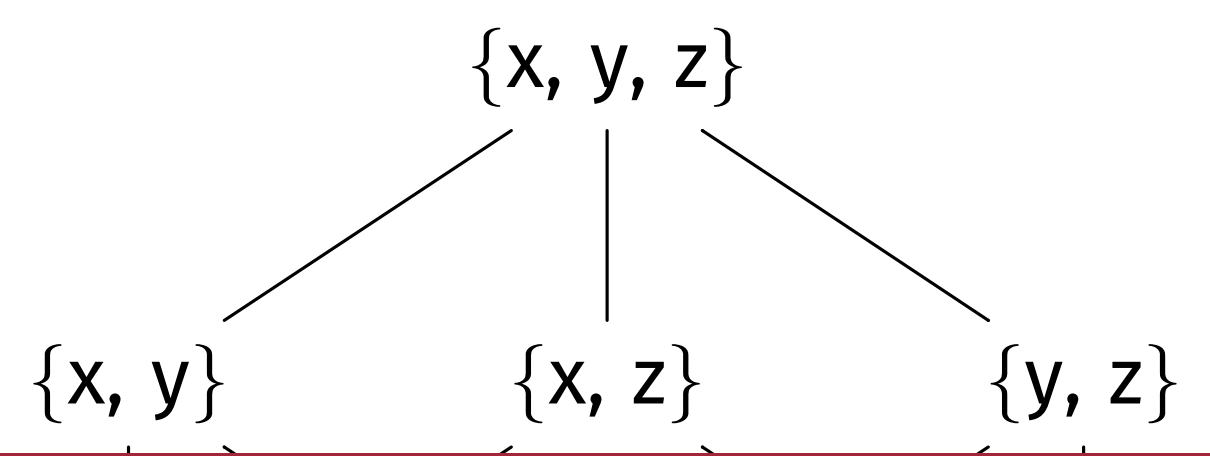


Abstract Interpretation

Software Engineering
Programming Languages

universität ulm

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;   ⦇ a ∈ {1} ⦈
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

$\{ a \in \{1\} \}$

$\{ r \in [0..10) \}$

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;   ⟅ a ∈ {1} ⟆
    if (r > 5) {                     ⟅ r ∈ [0..10) ⟆
       a = 2;
    }                                ⟅ a ∈ {2} ⟆
    System.out.println(a);
}
```

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

$\{ a \in \{1\} \}$

$\{ r \in [0..10) \}$

$\{ a \in \{2\} \}$

$\{ a \in \{1,2\} \}$

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

$\wr\, a \in \{1\} \,\wr$

$\wr\, r \in [0\,..\,10) \,\wr$

$\wr\, a \in \{2\} \,\wr$

$\wr\, a \in \{1, 2\} \,\wr \;\rightarrow$ Valid? Ok? Safe?

The Why

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

$\wr a \in \{1\} \int$

$\wr r \in [0..10) \int$

$\wr a \in \{2\} \int$

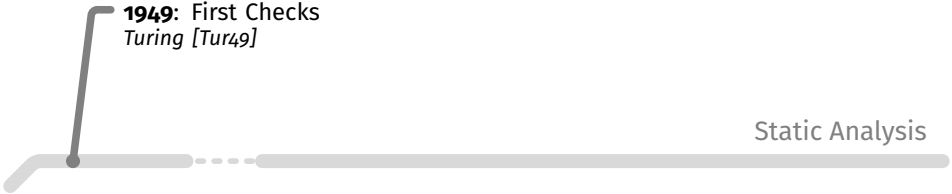
$\wr a \in \{1, 2\} \int \rightarrow$ Valid? Ok? Safe?

- We want to proof, that a program satisfies certain properties

Origins

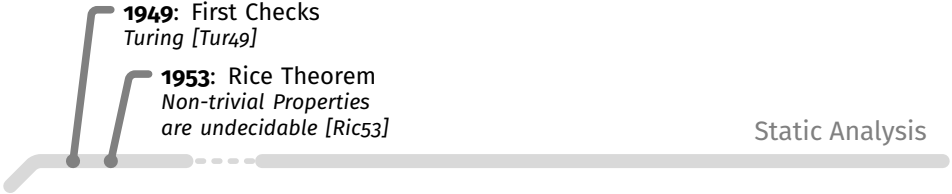
Based on the amazing “Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation” by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



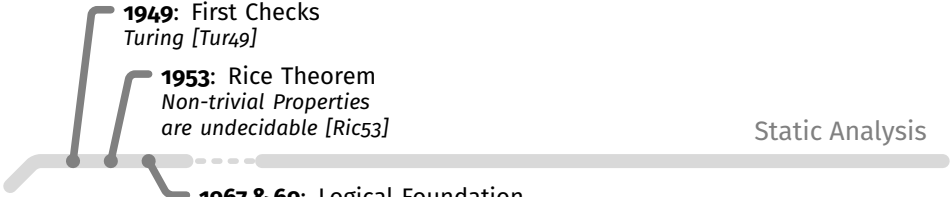
Based on the amazing “Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation” by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



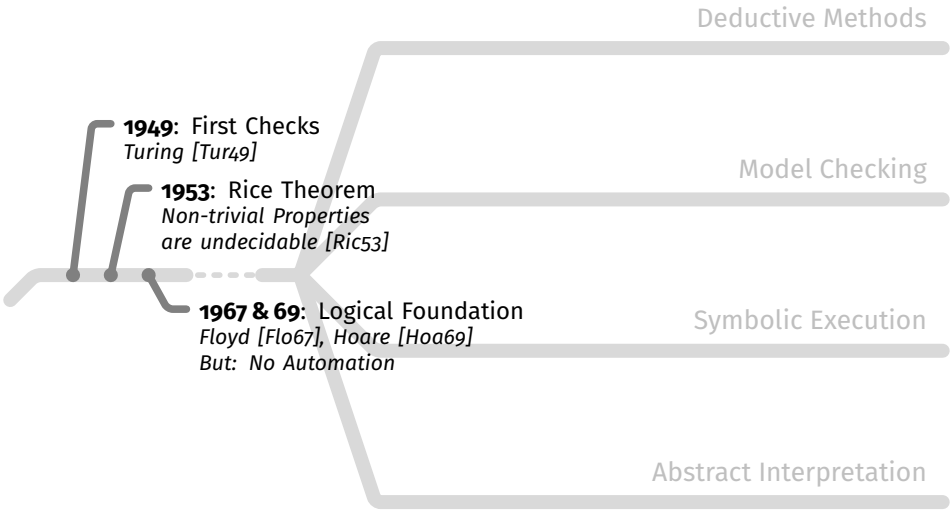
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



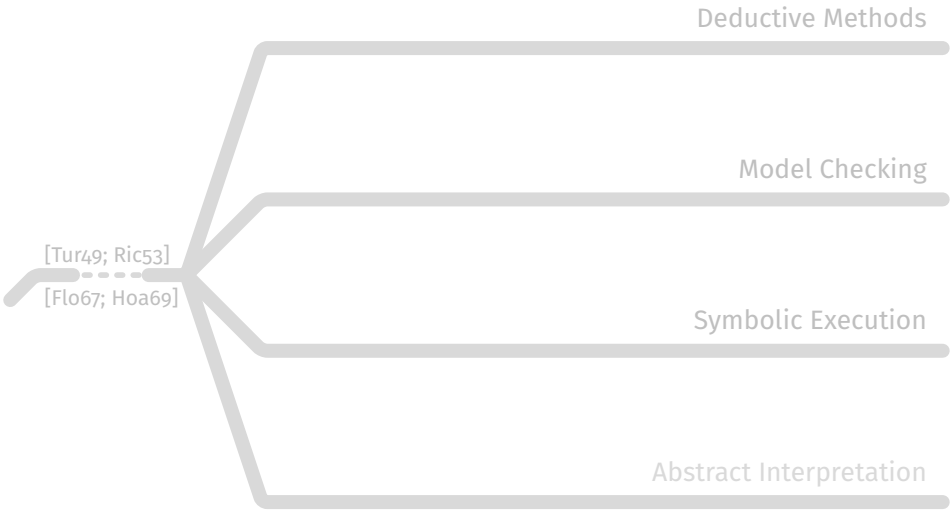
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



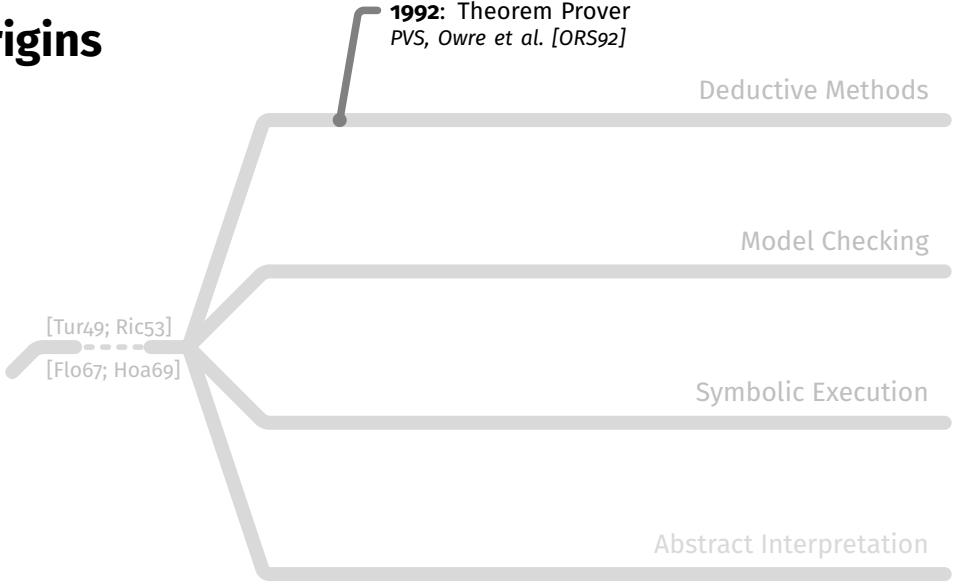
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



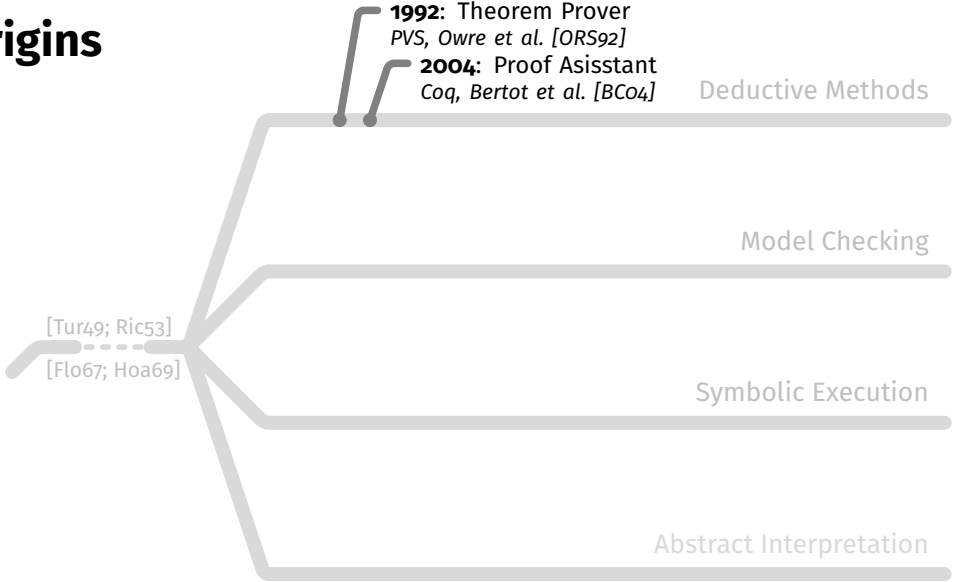
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



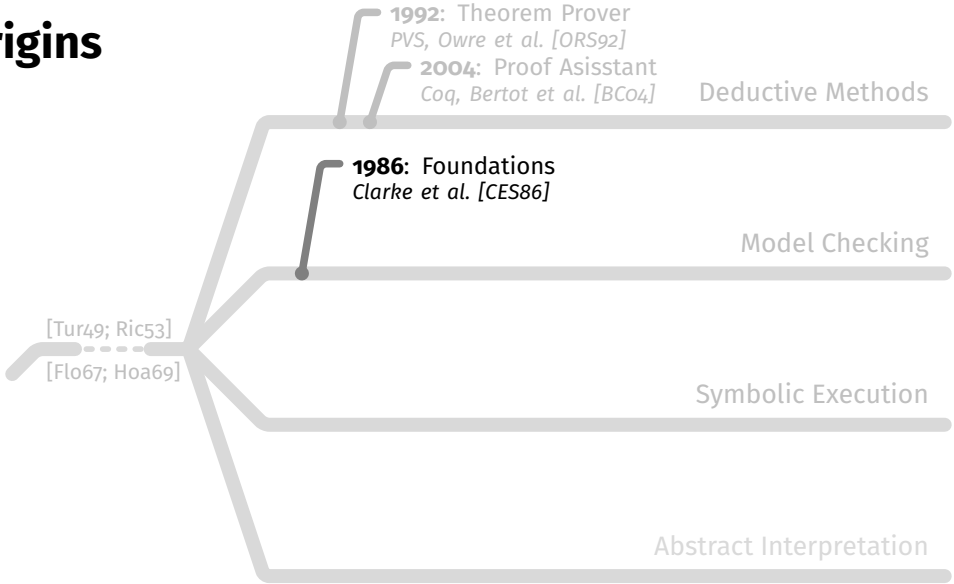
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



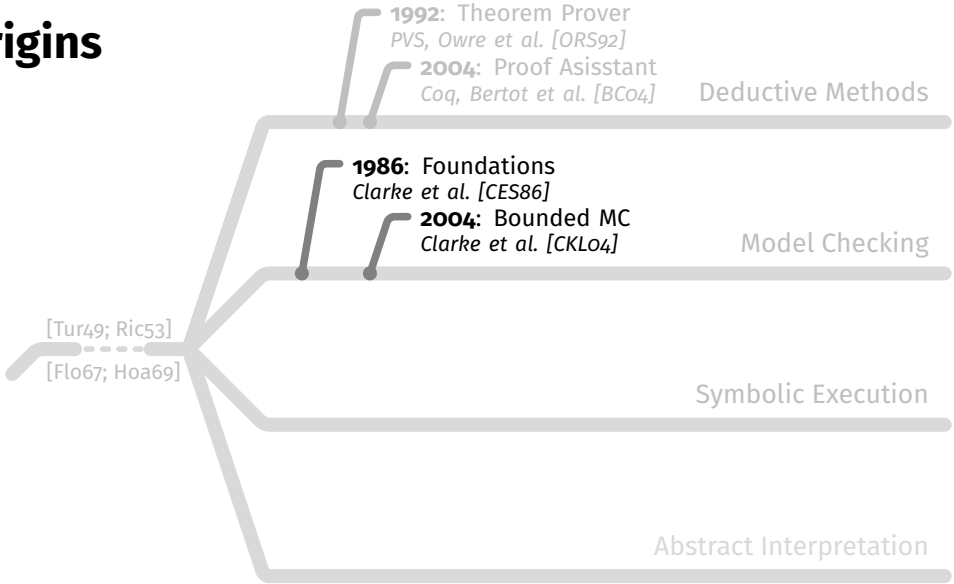
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



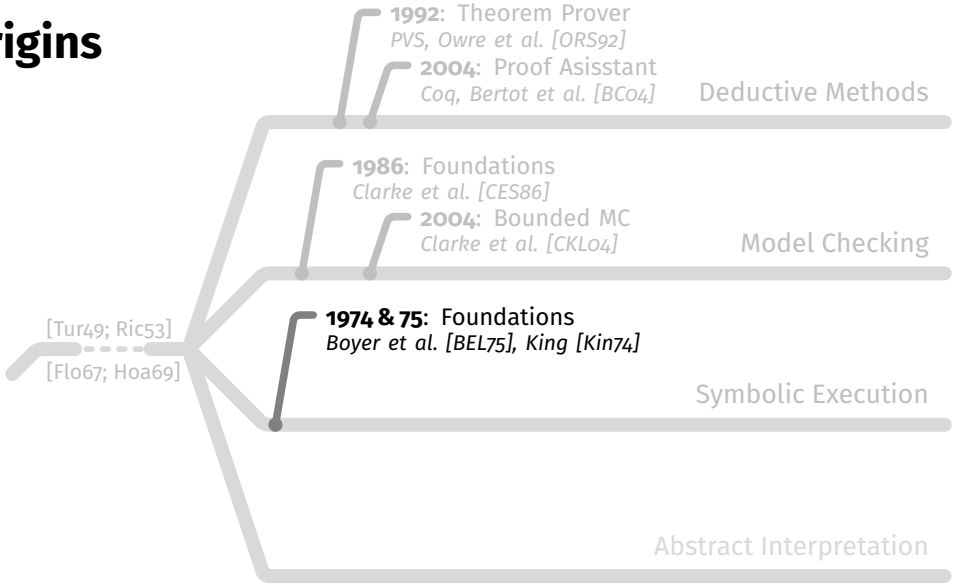
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



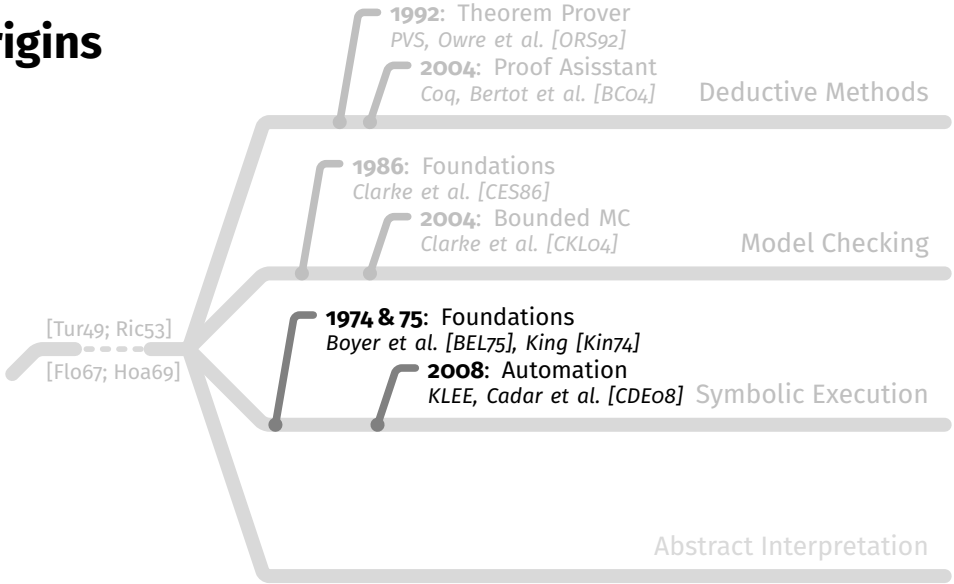
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



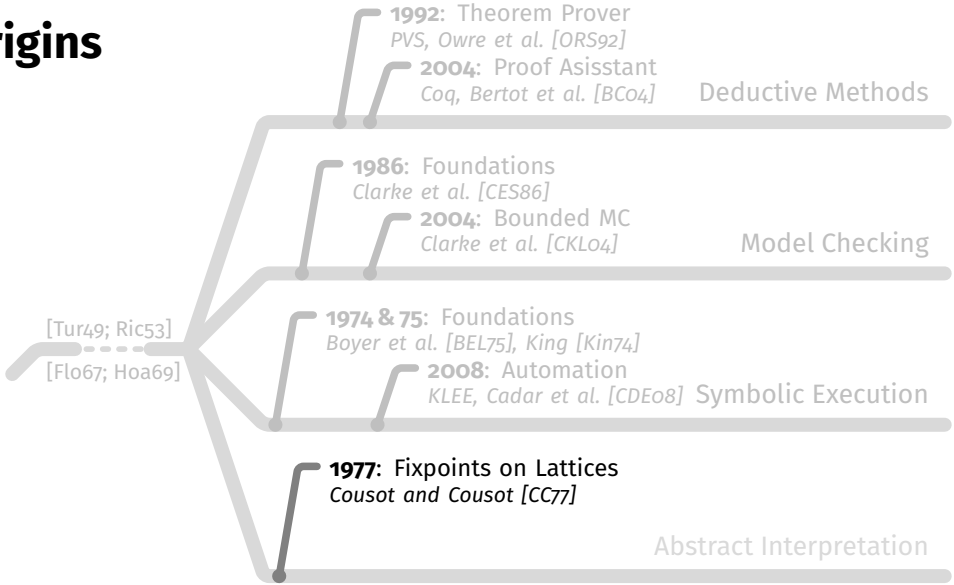
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



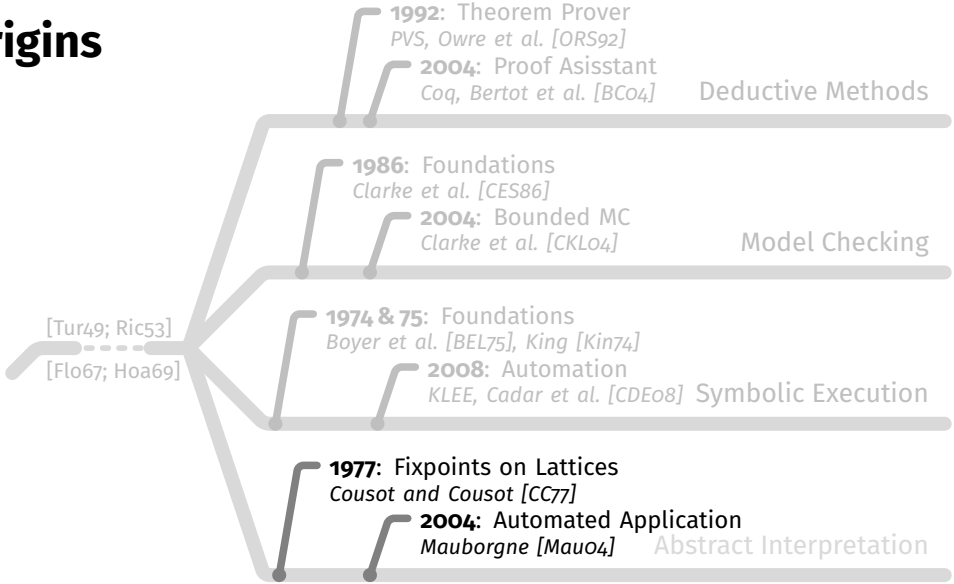
Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins



Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Origins

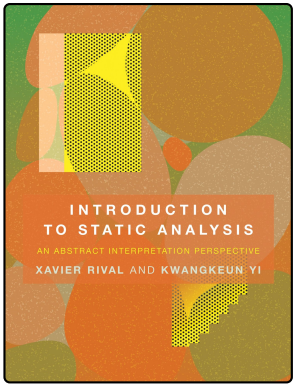


Based on the amazing "Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation" by Miné [Min17], https://www.di.ens.fr/~cousot/AI/, and [Bal+18; GR22]

Recommended Resources

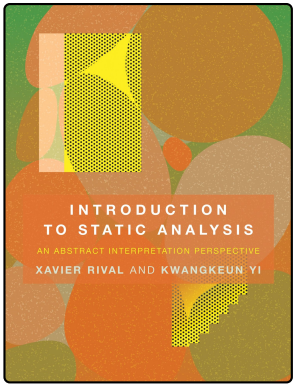
Recommended Resources

Using Analyses [RY20]

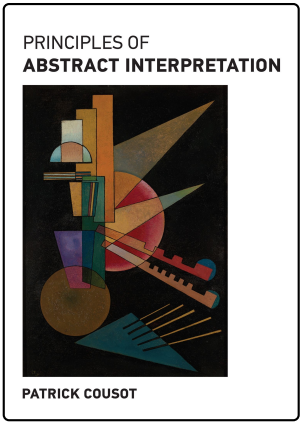


Recommended Resources

Using Analyses [RY20]

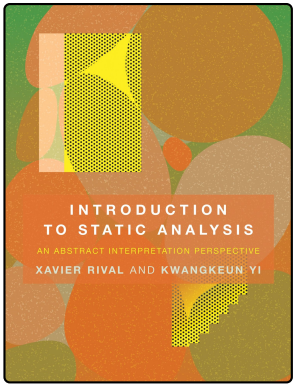


Formal Foundations [Cou21]

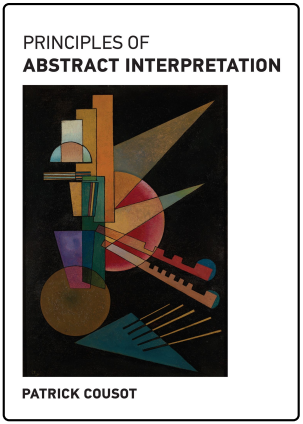


Recommended Resources

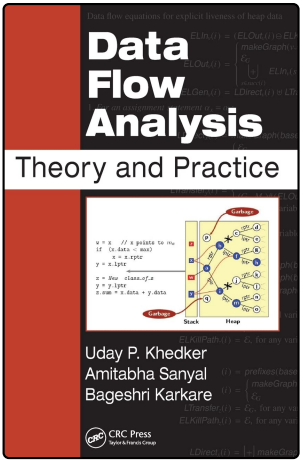
Using Analyses [RY20]



Formal Foundations [Cou21]



Dataflow Perspective [RY20]



Abstract Interpretation

Abstract Interpretation

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
       a = 2;
    }
    System.out.println(a);
}
```

$\wr a \in \{1\} \int$

$\wr r \in [0..10) \int$

$\wr a \in \{2\} \int$

$\wr a \in \{1,2\} \int \rightarrow$ Valid? Ok? Safe?

"Principles of Abstract Interpretation" [Cou21, p. 722],"History of Abstract Interpretation" [GR22, pp. 37]

Abstract Interpretation

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
        a = 2;
    }
    System.out.println(a);
}
```

$\{\!| a \in \{1\} |\!\}$

$\{\!| r \in [0\,..\,10) |\!\}$

$\{\!| a \in \{2\} |\!\}$

$\{\!| a \in \{1,2\} |\!\}$ → Valid? Ok? Safe?

"Principles of Abstract Interpretation" [Cou21, p. 722],"History of Abstract Interpretation" [GR22, pp. 37]

Abstract Interpretation

```
public static void main(String[] args) {
    int a = 1;
    double r = Math.random() * 10;
    if (r > 5) {
        a = 2;
    }
    System.out.println(a);
}
```

$\{ a \in \{1\} \}$

$\{ r \in [0..10) \}$

$\{ a \in \{2\} \}$

$\{ a \in \{1,2\} \}$ $\rightarrow$ Valid? Ok? Safe?

- We want to proof interesting properties of programs

"Principles of Abstract Interpretation" [Cou21, p. 722],"History of Abstract Interpretation" [GR22, pp. 37]

Abstract Interpretation

- We want to proof interesting properties of programs
  - *Dataflow Properties*
    Liveness, Fainting, Reaching Definitions, …

“Principles of Abstract Interpretation” [Cou21, p. 722],“History of Abstract Interpretation” [GR22, pp. 37]

Abstract Interpretation

- We want to proof interesting properties of programs
  - *Dataflow Properties*
    Liveness, Fainting, Reaching Definitions, …
  - *Safety Properties*
    No Null Dereference, No Division by Zero, …

“Principles of Abstract Interpretation” [Cou21, p. 722],“History of Abstract Interpretation” [GR22, pp. 37]

Abstract Interpretation

- We want to proof interesting properties of programs
  - *Dataflow Properties*
    Liveness, Fainting, Reaching Definitions, …
  - *Safety Properties*
    No Null Dereference, No Division by Zero, …
  - *Numerical Properties*
    Signs, Intervals, Octagons, Polyhedra, …

“Principles of Abstract Interpretation” [Cou21, p. 722],“History of Abstract Interpretation” [GR22, pp. 37]

Abstract Interpretation

- We want to proof interesting properties of programs
  - *Dataflow Properties*
    Liveness, Fainting, Reaching Definitions, …
  - *Safety Properties*
    No Null Dereference, No Division by Zero, …
  - *Numerical Properties*
    Signs, Intervals, Octagons, Polyhedra, …
  - …

“Principles of Abstract Interpretation” [Cou21, p. 722],“History of Abstract Interpretation” [GR22, pp. 37]

Abstract Interpretation

See “A casual introduction to Abstract Interpretation” [Cou12]

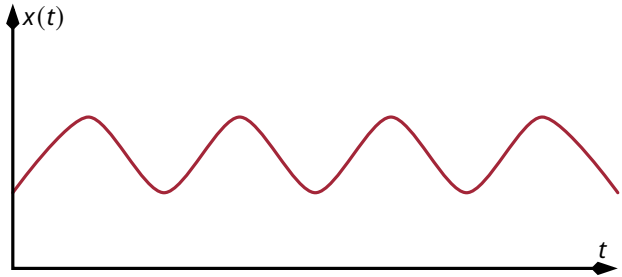
Concrete Interpretation

See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation

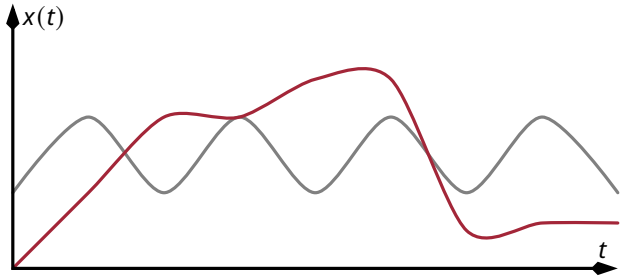
See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation



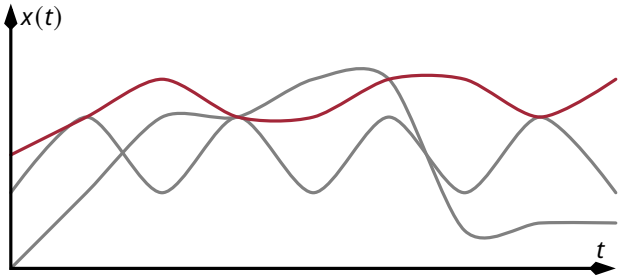
See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation



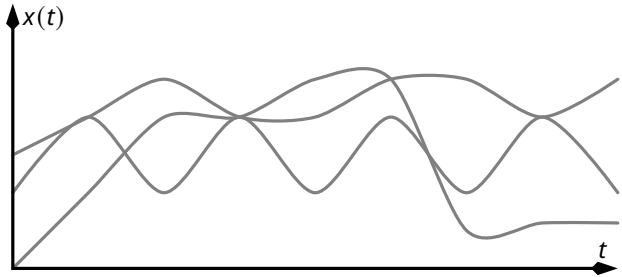
See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation



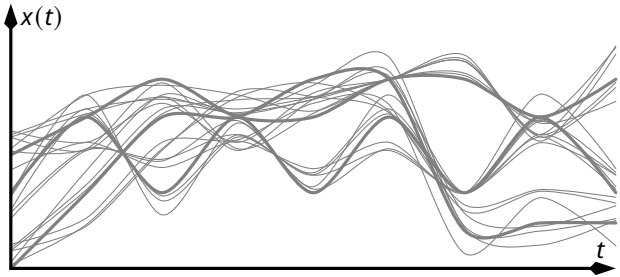
See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation



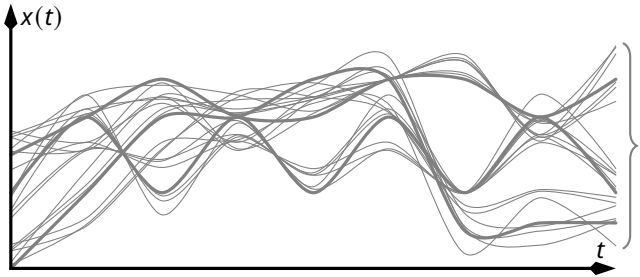
See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation



See “A casual introduction to Abstract Interpretation” [Cou12]

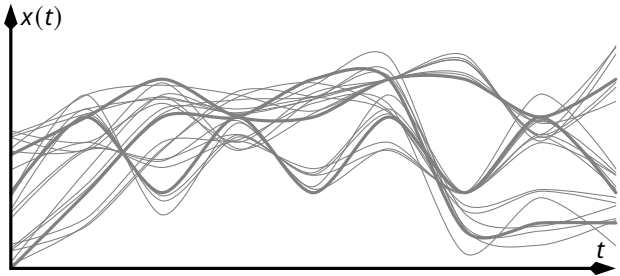
Concrete Interpretation



Collecting Semantics[Cou21, p. 91]

See “A casual introduction to Abstract Interpretation” [Cou12]

Concrete Interpretation

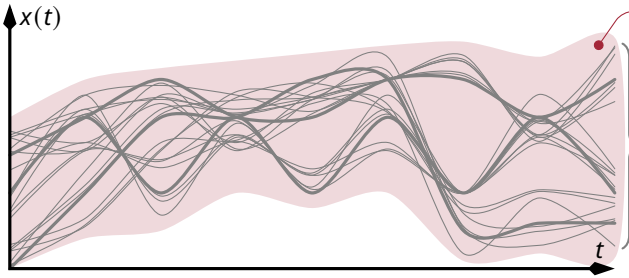


Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See “A casual introduction to Abstract Interpretation” [Cou12]

Abstract Interpretation



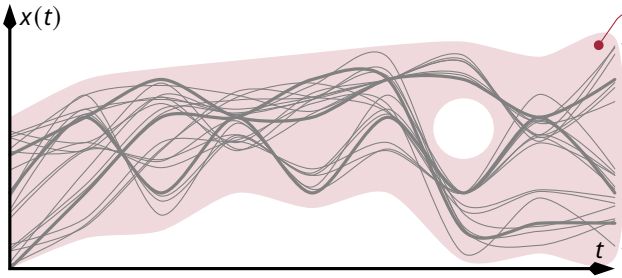
(Trace) Abstraction[Cou21, p. 92]
just one of many

Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See "A casual introduction to Abstract Interpretation" [Cou12]

Abstract Interpretation



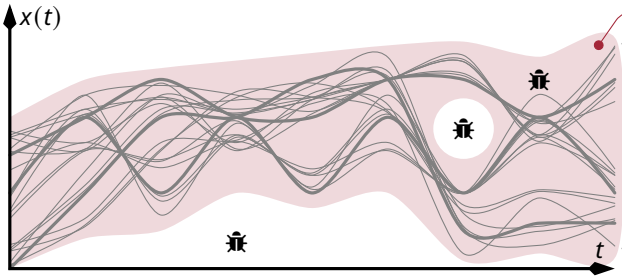
(Trace) Abstraction[Cou21, p. 92]
just one of many

Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See “A casual introduction to Abstract Interpretation” [Cou12]

Abstract Interpretation



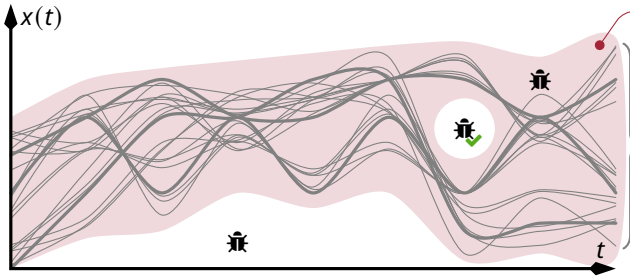
(Trace) Abstraction[Cou21, p. 92]
just one of many

Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See "A casual introduction to Abstract Interpretation" [Cou12]

Abstract Interpretation



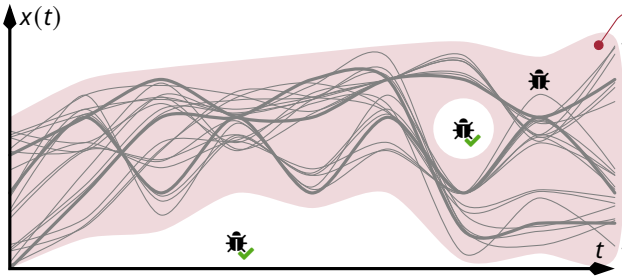
(Trace) Abstraction[Cou21, p. 92]
just one of many

Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See “A casual introduction to Abstract Interpretation” [Cou12]

Abstract Interpretation



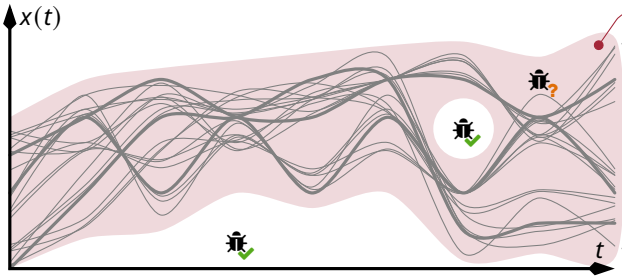
(Trace) Abstraction[Cou21, p. 92]
just one of many

Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See "A casual introduction to Abstract Interpretation" [Cou12]

Abstract Interpretation



(Trace) Abstraction[Cou21, p. 92]
just one of many

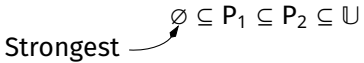
Collecting Semantics[Cou21, p. 91]

- Maybe impossible to compute statically
- … or very expensive (▸ *dynamic*)
- ▸ Abstract Interpretation to the rescue

See “A casual introduction to Abstract Interpretation” [Cou12]

Terminology

- **Property** — Set of states/traces that satisfy that property
  Even integers: $\mathsf{P} = \{ z \in \mathbb{Z} \mid \exists k \in \mathbb{Z} : z = 2k \} = \{ 0, 2, 4, 6, \dots \} \subseteq \mathcal{P}(\mathbb{Z})$



What is a Property? Set basis Poset etc. I have to abstract! Galois, Semantics Principles of Abstract Interpretation book

References I

[Bal+18] Roberto Baldoni et al. “A Survey of Symbolic Execution Techniques”. In: *ACM Comput. Surv.* 51.3 (2018), 50:1–50:39. DOI: 10.1145/3182657. URL: https://doi.org/10.1145/3182657.

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development - Coq’Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. ISBN: 978-3-642-05880-6. DOI: 10.1007/978-3-662-07964-5. URL: https://doi.org/10.1007/978-3-662-07964-5.

[BEL75] Robert S. Boyer, Bernard Elspas, and Karl N. Levitt. “SELECT - a formal system for testing and debugging programs by symbolic execution”. In: *Proceedings of the International Conference on Reliable Software 1975, Los Angeles, California, USA, April 21-23, 1975*. Ed. by Martin L. Shooman and Raymond T. Yeh. ACM, 1975, pp. 234–245. DOI: 10.1145/800027.808445. URL: https://doi.org/10.1145/800027.808445.

References II

[CC77] Patrick Cousot and Radhia Cousot. “Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints”. In: *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*. Ed. by Robert M. Graham, Michael A. Harrison, and Ravi Sethi. ACM, 1977, pp. 238–252. DOI: 10.1145/512950.512973. URL: https://doi.org/10.1145/512950.512973.

[CDE08] Cristian Cadar, Daniel Dunbar, and Dawson R. Engler. “KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs”. In: *8th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2008, December 8-10, 2008, San Diego, California, USA, Proceedings*. Ed. by Richard Draves and Robbert van Renesse. USENIX Association, 2008, pp. 209–224. URL: http://www.usenix.org/events/osdi08/tech/full%5C_papers/cadar/cadar.pdf.

References III

[CES86] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. “Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications”. In: *ACM Trans. Program. Lang. Syst.* 8.2 (1986), pp. 244–263. DOI: 10.1145/5397.5399. URL: https://doi.org/10.1145/5397.5399.

[CKL04] Edmund Clarke, Daniel Kroening, and Flavio Lerda. “A tool for checking ANSI-C programs”. In: *Tools and Algorithms for the Construction and Analysis of Systems: 10th International Conference, TACAS 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29-April 2, 2004. Proceedings 10*. Springer. 2004, pp. 168–176.

[Cou12] Patrick Cousot. “A casual introduction to Abstract Interpretation”. In: *CMACS Workshop on Systems Biology and Formals Methods (SBFM’12)* (2012). URL: https://pcousot.github.io/talks/PCousot-SBFM-2012-1-1.pdf (visited on 12/09/2024).

[Cou21] Patrick Cousot. “Principles of Abstract Interpretation”. In: (2021).

References IV

[Flo67] Robert W. Floyd. “Assigning Meanings to Programs”. In: *Proc. of the American Mathematical Society Symposia on Applied Mathematics*. Vol. 19. 1967, pp. 19–32.

[GR22] Roberto Giacobazzi and Francesco Ranzato. “History of Abstract Interpretation”. In: *IEEE Ann. Hist. Comput.* 44.2 (2022), pp. 33–43. DOI: 10.1109/MAHC.2021.3133136. URL: https://doi.org/10.1109/MAHC.2021.3133136.

[Hoa69] C. A. R. Hoare. “An Axiomatic Basis for Computer Programming”. In: *Commun. ACM* 12.10 (1969), pp. 576–580. DOI: 10.1145/363235.363259. URL: https://doi.org/10.1145/363235.363259.

[Kin74] James C. King. “A New Approach to Program Testing”. In: *Programming Methodology, 4th Informatik Symposium, IBM Germany, Wildbad, September 25-27, 1974*. Ed. by Clemens Hackl. Vol. 23. Lecture Notes in Computer Science. Springer, 1974, pp. 278–290. DOI: 10.1007/3-540-07131-8_30. URL: https://doi.org/10.1007/3-540-07131-8%5C_30.

References V

[Mau04] Laurent Mauborgne. “Astrée: verification of absence of run-time error”. In: *Building the Information Society, IFIP 18th World Computer Congress, Topical Sessions, 22-27 August 2004, Toulouse, France*. Ed. by René Jacquart. Vol. 156. IFIP. Kluwer/Springer, 2004, pp. 385–392. DOI: 10.1007/978-1-4020-8157-6_30. URL: https://doi.org/10.1007/978-1-4020-8157-6%5C_30.

[Min17] Antoine Miné. “Tutorial on Static Inference of Numeric Invariants by Abstract Interpretation”. In: *Found. Trends Program. Lang.* 4.3-4 (2017), pp. 120–372. DOI: 10.1561/2500000034. URL: https://doi.org/10.1561/2500000034.

[ORS92] Sam Owre, John M. Rushby, and Natarajan Shankar. “PVS: A Prototype Verification System”. In: *Automated Deduction - CADE-11, 11th International Conference on Automated Deduction, Saratoga Springs, NY, USA, June 15-18, 1992, Proceedings*. Ed. by Deepak Kapur. Vol. 607. Lecture Notes in Computer Science. Springer, 1992, pp. 748–752. DOI: 10.1007/3-540-55602-8_217. URL: https://doi.org/10.1007/3-540-55602-8%5C_217.

References VI

[Ric53] Henry Gordon Rice. “Classes of recursively enumerable sets and their decision problems”. In: *Transactions of the American Mathematical society* 74.2 (1953), pp. 358–366.

[RY20] Xavier Rival and Kwangkeun Yi. “Introduction to Static Analysis: An Abstract Interpretation Perspective”. In: (2020).

[Tur49] Alan Turing. “Checking a large routine”. In: *Report of a Conference on High Speed Automatic Calculating Machines*. 1949, pp. 67–69.