LOPSTR 2019

Tuesday, October 8th

Symposium on Formal Methods, FM'19, Porto, Portugal

On fixpoint/iteration/variant induction principles for proving total correctness of programs with denotational semantics

Patrick Cousot

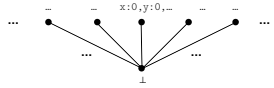
New York University, Courant Institute of Mathematics, Computer Science

pcousot@cs.nyu.edu cs.nyu.edu/~pcousot

Properties of Programs with Denotational Semantics

Denotational semantics of `while` iteration

- $\mathcal{D}$ domain of values of (vectors of) variables
- $\langle \mathcal{D}_\perp = \mathcal{D} \cup \{\perp\}, \sqsubseteq, \perp, \sqcup \rangle$ flat domain with Scott flat ordering



- Denotational semantics $[\![\mathtt{W}]\!]$ of iteration `W` = **`while (B) S`**
  - $B \in \mathcal{D} \to \{\mathrm{tt}, \mathrm{ff}\}$ is the semantics of boolean expression `B`
  - $S = [\![\mathtt{S}]\!] \in \mathcal{D} \to \mathcal{D}_\perp$ that of statement `S` (may contain conditionals and inner loops)
  - $(\!|\, \mathrm{tt} \;?\; a \;:\; b \,|\!) = a$ and $(\!|\, \mathrm{ff} \;?\; a \;:\; b \,|\!) = b$ is the conditional.
  - $F_{\mathtt{W}}(f)x = (\!|\, \neg B(x) \;?\; x \;:\; f(S(x)) \,|\!)$ $\dot{\sqsubseteq}$-upper-continuous loop body transformer[1]
  - $[\![\mathtt{W}]\!] = \mathrm{lfp}^{\dot{\sqsubseteq}} F_{\mathtt{W}}$

[1] *i.e.* **`while (B) S`** ≡ **`if (B) ; else S while (B) S`**

Termination property

- $f \in \mathcal{D} \to \mathcal{D}_\bot$ (semantics of a program)
- $T \subseteq \mathcal{D}$ (termination domain)
- $\mathcal{P}_T \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\bot \mid \forall x \in T \,.\, f(x) \neq \bot\}$ (termination property[2])
- $f \in \mathcal{P}_T$ (the program with semantics f terminates on T)

[2] A property is defined as the set of individuals with that property.

Termination property

- $f \in \mathcal{D} \to \mathcal{D}_\perp$ (semantics of a program)
- $T \subseteq \mathcal{D}$ (termination domain)
- $\mathcal{P}_T \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\perp \mid \forall x \in T \,.\, f(x) \neq \perp\}$ (termination property[2])
- $f \in \mathcal{P}_T$ (the program with semantics f terminates on T)
- The main difficulty is for recursive definitions involving fixpoints

$$\mathsf{lfp}^{\sqsubseteq}\ \boldsymbol{\lambda}\, f \cdot \boldsymbol{\lambda}\, x \cdot [\![\, \neg B(x)\ ?\ x\ :\ f(S(x)) \,]\!]$$

- We need an inductive reasoning
- Example: Jones' size-change termination method [Heizmann, Jones, and Podelski, 2010; Lee, Jones, and Ben-Amram, 2001].

[2] A property is defined as the set of individuals with that property.

Tarski fixpoint theorem and Park fixpoint induction

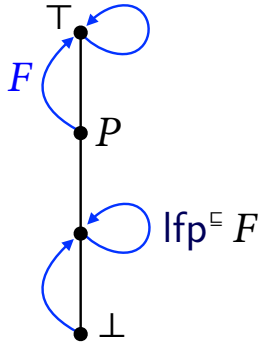
Tarski fixpoint theorem and Park fixpoint induction

Theorem 1 (Tarski fixpoint theorem [Tarski, 1955]) A monotonically increasing function $F \in L \nearrow L$ on a complete lattice $\langle L, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ has a least fixpoint $\mathrm{lfp}^{\sqsubseteq} F = \bigsqcap \{x \in L \mid F(x) \sqsubseteq x\}$. □

Theorem 2 (Park fixpoint induction) Let $F \in \mathcal{L} \nearrow \mathcal{L}$ be a monotonically increasing function on a complete lattice $\langle \mathcal{L}, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ and $P \in \mathcal{L}$. We have

$$\mathrm{lfp}^{\sqsubseteq} F \sqsubseteq P \quad \Leftrightarrow \quad \exists I \in \mathcal{L} \,.\, \begin{array}{ll} F(I) \sqsubseteq I & \text{(2.a)} \\ \wedge \; I \sqsubseteq P & \text{(2.b)} \end{array}$$ □

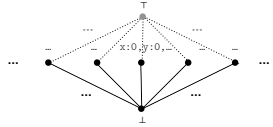
For completeness, the strongest invariant may be needed!



- but this situation is not general (Floyd/Hoare)
- and in this case, look below the fixpoint!

Can be used for proving partial correctness? termination?

- Complement with a top to get a complete lattice[3]



The semantics is the same as in the flat cpo ($\top$ is just never used!)

- partial correctness: e.g. the factorial $F_!(f) \triangleq \boldsymbol{\lambda}\, n \bullet (\!| n = 0 \mathbin{?} 1 \mathbin{:} n \times f(n-1) |\!)$ is partially correct is $\mathrm{lfp}^{\dot{\sqsubseteq}} F_! \mathrel{\dot{\sqsubseteq}} \boldsymbol{\lambda}\, n \bullet (\!| x \geqslant 0 \mathbin{?} n! \mathbin{:} \bot |\!)$

  $\rightarrow$ can be proved by Park induction

- termination: $\boldsymbol{\lambda}\, n \bullet (\!| x \geqslant 0 \mathbin{?} n! \mathbin{:} \bot |\!) \mathrel{\dot{\sqsubseteq}} \mathrm{lfp}^{\dot{\sqsubseteq}} F_!$

  $\rightarrow$ cannot be proved by Park induction or its dual (which is for $P \sqsubseteq \mathrm{gfp}^{\sqsubseteq} F$)

  $\rightarrow$ generalization is needed!

[3] as done in the original work of Dana Scott.

Tarski/Kleene/Scott iterative fixpoint theorem and Scott iteration induction

Fixpoint iteration and iteration induction

Theorem 3 (Tarski/Kleene/Scott iterative fixpoint theorem [Scott, 1970]) If $F \in \mathcal{L} \xrightarrow{uc} \mathcal{L}$ is an upper continuous function on a cpo $\langle \mathcal{L}, \sqsubseteq, \bot, \sqcup \rangle$ then F has a least fixpoint $\mathrm{lfp}^{\sqsubseteq} F = \bigsqcup_{n \in \mathbb{N}} F^n(\bot)$. □

Theorem 4 (Scott iterative fixpoint induction) If $\mathcal{P} \in \wp(\mathcal{D})$ is an *admissible* predicate, $\bot \in \mathcal{P}$, and $\forall d \in \mathcal{P} \,.\, F(d) \in \mathcal{P}$ then $\mathrm{lfp}^{\sqsubseteq} F \in \mathcal{P}$.

□

The predicate $\mathcal{P}$ is said to be *admissible* [Manna, Ness, and Vuillemin, 1973] or *inclusive* [Schmidt, 1988, p. 118] if and only if for all increasing enumerable chains of the cpo if the predicate holds all elements of the chain, it also holds for its limit: $F_0 \sqsubseteq F_1 \sqsubseteq \ldots \sqsubseteq F_i \sqsubseteq \ldots$, if $\forall i \in \mathbb{N} \,.\, F_i \in \mathcal{P}$ then $\bigsqcup_{i \in \mathbb{N}} F_i \in \mathcal{P}$.

Scott induction is incomplete

- Cannot prove termination

$$\mathrm{lfp}^{\sqsubseteq}\ \boldsymbol{\lambda}\, f \cdot \boldsymbol{\lambda}\, x \cdot [\![\, \neg B(x) \mathbin{?} x \mathbin{:} f(S(x)) \,]\!] \in \mathcal{P}_T$$

  on a termination domain $T \neq \varnothing$

- $\bot \in \mathcal{P}_T$!

Generalized iteration induction

Theorem 5 (Iteration induction) Let $F \in \mathcal{L} \xrightarrow{uc} \mathcal{L}$ be an upper-continuous function on a cpo $\langle \mathcal{L}, \sqsubseteq, \bot, \sqcup \rangle$ and $\mathcal{P} \in \wp(\mathcal{L})$.

$$\mathrm{lfp}^{\sqsubseteq} F \in \mathcal{P} \quad \Leftrightarrow \quad \exists \mathcal{Q} \in \wp(\mathcal{L}) \,.\, \begin{array}{ll} & \bot \in \mathcal{Q} \qquad (5.a) \\ \wedge & \forall x \in \mathcal{Q} \,.\, F(x) \in \mathcal{Q} \qquad (5.b) \\ \wedge & \text{for any } F\text{-maximal } \sqsubseteq\text{-increasing chain} \qquad (5.c) \\ & \langle x_i \in \mathcal{Q}, i \in \mathbb{N} \rangle \,.\, \bigsqcup_{i \in \mathbb{N}} x_i \in \mathcal{P} \end{array} \quad \square$$

- Sound and complete
- Note that, contrary to Scott, $\mathcal{P}$ may be different from $\mathcal{Q}$ (*e.g.* for termination)

F-maximal $\sqsubseteq$-increasing chain $\langle x_i,\ i \in \mathbb{N} \rangle$

- The sequence is infinite denumerable (hence non-empty),
- optionally, iterating F (*i.e.* $\forall i \in \mathbb{N} \,.\, x_{i+1} = F(x_i)$), and
- either strictly increasing (*i.e.* $\forall i, j \in \mathbb{N} \,.\, (i < j) \Rightarrow (x_i \sqsubsetneq x_j)$)
- or first strictly increasing and then stationary (*i.e.* $\exists k \in \mathbb{N} \,.\, \forall i, j \in \mathbb{N} \,.\, (i < j \leqslant k) \Rightarrow (x_i \sqsubsetneq x_j) \wedge (k \leqslant i) \Rightarrow (x_k = x_i)$).

The intuition is that only such sequences may correspond to iterates of F.

Example: Hoare logic

- The iteration $\mathtt{W} = \mathtt{while\ (B)\ S}$ has denotational semantics $[\![\mathtt{W}]\!] = \mathrm{lfp}^{\Subset} F_{\mathtt{W}}$ where $F_{\mathtt{W}}(f)x = (\!|\, \neg B(x) \mathbin{?} x \mathbin{:} f(S(x)) \,|\!)$
- Given $P, Q \in \wp(\mathcal{D})$, $\{\!|P|\!\}\ \mathtt{W}\ \{\!|Q|\!\}$ denotes $\forall x \in P\ .\ ([\![\mathtt{W}]\!]x \neq \bot) \Rightarrow ([\![\mathtt{W}]\!]x \in Q)$
- This is $[\![\mathtt{W}]\!] \in \mathcal{P}$ with property $\mathcal{P}_{P,Q} = \{f \mid \forall x \in P\ .\ (f(x) \neq \bot) \Rightarrow (f(x) \in Q)\}$
- Applying the iteration induction theorem Theorem 5 with $\mathcal{Q} \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\bot \mid \forall x \in I\ .\ f(x) \neq \bot \Rightarrow f(x) \in I\}$, we get Hoare rule

$$\frac{\{\!|I \cap B|\!\}\ \mathtt{S}\ \{\!|I|\!\}}{\{\!|I|\!\}\ \mathtt{W}\ \{\!|I \cap \neg B|\!\}} \tag{6}$$

Variant functions

Variant functions

- Variant functions are typically used for termination proofs [Floyd, 1967; Turing, 1949]
- Even for recursive functions (Jones size-change termination method [Heizmann, Jones, and Podelski, 2010; Lee, Jones, and Ben-Amram, 2001])

What may cause a recursive function not to terminate?

- The function body does not terminate (although all recursive calls do terminate)

$$F(f)x = \texttt{if } (x = 0)\ 1 \texttt{ else while } (\texttt{tt})\ \texttt{;} f(0)$$

- The recursive calls do not terminate (although the loop body always terminate)

$$F(f)x = \texttt{if } (x = 0)\ f(0) \texttt{ else } f(x)$$

- I want to distinguish these two cases;
- I need to define who calls what:

  $f(x)$ *calls* $f(y)$ *(*x *and* y *are given parameter values)*
  iff
  assuming all other recursive calls to f *do terminate then f(x) terminates iff* $f(y)$ *does terminate*

Parameter dependency

- $x \overset{F}{\longmapsto} y$: a call of $f = \mathrm{lfp}^{\dot{\sqsubseteq}} F$ for actual parameter x will recursively call f for y
- Formally ($\bot \neq v \in \mathcal{D}$)

$$x \overset{F}{\longmapsto} y \quad \triangleq \quad \text{let } f = \mathrm{lfp}^{\dot{\sqsubseteq}} F \text{ and } f'(z) = \llbracket\, f(z) = \bot \mathbin{?} v : f(z) \,\rrbracket \text{ in} \quad (7)$$
$$F(f'[y \leftarrow \bot])x = \bot \wedge F(f')x \neq \bot$$

- Example

$$f(n) = F(f)n \triangleq \llbracket\, n \in [0,1] \mathbin{?} 0 : f(n-1) + f(n-2) \,\rrbracket$$

- Usually $\overset{F}{\longmapsto}$ is over-approximated syntactically (*e.g.* in Jones' size-change termination method)

Hypothesis

- In recursive definitions $f(x) = (\mathsf{lfp}^{\sqsubseteq} F)x$, we assume that the function body $F(f)$ terminates if all recursive calls to f do terminate[4]

$$\forall f \in \mathcal{D} \to \mathcal{D}_\perp \,.\, \forall x \in \mathcal{D} \,.\, (F(f)x = \perp) \Rightarrow (\exists y \in \mathcal{D} \,.\, x \overset{F}{\longmapsto} y \wedge f(y) = \perp) \tag{8}$$

- Counter-example:

$$F(f)x = \texttt{if } (x = 0) \texttt{ } 1 \texttt{ else while } (\texttt{tt}) \texttt{ ; } f(0)$$

- Then, the only non-termination cause is recursion (not the function body)

Lemma 9 Let $f = \mathsf{lfp}^{\sqsubseteq} F$ where F satisfies (8) $f(x) = \perp$ if and only if $\exists y \in \mathcal{D} \,.\, x \overset{F}{\longmapsto} y \wedge f(y) = \perp$. □

[4] The hypothesis states that this requires a separate proof that we do not consider here.

Proving termination by a variant/convergence function

Theorem 10 (variant/convergence function proof principle for termination) Let $F \in (\mathcal{D} \to \mathcal{D}_\perp) \xrightarrow{uc} (\mathcal{D} \to \mathcal{D}_\perp)$ be an upper-continuous function on the cpo $\langle \mathcal{D} \to \mathcal{D}_\perp, \dot{\sqsubseteq}, \dot{\perp}, \dot{\sqcup} \rangle$ satisfying the function body termination hypothesis (8), $T \in \wp(\mathcal{D})$, and $\mathcal{P}_T \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\perp \mid \forall x \in T \,.\, f(x) \neq \perp\}$. Then

$$\begin{array}{lll} \mathrm{lfp}^{\sqsubseteq} F \in \mathcal{P}_T \quad \Leftrightarrow & & \exists D \in \wp(\mathcal{D}) \,.\, T \subseteq D \qquad (10.a) \\ & \wedge & \exists \preccurlyeq \in \wp(D \times D) \,.\, \langle D, \preccurlyeq \rangle \text{ is well-founded} \qquad (10.b) \\ & \wedge & \forall x \in D \,.\, \forall y \in \mathcal{D} \,.\, (x \overset{F}{\longmapsto} y) \Rightarrow (y \in D \wedge x \succ y) \qquad (10.c) \end{array}$$ □

D is the termination domain necessary for the proof (may be larger than T for which termination is desired)

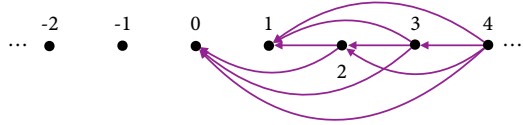
Example

- Program

$$f(n) = F(f)n \triangleq [\![n \in [0,1] \,?\, 0 : f(n-1) + f(n-2)]\!]$$

- Parameter dependency $\overset{F}{\longmapsto}$

- Termination domain $D = \mathbb{N}$
- Well-founded relation $\lessdot$



Equivalence of the termination proof by generalized iteration induction and by variant function induction

Equivalence of the termination proof by generalized iteration induction and by variant function induction

Theorem 11 Let $F \in \mathcal{L} \xrightarrow{uc} \mathcal{L}$ where $\mathcal{L} = \mathcal{D} \to \mathcal{D}_\perp$ which satisfies the function body termination hypothesis (8).

There exists a termination proof by the generalized iteration induction of Theorem 5 for F if and only if there exists one by the variant function induction of Theorem 10. □

The proof shows how to construct a proof by one method knowing a proof by the other method.

Extension to total correctness

Total correctness

Theorem 13 (The total correctness proof principle) Let $F \in \mathcal{D} \xrightarrow{uc} \mathcal{D}_\perp$ satisfying the function body termination hypothesis (8) be an upper-continuous function on the cpo $\langle \mathcal{D}_\perp, \sqsubseteq, \perp, \sqcup \rangle$, $P \in \wp(\mathcal{D})$, $Q \in \wp(\mathcal{D} \times \mathcal{D})$, and $\mathcal{P}_{P,Q} \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\perp \mid \forall x \in P \,.\, \langle x, f(x) \rangle \in Q\}$. Then

$$
\begin{array}{lll}
\mathrm{lfp}^{\dot{\sqsubseteq}} F \in \mathcal{P}_{P,Q} & \Leftrightarrow & \exists D \in \wp(\mathcal{D}) \,.\, \exists I \in \wp(\mathcal{D} \times \mathcal{D}) \,. \\
& & P \subseteq D \qquad (13.a) \\
& \wedge & \{\langle x, y \rangle \in I \mid x \in P\} \subseteq Q \qquad (13.b) \\
& \wedge & \exists \preccurlyeq \in \wp(\mathcal{D} \times \mathcal{D}) \,.\, \langle D, \preccurlyeq \rangle \text{ is well-founded} \qquad (13.c) \\
& \wedge & \forall x, y \in \mathcal{D} \,.\, (x \in D \wedge x \overset{F}{\longmapsto} y) \Rightarrow (y \in D \wedge x \succcurlyeq y) \qquad (13.d) \\
& \wedge & \text{let } \mathcal{P}_{D,I} \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\perp \mid \forall x \in D \,.\, (f(x) \neq \perp \Rightarrow \langle x, f(x) \rangle \in I)\} \text{ in} \qquad (13.e) \\
& & \forall f \in \mathcal{P}_{D,I} \,.\, F(f) \in \mathcal{P}_{D,I} \qquad \square
\end{array}
$$

Application to the iteration

Manna & Pnueli total correctness

- $(\!| P(x) |\!)\ \mathtt{W}\ (\!| Q(x,x') |\!)$ denotes $\mathsf{lfp}^{\stackrel{\cdot}{\sqsubseteq}} F_{\mathtt{W}} \in \mathcal{P}_{P,Q} \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\perp \mid \forall x \in P\ .\ \langle x,\ f(x)\rangle \in Q\}$
- Applying, the total correctness proof principle Theorem 13, we get

$$\frac{\begin{array}{rcl}
\text{consequence rule} \rightarrow & P(x) \Rightarrow D(x), \quad P(x) \wedge I(x,y) \Rightarrow Q(x,y), & (15.\text{a/b}) \\
\text{termination} \searrow & \exists \preccurlyeq \in \wp(\mathcal{D}\times\mathcal{D})\ .\ \langle D, \preccurlyeq\rangle \text{ is well-founded}, & (15.\text{c}) \\
& (\!| D(x) |\!)\ \mathtt{S}\ (\!| D(x') \wedge x \succcurlyeq x' |\!), & (15.\text{d}) \\
& (\!| D(x) \wedge B(x) |\!)\ \mathtt{S}\ (\!| I(x,x') \wedge \forall x''\ .\ I(x',x'') \Rightarrow I(x,x'') |\!), & (15.\text{e}) \\
& \forall x\ .\ D(x) \wedge \neg B(x) \Rightarrow I(x,x) & (15.\text{e}')
\end{array}}{(\!| P(x) |\!)\ \mathtt{W}\ (\!| Q(x,x') \wedge \neg B(x') |\!)}^{5}$$

(a variant of Manna & Pnueli rule incorporating Hoare's consequence rule)

[5] if $(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})x \neq \perp$ then $\neg B(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})$

Conclusion

Conclusion

- Fixpoint induction considers properties above the least fixpoint
- Iteration/variant induction consider properties below the least fixpoint
- These are different and complementary points of view

- Classical fixpoint/iteration/variant induction principles have *limitations*

- Roughly stated, the *generalized* iteration and variant induction principles are sound, complete and *equivalent* for proving total correctness of programs with denotational semantics
- They are the basis for the soundness/completeness of program logics

- Surprisingly, this was not well-understood for decades

From p.ohearn at ucl.ac.uk Fri Jun 14 11:23:24 2019
From: p.ohearn at ucl.ac.uk (O'Hearn, Peter)
Date: Fri, 14 Jun 2019 15:23:24 +0000
Subject: [TYPES] Variants and [Park or Scott] fixpoint Induction
Message-ID: <83B29CB1-1BE9-4034-AFDC-465BA8424607@ucl.ac.uk>

Two methods of reasoning about loops are provided by variants and by (Park or Scott) fixpoint induction. Is there a known relation or non-relation between them? My intuition is that fixpoint induction is not suitable for termination or liveness properties, but I am unsure whether this intuition is correct.

The Hoare rule for total correctness of while loops using variants is well explained in the wikipedia article:
   https://en.wikipedia.org/wiki/Hoare_logic#While_rule_for_total_correctness
There, you make sure a quantity in a well-founded set decreases on each loop iteration.

Here is Park induction:

   lfp(F) <= S iff   Exists I. FI <= I & I <= S

If you think of S as ?spec? and I as ?invariant?, then this can form the basis for reasoning about safety properties (as explained by Cousot here)
   http://web.mit.edu/16.399/www/lecture_11-b-fixpoints1/Cousot_MIT_2005_Course_11b_4-1.pdf

I am a bit worried that my intuition "fixpoint induction is not good for termination? might have some holes in it. In particular, Park induction is used in a known complete proof theory for modal mu-calculus
https://eprints.illc.uva.nl/569/1/PP-2016-33.text.pdf
and that logic is notable for being able to express liveness properties.

I asked a few experts who did not know a way to answer my question above, which is why I am posting it more widely here. In particular, if there is an explanation of how/why fixpoint induction could be good for reasoning about (say) liveness or termination properties of while loops, I?f be glad to hear about it.

Thanks!

Peter O'Hearn

Bibliography

References I

Floyd, Robert W. (1967). "Assigning meaning to programs". In: J.T. Schwartz, ed. *Proc. Symp. in Applied Math.* Vol. 19. Amer. Math. Soc., pp. 19–32 (17).

Heizmann, Matthias, Neil D. Jones, and Andreas Podelski (2010). "Size-Change Termination and Transition Invariants". In: *SAS*. Vol. 6337. Lecture Notes in Computer Science. Springer, pp. 22–50 (4, 5, 17).

Lee, Chin Soon, Neil D. Jones, and Amir M. Ben-Amram (2001). "The size-change principle for program termination". In: *POPL*. ACM, pp. 81–92 (4, 5, 17).

Manna, Zohar, Stephen Ness, and Jean Vuillemin (1973). "Inductive Methods for Proving Properties of Programs". *Commun. ACM* 16.8, pp. 491–502 (11).

Schmidt, David W. (June 1988). *Denotational Semantics: A Methodology for Language Development*. William C. Brown Publishers, Dubuque, IA, USA. URL: http://people.cs.ksu.edu/~schmidt/text/DenSem-full-book.pdf (11).

References II

Scott, Dana S. (Mar. 1970). "Outline of a mathematical theory of computation". In: *Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems*. Princeton University, pp. 169–176 (11).

Tarski, Alfred (1955). "A lattice theoretical fixpoint theorem and its applications". *Pacific J. of Math.* 5, pp. 285–310 (7).

Turing, Alan (1949). "Checking a large routine". In: *Report of a Conference on High Speed Automatic Calculating Machines, University of Cambridge Mathematical Laboratory, Cambridge, England*, pp. 67–69. URL: http://www.turingarchive.org/browse.php/b/8 (17).

The End, Thank you