Undecidable Problems

Martin Čermák, Jiří Koutný and Alexander Meduna

Deparment of Information Systems
Faculty of Informatin Technology
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, Brno 612 00, Czech Republic

Advanced Topics of Theoretical Computer Science

FRVŠ MŠMT FR2581/2010/G1

Diagolization

Diagolization–based proof is schematically performed in the following way:

(1) Assume that ${}_PL$ is decidable, and consider a Turing decider D such that $L(D) = {}_PL$.

(2) From D, construct another Turing decider O; then, by using diagolization technique, apply O on its own description $\langle O \rangle$ so this application results into a contradiction.

(3) The contradiction obtained in (2) implies that the assumption in (1) is incorrect, so ${}_PL$ is undecidable.

TM–Halting

Problem: *TM–Halting*
Question: Let $M \in {}_{TM}\Psi$ and $w \in \triangle^*$. Does M halts on w?
Language: ${}_{TM\text{–}Halting}L = \{\langle M, w\rangle \mid M \in {}_{TM}\Psi, w \in \triangle^*, M \text{ halts on } w\}$.

Theorem

${}_{TM\text{–}Halting}L \notin {}_{TD}\Phi$

Proof I/II:

- Assume that ${}_{TM\text{–}Halting}L$ is decidable.
- Then, there exists a Turing decider D such that $L(D) = {}_{TM\text{–}Halting}L$.
- From D, construct another Turing decider O that works on every input w, where $w = \langle M\rangle$ with $M \in {}_{TM}\Psi$ as follows:
  - O replaces w with $\langle M, M\rangle$;
  - O runs D on $\langle M, M\rangle$;
  - O accepts iff D rejects, and O rejects iff D accepts.

Proof II/II:

- That is, O accepts $\langle M \rangle$ iff M loops on $\langle M \rangle$.
- As O works on every input w, it also works on $w = \langle O \rangle$.
- Since O accepts $\langle M \rangle$ iff M loops on $\langle M \rangle$ for every $w = \langle M \rangle$.
- This equivalence holds for $w = \langle O \rangle$ as well.
- O accepts $\langle O \rangle$ iff O loops on $\langle O \rangle$.
- Thus, $\langle O \rangle \in L(O)$ iff $\langle O \rangle \notin L(O)$—a contradiction.
- Therefore, ${}_{TM-Halting}L$ is undecidable.

Theorem

${}_{TD}\Phi \subset {}_{TM}\Phi$

Proof:

- Clearly, ${}_{TD}\Phi \subseteq {}_{TM}\Phi$.
- We know that ${}_{TM-Halting}L \in {}_{TD}\Phi - {}_{TM}\Phi$.
- Therefore, ${}_{TD}\Phi \subset {}_{TM}\Phi$.

TM–Looping

Problem: *TM–Looping*
Question: Let $M \in {}_{TM}\Psi$ and $w \in \triangle^*$ Does M loops on w?
Language: ${}_{TM-Looping}L = \{\langle M, w\rangle |\ M \in {}_{TM}\Psi, w \in \triangle^*, M \text{ loops on } w\}$.

To prove the undecidability of ${}_{TM-Looping}L$, we first establish the following two theorems.

Theorem

${}_{TM-Looping}L$ is the complement of ${}_{TM-Halting}L$.

Theorem

Let $L \subseteq \triangle^*$. $L \in {}_{TD}\Phi$ iff both L and $\sim L$ are in ${}_{TD}\Phi$.

Proof (only if part):

- Let L be a decidable language.
- Then, there is $M \in {}_{TD}\Psi$ such that $L(M) = L$.
- Change M on Turing machine $N \in {}_{TM}\Psi$ where N enters a non–final state in which it keeps looping exactly when M enters the final state.

Proof (if part):

- Let $L, \sim L \in {}_{TM}\Psi$.
- Then, there exist $N, O \in {}_{TM}\Psi$ such that $L(N) = L$ and $L(O) = \sim L$.
- Clearly, for every $w \in \triangle^*$, $w \in L(N)$ or $w \in L(O)$ and $L(N) \cap L(O) = \emptyset$.
- Construct Turing decider M works on every $w \in \triangle^*$ in the following way:
    - (1) M simultaneously runs N and O on w so M executes by turns one move in N and O.
    - (2) M continues the simulation described in (1) until a move that would take N or O an accepting configuration, where $w \in L(N)$ or $w \in L(O)$.
    - (3) Mhalts and either accepts if $w \in L(N)$ or rejects if $w \in L(O)$.
- Observe that $L(M) = L$. Futhermore, M always halts, so $M \in {}_{TD}\Psi$ and $L \in {}_{TD}\Phi$.

Theorem

${}_{TM\text{—}Looping}L \notin {}_{TM}\Phi$.

Proof:

- Assume, ${}_{TM\text{—}Looping}L \in {}_{TM}\Phi$.
- ${}_{TM\text{—}Looping}L$ is the complement of ${}_{TM\text{—}Halting}L$.
- ${}_{TM\text{—}Halting}L \in {}_{TM}\Phi$.
- ${}_{TM\text{—}Looping}L \notin {}_{TM}\Phi$, but by assumption, ${}_{TM\text{—}Looping}L \in {}_{TM}\Phi$—a contradiction.

Corollary

${}_{TM\text{—}Looping}L \notin {}_{TD}\Phi$.

Reduction

Reduction–based proof is schematically performed in the following way:

(1) Assume that ${}_PL$ is decidable, and consider a Turing decider D such that $L(D) = {}_PL$.

(2) Modify D to another Turing decider that would decide a well–known undecidable language—a contradiction.

(3) The contradiction obtained in (2) implies that the assumption in (1) is incorrect, so ${}_PL$ is undecidable.

TM–Membership

Problem: *TM–Membership*
Question: Let $M \in {}_{TM}\Psi$ and $w \in \triangle^*$ Is w a member of $L(M)$?
Language: ${}_{TM\text{—}Membership}L = \{\langle M, w\rangle |\ M \in {}_{TM}\Psi, w \in \triangle^*, w \in L(M)\}$.

Theorem

${}_{TM\text{—}Membership}L \notin {}_{TD}\Phi$.

Proof:

- Given $\langle M, w\rangle$.
- Construct a Turing machine N that coincides with M except that N accepts x iff M halts on x.
- If there were a Turing decider D for ${}_{TM\text{—}Membership}L$, we could use D and this equivalence to decide ${}_{TM\text{—}Halting}L$.
- Therefore, D can not exist.
- Thus, there is no Turing decider for ${}_{TM\text{—}Membership}L$.

Non–TM–Membership

Problem: *Non–TM–Membership*
Question: Let $M \in {}_{TM}\Psi$ and $w \in \triangle^*$ Is w out of $L(M)$?
Language:
${}_{Non\text{—}TM\text{—}Membership}L = \{\langle M, w\rangle |\ M \in {}_{TM}\Psi, w \in \triangle^*, w \notin L(M)\}$.

Theorem

${}_{Non\text{—}TM\text{—}Membership}L \notin {}_{TM}\Phi$.

Proof:

- Suppose that ${}_{Non\text{—}TM\text{—}Membership}L \in {}_{TM}\Phi$.
- Clearly, ${}_{TM\text{—}Membership}L \in {}_{TM}\Phi$.
- As obvious, ${}_{Non\text{—}TM\text{—}Membership}L$ is complement of ${}_{TM\text{—}Membership}L$
- Thus, ${}_{TM\text{—}Membership}L$ would belong to ${}_{TM}\Phi$.

Corollary

${}_{Non\text{—}TM\text{—}Membership}L \notin {}_{TD}\Phi$.

TM–Regularness

Problem: *TM–Regularness*
Question: Let $M \in {}_{TM}\Psi$ and $w \in \triangle^*$ Is $L(M)$ reular?
Language: ${}_{TM\text{—}Regularness}L = \{\langle M\rangle |\ M \in {}_{TM}\Psi, L(M) \text{ is regular}\}$.

Theorem

${}_{Non\text{—}TM\text{—}Regularness}L \notin {}_{TM}\Phi$.

Other Undecidable Problems I/II

Problem: *CF–Equivalence*
Question: Let $G, H \in {}_{CF}\Psi$. Are G and H equivalent?
Language: ${}_{CF\text{—}Equivalence}L = \{\langle G, H\rangle |\ G, H \in {}_{CF}\Psi, L(G) = L(H)\}$.

Problem: *CF–Containment*
Question: Let $G, H \in {}_{CF}\Psi$. Does $L(G)$ contains $L(H)$?
Language: ${}_{CF\text{—}Containment}L = \{\langle G, H\rangle |\ G, H \in {}_{CF}\Psi, L(H) \subseteq L(G)\}$.

Other Undecidable Problems II/II

Problem: *CF–Intersection*
Question: Let $G, H \in {}_{CF}\Psi$. Is the intersection of G and H empty?
Language: ${}_{CF\text{–}Intersection}L = \{\langle G, H\rangle \mid G, H \in {}_{CF}\Psi, L(G) \cap L(H) = \emptyset\}$.

Problem: *CF–Universality*
Question: Let $G \in {}_{CF}\Psi$. Is $L(G)$ equal to ${}_{G}\triangle^*$?
Language: ${}_{CF\text{–}Universality}L = \{\langle G\rangle \mid G \in {}_{CF}\Psi, L(G) = {}_{G}\triangle^*\}$.

Problem: *CF–Ambiguity*
Question: Let $G \in {}_{CF}\Psi$. Is G ambiguous?
Language: ${}_{CF\text{–}Ambiguity}L = \{\langle G\rangle \mid G \in {}_{CF}\Psi, G \text{ is ambiguous}\}$.

References

Wayne Goddard.
Introducing the Theory of Computation.
Jones Bartlett Publishers, 2008.

Jeffrey D. Ullman John E. Hopcroft, Rajeev Motwani.
Introduction to Automata Theory, Languages, and Computation.
Addison Wesley, 2006.

Dexter C. Kozen.
Automata and Computability.
Springer, 2007.

Dexter C. Kozen.
Theory of Computation.
Springer, 2010.

John C. Martin.
Introduction to Languages and the Theory of Computation.
McGraw-Hill Science/Engineering/Math, 2002.

Thank you for your attention!

End