Computational Complexity of Input/Output Logic

Xin Sun[1] and Diego Agustín Ambrossio[1][2]

[1] Faculty of Science, Technology and Communication, University of Luxembourg,
Luxembourg
xin.sun@uni.lu
[2] ICR, SnT
diego.ambrossio@uni.lu

Abstract. Input/output logics are abstract structures designed to represent conditional norms. The complexity of input/output logic has been sparsely developed. In this paper we study the complexity of input/output logics. We show that the lower bound of the complexity of the fulfillment problem of 4 input/output logics is coNP, while the upper bound is either coNP or P^{NP}.[3]

Key words: input/output logic, norm, complexity

1 Introduction

In recent years, normative multi-agent system [7, 3] arises as a new interdisciplinary academic area bringing together researchers from multi-agent system [17, 22, 21], deontic logic [9] and normative system [1, 11, 2]. Norms play an important role in normative multi-agent systems. They are heavily used in agent cooperation and coordination, group decision making, multi-agent organizations, electronic institutions, and so on.

In the first volume of the handbook of deontic logic and normative systems [9], input/output logic [12–15] appears as one of the new achievements in deontic logic of this century. Input/output logic takes its origin in the study of conditional norms. Unlike the modal logic framework, which usually uses possible world semantics, input/output logic adopts mainly operational semantics: a normative system is conceived in input/output logic as a deductive machine, like a black box which produces normative statements as output, when we feed it descriptive statements as input.

Boella and van der Torre [6] extends input/output logic to reason about constitutive norms. Tosatto *et al.* [8] adapts it to represent and reason about abstract normative systems. For a comprehensive introduction to input/output logic, see Parent and van der Torre [15]. A technical toolbox to build input/output logic can be found in Sun [19].

While the semantics and application of input/output logic has been well developed in recent years, the complexity of input/output logic has not been studied yet. In this paper we fill this gap. We show that the lower bound of the complexity of the fulfillment problem of 4 input/output logics are coNP, while the upper bound is either coNP or P^{NP}.

The structure of this paper is as follows we present a summary of basic concepts and results in input/output logic and some notes in complexity theory, in Section 2. In

[3] This paper is an extension of a short paper [20] by the same authors.

Section 3 we study the complexity of input/output logic. We point out some directions for future work and conclude this paper in Section 4.

2 Background

2.1 Input/output logic

Makinson and van der Torre introduce input/output logic as a general framework for reasoning about the detachment of obligations, permissions and institutional facts from conditional norms. Strictly speaking input/output logic is not a single logic but a family of logics, just like modal logic is a family of logics containing systems K, KD, S4, S5, ... We refer to the family as the input/output framework. The proposed framework has been applied to domains other than normative reasoning, for example causal reasoning, argumentation, logic programming and non-monotonic logic, see Bochman [5].

Let $\mathbb{P} = \{p_0, p_1, \ldots\}$ be a countable set of propositional letters and $L_{\mathbb{P}}$ be the propositional language built upon $\mathbb{P}$. Let $N \subseteq L_{\mathbb{P}} \times L_{\mathbb{P}}$ be a set of ordered pairs of formulas of $L_{\mathbb{P}}$. We call N a normative system. A pair $(a, x) \in N$, call it a *norm*, is read as "given a, it ought to be x". N can be viewed as a function from $2^{L_{\mathbb{P}}}$ to $2^{L_{\mathbb{P}}}$ such that for a set of formulas $A \subseteq L_{\mathbb{P}}$, $N(A) = \{x \in L_{\mathbb{P}} : (a, x) \in N \text{ for some } a \in A\}$. Intuitively, N can be interpreted as a *normative code* composed of conditional norms and the set A serves as explicit input.

Makison and van der Torre [12] define the semantics of input/output logics from O_1 to O_4 as follows:

- $O_1(N, A) = Cn(N(Cn(A)))$.
- $O_2(N, A) = \bigcap\{Cn(N(V)) : A \subseteq V, V \text{ is complete}\}$.
- $O_3(N, A) = \bigcap\{Cn(N(B)) : A \subseteq B = Cn(B) \supseteq N(B)\}$.
- $O_4(N, A) = \bigcap\{Cn(N(V) : A \subseteq V \supseteq N(V)), V \text{ is complete}\}$.

Here Cn is the classical consequence operator of propositional logic, and a set of formulas is *complete* if it is either *maximal consistent* or equal to $L_{\mathbb{P}}$. These four operators are called *simple-minded output*, *basic output*, *simple-minded reusable output* and *basic reusable output* respectively. For each of these four operators, a *throughput* version that allows inputs to reappear as outputs, defined as $O_i^+(N, A) = O_i(N_{id}, A)$, where $N_{id} = N \cup \{(a, a) \mid a \in L_{\mathbb{P}}\}$. When A is a singleton, we write $O_i(N, a)$ for $O_i(N, \{a\})$.

Input/output logics are given a proof theoretic characterization. We say that an ordered pair of formulas is derivable from a set N iff (a, x) is in the smallest set that extends N and is closed under a number of derivation rules. The following are the rules we need to define O_1 to O_4^+:

- SI (strengthening the input): from (a, x) to (b, x) whenever $b \vdash a$. Here $\vdash$ is the classical entailment relation of propositional logic.
- OR (disjunction of input): from (a, x) and (b, x) to $(a \vee b, x)$.
- WO (weakening the output): from (a, x) to (a, y) whenever $x \vdash y$.
- AND (conjunction of output): from (a, x) and (a, y) to $(a, x \wedge y)$.

- CT (cumulative transitivity): from (a, x) and $(a \wedge x, y)$ to (a, y).
- ID (identity): from nothing to (a, a).

The derivation system based on the rules SI, WO and AND is called D_1. Adding OR to D_1 gives D_2. Adding CT to D_1 gives D_3. The five rules together give D_4. Adding ID to D_i gives D_i^+ for $i \in \{1, 2, 3, 4\}$. $(a, x) \in D_i(N)$ is used to denote the norms (a, x) is derivable from N using rules of derivation system D_i. In Makinson and van der Torre [12], the following soundness and completeness theorems are given:

Theorem 1 ([12]). *Given an arbitrary normative system N and formula a,*

- $x \in O_i(N, a)$ *iff* $(a, x) \in D_i(N)$*, for* $i \in \{1, 2, 3, 4\}$.
- $x \in O_i^+(N, a)$ *iff* $(a, x) \in D_i^+(N)$*, for* $i \in \{1, 2, 3, 4\}$.

2.2 Complexity theory

Complexity theory is the theory to investigate the time, memory, or other resources required for solving computational problems. In this subsection we briefly review those concepts and results from complexity theory which will be used in this paper. More comprehensive introduction of complexity theory can be found in [18, 4]

We assume the readers are familiar with notions like Turing machine and the complexity class P, NP and coNP. Oracle Turing machine and a complexity class related to oracle Turing machine will be used in this paper.

Definition 1 (oracle Turing machine). *An* oracle *for a language L is device that is capable of reporting whether any string w is a member of L. An (resp. non-deterministic) oracle Turing machine M^L is a modified (resp. non-deterministic) Turing machine that has the additional capability of querying an oracle. Whenever M^L writes a string on a special oracle tape it is informed whether that string is a member of L, in a single computation step.*

Definition 2. *P^{NP} is the class of languages decidable with a polynomial time oracle Turing machine that uses oracle $L \in NP$.*

3 Complexity of input/output logic

The complexity of input/output logic has been sparsely studied in the past. Although the reversibility of derivations rules as a proof re-writing mechanism has been studied for input/output logic framework [12], the length or complexity of such proofs have not been developed. We approach the complexity of input/output logic from a semantic point of view. We focus on the following fulfillment problem:

Given a finite set of norms N, a finite set of formulas A and a formula x, is
$x \in O(N, A)$?

The aim of the fulfillment problem is to check whether the formula x appears among the obligations detached from the normative system N and facts A.

3.1 Simple-minded O_1

Theorem 2. *The fulfillment problem of simple-minded input/output logic is coNP-complete.*

Proof: Concerning the coNP hardness, we prove by reducing the validity problem of propositional logic to the fulfillment problem of simple-minded input/output logic: given an arbitrary $x \in L_{\mathbb{P}}$, $\vdash x$ iff $x \in Cn(\top)$ iff $x \in Cn(N(Cn(A)))$ where $N = \emptyset$ iff $x \in O_1(N, A)$ where $N = \emptyset$.

Now we prove the coNP membership. We provide the following non-deterministic Turing machine to solve the complement of our problem. Let $N = \{(a_1, x_1), \ldots, (a_n, x_n)\}$, A be a finite set of formulas and x be a formula.

1. Guess a sequence of valuations $V_1, \ldots, V_n$ and V' on the propositional letters appears in $A \cup \{a_1, \ldots, a_n\} \cup \{x_1, \ldots, x_n\} \cup \{x\}$.
2. Let $N' \subseteq N$ be the set of norms which contains all (a_i, x_i) such that $V_i(A) = 1$ and $V_i(a_i) = 0$.
3. Let $X = \{x : (a, x) \in N - N'\}$.
4. If $V'(X) = 1$ and $V'(x) = 0$. Then return "accept" on this branch. Otherwise return "reject" on this branch.

The main intuition of the proof is: N' collects all norms which *cannot* be triggered by A. [4] On some branches we must have that N' contains exactly those norms which are not triggered by A. In those lucky branches X is the same as $N(Cn(A))$. If there is a valuation V' such that $V'(X) = 1$ and $V'(x) = 0$, then we know $x \notin Cn(X) = Cn(N(Cn(A)))$.

It can be verified that $x \notin Cn(N(Cn(A)))$ iff the algorithm returns "accept" on some branches and the time complexity of the non-deterministic Turing machine is polynomial. $\dashv$

3.2 Simple-minded throughput O_1^+

Lemma 1. $O_1^+(N, A) = Cn(A \cup N(Cn(A)))$.

Proof: The proof is routine and left to the readers. $\dashv$

Theorem 3. *The fulfillment problem of simple-minded throughput input/output logic is coNP-complete.*

Proof: Concerning the lower bound, we prove by a reduction from the validity problem of propositional logic: given arbitrary $x \in L_{\mathbb{P}}$, $\vdash x$ iff $x \in Cn(\top)$ iff $x \in Cn(A \cup N(Cn(A)))$ where $N = \emptyset = A$ iff $x \in O_1^+(N, A)$ where $N = \emptyset = A$.

Concerning the upper bound, we prove by giving a non-deterministic Turing machine similar to the one in the proof of Theorem 2. The only change is now in step 4 we test if $V'(A \cup X) = 1$ and $V'(x) = 0$. It can be verified that $x \notin Cn(A \cup N(Cn(A)))$ iff the non-deterministic Turing machine returns "accept" on some branch. By Lemma 1 we know this Turing machine solves our problem. $\dashv$

[4] We say a norm (a, x) is triggered by A if $a \in Cn(A)$).

3.3 Simple-minded reusable O_3

Given a set N of norms and a set A of formulas, we define a function $f_A^N : 2^{L_{\mathbb{P}}} \to 2^{L_{\mathbb{P}}}$ such that $f_A^N(X) = Cn(A \cup N(X))$. It can be proved that f_A^N is monotonic with respect to the set theoretical $\subseteq$ relation, and $(2^{L_{\mathbb{P}}}, \subseteq)$ is a complete lattice. Then by Tarski's fixed point theorem there exists a least fixed point of f_A^N. The following proposition shows that the least fixed point can be constructed in an inductive manner.

Proposition 1 ([19]). *Let B_A^N be the least fixed point of the function f_A^N. Then $B_A^N = \bigcup_{i=0}^{\infty} B_{A,i}^N$, where $B_{A,0}^N = Cn(A), B_{A,i+1}^N = Cn(A \cup N(B_{A,i}^N))$.*

Using the least fixed point, a more constructive semantics of O_3 and O_3^+ are stated as follows, such semantics gives us insights to develop algorithms to solve the fulfillment problem of reusable input/output logic:

Theorem 4 ([19]). *For a set of norms N and a formula a,*

1. $(a, x) \in D_3(N)$ *iff* $x \in Cn(N(B_{\{a\}}^N))$.
2. $(a, x) \in D_3^+(N)$ *iff* $x \in Cn(N_{id}(B_{\{a\}}^{N_{id}}))$.

Theorem 5. *The fulfillment problem of simple-minded reusable input/output logic is between coNP and P^{NP}.*

Proof: The lower bound is easy, here we omit it.

Concerning the upper bound, we provide the following algorithm on a oracle Turing machine with oracle SAT.

Let $N = \{(a_1, x_1), \dots, (a_n, x_n)\}$, A be a finite set of formulas and x be a formula.

1. Let $X = A, Y = Z = N, U = \emptyset$.
2. for each $(a_i, x_i) \in Y$, ask the oracle if $\neg(\bigwedge X \to a_i)$ is satisfiable.
   (a) If "no", then let $X = X \cup \{x_i\}$, $Z = Z - \{(a_i, x_i)\}$.
   (b) Otherwise do nothing.
3. If $Y == Z$, goto 4. Otherwise let $Y = Z$, goto step 2.
4. for each $(a_i, x_i) \in N$, ask the oracle if $\neg(\bigwedge X \to a_i)$ is satisfiable.
   (a) If "no", then let $U = U \cup \{x_i\}$.
   (b) Otherwise do nothing
5. Ask the oracle if $\neg(\bigwedge U \to x)$ is satisfiable.
   (a) If "no", then return "accept".
   (b) Otherwise return "reject".

The correctness of the above algorithm is routine to prove and we leave it to the readers. Concerning the time complexity, the times of loop in step 2 is at most n. Each loop can be finished in polynomial time. Therefore all the loops in step 2 can be done in polynomial time. Step 3 call for step 2 for at most n times. Therefore it can still be done in polynomial time. The times of loop in step 4 is exactly n. Each loop can be finished in polynomial time. Therefore all the loops in step 4 can be done in polynomial time. Step 5 can be done in polynomial time. Therefore the algorithm is polynomial. $\dashv$

3.4 Simple-minded reusable throughput O_3^+

Theorem 6. *The fulfillment problem of simple-minded reusable throughput input/output logic is between coNP and P^{NP}.*

Proof: The lower bound is easy, here we omit it.

Concerning the upper bound, we prove by giving an algorithm similar to the one in the proof of Theorem 5. We make the following change:

- In step 2 and 4 we ask the oracle if $\neg(\bigwedge A \wedge \bigwedge X \to a_i)$ is satisfiable.
- In step 5 we ask the oracle if $\neg(\bigwedge A \wedge \bigwedge U \to a_i)$ is satisfiable. $\dashv$

4 Conclusion and future work

In this paper we develop complexity results of input/output logic. We show that four input/output logics have lower bound coNP and upper bound either coNP or P^{NP}. There are several natural directions for future work:

1. What is the tight complexity results of reusable input/output logic, as well as other input/output logics?
2. What is the complexity of constraint input/output logic? Constraint input/output logic [13] is developed to deal with the inconsistency of output. The semantics of constraint input/output logic is more complicated than those input/output loic discussed in this paper. This might increase the complexity of the fulfillment problem. Constraint input/output logic based on O_3^+ has close relation with Reiter's default logic [16]. Gottlob [10] presents some complexity results of Reiter's default logic, which will give us insights on the complexity of constraint input/output logic.
3. What is the complexity of different types of permission? Three different of permissions are introduced in Makinson and van der Torre [14]. The semantics of these three logics are different, which suggests different complexity for the problems related to permissions.

References

1. Ågotnes, T., van der Hoek, W., Rodríguez-Aguilar, J.A., Sierra, C., Wooldridge, M.: On the logic of normative systems. In: Veloso, M.M. (ed.) IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007. pp. 1175–1180 (2007), http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-190.pdf
2. Alechina, N., Dastani, M., Logan, B.: Reasoning about normative update. In: Rossi, F. (ed.) IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013. IJCAI/AAAI (2013), http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6884
3. Andrighetto, G., Governatori, G., Noriega, P., van der Torre, L.W.N. (eds.): Normative Multi-Agent Systems, Dagstuhl Follow-Ups, vol. 4. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2013), http://drops.dagstuhl.de/opus/portals/dfu/index.php?semnr=13003
4. Arora, S., Barak, B.: Computational Complexity: A Modern Approach. Cambridge University Press, New York, USA (2009)

5. Bochman, A.: A causal approach to nonmonotonic reasoning. Artificial intelligence 160(1-2), 105–143 (2004)
6. Boella, G., van der Torre, L.: A logical architecture of a normative system. In: Goble, L., Meyer, J.J.C. (eds.) Deontic Logic and Artificial Normative Systems. Lecture Notes in Computer Science, vol. 4048, pp. 24–35. Springer, Utrecht, The Netherlands (2006)
7. Boella, G., van der Torre, L., Verhagen, H.: Introduction to the special issue on normative multiagent systems. Autonomous Agents and Multi-Agent Systems 17(1), 1–10 (2008), http://dx.doi.org/10.1007/s10458-008-9047-8
8. Colombo Tosatto, S., Boella, G., van der Torre, L., Villata, S.: Abstract normative systems: Semantics and proof theory. In: Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning. pp. 358–368 (2012)
9. Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L. (eds.): Handbook of Deontic Logic and Normative Systems. College Publications, London (2013)
10. Gottlob, G.: Complexity results for nonmonotonic logics. J. Log. Comput. 2(3), 397–425 (1992), http://dx.doi.org/10.1093/logcom/2.3.397
11. Herzig, A., Lorini, E., Moisan, F., Troquard, N.: A dynamic logic of normative systems. In: Walsh, T. (ed.) IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 228–233. IJCAI/AAAI (2011), http://ijcai.org/papers11/Papers/IJCAI11-049.pdf
12. Makinson, D., van der Torre, L.: Input-output logics. Journal of Philosophical Logic 29, 383–408 (2000)
13. Makinson, D., van der Torre, L.: Constraints for input/output logics. Journal of Philosophical Logic 30(2), 155–185 (2001)
14. Makinson, D., van der Torre, L.: Permission from an input/output perspective. Journal of Philosophical Logic 32, 391–416 (2003)
15. Parent, X., van der Torre, L.: I/O logic. In: Horty, J., Gabbay, D., Parent, X., van der Meyden, R., van der Torre, L. (eds.) Handbook of Deontic Logic and Normative Systems. College Publications (2013)
16. Reiter, R.: A logic for default reasoning. Artif. Intell. 13(1-2), 81–132 (1980), http://dx.doi.org/10.1016/0004-3702(80)90014-4
17. Shoham, Y., Leyton-Brown, K.: Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press (2009)
18. Sipser, M.: Introduction to the theory of computation. Cengage Learning, Boston, USA, 3 edn. (2012)
19. Sun, X.: How to build input/output logic. In: Bulling, N., van der Torre, L.W.N., Villata, S., Jamroga, W., Vasconcelos, W. (eds.) Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV, Prague, Czech Republic, August 18-19, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8624, pp. 123–137. Springer (2014), http://dx.doi.org/10.1007/978-3-319-09764-0_8
20. Sun, X., Ambrossio, D.A.: On the complexity of input/output logic. In: the Proceedings of The Fifth International Conference on Logic, Rationality and Interaction (LORI) (2015)
21. Weiss, G. (ed.): Multiagent systems. MIT press, 2 edn. (2013)
22. Wooldridge, M.J.: An Introduction to MultiAgent Systems (2. ed.). Wiley (2009)