A Decision Algorithm for the CL15 Fragment of Computability Logic

Stella Spadoni

IMT School for Advanced Studies Lucca, Lucca, Tuscany, Italy
stella.spadoni@imtlucca.it

Computability Logic (CoL) is a constructive and resource-conscious game semantical approach to interactive computability which captures dynamic and uncertain multi-agent interactions [5, 6]. By replacing the classical notion of truth with that of computability, CoL transitions from a classical logic of states to one of strategies, thus entailing a shift in the understanding of logical validity [9].

In this framework, formulas are interpreted as interactive computational problems, i.e. games played between two pre-defined agents: Machine $\mathcal{M}$, the honest player, and Environment $\mathcal{E}$, the deceitful adversary. A formula is considered valid if there exists an effective strategy (an algorithm) for $\mathcal{M}$ to win the game against any possible behaviour of $\mathcal{E}$.

The question of decidability for several of expressive fragments of CoL has remained an open challenge in the last few years. This work focusses on the propositional fragment CL15, a system whose signature includes negation ($\neg$), parallel disjunction ($\vee$), parallel conjunction ($\wedge$) and branching recurrence operators (⫰, ⫯).

The core of the present contribution is the formulation of a decision algorithm for CL15.

Theorem 1 (Decidability of CL15 [14])**.** *The propositional fragment CL15 is decidable. Given a formula F, there exists an effective procedure that determines in a finite number of steps whether F is provable in CL15, i.e.*

$$\mathit{CL15} \vdash F \Longleftrightarrow F \in \mathit{CL15}.$$

CL15 is a provably sound and complete system within the cirquent calculus (CC) framework [7, 8]. Unlike traditional sequent calculus, CC is a deep-inference deductive system based on graph-like structures that allow for the explicit sharing of resources. A CL15-cirquent is defined as a triple $C = (\vec{F}, \vec{U}, \vec{O})$, consisting of a sequence of formulas, undergroups, and overgroups all connected by arcs, which enable the system to account for internal and external resource dependencies of interactive games.

The inference rules of CL15 include standard structural rules alongside specific operational ones for disjunction, conjunction and recurrence introduction. The unrestricted application of the contraction rule affects decidability proofs, as it potentially leads to infinite proof-search branches.

Definition 1 (CL15 rules of inference)**.**

- ***Axiom*** (A) : *The conclusion of this premiseless rule looks like an array of n diamonds (with $n \geq 1$) where the formulas are $\neg F$ and F, for some formula F;*

- ***Exchange*** (E) : *Allows for the bottom-up swap of any two adjacent objects in a cirquent, while preserving all formulas, groups and arcs;*

- ***Weakening*** (W) : *The premise is obtained by deleting an arc between some undergroup and a formula;*

- ***Contraction*** (C) : *The premise is obtained by replacing a recurrence formula of the form* $⫯F$ *with two adjacent identical ones* $⫯F, ⫯F$ *(with the condition of including them in the exact same groups the original formula was in);*

- ***Duplication*** (D) : *The premise is obtained by merging any two undergroups or overgroups from the conclusion and by keeping the same contents;*

- ***Merging*** (M) : *The premise is obtained by duplicating any overgroup from the conclusion and by keeping the same contents.*

- ***Disjunction introduction*** $(\vee)$: *The premise is obtained by replacing a disjunctive formula of the form* $F \vee G$ *with the two adjacent disjuncts* F, G *and by including them in the same groups of the original formula;*

- ***Conjunction introduction*** $(\wedge)$: *The premise is obtained by replacing a conjunctive formula of the form* $F \wedge G$ *with the two adjacent conjuncts* F, G *and by replacing the original undergroup with two adjacent undergroups singularly containing the two separate conjuncts;*

- ***Recurrence introduction*** $(⫰)$: *The premise is obtained by replacing a recurrence formula of the form* $⫰F$ *with the formula* F *and then by inserting anywhere a new overgroup that contains only* F*;*

- ***Corecurrence introduction*** $(⫯)$: *The premise is obtained by replacing a recurrence formula of the form* $⫯F$ *with the formula* F *and by including it in any number of existing overgroups.*

Our algorithm operates by conducting a proof search in a modified version of the system, designated as CL15$^{\overline{C}}$, where the rule of contraction is bounded to one use per branch in the proof tree. The convergence of this search is established in three steps, i.e. by:

1. Considering CL15 without the contraction rule and showing that it is decidable;

2. Proving that CL15$^{\overline{C}}$ is equivalent to CL15;

3. Showing that CL15$^{\overline{C}}$ is, indeed, decidable.

Crucially, we ultimately assess the decidability of the fragment by virtue of three elements [14]: the finiteness of inference rules; the strictly decreasing size of a novel complexity measure counting recurrence occurrences in each cirquent; and a bound on contraction application (which provably ensures the same deductive power by avoiding potential infinite loops).

Our proofs are constructive and syntactic in nature, thus providing a concrete effective procedure for determining CL15-provability, so that we obtain an algorithm that is inherently implementable. As CoL is designed to serve as a tool for managing interactive computational tasks, a decision algorithm provides a reliable engine for automated planning, strategy synthesis and formal verification where guaranteed algorithmic termination is required. The methodology developed here for CL15 may also provide a blueprint for tackling the decidability of further fragments of CoL and their applications in different domains [3, 10, 11, 12, 13].

References

[1] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. In: Journal of Symbolic Logic 59 (2), pp. 543–574, 1994. Available at https://arxiv.org/abs/1311.6057.

[2] M. Bauer. The computational complexity of propositional cirquent calculus. In: Logical Methods in Computer Science 11[1:12], pp. 1–16, 2015. Available at https://arxiv.org/abs/1401.1849.

[3] C. Perini Brogi and S. Spadoni. Cutting out the middle man: A game-theoretic analysis in computability logic of the needham-schroeder protocol. In: D. Maiorca, P. Samarati (Eds.), Proceedings of the Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), Cagliari, Italy, February 09-13, 2026, volume 4198 of CEUR Workshop Proceedings, CEUR-WS.org, 2026. Available at https://ceur-ws.org/Vol-4198/paper57.pdf.

[4] A. Guglielmi. A system of interaction and structure. In: ACM Transactions on Computational Logic 8, no.1, pp. 1–64, 2007. Available at https://dl.acm.org/doi/pdf/10.1145/1182613.1182614.

[5] G. Japaridze. Introduction to computability logic. Annals of Pure and Applied Logic, vol. 123, no. 1–3, pp. 1–99, 2003. Available at https://www.sciencedirect.com/science/article/pii/S016800720300023X.

[6] G. Japaridze. Computability logic: a formal theory of interaction. In: D. Goldin, S. Smolka, P. Wegner (Eds.), Interactive Computation: The New Paradigm, Springer, pp. 183–223, 2006. Available at https://arxiv.org/abs/cs/0404024.

[7] G. Japaridze. The taming of recurrences in computability logic through cirquent calculus, part i. Archive for Mathematical Logic, vol. 52, no. 2, pp. 173–212, 2013. Available at https://arxiv.org/abs/1105.3853.

[8] G. Japaridze. The taming of recurrences in computability logic through cirquent calculus, part ii. Archive for Mathematical Logic, vol. 52, no. 2, pp. 213–259, 2013. Available at https://arxiv.org/abs/1106.3705.

[9] G. Japaridze. Fundamentals of computability logic. In: M. Fitting, M. Pourmahdian, A. Rezus, A. Daghighi (Eds.), Research Trends in Contemporary Logic, College Publications, pp. 1–37, 2020. Available at https://arxiv.org/abs/1904.01431.

[10] K. Kwon. Expressing algorithms as concise as possible via computability logic. In: IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E97-A, pp. 1385–1387, 2014. Available at https://arxiv.org/abs/1305.2004.

[11] K. Kwon. Towards distributed logic programming based on computability logic, 2019. Available at https://arxiv.org/abs/1909.07036.

[12] K. Kwon. Computability-logic web: an alternative to deep learning, 2020. Available at https://arxiv.org/abs/2101.09222.

[13] K. Kwon. What is an algorithm?: a modern view, 2021. Available at https://arxiv.org/abs/2108.10728.

[14] S. Spadoni. The fertile steppe: Computability logic and the decidability of one of its fragments, 2025. Available at https://arxiv.org/abs/2503.05826.

[15] W. Xu and S. Liu. The parallel versus branching recurrences in computability logic. In: Notre Dame Journal of Formal Logic, vol. 54, no.1, pp. 61–78, 2013. Available at https://arxiv.org/abs/1107.2284.