Simpson's Proof Systems for Process Verification: A Fine-tuning (short paper)

Cosimo Perini Brogi[1,*,†], Rocco De Nicola[1,†] and Omar Inverso[2,†]

[1] *IMT School for Advanced Studies Lucca, Italy*
[2] *Gran Sasso Science Institute, Italy*

Abstract

We refine Alex Simpson's approach to formal verification of properties of concurrent systems via proof systems. Our new sequent calculus $\mathsf{G3HML}_{\mathsf{GSOS}}$ seamlessly harmonises the GSOS semantics for process calculi (for system formalisation) with Hennessy-Milner logic (for formal property specification) in a pure G3-style system, from which the cut-rule is effectively eliminated. We achieve this substantial improvement by applying the geometrisation of formal theories introduced in the proof-theoretic literature by Roy Dyckhoff and Sara Negri. We communicate here our design methodology to fine-tune Simpson's calculi, which we consider promising to cover, in the future, more expressive process algebra's specification formalisms and logics in a principled and uniform manner.

Keywords

Formal methods, Concurrent systems, Structural proof theory, Process algebras, Structural operational semantics, Cut-elimination, Compositional verification.

1. Introduction

It is well-known that computational/behavioural equivalence of concurrent processes can be logically characterised by (possibly variants and extensions of) Hennessy-Milner modal systems [1, 2, 3],[4],[5, 6]. These results widened the process verification possibilities through model-checking system properties expressed in the language of suited modal logics [7, 8, 9]. What about the other coin side of verification, i.e., proof development?

1.1. Proof systems for process verification

In his [10], Colin Stirling addressed the research question of finding "*compositional, syntax-directed* proof systems" for verifying properties of concurrent systems expressed in the language of modal logics. Compositional here means proving that process p satisfies property A only involves subproofs for subprocesses of p satisfying subformulas of A.

ICTCS'24: Italian Conference on Theoretical Computer Science, September 11–13, 2024, Torino, Italy

* Corresponding author.
† These authors contributed equally.
cosimo.perinibrogi@imtlucca.it (C. Perini Brogi); rocco.denicola@imtlucca.it (R. De Nicola); omar.inverso@gssi.it (O. Inverso)
https://sysma.imtlucca.it/people/cosimo-perini-brogi (C. Perini Brogi);
https://sysma.imtlucca.it/people/rocco-de-nicola (R. De Nicola);
https://www.gssi.it/people/professors/lectures-computer-science/item/1018-inverso-omar (O. Inverso)
0000-0001-7883-5727 (C. Perini Brogi); 0000-0003-4691-7570 (R. De Nicola); 0000-0002-9348-1979 (O. Inverso)

Stirling partially achieved such a proof system for CCS at the price of breaking the analyticity of the calculus in order to handle restriction and parallel composition operators. The methodology proposed in that seminal paper thus generated calculi that are not structurally complete. As a practical consequence, verification of process properties through proof-search cannot be automated even for recursion-free CCS.

Years later, in [11, 12], Alex Simpson suggested to overcome this weakness by introducing (what are now commonly called) labelled sequent calculi for Hennessy-Milner logic [1, 2] applied to arbitrary GSOS processes [13].

In particular, the paper [12] identifies some minimal requirements for proof systems suitable for process verification, namely: (1) soundness; (2) completion of ordinary verification tasks; (3) parametrised verification; (4) natural implementation of compositional reasoning (as required by Stirling); (5) semantic completeness (within the limits of the logic under consideration); (6) structural completeness and proof analyticity; (7) terminating proof-search (whenever the logic under consideration is decidable).

First, the availability of a cut rule is crucial for implementing a formal verification that satisfies most of these desiderata and is also modular or compositional in the original sense first envisaged by Stirling in [10]. Specifically, whenever we have a derivation of a parametrised property $x_1 : A_1, \cdots, x_n : A_n \Rightarrow \mathit{op}(x_1, \cdots, x_n) : B$ (where *op* is a process operator) and derivations of $\Rightarrow p_1 : A_1, \cdots, \Rightarrow p_n : A_n$, we can then apply substitution and cut to obtain a derivation of $\Rightarrow \mathit{op}(p_1, \cdots, p_n) : B$ (Requirement (4)). Concurrently, the admissibility of the cut rule in the proof systems guarantees that we can, in principle, prove that a given process satisfies a required property through goal-directed verification (Requirement (6)). This capability entails a root-first proof search, guided solely by the structure of the process and the formula expressing the desired property, which may eventually be mechanised to automate the verification task (Requirement (7)). When successful, each derivation step can be directly interpreted in the semantics for the process behaviour (Requirement (1)); in case of failure, it might be possible to extract a countermodel to the checked process property from the aborted proof-search (Requirement (5)).

1.2. Our contribution

In his original papers, Simpson provides a *semantic* proof of cut admissibility – i.e. he derives the technical contents of Requirement (6) from Requirement (5) – in a labelled sequent calculus for Hennessy-Milner logic specifying properties of GSOS processes.

We think we can obtain more general results – covering more general process formalisms and more expressive modal logics for processes – by a more principled proof system design methodology that refines Simpson's calculi and builds on recent advances in structural proof theory.

More precisely, we propose *different sequent rules for process operators*, based on the geometrisation methods discussed in detail by Roy Dyckhoff and Sara Negri [14]. We use them to define a *new* G3-*style labelled sequent calculus*, that we named $\mathsf{G3HML}_{\mathrm{GSOS}}$, targeting the same class of processes and logic as Simpson's work, i.e. Hennessy-Milner logic [1, 2] applied to arbitrary GSOS processes [13].

Our main technical advancement in this proof-theoretic approach to process verification is the *constructive* proof of structural completeness of the calculus. More relevantly, we provide a *cut-elimination algorithm* for G3HML$_{\text{GSOS}}$(Theorem 1), which is an essential first result towards implementing the compositional verification initially discussed by Stirling in [10] and his later work, such as [15, 16].

In other terms, by adopting our calculus design method, we provide a *direct and constructive proof* of Requirement (4) (analyticity) without compromising Requirement (6) (compositionality), still making Requirements (2-3) hold by design (ordinary and parametrised verification). Requirement (1) (soundness) is then straightforward to prove,[1] and Requirements (5) and (7) (semantic completeness, and termination of proof-search for the expected fragments of GSOS, resp.) are proven by Tait-Schütte-Takeuti saturation method.[2]

It is a first but relevant and promising fine-tuning of Simpson's original answer to Stirling's research question. The results we communicate here may prelude to further extensions for more expressive logics (such as those proposed in [3, 20]) and transition system specification formats (aiming to capture in our proof system design the PANTH format [4] seamlessly), by possibly considering recent proof-theoretic advances for e.g. temporal logics [21, 22] and μ-calculi [16, 23, 24]. On the applicative side, they are central for a future development (on the lines of [25, 26]) of certified theorem provers and countermodel constructors for process verification based on our calculi.

In the following pages, we overview and discuss our design choices for G3HML$_{\text{GSOS}}$, focusing on the main difference from Simpson's calculi – i.e. sequent rules for GSOS process – which enables us to define a pure G3-style calculus using only rules that directly express the logical connectives and process operators, for which we can directly prove cut-elimination and structural completeness.

2. Calculus design, structural analysis, and first results

We need first to fix the basic language of G3HML$_{\text{GSOS}}$.
We borrow from the process algebra literature some notations and formal definitions concerning **signatures**, **process terms**, **labelled transition systems** (LTS), and **transition system specifications** (TSS): we refer the reader to e.g. [4] and [6] for that background material. In particular, we denote by $\mathcal{A}_\tau$ the set of **actions**, together with a "silent" action τ, that processes can perform during execution. **Hennessy-Milner logic** consists of a multimodal version of the minimal normal logic K, with modalities indexed by actions in $\mathcal{A}_\tau$. Finally, let us recall that a transition is specified in the **GSOS format** if, and only if, it is defined in a transition system specification via a rule having the following form:

$$(\star)\quad \frac{\{x_i \xrightarrow{\mu_{ij}} y_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\} \quad \cup \quad \{x_i \stackrel{\nu_{ik}}{\not\rightarrow} \mid 1 \leq i \leq n, 1 \leq k \leq \ell_i\}}{f(\vec{x}) \xrightarrow{\pi} p(\vec{x}, \vec{y})}$$

where the x_i's and the y_{ij}'s ($1 \leq i \leq n$ and $1 \leq j \leq m_i$) are all distinct process variables; n, m_i and ℓ_i are natural numbers; $p(\vec{x}, \vec{y})$ is a process term with variables including at most the

[1] See the proof techniques discussed in [17].
[2] See [18] and [19].

x_i's and y_{ij}'s; and the μ_{ij}'s, ν_{ik}'s and π are actions from $\mathcal{A}_\mathcal{T}$. We call any expression of the form $p \xrightarrow{\mu} p'$ a **positive relational atom** – formalising the fact that process p evolves through action μ into process p' – and any expression of the form $p \stackrel{\mu}{\not\rightarrow}$ a **negative relational atom** – formalising the fact that process p does not evolve into any process through action μ – where p, p' are process terms.[3]

We are ready to define the syntax of our proof system $\mathsf{G3HML}_{\text{GSOS}}$.

Definition 1. ***Structural atoms*** *are positive relational atoms, negative relational atoms, or atoms of shape* $p \equiv q$*, where* $\mu \in \mathcal{A}_\mathcal{T}$ *and* $\equiv$ *denotes a given congruence relation (possibly, syntactic equality) between process terms.*
Labelled formulas *are either structural atoms or formulas of shape* $p : A$ *(read as "formula* A *is forced by process* p*"), where* A *is a formula in Hennessy-Milner logic and* p *is a process term. We denote by* φ *a generic labelled formula.*
Sequents *of* $\mathsf{G3HML}_{\text{GSOS}}$ *are expressions* $\Gamma \Rightarrow \Delta$*, where* Γ, Δ *are finite multisets of labelled formulas, and structural atoms may occur only in* Γ*.*

We shall now describe the rules defining our proof system.

Logical rules. The logical rules are the multimodal version of the standard rules for G3K [17], based on the semantic clauses for local forcing relation of Hennessy-Milner formulas on a given labelled transition system. We omit the standard rules for classical propositional connectives, to recall only the modal rules:

$$\frac{q : A, p \xrightarrow{\mu} q, p : [\mu]A, \Gamma \Rightarrow \Delta}{p \xrightarrow{\mu} q, p : [\mu]A, \Gamma \Rightarrow \Delta}\, L\Box \qquad \frac{p \xrightarrow{\mu} y, \Gamma \Rightarrow \Delta, y : A}{\Gamma \Rightarrow \Delta, p : [\mu]A}\, R\Box_{(!y)}$$

The rule $R\Box$ corresponds to the right-to-left direction of the definition of local forcing for $[\mu]A$: the side condition $(!y)$ denotes the requirement on y not to occur in Γ, Δ; in this situation, y is said the *eigenvariable* of the rule. The rule $L\Box$ corresponds to the left-to-right direction of the same definition of local forcing. Notice that this is a *first difference* from the system introduced by Simpson, as, contrary to his system, we can handle modal operators keeping structural atoms only on the *left-hand side* of sequents.

Compositional rules. For each process operator, we need to introduce in $\mathsf{G3HML}_{\text{GSOS}}$ some rules characterising it in terms of the associated GSOS specification. We notice first that each of these rules can be translated into geometric formulas. In fact, any GSOS rule of shape $(\star)$ above states *at the meta-level* that the following formulas $(\circ)$ and $(\circ\circ)$ hold:

$$(\circ)\ \forall \vec{x}, \vec{y} : \left[\left(\bigwedge_{1 \le i \le n, 1 \le j \le m_i} (x_i \xrightarrow{\mu_{ij}} y_{ij}) \ \& \bigwedge_{1 \le i \le n, 1 \le k \le \ell_i} (x_i \stackrel{\nu_{ik}}{\not\rightarrow}\ ) \right) \supset (f(\vec{x}) \xrightarrow{\pi} p(\vec{x}, \vec{y})) \right]$$

[3] A GSOS specification system naturally defines an LTS over the closed terms of a signature by structural induction. Refer to, e.g., [4].

$$(\circ\circ)\ \forall \vec{r}, \vec{y}, z : \left[(f(\vec{r}) \xrightarrow{\pi} z) \supset \left(\exists \vec{y} : p(\vec{r}, \vec{y}) \equiv z \ \& \bigwedge_{1 \le i \le n, 1 \le j \le m_i} (r_i \xrightarrow{\mu_{ij}} y_{ij}) \ \& \bigwedge_{1 \le i \le n, 1 \le k \le \ell_i} (r_i \overset{\nu_{ik}}{\not\to}\ ) \right) \right]$$

These formulas become geometric by adopting the "semidefinitional extension trick" of [14] to turn the generic negative relational atom $p \overset{\mu}{\not\to}$ into $\forall q : (p \xrightarrow{\mu} q \supset \bot)$. Following the methodology of [17, 14], we can then introduce the following rules of our G3HML$_{\text{GSOS}}$ for the GSOS specification of processes[4]

$$\frac{}{p \overset{\mu}{\not\to}\ , p \xrightarrow{\mu} q, \Gamma \Rightarrow \Delta} \overset{\mu}{\not\to} Def$$

$$\frac{f(\vec{x}) \xrightarrow{\pi} p(\vec{x}, \vec{y}),\ \{x_i \xrightarrow{\mu_{ij}} y_{ij}\}_{1 \le i \le n, 1 \le j \le m_i},\ \{x_i \overset{\nu_{ik}}{\not\to}\ \}_{1 \le i \le n, 1 \le k \le \ell_i}, \Gamma \Rightarrow \Delta}{\{x_i \xrightarrow{\mu_{ij}} y_{ij}\}_{1 \le i \le n, 1 \le j \le m_i},\ \{x_i \overset{\nu_{ik}}{\not\to}\ \}_{1 \le i \le n, 1 \le k \le \ell_i}, \Gamma \Rightarrow \Delta} f\circ$$

$$\frac{\left\{ p_h(\vec{r}, \vec{y}) \equiv z,\ \{r_i \xrightarrow{\mu_{ij}} y_{ij}\}_{1 \le i \le n, 1 \le j \le m_i},\ \{x_i \overset{\nu_{ik}}{\not\to}\ \}_{1 \le i \le n, 1 \le k \le \ell_i}, f(\vec{r}) \xrightarrow{\pi} z, \Gamma \Rightarrow \Delta \right\}_{1 \le h \le N}}{f(\vec{r}) \xrightarrow{\pi} z, \Gamma \Rightarrow \Delta} f\circ\circ_{(!\vec{y})}$$

$$\frac{Atm(q), Atm(p), q \equiv p, \Gamma \Rightarrow \Delta}{Atm(p), q \equiv p, \Gamma \Rightarrow \Delta} \textsc{Repl}_1 \qquad \frac{Atm(p), Atm(q), q \equiv p, \Gamma \Rightarrow \Delta}{Atm(q), q \equiv p, \Gamma \Rightarrow \Delta} \textsc{Repl}_2$$

In the replacement rules, $Atm(x)$ stands for $x \xrightarrow{\mu} p$, or $p \xrightarrow{\mu} x$, or $x \overset{\mu}{\not\to}\ $, or $x \equiv p$. In the rule $f \circ \circ$, N is the number of GSOS rules in the TSS for the operator f and action π; the side condition $(!\vec{y})$ denotes the requirement on y's not to occur in the conclusion of $f \circ \circ$ (*eigenvariable* condition).

These rules mark the *main difference* from Simpson's original calculus [12, p. 301], as we are, once again, able to handle process operators by keeping structural atoms only on the left-hand side of sequents, so that we can dismiss all the additional non-logical rules of [12, Fig. 4], proving straightforwardly the **structural completeness of our** G3HML$_{\text{GSOS}}$**.**

More relevantly, contrary to what happens for Simpson's system, we can give a *direct and constructive proof of cut-elimination* by standard double induction:[5]

Theorem 1 (Cut-elimination algorithm)**.**
The rule of cut

$$\frac{\Gamma \Rightarrow \Delta, p : A \qquad p : A, \Gamma' \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'} Cut$$

(where $p : A$ *is called the* cut formula *of the rule) is admissible in* G3HML$_{\mathit{GSOS}}$.

[4]For some operators, further rules (omitted here) should be added to deal with those instances presenting a duplication of the atoms in the conclusion in order to have contraction rules height-preserving admissible. We refer the reader to [17] for an extensive discussion.

[5]From cut admissibility, we easily derive: admissibility of generalised replacement rules; axiomatic completeness w.r.t Hennessy-Milner logic; (semantic) ω-completeness via Tait-Schütte-Takeuti saturation method.

Acknowledgments

This work was partially funded by: the project SERICS – Security and Rights in CyberSpace PE0000014, financed within PNRR, M4C2 I.1.3, funded by the European Union - NextGenerationEU (MUR Code: 2022CY2J5S); the MIUR project PRIN 2017FTXR7S IT MATTERS (Methods and Tools for Trustworthy Smart Systems); Istituto Nazionale di Alta Matematica – INdAM group GNSAGA.

We thank three anonymous referees for their comments and feedback on the first submission.

References

[1] M. Hennessy, R. Milner, On observing nondeterminism and concurrency, in: J. W. de Bakker, J. van Leeuwen (Eds.), Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherlands, July 14-18, 1980, Proceedings, volume 85 of *Lecture Notes in Computer Science*, Springer, 1980, pp. 299–309. URL: https://doi.org/10.1007/3-540-10003-2_79. doi:10.1007/3-540-10003-2_79.

[2] M. Hennessy, R. Milner, Algebraic laws for nondeterminism and concurrency, Journal of the ACM (JACM) 32 (1985) 137–161.

[3] R. De Nicola, F. W. Vaandrager, Three logics for branching bisimulation, J. ACM 42 (1995) 458–487. URL: https://doi.org/10.1145/201019.201032. doi:10.1145/201019.201032.

[4] L. Aceto, W. J. Fokkink, C. Verhoef, Structural operational semantics, in: J. A. Bergstra, A. Ponse, S. A. Smolka (Eds.), Handbook of Process Algebra, North-Holland / Elsevier, 2001, pp. 197–292. URL: https://doi.org/10.1016/b978-044482830-9/50021-7. doi:10.1016/B978-044482830-9/50021-7.

[5] R. De Nicola, Behavioral equivalences, in: Encyclopedia of Parallel Computing, Springer, 2011, pp. 120–127.

[6] R. De Nicola, Process algebras, in: Encyclopedia of Parallel Computing, Springer, 2011, pp. 1624–1636.

[7] E. Clarke, O. Grumberg, D. Peled, D. Peled, Model Checking, The Cyber-Physical Systems Series, MIT Press, 1999.

[8] L. Aceto, A. Ingólfsdóttir, K. G. Larsen, J. Srba, Reactive systems: modelling, specification and verification, Cambridge University Press, 2007.

[9] J. R. Andersen, N. Andersen, S. Enevoldsen, M. M. Hansen, K. G. Larsen, S. R. Olesen, J. Srba, J. K. Wortmann, CAAL: concurrency workbench, Aalborg edition, in: Theoretical Aspects of Computing-ICTAC 2015: 12th International Colloquium, Cali, Colombia, October 29-31, 2015, Proceedings 12, Springer, 2015, pp. 573–582.

[10] C. Stirling, Modal logics for communicating systems, Theoretical Computer Science 49 (1987) 311–347.

[11] A. K. Simpson, Compositionality via Cut-Elimination: Hennessy-Milner Logic for an Arbitrary GSOS, in: Proceedings, 10th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, USA, June 26-29, 1995, IEEE Computer Society, 1995, pp. 420–430. URL: https://doi.org/10.1109/LICS.1995.523276. doi:10.1109/LICS.1995.523276.

[12] A. K. Simpson, Sequent calculi for process verification: Hennessy-Milner logic for an

arbitrary GSOS, J. Log. Algebraic Methods Program. 60-61 (2004) 287–322. URL: https://doi.org/10.1016/j.jlap.2004.03.004. doi:10.1016/J.JLAP.2004.03.004.

[13] B. Bloom, S. Istrail, A. R. Meyer, Bisimulation can't be traced, J. ACM 42 (1995) 232–268. URL: https://doi.org/10.1145/200836.200876. doi:10.1145/200836.200876.

[14] R. Dyckhoff, S. Negri, Geometrisation of first-order logic, Bull. Symb. Log. 21 (2015) 123–163. URL: https://doi.org/10.1017/bsl.2015.7. doi:10.1017/BSL.2015.7.

[15] C. Stirling, A proof system with names for modal mu-calculus, in: A. Banerjee, O. Danvy, K. Doh, J. Hatcliff (Eds.), Semantics, Abstract Interpretation, and Reasoning about Programs: Essays Dedicated to David A. Schmidt on the Occasion of his Sixtieth Birthday, Manhattan, Kansas, USA, 19-20th September 2013, volume 129 of *EPTCS*, 2013, pp. 18–29. URL: https://doi.org/10.4204/EPTCS.129.2. doi:10.4204/EPTCS.129.2.

[16] C. Stirling, A tableau proof system with names for modal mu-calculus, in: A. Voronkov, M. V. Korovina (Eds.), HOWARD-60: A Festschrift on the Occasion of Howard Barringer's 60th Birthday, volume 42 of *EPiC Series in Computing*, EasyChair, 2014, pp. 306–318. URL: https://doi.org/10.29007/lwqm. doi:10.29007/lwqm.

[17] S. Negri, J. Von Plato, Proof analysis: a contribution to Hilbert's last problem, Cambridge University Press, 2011.

[18] A. S. Troelstra, H. Schwichtenberg, Basic proof theory, Second Edition, volume 43 of *Cambridge tracts in theoretical computer science*, Cambridge University Press, 2000.

[19] A. Indrzejczak, Sequents and trees, Studies in Universal Logic, Birkhäuser, Cham (2021).

[20] R. De Nicola, M. Loreti, A modal logic for mobile agents, ACM Trans. Comput. Log. 5 (2004) 79–128. URL: https://doi.org/10.1145/963927.963930. doi:10.1145/963927.963930.

[21] B. Afshari, L. Grotenhuis, G. E. Leigh, L. Zenger, Ill-founded proof systems for intuitionistic linear-time temporal logic, in: R. Ramanayake, J. Urban (Eds.), Automated Reasoning with Analytic Tableaux and Related Methods - 32nd International Conference, TABLEAUX 2023, Prague, Czech Republic, September 18-21, 2023, Proceedings, volume 14278 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 223–241. URL: https://doi.org/10.1007/978-3-031-43513-3_13. doi:10.1007/978-3-031-43513-3_13.

[22] B. Afshari, G. E. Leigh, G. M. Turata, A cyclic proof system for full computation tree logic, in: B. Klin, E. Pimentel (Eds.), 31st EACSL Annual Conference on Computer Science Logic, CSL 2023, February 13-16, 2023, Warsaw, Poland, volume 252 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, pp. 5:1–5:19. URL: https://doi.org/10.4230/LIPIcs.CSL.2023.5. doi:10.4230/LIPICS.CSL.2023.5.

[23] B. Afshari, G. E. Leigh, G. M. Turata, Demystifying μ, CoRR abs/2401.01096 (2024). URL: https://doi.org/10.48550/arXiv.2401.01096. doi:10.48550/ARXIV.2401.01096. arXiv:2401.01096.

[24] B. Afshari, S. Enqvist, G. E. Leigh, Cyclic proofs for the first-order μ-calculus, Log. J. IGPL 32 (2024) 1–34. URL: https://doi.org/10.1093/jigpal/jzac053. doi:10.1093/JIGPAL/JZAC053.

[25] M. Maggesi, C. Perini Brogi, A formal proof of modal completeness for provability logic, in: L. Cohen, C. Kaliszyk (Eds.), 12th International Conference on Interactive Theorem Proving, ITP 2021, June 29 to July 1, 2021, Rome, Italy (Virtual Conference), volume 193 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 26:1–26:18. URL: https://doi.org/10.4230/LIPIcs.ITP.2021.26. doi:10.4230/LIPICS.ITP.2021.26.

[26] M. Maggesi, C. Perini Brogi, Mechanising Gödel-Löb provability logic in HOL light, J.

Autom. Reason. 67 (2023) 29. URL: https://doi.org/10.1007/s10817-023-09677-z. doi:10.1007/S10817-023-09677-Z.
[27] L. Viganò, Labelled non-classical logics, Springer Science & Business Media, 2013.
[28] A. K. Simpson, The proof theory and semantics of intuitionistic modal logic, University of Edinburgh. College of Science and Engineering. (1994).