University of Birmingham
Research at Birmingham

Cyclic Hypersequent System for Transitive Closure Logic

Das, Anupam; Girlando, Marianna

DOI:
10.1007/s10817-023-09675-1

License:
Creative Commons: Attribution (CC BY)

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Das, A & Girlando, M 2023, 'Cyclic Hypersequent System for Transitive Closure Logic', *Journal of Automated Reasoning*, vol. 67, no. 3, 27. https://doi.org/10.1007/s10817-023-09675-1

Link to publication on Research at Birmingham portal

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 29. Apr. 2024

Journal of Automated Reasoning (2023) 67:27
https://doi.org/10.1007/s10817-023-09675-1

Cyclic Hypersequent System for Transitive Closure Logic

Anupam Das[1] · **Marianna Girlando**[2]

Received: 3 February 2023 / Accepted: 19 June 2023
© The Author(s) 2023

Abstract

We propose a cut-free cyclic system for transitive closure logic (TCL) based on a form of *hypersequents*, suitable for automated reasoning via proof search. We show that previously proposed sequent systems are cut-free incomplete for basic validities from Kleene Algebra (KA) and propositional dynamic logic (PDL), over standard translations. On the other hand, our system faithfully simulates known cyclic systems for KA and PDL, thereby inheriting their completeness results. A peculiarity of our system is its richer correctness criterion, exhibiting 'alternating traces' and necessitating a more intricate soundness argument than for traditional cyclic proofs.

Keywords Cyclic proofs · Transitive closure logic · Hypersequents · Propositional dynamic logic · Proof theory

1 Introduction

Transitive closure logic (TCL) is the extension of first-order logic by an operator computing the transitive closure of definable binary relations. It has been studied by numerous authors, e.g. [15–17], and in particular has been proposed as a foundation for the mechanisation and automation of mathematics [18].

Recently, Cohen and Rowe have proposed *non-wellfounded* and *cyclic* systems for TCL [8, 10]. These systems differ from usual ones by allowing proofs to be infinite (finitely branching) trees, rather than finite ones, under some appropriate global correctness condition (the 'progressing criterion'). One particular feature of the cyclic approach to proof theory is the facilitation of automation, since complexity of inductive invariants is effectively traded off for a richer proof structure. In fact this trade off has recently been made formal, cf. [1, 11], and has led to successful applications to automated reasoning, e.g. [6, 7, 25, 28, 29].

In this work we investigate the capacity of cyclic systems to automate reasoning in TCL (refer to Fig. 1 for a summary of our contributions). Our starting point is the demonstration of

---

✉ Marianna Girlando
m.girlando@uva.nl

Anupam Das
a.das@bham.ac.uk

[1] University of Birmingham, Birmingham, UK

[2] ILLC, University of Amsterdam, Amsterdam, The Netherlands

Published online: 16 August 2023

$$\begin{tikzcd}
\text{Logics} & \mathrm{PDL}^+ / \mathrm{PDL} \arrow[r, hook] \arrow[d, Leftrightarrow] & \mathrm{TCL} \ / \ \mathrm{TCL}^= \\
\text{Cyclic proof systems} & \mathsf{LPD}^+ / \mathsf{LPD} \arrow[r, hook, "(\star)"] \arrow[dr, "(\star)"', "|" marking] & \mathsf{HTC} \ / \ \mathsf{HTC}^= \arrow[u, Rightarrow, "(\star)"'] \\
& & \mathsf{TC}_G \ / \ \mathsf{TC}_G^= \arrow[u, hook, "(\star)"']
\end{tikzcd}$$

Fig. 1 The diagram displays results from the literature together with our contributions, marked with $(\star)$. Double arrows represent soundness and completeness results, which for PDL^+ and the class of cyclic sequent proofs LPD is known from [21] (cfr. Sect. 6.1). Hooked arrows represent simulations via translations: PDL^+ can be simulated by TCL, under the standard translation (cfr. Sect. 3.2). TC_G is the class of cyclic sequent proofs for TCL, introduced in [10], that cannot simulate LPD proofs (cfr. Sect. 3.3). Our contribution is the hypersequential and cyclic proof system HTC, for which we prove soundness (Sect. 5) and completeness via simulation of LPD (Sect. 6). These results can be extended to full PDL and $\mathrm{TCL}^=$ (with identity), indicated by the right components of each node (in blue)

a key shortfall of Cohen and Rowe's system: its cut-free fragment, here called TC_G, is unable to cyclically prove even standard theorems of relational algebra, e.g. $(a \cup b)^* = a^*(ba^*)^*$ and $(aa \cup aba)^+ \leq a^+((ba^+)^+ \cup a))$ (Theorem 3.7). An immediate consequence of this is that cyclic proofs of TC_G do not enjoy cut-admissibility (Corollary 3.14). On the other hand, these (in)equations are theorems of Kleene Algebra (KA) [19, 20], a decidable theory which admits automation-via-proof-search thanks to the recent cyclic system of Das and Pous [13].

What is more, TCL is well-known to interpret Propositional Dynamic Logic (PDL), a modal logic whose modalities are just terms of KA, by a natural extension of the 'standard translation' from (multi)modal logic to first-order logic (see, e.g., [2, 3]). Incompleteness of cyclic-TC_G for PDL over this translation is inherited from its incompleteness for KA. This is in stark contrast to the situation for modal logics without fixed points: the standard translation from K (and, indeed, all logics in the 'modal cube') to first-order logic actually *lifts* to cut-free proofs for a wide range of modal logic systems, cf. [22, 23].

A closer inspection of the systems for KA and PDL reveals the stumbling block to any simulation: these systems implicitly conduct a form of 'deep inference', by essentially reasoning underneath $\exists$ and $\wedge$. Inspired by this observation, we propose a form of *hypersequents* for predicate logic, with extra structure admitting the deep reasoning required. We present the cut-free system HTC and a novel notion of cyclic proof for these hypersequents. In particular, the incorporation of some deep inference at the level of the rules necessitates an 'alternating' trace condition corresponding to *alternation* in automata theory.

Our first main result is the Soundness Theorem (Theorem 5.1): non-wellfounded proofs of HTC are sound for *standard semantics*. The proof is rather more involved than usual soundness arguments in cyclic proof theory, due to the richer structure of hypersequents and the corresponding progress criterion. Our second main result is the Simulation Theorem (Theorem 6.1): HTC is complete for PDL over the standard translation, by simulating a cut-free cyclic system for the latter. This result can be seen as a formal interpretation of cyclic modal proof theory within cyclic predicate proof theory, in the spirit of [22, 23]. To simplify the exposition, we shall mostly focus on equality-free TCL and 'identity-free' PDL during this paper, though we indeed present an extension to the general case (for TCL with equality and PDL with tests) towards the end of this paper in Sect. 7.

The paper is structured as follows. Section 2 introduces TCL, its semantics, and the cyclic sequent calculus for TCL from Cohen and Rowe [10]. Section 3 introduces PDL^+, the identity-free version of PDL, and the standard translation, and shows that the cyclic system for

TCL by Cohen and Rowe is incomplete for PDL^+. Section 4 presents the cyclic hypersequent calculus for TCL, Sect. 5 shows that it is sound and Sect. 6 proves its completeness over PDL^+ with respect to the standard translation via a cyclic sequent calculus for PDL^+. Finally, Sect. 7 discusses the extension of our calculus to full TCL and PDL, and Sect. 8 presents further insights and conclusions.

This paper is a full version of the conference paper [12] published at *IJCAR '22*. It extends the conference version by providing full definitions, detailed proofs and additional examples.

2 Preliminaries

We shall work with a fixed first-order vocabulary consisting of a countable set Pr of unary *predicate* symbols, written p, q, etc., and of a countable set Rel of binary *relation* symbols, written a, b, etc. We build formulas from this language differently in the modal and predicate settings, but all our formulas may be formally evaluated within *structures*:

Definition 2.1 *(Structures)* A *structure* $\mathcal{M}$ consists of a set D, called the *domain* of $\mathcal{M}$, which we sometimes denote by $|\mathcal{M}|$; a subset $p^{\mathcal{M}} \subseteq D$ for each $p \in \mathsf{Pr}$; and a subset $a^{\mathcal{M}} \subseteq D \times D$ for each $a \in \mathsf{Rel}$.

As above, we shall generally distinguish the words 'predicate' (unary) and 'relation' (binary). We could include further relational symbols too, of higher arity, but choose not to in order to calibrate the semantics of both our modal and predicate settings.

2.1 Transitive Closure Logic

In addition to the language introduced at the beginning of this section, in the predicate setting we further make use of a countable set of *function* symbols, written f^i, g^j, etc., where the superscripts $i, j \in \mathbb{N}$ indicate the *arity* of the function symbol and may be omitted when it is not ambiguous. Nullary function symbols (aka *constant* symbols), are written c, d etc. We shall also make use of *variables*, written x, y, etc., typically bound by quantifiers. *Terms*, written s, t, etc., are generated as usual from variables and function symbols by function application. A term is *closed* if it has no variables.

We consider the usual syntax for first-order logic formulas over our language, with an additional operator for transitive closure (and its dual). Formally TCL formulas, written A, B, etc., are generated as follows:

$$\begin{aligned} A, B \ ::= \ & p(t) \mid \bar{p}(t) \mid a(s,t) \mid \bar{a}(s,t) \mid (A \wedge B) \mid (A \vee B) \mid \forall x A \mid \exists x A \mid \\ & TC(\lambda x, y.A)(s,t) \mid \overline{TC}(\lambda x, y.A)(s,t) \end{aligned}$$

When variables x, y are clear from context, we may write $TC(A(x,y))(s,t)$ or $TC(A)(s,t)$ instead of $TC(\lambda x, y.A)(s,t)$, as an abuse of notation, and similarly for $\overline{TC}$. Without loss of generality, we assume that the same variable cannot occur free and bound within the scope of the quantifiers, TC- and $\overline{TC}$-formulas. We write $A[t/x]$ for the formula obtained from A by replacing every free occurrence of the variable x by the term t. The choice of allowing negation only on atomic proposition, and not including implication as a primitive operator in the language, is motivated by the fact that we will opt for a one-sided definition of sequents.

Remark 2.2 (Equality) Note that we do not include term equality among our atomic formulas at this stage. Later we shall indeed consider such extensions, for which the syntax and semantics are as usual for predicate logic.

Definition 2.3 *(Duality)* For a formula A we define its *complement*, $\bar{A}$, by:

$$\begin{array}{rlrlrl}
\overline{p(t)} &:= \bar{p}(t) & \overline{\forall x A} &:= \exists x \bar{A} & & \\
\overline{\bar{p}(t)} &:= p(t) & \overline{\exists x A} &:= \forall x \bar{A} & \overline{TC(A)(s,t)} &:= \overline{TC}(\bar{A})(s,t) \\
\overline{a(s,t)} &:= \bar{a}(s,t) & \overline{A \wedge B} &:= \bar{A} \vee \bar{B} & \overline{\overline{TC}(A)(s,t)} &:= TC(\bar{A})(s,t) \\
\overline{\bar{a}(s,t)} &:= a(s,t) & \overline{A \vee B} &:= \bar{A} \wedge \bar{B} & &
\end{array}$$

We shall employ standard logical abbreviations, e.g. $A \to B$ for $\bar{A} \vee B$. We may evaluate formulas with respect to a structure, but we need additional data for interpreting function symbols:

Definition 2.4 *(Interpreting function symbols)* Let $\mathcal{M}$ be a structure with domain D. An *interpretation* is a map ρ that assigns to each function symbol f^n a function $D^n \to D$. We may extend any interpretation ρ to an action on (closed) terms by setting recursively $\rho(f(t_1, \ldots, t_n)) := \rho(f)(\rho(t_1), \ldots, \rho(t_n))$.

We only consider *standard* semantics in this work: TC (and $\overline{TC}$) is always interpreted as the *real* transitive closure (and its dual) in a structure, rather than being axiomatised by some induction (and coinduction) principle.

In order to facilitate the formal definition of satisfaction, namely for the quantifier and reflexive transitive closure cases, we shall adopt a standard convention of assuming among our constant symbols arbitrary parameters from the model $\mathcal{M}$. Formally this means that we construe each $v \in D$ as a constant symbol for which we shall always set $\rho(v) = v$.

Definition 2.5 *(Semantics)* Given a structure $\mathcal{M}$ with domain D and an interpretation ρ, the judgement $\mathcal{M}, \rho \models A$ is defined as follows:

- $\mathcal{M}, \rho \models p(t)$ if $\rho(t) \in p^{\mathcal{M}}$.
- $\mathcal{M}, \rho \models \bar{p}(t)$ if $\rho(t) \notin p^{\mathcal{M}}$.
- $\mathcal{M}, \rho \models a(s,t)$ if $(\rho(s), \rho(t)) \in a^{\mathcal{M}}$.
- $\mathcal{M}, \rho \models \bar{a}(s,t)$ if $(\rho(s), \rho(t)) \notin a^{\mathcal{M}}$.
- $\mathcal{M}, \rho \models A \wedge B$ if $\mathcal{M}, \rho \models A$ and $\mathcal{M}, \rho \models B$.
- $\mathcal{M}, \rho \models A \vee B$ if $\mathcal{M}, \rho \models A$ or $\mathcal{M}, \rho \models B$.
- $\mathcal{M}, \rho \models \forall x A$ if, for every $v \in D$, we have $\mathcal{M}, \rho \models A[v/x]$.
- $\mathcal{M}, \rho \models \exists x A$ if, for some $v \in D$, we have $\mathcal{M}, \rho \models A[v/x]$.
- $\mathcal{M}, \rho \models TC(A)(s,t)$ if there are $v_0, \ldots, v_{n+1} \in D$ with $\rho(s) = v_0$, $\rho(t) = v_{n+1}$, such that for every $i \leq n$ we have $\mathcal{M}, \rho \models A(v_i, v_{i+1})$.
- $\mathcal{M}, \rho \models \overline{TC}(A)(s,t)$ if for all $v_0, \ldots, v_{n+1} \in D$ with $\rho(s) = v_0$ and $\rho(t) = v_{n+1}$, there is some $i \leq n$ such that $\mathcal{M}, \rho \models A(v_i, v_{i+1})$.

If $\mathcal{M}, \rho \models A$ for all $\mathcal{M}$ and ρ, we simply write $\models A$.

As expected, we have $\mathcal{M}, \rho \not\models TC(A)(s,t)$ just if $\mathcal{M}, \rho \models \overline{TC}(\bar{A})(s,t)$, and so the two operators are semantically dual. The following statement, that easily follows from the semantics clauses defined above, demonstrates that TC and $\overline{TC}$ duly correspond to least and greatest fixed points.

Fact 2.6 *(TC and $\overline{TC}$ as least and greatest fixed points)* The following hold, for arbitrary $\mathcal{M}$, ρ and x:

$$\mathcal{M}, \rho \models TC(A)(s,t) \iff \mathcal{M}, \rho \models A(s,t) \vee \exists x (A(s,x) \wedge TC(A)(x,t)), \tag{1}$$

$$\mathcal{M}, \rho \models \overline{TC}(A)(s,t) \iff \mathcal{M}, \rho \models A(s,t) \wedge \forall x (A(s,x) \vee \overline{TC}(A)(x,t)). \tag{2}$$

$$\mathsf{id}\ \frac{}{\Gamma, p(t), \bar{p}(t)} \qquad \mathsf{wk}\ \frac{\Gamma}{\Gamma, \Gamma'} \qquad \sigma\ \frac{\Gamma}{\sigma(\Gamma)}$$

$$\vee_i\ \frac{\Gamma, A_i}{\Gamma, A_0 \vee A_1}\ i \in \{0,1\} \qquad \wedge\ \frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B} \qquad \exists\ \frac{\Gamma, A(t)}{\Gamma, \exists x A(x)} \qquad \forall\ \frac{\Gamma, A[c/x]}{\Gamma, \forall x A}\ c \text{ fresh}$$

$$TC_0\ \frac{\Gamma, A(s,t)}{\Gamma, TC(A)(s,t)} \qquad TC_1\ \frac{\Gamma, A(s,r) \quad \Gamma, TC(A)(r,t)}{\Gamma, TC(A)(s,t)}$$

$$\overline{TC}\ \frac{\Gamma, A(s,t) \quad \Gamma, A(s,c), \overline{TC}(A)(c,t)}{\Gamma, \overline{TC}(A)(s,t)}\ c \text{ fresh}$$

……

$$=\ \frac{}{\Gamma, t = t} \qquad \neq\ \frac{\Gamma, A(t)}{\Gamma, s \neq t, A(s)} \qquad \neq\ \frac{\Gamma, A(s)}{\Gamma, s \neq t, A(t)}$$

Fig. 2 Above: Sequent calculus TC_G. The first two lines of the Figure contain the rules of the Tait-style sequent system for first-order predicate logic, without equality. The constant symbol c in the $\forall$-rule and the $\overline{TC}$-rule is called an *eigenvariable*. Below: The standard rules for equality. When added to TC_G, they give the sequent calculus $\mathsf{TC}_G^=$. Colours define traces (see Remark 2.9)

We have included both TC and $\overline{TC}$ as primitive so that we can reduce negation to atomic formulas, allowing a one-sided formulation of proofs. Let us point out that our $\overline{TC}$ operator is not the same as Cohen and Rowe's transitive 'co-closure' operator TC^{op} in [9]. As they already note there, TC^{op} cannot be defined in terms of TC (using negations), whereas $\overline{TC}$ is the formal De Morgan dual of TC and, in the presence of negation, are indeed interdefinable, cf. Definition 2.3.

2.2 Cohen–Rowe Cyclic System for TCL

Cohen and Rowe proposed in [8, 10] a non-wellfounded sequent system for TCL (with equality) extending a standard sequent calculus $\mathsf{LK}^=$ for first-order logic with equality and substitution by rules for TC inspired by its characterisation as a least fixed point, cf. Fact 2.6. A non-wellfounded proof system allows for infinitely long branches, provided that they satisfy a logic-specific progress condition. Here we present a one-sided variation of (the cut-free fragment of) their system, both with and without equality.

Definition 2.7 *(System)* A *sequent*, written Γ, Δ etc., is a set of formulas. The systems TC_G and $\mathsf{TC}_G^=$ are given in Fig. 2: $\mathsf{TC}_G^=$ consists of all the rules displayed, while TC_G does not include the $=$ or $\neq$ rules. $\mathsf{TC}_G^{(=)}$-*preproofs* are possibly infinite trees of sequents generated by the rules of $\mathsf{TC}_G^{(=)}$ (colours may be ignored for now). A preproof is *regular* if it has only finitely many distinct sub-preproofs.

In Fig. 2 σ is a map ("substitution") from constants to terms and other function symbols to function symbols of the same arity, extended to terms, formulas and sequents in the natural way. The substitution rule is redundant for usual provability, but facilitates the definition of 'regularity' in predicate cyclic proof theory.

The notions of non-wellfounded and cyclic proofs for $\mathsf{TC}_G^{(=)}$ are formulated similarly to those for first-order logic with (ordinary) inductive definitions [5]:

Definition 2.8 *(Traces and proofs)* Given a $\mathsf{TC}_G^{(=)}$ preproof $\mathcal{D}$ and a branch $\mathcal{B} = (\mathsf{r}_i)_{i\in\omega}$ (where each r_i is an inference step), a *trace* is a sequence of formulas of the form $(\overline{TC}(A)(s_i, t_i))_{i\geq k}$ such that for all $i \geq k$ either:

1. r_i is not a substitution step and $(s_{i+1}, t_{i+1}) = (s_i, t_i)$; or,
2. r_i is a $\overline{TC}$ step with principal formula $\overline{TC}(A)(s_i, t_i)$ and $(s_{i+1}, t_{i+1}) = (c, t_i)$, where c is the eigenvariable of r_i; or,
3. r_i is a substitution step with respect to σ and $(\sigma(s_{i+1}), \sigma(t_{i+1})) = (s_i, t_i)$.

We say that the trace is *progressing* if the case 2 above happens infinitely often along it. A $\mathsf{TC}_G^{(=)}$-preproof $\mathcal{D}$ is a *proof* if each of its infinite branches has a progressing trace. If $\mathcal{D}$ is regular we call it a *cyclic proof*. We write $\mathsf{TC}_G^{(=)} \vdash_{cyc} A$ if there is a cyclic proof in $\mathsf{TC}_G^{(=)}$ of A.

Remark 2.9 (Traces via colours) Fig. 2 codes the notion of trace by means of colours: along any infinite branch a trace is a monochromatic sequence of formulas (with inference steps as displayed in Fig. 2); if the trace hits a formula in the context Γ in the conclusion of an inference step, it must hit the same formula in the premiss it follows.

Proposition 2.10 (Soundness, [8, 10]) *If* $\mathsf{TC}_G^{(=)} \vdash_{cyc} A$ *then* $\models A$.

In fact (the equality-free version of) this result is subsumed by our main soundness result for HTC (Theorem 5.1) and its simulation of TC_G (Theorem 4.11). A partial converse of Proposition 2.10 is available in the presence of a cut rule:

$$\mathsf{cut}\,\frac{\Gamma, A \qquad \bar{A}, \Gamma}{\Gamma, \Gamma'}$$

Namely, cyclic $\mathsf{TC}_G^{(=)}$ proofs are 'Henkin complete', i.e. complete for all models of a particular axiomatisation of TCL (with or withour equality, resp.) based on (co)induction principles [8, 10]. However, the counterexample we present in the next section implies that cut is not eliminable (Corollary 3.14).

2.3 Differences to [8, 10]

Our formulation of $\mathsf{TC}_G^{(=)}$ differs slightly from the original presentation in [8, 10], but in no essential way. Nonetheless, let us survey these differences now.

One-Sided vs. Two-Sided

Cohen and Rowe employ a two-sided calculus as opposed to our one-sided one, but the difference is purely cosmetic. Sequents in their calculus are written $A_1, \ldots, A_m \Rightarrow B_1, \ldots, B_n$, which may be duly interpreted in our calculus as $\bar{A}_1, \ldots, \bar{A}_m, B_1, \ldots, B_n$. Indeed we may write sequents in this two-sided notation at times in order to facilitate the reading of a sequent and to distinguish left and right formulas. For this reason, Cohen and Rowe do not include a $\overline{TC}$ operator in their calculus, but are able to recover it thanks to a formal negation symbol, cf. Definition 2.3.

TC vs. RTC

Cohen and Rowe's system is originally called RTC_G, rather using a 'reflexive' version RTC of the TC operator. As they mention, this makes no difference in the presence of equality. Semantically we have $RTC(A)(s,t) \iff s = t \vee TC(A)(s,t)$, but this encoding does not lift to proofs, i.e. the RTC rules of [8] are not locally derived in $\mathsf{TC}_G^{=}$ modulo this encoding. However, the encoding $RTC(A)(s,t) := TC((x = y \vee A))(s,t)$ suffices for this purpose.

Alternative Rules and Fixed Point Characterisations

Cohen and Rowe use a slightly different fixed point formula to induce rules for RTC and $\overline{RTC}$ (i.e. RTC on the left) based on the fixed point characterisation,

$$RTC(A)(s,t) \iff s = t \vee \exists x (RTC(A)(s,x) \wedge A(x,t)), \tag{1}$$

decomposing paths 'from the right' rather than the left. These alternative rules induce analogous notions of trace and progress for preproofs such that progressing preproofs enjoy a similar soundness theorem, cf. Proposition 2.10. The reason we employ a slight variation of Cohen and Rowe's system is to remain consistent with how the rules of LPD^+ (or LPD) and HTC (or $\mathsf{HTC}^=$) are devised later. To the extent that we prove things *about* $\mathsf{TC}_G^{(=)}$, namely its (cut-free) regular *in*completeness in Theorem 3.7, the particular choice of rules turns out to be unimportant. The counterexample we present there is robust: it applies to systems with any (and indeed all) of the above rules.

3 Interlude: Motivation from PDL

Given the TCL sequent system proposed by Cohen and Rowe, why do we propose a hypersequential system? Our main argument is that proof search in $\mathsf{TC}_G^{(=)}$ is rather weak, to the extent that cut-free cyclic proofs are unable to simulate a basic (cut-free) system for modal logic PDL (regardless of proof search strategy). At least one motivation here is to 'lift' the *standard translation* from cut-free cyclic proofs for PDL to cut-free cyclic proofs in an adequate system for TCL (with equality).

3.1 Identity-Free PDL

Identity-free propositional dynamic logic (PDL^+) is a version of the modal logic PDL without tests or identity, thereby admitting an 'equality-free' standard translation into predicate logic. Formally, PDL^+ *formulas*, written A, B, etc., and *programs*, written α, β, etc., are generated by the following grammars:

$$\begin{array}{rcl} A, B & ::= & p \mid \overline{p} \mid A \wedge B \mid A \vee B \mid [\alpha]A \mid \langle\alpha\rangle A \\ \alpha, \beta & ::= & a \mid \alpha\beta \mid \alpha \cup \beta \mid \alpha^+ \end{array}$$

Remark 3.1 (Formula metavariables) We are using the same metavariables A, B, etc. to vary over both PDL^+ and TCL formulas. This should never cause confusion due to the context in which they appear. Moreover, this coincidence is suggestive, since many notions we consider, such as duality and satisfaction, are defined in a way that is compatible with both notions of formula.

Definition 3.2 *(Duality)* For a formula A we define its *complement*, $\bar{A}$, by:

$$\bar{\bar{p}} := p \qquad \begin{array}{r}\overline{A \wedge B} := \bar{A} \vee \bar{B} \\ \overline{A \vee B} := \bar{A} \wedge \bar{B}\end{array} \qquad \begin{array}{r}\overline{[\alpha]A} := \langle\alpha\rangle\bar{A} \\ \overline{\langle\alpha\rangle A} := [\alpha]\bar{A}\end{array}$$

We *evaluate* PDL^+ formulas using the traditional relational semantics of modal logic, by associating each program with a binary relation in a structure. Again, we only consider 'standard' semantics:

Definition 3.3 *(Semantics)* Fix a structure $\mathcal{M}$ with domain D. For elements $v \in D$ and programs α we define $\alpha^{\mathcal{M}} \subseteq D \times D$ by:

- ($a^{\mathcal{M}}$ is already given in the specification of $\mathcal{M}$, cf. Definition 2.1).
- $(\alpha\beta)^{\mathcal{M}} := \{(u, v) : \text{there is } w \in D \text{ s.t. } (u, w) \in \alpha^{\mathcal{M}} \text{ and } (w, v) \in \beta^{\mathcal{M}}\}$.
- $(\alpha \cup \beta)^{\mathcal{M}} := \{(u, v) : (u, v) \in \alpha^{\mathcal{M}} \text{ or } (u, v) \in \beta^{\mathcal{M}}\}$.
- $(\alpha^+)^{\mathcal{M}} := \{(u, v) : \text{there are } w_0, \ldots, w_{n+1} \in D \text{ s.t. } u = w_0, v = w_{n+1} \text{ and for every } i \leq n \text{ we have } (w_i, w_{i+1}) \in \alpha^{\mathcal{M}}\}$.

For elements $v \in D$ and formulas A we also define the judgement $\mathcal{M}, v \models A$ by:

- $\mathcal{M}, v \models p$ if $v \in p^{\mathcal{M}}$.
- $\mathcal{M}, v \models \overline{p}$ if $v \notin p^{\mathcal{M}}$.
- $\mathcal{M}, v \models A \wedge B$ if $\mathcal{M}, v \models A$ and $\mathcal{M}, v \models B$.
- $\mathcal{M}, v \models A \vee B$ if $\mathcal{M}, v \models A$ or $\mathcal{M}, v \models B$.
- $\mathcal{M}, v \models [\alpha]A$ if for every $(v, w) \in \alpha^{\mathcal{M}}$ we have $\mathcal{M}, w \models A$.
- $\mathcal{M}, v \models \langle\alpha\rangle A$ if there exists $(v, w) \in \alpha^{\mathcal{M}}$ with $\mathcal{M}, w \models A$.

If $\mathcal{M}, v \models A$ for all $\mathcal{M}$ and $v \in D$, then we write $\models A$.

Note that we are overloading the satisfaction symbol $\models$ here, for both PDL^+ and TCL. This should never cause confusion, in particular since the two notions of satisfaction are 'compatible', given that we employ the same underlying language and structures. In fact such overloading is convenient for relating the two logics, as we shall now see.

3.2 The Standard Translation

The so-called "standard translation" of modal logic into predicate logic is induced by reading the semantics of modal logic as first-order formulas. We now give a natural extension of this that interprets PDL^+ into TCL. At the logical level our translation coincides with the usual one for basic modal logic; our translation of programs, as expected, requires the *TC* operator to interpret the $+$ of PDL^+.

Definition 3.4 For a PDL^+ formula A and program α, we define the *standard translations* $\mathsf{ST}(A)(x)$ and $\mathsf{ST}(\alpha)(x, y)$ as TCL-formulas with free variables x and x, y, resp., inductively as follows,

$$\begin{array}{rlrl}
\mathsf{ST}(p)(x) & := p(x) & \mathsf{ST}(a)(x, y) & := a(x, y) \\
\mathsf{ST}(\bar{p})(x) & := \bar{p}(x) & \mathsf{ST}(\bar{a})(x, y) & := \bar{a}(x, y) \\
\mathsf{ST}(A \vee B)(x) & := \mathsf{ST}(A)(x) \vee \mathsf{ST}(B)(x) & \mathsf{ST}(\alpha \cup \beta)(x, y) & := \mathsf{ST}(\alpha)(x, y) \vee \mathsf{ST}(\beta)(x, y) \\
\mathsf{ST}(A \wedge B)(x) & := \mathsf{ST}(A)(x) \wedge \mathsf{ST}(B)(x) & \mathsf{ST}(\alpha\beta)(x, y) & := \exists z(\mathsf{ST}(\alpha)(x, z) \wedge \mathsf{ST}(\beta)(z, y)) \\
\mathsf{ST}(\langle\alpha\rangle A)(x) & := \exists y(\mathsf{ST}(\alpha)(x, y) \wedge \mathsf{ST}(A)(y)) & \mathsf{ST}(\alpha^+)(x, y) & := TC(\mathsf{ST}(\alpha))(x, y) \\
\mathsf{ST}([\alpha]A)(x) & := \forall y(\overline{\mathsf{ST}(\alpha)(x, y)} \vee \mathsf{ST}(A)(y)) & &
\end{array}$$

where we have written simply $TC(\mathsf{ST}(\alpha))$ instead of $TC(\lambda x, y.\mathsf{ST}(\alpha)(x, y))$.

Example 3.5 By means of example, consider the following formulas:

$$\mathsf{ST}([a \cup b]q)(x) := \forall y((\bar{a}(x, y) \wedge \bar{b}(x, y)) \vee q(y)).$$
$$\mathsf{ST}(\langle (ab)^+ \rangle p)(x) := \exists y(TC(\exists z(a(x, z) \wedge b(z, y))) \wedge p(y)).$$

It is routine to show that $\overline{\mathsf{ST}(A)(x)} = \mathsf{ST}(\bar{A})(x)$, by structural induction on A, justifying our overloading of the notation $\bar{A}$, in both TCL and PDL$^+$. Yet another advantage of using the same underlying language for both the modal and predicate settings is that we can state the following (expected) result without the need for encodings, following by a routine structural induction (see, e.g., [3]):

Theorem 3.6 *For PDL$^+$ formulas A, we have $\mathcal{M}, v \models A$ iff $\mathcal{M} \models \mathsf{ST}(A)(v)$.*

3.3 Cohen–Rowe System is Not Complete for PDL$^+$

PDL$^+$ admits a standard cut-free cyclic proof system LPD^+ (see Sect. 6.1) which is both sound and complete (cf. Theorem 6.4). However, a shortfall of TC_G is that it is unable to cut-free simulate LPD^+. In fact, we can say something stronger:

Theorem 3.7 (Incompleteness) *There exist a PDL$^+$ formula A such that $\models A$ but $\mathsf{TC}_G \nvdash_{cyc} \mathsf{ST}(A)(x)$ (in the absence of cut).*

This means not only that TC_G is unable to locally cut-free simulate the rules of LPD^+, but also that there are some validities for which there are no cut-free cyclic proofs at all in TC_G. One example of such a formula is:

$$\langle (aa \cup aba)^+ \rangle p \rightarrow \langle a^+((ba^+)^+ \cup a) \rangle p. \tag{2}$$

This formula is derived from the well-known PDL validity $\langle (a \cup b)^* \rangle p \rightarrow \langle a^*(ba^*)^* \rangle p$ by identity-elimination. This in turn is essentially a theorem of relational algebra, namely $(a \cup b)^* \leq a^*(ba^*)^*$, which is often used to eliminate $\cup$ in (sums of) regular expressions. The same equation was (one of those) used by Das and Pous in [13] to show that the sequent system LKA for Kleene Algebra is cut-free cyclic incomplete.

In the remainder of this subsection, we shall give a proof of Theorem 3.7. The argument is much more involved than the one from [13], due to the fact we are working in predicate logic, but the underlying basic idea is similar. At a very high level, the right-hand side of (2) (viewed as a relational inequality) is translated to an existential formula $\exists z(\mathsf{ST}(a^+)(x, z) \wedge \mathsf{ST}((ba^+)^+ \cup a)(z, y)$ that, along some branch (namely the one that always chooses aa when decomposing the LHS of (2)) can never be instantiated while remaining valid. This branch witnesses the non-regularity of any proof.

3.3.1 Some Closure Properties for Cyclic Proofs

Demonstrating that certain formulas do *not* have (cut-free) cyclic proofs is a delicate task, made more so by the lack of a suitable model-theoretic account (indeed, cf. Corollary 3.14). In order to do so formally, we first develop some closure properties of cut-free cyclic provability.

Proposition 3.8 (Inversions) *We have the following:*

1. *If $\mathsf{TC}_G \vdash_{cyc} \Gamma, A \vee B$ then $\mathsf{TC}_G \vdash_{cyc} \Gamma, A, B$.*

2. *If* $\mathsf{TC}_G \vdash_{cyc} \Gamma, A \wedge B$ *then* $\mathsf{TC}_G \vdash_{cyc} \Gamma, A$ *and* $\mathsf{TC}_G \vdash_{cyc} \Gamma, B$.
3. *If* $\mathsf{TC}_G \vdash_{cyc} \Gamma, \forall x A(x)$ *then* $\mathsf{TC}_G \vdash_{cyc} \Gamma, A(c)$*, as long as c is fresh.*

Proof Sketch All three statements are proved similarly.

For item 1, replace every direct ancestor of $A \vee B$ with A, B. The only critical steps are when $A \vee B$ is principal, in which case we delete the step, or is weakened, in which case we apply two weakenings, one on A and one on B. If the starting proof had only finitely many distinct subproofs (up to substitution), say n, then the one obtained by this procedure has at most $2n$ distinct subproofs (up to substitution), since we simulate a weakening on $A \vee B$ by two weakenings.

For item 2, replace every direct ancestor of $A \wedge B$ with A or B, respectively. The only critical steps are when $A \wedge B$ is principal, in which case we delete the step and take the left or right subproof, respectively, or is weakened, in which case we simply apply a weakening on A or B, respectively. The proof we obtain has at most the same number of distinct subproofs (up to substitution) as the original one.

For item 3, replace every direct ancestor of $\forall x A(x)$ with $A(c)$. The only critical steps are when $\forall x A(x)$ is principal, in which case we delete the step and rename the eigenvariable in the remaining subproof everywhere with c, or is weakened, in which case we simply apply a weakening on $A(c)$. The proof we obtain has at most the same number of distinct subproofs (up to substitution) as the original one.

Proposition 3.9 (Predicate admissibility) *Suppose* $\mathsf{TC}_G \vdash_{cyc} \Gamma, p(t)$ *or* $\mathsf{TC}_G \vdash_{cyc} \Gamma, \bar{p}(t)$*, where* $\bar{p}$ *or* p *(respectively) does not occur in* Γ*. Then it holds that* $\mathsf{TC}_G \vdash_{cyc} \Gamma$.

Proof sketch Delete every ancestor of $p(t)$ or $\bar{p}(t)$, respectively. The only critical case is when one of the formulas is weakened, in which case we omit the step. Note that there cannot be any identity on p, due to the assumption on Γ, and by the subformula property.

3.3.2 Reducing to a Relational Tautology

Here, and for the remainder of this subsection, we shall simply construe PDL$^+$ programs α and formulas A as TCL formulas with two free variables and one free variable, respectively, by identifying them with their standard translations $\mathsf{ST}(\alpha)(x, y)$ and $\mathsf{ST}(A)(x)$, respectively. This modest abuse of notation will help suppress much of the notation in what follows.

Lemma 3.10 *If* $\mathsf{TC}_G \vdash_{cyc} (\langle (aa \cup aba)^+ \rangle p \to \langle a^+((ba^+)^+ \cup a) \rangle p)(c)$ *then also* $\mathsf{TC}_G \vdash_{cyc} (aa \cup aba)^+(c, d) \to (a^+((ba^+)^+ \cup a))(c, d)$.

Proof sketch Suppose $\mathsf{TC}_G \vdash_{cyc} (\langle (aa \cup aba)^+ \rangle p \to \langle a^+((ba^+)^+ \cup a) \rangle p)(c)$ so, by unwinding the definition of ST and since duality commutes with the standard translation, cf. Sect. 3.2, we have that $\mathsf{TC}_G \vdash_{cyc} ([(aa \cup aba)^+]\bar{p})(c) \vee (\langle a^+((ba^+)^+ \cup a) \rangle p)(c)$. By $\vee$-inversion (Proposition 3.8.1) we have:

$$\mathsf{TC}_G \vdash_{cyc} ([(aa \cup aba)^+]\bar{p})(c), (\langle a^+((ba^+)^+ \cup a) \rangle p)(c).$$

Again unwinding the definition of ST, and by the definition of duality, we thus have:

$$\mathsf{TC}_G \vdash_{cyc} \forall x (\overline{(aa \cup aba)^+}(c, x)) \vee \bar{p}(x)), \exists y ((a^+((ba^+)^+ \cup a)(c, y)) \wedge p(y)).$$

Now, by $\forall$-inversion and $\vee$-inversion, Prop. 3.8, we have:

$$\mathsf{TC}_G \vdash_{cyc} \overline{(aa \cup aba)^+}(c, d), \bar{p}(d), \exists y ((a^+((ba^+)^+ \cup a)(c, y)) \wedge p(y)).$$

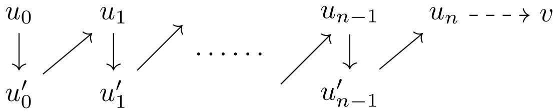

Fig. 3 The adversarial model from Definition 3.12. Solid arrows represent $a^{\mathcal{A}_n}$-relations, dashed arrows $b^{\mathcal{A}_n}$-relations

Without loss of generality we may instantiate the $\exists y$ by d and so by $\wedge$-inversion, Prop. 3.8.2, we have:

$$\mathsf{TC}_G \vdash_{cyc} \overline{(aa \cup aba)^+}(c, d),\ \bar{p}(d),\ a^+((ba^+)^+ \cup a)(c, d).$$

Since there is no occurrence of p above, by Prop. 3.9 we conclude

$$\mathsf{TC}_G \vdash_{cyc} \overline{(aa \cup aba)^+}(c, d),\ a^+((ba^+)^+ \cup a)(c, d),$$

as required.

3.3.3 Irregularity via an Adversarial Model

In the previous subsubsection we reduced the incompleteness of cut-free cyclic sequent proofs for TCL over the image of the standard translation on PDL^+ to the non-regular cut-free provability of a particular relational validity. Unwinding this a little, the sequent that we shall show has no (cut-free) cyclic proof in TC_G can be written in ‘two-sided notation’ (cf. Sect. 2.3) as follows:

$$TC(aa \vee aba)(c, d) \Rightarrow \exists z(a^+(c, z) \wedge ((ba^+)^+ \cup a)(z, d)). \tag{3}$$

This two-sided presentation is simply a notational variant that allows us to more easily reason about the proof search space (e.g. referring to ‘LHS’ and ‘RHS’). Formally:

Convention 3.11 (Two-sided notation) We may write $\Gamma \Rightarrow \Delta$ as shorthand for the sequent $\bar{\Gamma}, \Delta$, where $\bar{\Gamma} = \{\bar{A} : A \in \Gamma\}$. References to the ‘left-hand side (LHS)’ and ‘right-hand side (RHS)’ have the obvious meaning, always with respect to the delimiter $\Rightarrow$.

To facilitate our argument, we shall only distinguish sequents ‘modulo substitution’ rather than allowing explicit substitution steps when reasoning about (ir)regularity of a proof.

We shall design a family of ‘adversarial’ models, and instantiate proof search to just these models. In this way, we shall show that any non-wellfounded TC_G proof of the sequent (3) must have arbitrarily long branches without a repetition (up to substitution). Since TC_G is finitely branching, by König’s Lemma this means that any non-wellfounded TC_G proof of (3) has an infinite branch with no repetitions (up to substitution), as required.

Definition 3.12 *(An adversarial model)* For $n \in \mathbb{N}$, define the structure $\mathcal{A}_n$ by:

- The domain of $\mathcal{A}_n$ is $\{u_0, u'_0, \ldots, u_{n-1}, u'_{n-1}, u_n, v\}$.
- $a^{\mathcal{A}_n} = \{(u_i, u'_i), (u'_i, u_{i+1})\}_{i<n}$.
- $b^{\mathcal{A}_n} = \{(u_n, v)\}$.

Note that, since the sequent (3) that we are considering is purely relational, it does not matter what sets $\mathcal{A}_n$ assigns to the predicate symbols, so we refrain from specifying such data.

Lemma 3.13 *Let $n \in \mathbb{N}$. Any TC_G proof $\mathcal{D}$ of (3) has a branch with no repetitions (up to substitutions) among its first n sequents.*

Proof Set $c_0 = c$. Consider some (possibly finite, but maximal) branch $\mathcal{B} = (\mathsf{r}_i)_{i \leq \nu}$ (with $\nu \leq \omega$) of $\mathcal{D}$ satisfying:

- whenever TC on the LHS is principal (formally speaking, for a $\overline{TC}$ step), the right premiss is followed; and,
- whenever $(aa)(s,t) \vee (aba)(s,t)$ is principal for any s and t on the LHS (formally speaking, for a $\wedge$ step) the left premiss (corresponding to $(aa)(s,t)$) is followed.

Let $k \leq n$ be maximal such that, for each $i \leq k$, r_i has principal formula on the LHS. Now:

1. For $i \leq k$, each r_i has conclusion with LHS of the form:
$$\{\Gamma_j(c_{j-1}, c_j)\}_{j=1}^{l}, TC(aa \vee aba)(c_l, d), \tag{4}$$
for some $l \leq i$ and distinct $c_0, \dots, c_l$ and where each $\Gamma_j(c_{j-1}, c_j)$ has the form $a(c_{j-1}, c'_{j-1}), a(c'_{j-1}, c_j)$ or $a(c_{j-1}, c'_{j-1}) \wedge a(c'_{j-1}, c_j)$ or $(aa)(c_{j-1}, c_j)$ or $(aa)(c_{j-1}, c_j) \vee (aba)(c_{j-1}, c_j)$.
To see this, proceed by induction on $i \leq k$:
    - The base case is immediate, by setting $l = 0$.
    - For the inductive step, note that the principal formula of r_i must be on the LHS, since $i \leq k$. Thus by the inductive hypothesis the principal formula of r_i must have the form:
        - $a(c_{j-1}, c'_{j-1}) \wedge a(c'_{j-1}, c_j)$, in which case the premiss of r_i (which is a $\vee$ step) replaces it by $a(c_{j-1}, c'_{j-1}), a(c'_{j-1}, c_j)$; or,
        - $(aa)(c_{j-1}, c_j)$, in which case the premiss of r_i (which is a $\forall$ step) replaces it by $a(c_{j-1}, c'_{j-1}) \wedge a(c'_{j-1}, c_j)$, for c'_{j-1} a fresh symbol; or,
        - $(aa)(c_{j-1}, c_j) \vee (aba)(c_{j-1}, c_j)$, in which case, by definition of $\mathcal{B}$, the $\mathcal{B}$-premiss of r_i (which is a left-$\vee$ step) replaces this formula by some $a(c_{j-1}, c_j)$; or,
        - $TC(aa \vee aba)(c_l, d)$ for some $l \leq i$, in which case, by definition of $\mathcal{B}$, the $\mathcal{B}$-premiss of r_i (which is a left-TC step) replaces it by the cedent $(aa)(c_l, c_{l+1}) \vee (aba)(c_l, c_{l+1}), TC(aa \vee aba)(c_{l+1}, d)$.
2. Moreover, for $i < i' \leq k$, the conclusion of r_i and $\mathsf{r}_{i'}$ are not equal (up to substitution). To see this, note that any rule principal on an LHS of form (4) either decreases the size of some $\Gamma_j(c_{j-1}, c_j)$ (when it is a $\vee$, $\forall$ or $\wedge$ step) or increases the number of eigenvariables in the sequent (when it is a left TC step), in particular the index l of $TC(aa \vee aba)(c_l, d,)$.
3. Since proofs must be sound for all models (by soundness), we shall work in $\mathcal{A}_n$ with respect to an interpretation ρ_n satisfying $c_i \mapsto u_i$ for $i \leq n$ and $c'_i \mapsto u'_i$ for $i < n$ and $d \mapsto v$. It follows by inspection of (4) that, for $i \leq k$, each formula on the LHS of the conclusion of r_i is true in $(\mathcal{A}_n, \rho_n)$.
4. Along $\mathcal{B}$, the RHS cannot be principal unless $l \geq n$ in (4), so in particular $k \geq n$. To see this:
    - Recall that the interpretation ρ_n assigns to $c_0, c'_0, \dots, c'_{n-1}, c_n$ the worlds $u_0, u'_0, \dots, u'_{n-1}, u_n$ respectively.
    - If the existential formula on the RHS is instantiated by some c_i or c'_i with $i < n$ then the resulting sequent is false in $(\mathcal{A}_n, \rho_n)$ (recall that, by Item 3, every formula on the LHS is true, so we require the RHS to be true too). To see this, note that the RHS

in particular would imply $(ba^+)^+(c_i, d)$ or $a(c_i, d)$ or $(ba^+)^+(c'_i, d)$ or $a(c'_i, d)$. However when $i < n$ none of these formulas are true with respect to $(\mathcal{A}_n, \rho_n)$.
- If the existential formula on the RHS is instantiated by d then the resulting sequent is again false, by the same analysis as above.

By Item 4, we have that $k \geq n$ and so, since we assumed $k \leq n$ at the start, indeed $k = n$. Thus, by Item 1 and Item 2, there are no repeated sequents (up to substitution) in $(\mathsf{r}_i)_{i \leq n}$, as required.

3.3.4 Putting It All Together

We are now ready to give the proof of the main result of this section.

Proof of Theorem 3.7, Sketch Since the choice of n in Lemma 3.13 was arbitrary, any TC_G proof $\mathcal{D}$ of (3) must have branches with arbitrarily long initial segments without any repetition (up to substitution). Since the system is finitely branching, by König's Lemma we have that there is an infinite branch through $\mathcal{D}$ without any repetition (up to substitution), and thus $\mathcal{D}$ is not regular. Thus $\mathsf{TC}_G \nvdash_{cyc}$ (3). Finally, by contraposition of Lemma 3.10, we have, as required:

$$\mathsf{TC}_G \nvdash_{cyc} (\langle (aa \cup aba)^+ \rangle p \rightarrow \langle a^+((ba^+)^+ \cup a)\rangle p)(c).$$

An immediate consequence of Theorem 3.7 and Henkin-completeness of TC_G with cut [8, 10] is:

Corollary 3.14 *The class of cyclic proofs of* TC_G *does not enjoy cut-admissibility.*

4 Hypersequent Calculus for TCL

In light of the preceding subsection, let us take a moment to examine how a 'local' simulation of LPD^+ by TC_G fails, in order to motivate the main system that we shall present. The program rules, in particular the $\langle\,\rangle$-rules, require a form of *deep inference* to be correctly simulated, over the standard translation. For instance, let us consider the action of the standard translation on two rules we shall see later in LPD^+ (cf. Sect. 6.1):

$$\langle\cup\rangle_0\,\frac{\Gamma, \langle a_0\rangle p}{\Gamma, \langle a_0 \cup a_1\rangle p} \quad\rightsquigarrow\quad \frac{\mathsf{ST}(\Gamma)(c), \exists x(a_0(c, x) \wedge p(x))}{\mathsf{ST}(\Gamma)(c), \exists x((a_0(c, x) \vee a_1(c, x)) \wedge p(x))}$$

$$\langle\,\rangle\,\frac{\Gamma, \langle a\rangle\langle b\rangle p}{\Gamma, \langle ab\rangle p} \quad\rightsquigarrow\quad \frac{\mathsf{ST}(\Gamma)(c), \exists y(a(c, y) \wedge \exists x(b(y, x) \wedge p(x)))}{\mathsf{ST}(\Gamma)(c), \exists x(\exists y(a(c, y) \wedge b(y, x)) \wedge p(x))}$$

The first case above suggests that any system to which the standard translation lifts must be able to reason *underneath* $\exists$ *and* $\wedge$, so that the inference indicated in blue is 'accessible' to the prover. The second case above suggests that the existential-conjunctive meta-structure necessitated by the first case should admit basic equivalences, in particular certain *prenexing*. This section is devoted to the incorporation of these ideas (and necessities) into a bona fide proof system.

$$\mathsf{init}\ \frac{}{\{\ \}^{\varnothing}} \qquad \mathsf{wk}\ \frac{\mathbf{S}}{\mathbf{S}, \mathbf{S}'} \qquad \sigma\ \frac{\mathbf{S}}{\sigma(\mathbf{S})} \qquad \cup\ \frac{\mathbf{S}, \{\Gamma\}^{\mathbf{x}} \quad \mathbf{S}, \{\Delta\}^{\mathbf{y}}}{\mathbf{S}, \{\Gamma, \Delta\}^{\mathbf{x},\mathbf{y}}}\ \begin{array}{l}\mathsf{fv}(\Delta) \cap \mathbf{x} = \varnothing \\ \mathsf{fv}(\Gamma) \cap \mathbf{y} = \varnothing\end{array}$$

$$\mathsf{id}\ \frac{\mathbf{S}, \{\Gamma\}^{\mathbf{x}}}{\mathbf{S}, \{\Gamma, A\}^{\mathbf{x}}, \{\overline{A}\}^{\varnothing}}\ A \text{ closed} \qquad \wedge\ \frac{\mathbf{S}, \{\Gamma, A, B\}^{\mathbf{x}}}{\mathbf{S}, \{\Gamma, A \wedge B\}^{\mathbf{x}}} \qquad \vee\ \frac{\mathbf{S}, \{\Gamma, A_i\}^{\mathbf{x}}}{\mathbf{S}, \{\Gamma, A_0 \vee A_1\}^{\mathbf{x}}}\ i \in \{0, 1\}$$

$$\mathsf{inst}\ \frac{\mathbf{S}, \{\Gamma(t)\}^{\mathbf{x}}}{\mathbf{S}, \{\Gamma(y)\}^{\mathbf{x},y}} \qquad \exists\ \frac{\mathbf{S}, \{\Gamma, A(y)\}^{\mathbf{x},y}}{\mathbf{S}, \{\Gamma, \exists x(A(x))\}^{\mathbf{x}}}\ y \text{ fresh} \qquad \forall\ \frac{\mathbf{S}, \{\Gamma, A(f(\mathbf{x}))\}^{\mathbf{x}}}{\mathbf{S}, \{\Gamma, \forall x(A(x))\}^{\mathbf{x}}}\ f \text{ fresh}$$

$$TC\ \frac{\mathbf{S}, \{\Gamma, A(s,t)\}^{\mathbf{x}}, \{\Gamma, A(s,z), TC(A)(z,t)\}^{\mathbf{x},z}}{\mathbf{S}, \{\Gamma, TC(A)(s,t)\}^{\mathbf{x}}}\ z \text{ fresh}$$

$$\overline{TC}\ \frac{\mathbf{S}, \{\Gamma, A(s,t), A(s, f(\mathbf{x}))\}^{\mathbf{x}}, \{\Gamma, A(s,t), \overline{TC}(A)(f(\mathbf{x}), t)\}^{\mathbf{x}}}{\mathbf{S}, \{\Gamma, \overline{TC}(A)(s,t)\}^{\mathbf{x}}}\ f \text{ fresh}$$

Fig. 4 Hypersequent calculus HTC, where σ is a substitution map from constants to terms and a renaming of other function symbols and variables, extended to terms, formulas, cedents and hypersequents in the natural way. For Γ set of formulas, $\mathsf{fv}(\Gamma)$ denotes the set of free variables occurring in formulas in Γ. Colours define ancestry (see Remark 4.4)

4.1 Annotated Hypersequents

An *annotated cedent*, or simply *cedent*, written S, S' etc., is an expression $\{\Gamma\}^{\mathbf{x}}$, where Γ is a set of formulas and the *annotation* $\mathbf{x}$ is a set of variables. We sometimes construe annotations as lists rather than sets when it is convenient, e.g. when taking them as inputs to a function. Each cedent may be intuitively read as a TCL formula, under the following interpretation:

$$fm(\{\Gamma\}^{x_1,\dots,x_n}) \ := \ \exists x_1 \dots \exists x_n \bigwedge \Gamma.$$

When $\mathbf{x} = \varnothing$ then there are no existential quantifiers above, and when $\Gamma = \varnothing$ we simply identify $\bigwedge \Gamma$ with $\top$. We also sometimes write simply A for the annotated cedent $\{A\}^{\varnothing}$.

A *(annotated) hypersequent*, written $\boldsymbol{S}, \boldsymbol{S}'$ etc., is a set of annotated cedents. Each hypersequent may be intuitively read as the disjunction of its cedents. Namely we set: $fm(\{\Gamma_1\}^{\mathbf{x}_1}, \dots, \{\Gamma_n\}^{\mathbf{x}_n}) := fm(\{\Gamma_1\}^{\mathbf{x}_1}) \vee \dots \vee fm(\{\Gamma_n\}^{\mathbf{x}_n})$. With a slight abuse of notation, we sometimes identify $\boldsymbol{S}$ and $fm(\boldsymbol{S})$.

4.2 Non-wellfounded Hypersequent Proofs

We now present our hypersequential system for TCL and its corresponding notion of 'non-wellfounded proof'.

Definition 4.1 *(System)* The rules of HTC are given in Fig. 4 (the colours may be ignored for now). A HTC *preproof* is a possibly infinite tree of sequents generated by the rules of HTC. A preproof is *regular* if it has only finitely many distinct subproofs.

The substitution rule σ is needed to guarantee regularity of non-wellfounded branches. While we have included an explicit substitution rule we shall, as in earlier sections, often work 'modulo substitution' when writing down cyclic preproofs. Propositional rules, as well as init, are standard, recalling the formula interpretation of hypersequents defined in the previous section. The $\cup$ rule is the only branching rule of the system, while rule id allows us to eliminate (bottom-up) a closed formula A from one of the cedents (thus from a conjunction, wrt the formula interpretation) provided that the dual of A occurs in a singleton cedent with empty annotation. The usual sequent rule for the existential quantifier is factored into two HTC rules: $\exists$, which introduces a fresh variable in the annotation of a cedent, and inst, which instantiates a variable in the annotation with a term. Similarly the usual sequent rule for $\wedge$ is factored in HTC by the rules $\wedge$ and $\cup$. The rules for TC and $\overline{TC}$ are induced by the characterisation of TC as a least fixed point in (1). Note that the rules $\overline{TC}$ and $\forall$ introduce, bottom-up, the fresh function symbol f, which plays the role of the *Herbrand* function of the corresponding $\forall$ quantifier: just as $\forall \mathbf{x} \exists x A(x)$ is equisatisfiable with $\forall \mathbf{x} A(f(\mathbf{x}))$, when f is fresh, by Skolemisation, by duality $\exists \mathbf{x} \forall x A(x)$ is equivalid with $\exists \mathbf{x} A(f(\mathbf{x}))$, when f is fresh, by Herbrandisation. Note that the usual $\forall$ rule of the sequent calculus is just a special case of this, when $\mathbf{x} = \varnothing$, and so f is a constant symbol.

Our notion of ancestry, as compared to traditional sequent systems, must account for the richer structure of hypersequents. Specifically, since formulas now occur within cedents, tracing ancestry only for formulas no longer suffices. Instead, we define a notion of ancestry for cedents, and then trace formulas *within* cedent-paths. In line with the formula interpretation, our notion of 'progress' needs to take into account *all* infinite traces occurring within such cedent-paths.

Definition 4.2 *(Ancestry for cedents)* Fix an inference step r, as typeset in Fig. 4. We say that a cedent S in a premiss of r is an *immediate ancestor* of a cedent S' in the conclusion of r if either:

1. $\mathsf{r} \neq \sigma$ and $S = S' \in \boldsymbol{S}$, i.e. S and S' are identical 'side' cedents of r; or,
2. $\mathsf{r} = \sigma$ and $S' = \sigma(S)$.
3. $\mathsf{r} \neq \mathsf{id}$, $\mathsf{r} \neq \sigma$, and S' is the (unique) cedent distinguished in the conclusion of r, and S is a cedent indicated in a premiss of r; or,
4. $\mathsf{r} = \mathsf{id}$ and S is the (unique) cedent distinguished in the premiss of id and S' is the cedent $\{\Gamma, A\}^{\mathbf{x}}$ dsitinguished in the conclusion of id.

Note in particular that in id, as typeset in Fig. 4, $\{\Gamma\}^{\mathbf{x}}$ is *not* an immediate ancestor of $\{\bar{A}\}^{\varnothing}$.

Definition 4.3 *(Ancestry for formulas)* Fix an inference step r, as typeset in Fig. 4. We say that a formula F in a premiss of r is an *immediate ancestor* of a formula F' in the conclusion of r if either:

(a) $\mathsf{r} \neq \sigma$ and $F = F'$ occur in some cedent $S \in \boldsymbol{S}$; or,
(b) $\mathsf{r} = \sigma$ and $F' = \sigma(F)$ occurs in some $S' = \sigma(S)$ where F occurs in $S \in \boldsymbol{S}$; or,
(c) $\mathsf{r} = \cup$ and $F = F' \in \Gamma$ or $F = F' \in \Delta$; or,
(d) F is one of the formulas explicitly distinguished in the premiss of r and F' is the (unique) formula explicitly distinguished in the conclusion of r.

Remark 4.4 (Ancestry via colours) Again we may understand cedent ancestry and formula ancestry by the colouring in Fig. 4. A formula C in the premiss is an *immediate ancestor* of a formula C' in the conclusion if they have the same colour; if $C, C' \in \Gamma$ then we further require $C = C'$, and if C, C' occur in $\boldsymbol{S}$ then $C = C'$ occur in the same cedent. A cedent S in the premiss is an immediate ancestor of a cedent S' in the conclusion if some formula in S is an immediate ancestor of some formula in S'.

Immediate ancestry on both formulas and cedents is a binary relation, inducing a directed graph whose paths form the basis of our correctness condition:

Definition 4.5 *((Hyper)traces)* A *hypertrace* is a maximal path in the graph of immediate ancestry on cedents. A *trace* is a maximal path in the graph of immediate ancestry on formulas.

Thus, in the id rule, as typeset in Fig. 4, no (infinite) trace can include the distinguished A or $\bar{A}$. From the above definitions it follows that whenever a cedent S in the premiss of a rule r is an immediate ancestor of a cedent S' in the conclusion, then some formula in S is an immediate ancestor of some formula in S'. Thus, for a hypertrace $(S_i)_{i<\omega}$, there is at least one trace $(F_i)_{i<\omega}$ which lies 'within' or 'along' the hypertrace, i.e., such that $F_i \in S_i$ for all i.

Definition 4.6 *(Progress and proofs)* Fix a preproof $\mathcal{D}$. A (infinite) trace $(F_i)_{i\in\omega}$ is *progressing* if there is k such that, for all $i > k$, F_i has the form $\overline{TC}(A)(s_i, t_i)$ and is infinitely often principal.[1] A (infinite) hypertrace $\mathcal{H}$ is *progressing* if every infinite trace along it is progressing. A (infinite) branch is progressing if it has a progressing hypertrace. $\mathcal{D}$ is a *proof* if every infinite branch is progressing. If, furthermore, $\mathcal{D}$ is regular, we call it a *cyclic proof*.

We write HTC $\vdash_{nwf} \boldsymbol{S}$ (or HTC $\vdash_{cyc} \boldsymbol{S}$) if there is a proof (or cyclic proof, respectively) in HTC of the hypersequent $\boldsymbol{S}$.

4.3 Some Examples

Let us consider some examples of cyclic proofs in HTC and compare the system to TC_G. As mentioned in Sect. 4.2, for convenience we here write cyclic (pre)proofs modulo substitution.

Example 4.7 (Fixed point identity) Here is a cyclic proof in TC_G that reduces the identity $\overline{TC}(a)(c,d), TC(\bar{a})(c,d)$ to simpler identities,

$$\overline{TC}\,\dfrac{TC\,\dfrac{\mathsf{id}\,\dfrac{}{a(c,d), \bar{a}(c,d)}}{a(c,d), TC(\bar{a})(c,d)} \qquad TC\,\dfrac{\mathsf{id}\,\dfrac{}{a(c,e), \bar{a}(c,e)} \qquad \overline{TC}\,\dfrac{\vdots}{\overline{TC}(a)(e,d), TC(\bar{a})(e,d)}\,\bullet}{a(c,e), \overline{TC}(a)(e,d), TC(\bar{a})(c,d)}}{\overline{TC}(a)(c,d), TC(\bar{a})(c,d)}\,\bullet$$

where we have indicated roots of identical subproofs with •, and an infinite progressing trace along the (unique) infinite branch in blue.

There is not much choice in the construction of this cyclic proof, bottom-up: we must apply $\overline{TC}$ first and branch before applying TC differently on each branch. This cyclic proof is naturally simulated by the following HTC one, where the progressing hypertrace (along the unique infinite branch) is marked in blue:

$$\overline{TC}\,\dfrac{2\cup\,\dfrac{TC\,\dfrac{\mathsf{id}\,\dfrac{\mathsf{init}\,\dfrac{}{\{\,\}^{\varnothing}}}{\{a(c,d)\}^{\varnothing}, \{\bar{a}(c,d)\}^{\varnothing}}}{\{a(c,d)\}^{\varnothing}, \{TC(\bar{a})(c,d)\}^{\varnothing}} \qquad TC\,\dfrac{\mathsf{inst}\,\dfrac{\cup\,\dfrac{\mathsf{id}\,\dfrac{\mathsf{init}\,\dfrac{}{\{\,\}^{\varnothing}}}{\{a(c,e)\}^{\varnothing}, \{\bar{a}(c,e)\}^{\varnothing}} \qquad \overline{TC}\,\dfrac{\vdots}{\{\overline{TC}(a)(e,d)\}^{\varnothing}, \{TC(\bar{a})(e,d)\}^{\varnothing}}\,\bullet}{\{a(c,e)\}^{\varnothing}, \{\overline{TC}(a)(e,t)\}^{\varnothing}, \{\bar{a}(c,e), TC(\bar{a})(e,d)\}^{\varnothing}}}{\{a(c,e)\}^{\varnothing}, \{\overline{TC}(a)(e,t)\}^{\varnothing}, \{\bar{a}(c,x), TC(\bar{a})(x,d)\}^{x}}}{\{a(c,e)\}^{\varnothing}, \{\overline{TC}(a)(e,t)\}^{\varnothing}, \{TC(\bar{a})(c,d)\}^{\varnothing}}}{\{a(c,d), a(c,e)\}^{\varnothing}, \{a(c,d), \overline{TC}(a)(e,d)\}^{\varnothing}, \{TC(\bar{a})(c,d)\}^{\varnothing}}}{\{\overline{TC}(a)(c,d)\}^{\varnothing}, \{TC(\bar{a})(c,d)\}^{\varnothing}}\,\bullet$$

---

[1] In fact, by a a simple well-foundedness argument, it is equivalent to say that $(F_i)_{i<\omega}$ is progressing if it is infinitely often principal for a $\overline{TC}$-formula.

Due to the granularity of the inference rules of HTC, we actually have some liberty in how we implement such a derivation. Example, the HTC-proof below applies TC rules below $\overline{TC}$ ones, and delays branching until the 'end' of proof search, which is impossible in TC_G. The only infinite branch, looping on $\bullet$, is progressing by the blue hypertrace.

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{init}
\UnaryInfC{$\{\,\}^{\varnothing}$}
\LeftLabel{id}
\UnaryInfC{$\{a(c,d)\}^{\varnothing}, \{\bar{a}(c,d)\}^{\varnothing}$}
\AxiomC{}
\LeftLabel{init}
\UnaryInfC{$\{\,\}^{\varnothing}$}
\LeftLabel{id}
\UnaryInfC{$\{a(c,e)\}^{\varnothing}, \{\bar{a}(c,e)\}^{\varnothing}$}
\AxiomC{$\vdots$}
\LeftLabel{TC}
\RightLabel{$\bullet$}
\UnaryInfC{${\color{blue}\{\overline{TC}(a)(e,d)\}^{\varnothing}}, \{TC(\bar{a})(e,d)\}^{\varnothing}$}
\LeftLabel{$\cup$}
\BinaryInfC{$\{a(c,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(a)(e,d)\}^{\varnothing}}, \{\bar{a}(c,e), TC(\bar{a})(e,d)\}^{\varnothing}$}
\LeftLabel{$2\cup$}
\BinaryInfC{$\{a(c,d), a(c,e)\}^{\varnothing}, {\color{blue}\{a(c,d), \overline{TC}(a)(e,d)\}^{\varnothing}}, \{\bar{a}(c,d)\}^{\varnothing}, \{\bar{a}(c,e), TC(\bar{a})(e,d)\}^{\varnothing}$}
\LeftLabel{inst}
\UnaryInfC{$\{a(c,d), a(c,e)\}^{\varnothing}, {\color{blue}\{a(c,d), \overline{TC}(a)(e,d)\}^{\varnothing}}, \{\bar{a}(c,d)\}^{\varnothing}, \{\bar{a}(c,x), TC(\bar{a})(x,d)\}^{x}$}
\LeftLabel{$\overline{TC}$}
\UnaryInfC{${\color{blue}\{\overline{TC}(a)(c,d)\}^{\varnothing}}, \{\bar{a}(c,d)\}^{\varnothing}, \{\bar{a}(c,x), TC(\bar{a})(x,d)\}^{x}$}
\LeftLabel{TC}
\RightLabel{$\bullet$}
\UnaryInfC{${\color{blue}\{\overline{TC}(a)(c,d)\}^{\varnothing}}, \{TC(\bar{a})(c,d)\}^{\varnothing}$}
\end{prooftree}
$$

This is an example of the more general 'rule permutations' available in HTC, hinting at a more flexible proof theory (we discuss this further in Sect. 8).

Let us now consider a more complex example whose relevance will become significant shortly:

Example 4.8 We give a cyclic HTC proof $\mathcal{D}$ of the following hypersequent:

$$\{\overline{TC}(\overline{\alpha})(c,d)\}^{\varnothing}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\beta)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\beta)(y,d)\}^{y},$$

where $\alpha(c,d) = \mathsf{ST}(aa \cup aba)(c,d)$ and $\gamma(c,d) = \mathsf{ST}(ba^{+})(c,d)$.

$$
\begin{prooftree}
\AxiomC{$\mathbf{Q}_1$}
\AxiomC{$\mathbf{Q}_2$}
\AxiomC{$\mathbf{Q}_3$}
\AxiomC{$\vdots$}
\RightLabel{$\circ$}
\UnaryInfC{${\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(e,d)\}^{\varnothing}, \{TC(\gamma)(e,d)\}^{\varnothing}, \{TC(a)(e,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{inst,id}
\UnaryInfC{$\{\overline{a}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(e_1,z_2), TC(a)(z_2,d)\}^{z_2}, \{a(e_1,y), TC(\gamma)(y,d)\}^{y}, \{a(e_1,z_3), TC(a)(z_3,y), TC(\gamma)(y,d)\}^{y,z_3}$}
\LeftLabel{TC}
\UnaryInfC{$\{\overline{a}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(e_1,z_2), TC(a)(z_2,d)\}^{z_2}, \{TC(a)(e_1,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{TC,wk}
\UnaryInfC{$\{\overline{a}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(e_1,d)\}^{\varnothing}, \{TC(a)(e_1,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{inst,id}
\UnaryInfC{$\{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{a}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(c,z), TC(a)(z,d)\}^{z}, \{a(c,z_1), TC(a)(z_1,y), TC(\gamma)(y,d)\}^{y,z_1}$}
\LeftLabel{TC,wk}
\UnaryInfC{$\{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{a}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(c,z), TC(a)(z,d)\}^{z}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{TC,wk}
\UnaryInfC{$\{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{a}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{$\forall,\vee$}
\dottedLine
\UnaryInfC{$\{\overline{aa}(c,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{wk}
\UnaryInfC{$\{\overline{aa}(c,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{$\cup$}
\BinaryInfC{$\{\overline{aa}(c,e), \overline{aba}(c,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{$\wedge$}
\dottedLine
\UnaryInfC{$\{\overline{\alpha}(c,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{$\cup$}
\BinaryInfC{$\{\overline{\alpha}(c,e)\}^{\varnothing}, {\color{red}\{\overline{\alpha}(c,d), \overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{$\cup$}
\BinaryInfC{$\{\overline{\alpha}(c,d), \overline{\alpha}(c,e)\}^{\varnothing}, {\color{red}\{\overline{\alpha}(c,d), \overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\LeftLabel{$\overline{TC}$}
\RightLabel{$\circ$}
\UnaryInfC{${\color{red}\{\overline{TC}(\overline{\alpha})(c,d)\}^{\varnothing}}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}$}
\end{prooftree}
$$

where:

$$
\begin{aligned}
\mathbf{Q}_1 &= \{\overline{\alpha}(c,d)\}^{\varnothing}, \{\overline{\alpha}(c,d), \overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,d)\}^{\varnothing},\\
&\quad \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}\\
\mathbf{Q}_2 &= \{\overline{\alpha}(c,e)\}^{\varnothing}, \{\overline{\alpha}(c,e)\}^{\varnothing}, \{TC(a)(c,d)\}^{\varnothing}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}\\
\mathbf{Q}_3 &= \{\overline{aba}(c,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,d)\}^{\varnothing},\\
&\quad \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}.
\end{aligned}
$$

We do not show the finite derivations of hypersequents $\mathbf{Q}_1$ and $\mathbf{Q}_2$, but here is the subproof of $\mathbf{Q}_3$:

$$
\begin{array}{rc}
 & \vdots \\ \hline
 & {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(e,d)\}^{\varnothing}\{TC(\gamma)(e,d)\}^{\varnothing}, \{TC(a)(e,z), TC(\gamma)(z,d)\}^{z} \;\circ \\ \hline
\text{inst,id} & \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(e_2,w)TC(a)(w,d)\}^{w}\{a(e_2,z), TC(\gamma)(z,d)\}^{z}, \{a(e_2,k), TC(a)(k,z), TC(\gamma)(z,d)\}^{z,k} \\ \hline
TC & \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(e_2,w)TC(a)(w,d)\}^{w}\{TC(a)(e_2,z), TC(\gamma)(z,d)\}^{z} \\ \hline
TC,\text{wk} & \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(e_2,d)\}^{\varnothing}\{TC(a)(e_2,z), TC(\gamma)(z,d)\}^{z} \\ \hline
\text{inst,id} & \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{b(y,y_1), TC(a)(y_1,d)\}^{y,y_1}\{b(y,z_1), TC(a)(z_1,z), TC(\gamma)(z,d)\}^{y,z,z_1} \\ \hdashline
\exists,\wedge & \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{\mathsf{ST}(ba^{+})(y,d)\}^{y}\{\mathsf{ST}(ba^{+})(y,z), TC(\gamma)(z,d)\}^{y,z} \\ \hline
TC & \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(\gamma)(y,d)\}^{y} \\ \hline
\text{inst,id} & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(c,y), TC(\gamma)(y,d)\}^{y} \\ \hline
TC,\text{wk} & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hline
 & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{ba}(e_2,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hdashline
\forall,\vee & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{ba}(e_1,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hdashline
\forall,\vee & \{\overline{aba}(c,d)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hline
\text{wk} & \{\overline{aba}(c,e)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hline
 & \{\overline{aba}(c,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, {\color{red}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(\gamma)(c,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}
\end{array}
$$

Note the multiple occurrences of the ‘backpointer’ ∘ (we have omitted explicit substitution steps here), resulting in uncountably many infinite branches. Specifically, the preproof contains two non-wellfounded branches: $\mathbf{Q}_3$ and the branch displayed in $\mathcal{D}$. Both branches are regular, as they are identical (modulo substitution) to the sequent generated in the last-but-one bottom sequent in $\mathcal{D}$. Since $\mathbf{Q}_3$ is generated by the branch displayed in $\mathcal{D}$, and this latter is repeated infinitely often, the preproof contains uncountably many infinite branches. In all cases, the cedents marked in red induce progressing hypertraces along any infinite branch.

Finally, it is pertinent to revisit the ‘counterexample’ (2) from Sect. 3.3 that witnessed incompleteness of TC_G for PDL^+. The following result is, in fact, already implied by our later completeness result, Theorem 6.1, but it is useful to give it explicitly nonetheless:

Proposition 4.9 HTC $\vdash_{cyc} \mathsf{ST}((aa \cup aba)^+)(c,d) \to \mathsf{ST}(a^+((ba^+)^+ \cup a))(c,d)$.

Proof Using the abbreviations $\alpha(c,d) = \mathsf{ST}(aa \cup aba)(c,d)$ and $\beta(c,d) = \mathsf{ST}((ba^+)^+ \cup a)(c,d)$, we give the following cyclic proof, where • marks roots of identical subproofs (we omit explicit substitution steps),

$$
\begin{array}{rc}
 & \vdots \\ \hline
 & {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(j,e), \beta(yd)\}^{y} \;\bullet \\ \hline
\text{id} & \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(f,g), TC(a)(g,y), \beta(y,d)\}^{y} \\ \hline
\text{inst} & \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(f,k), TC(a)(k,y), \beta(y,d)\}^{y,k} \;[e/k] \\ \hline
\text{wk} & \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,\ d)\}^{\varnothing}}, \{a(f,y), \beta(y,d)\}^{y}, \{a(f,k), TC(a)(k,y), \beta(y,d)\}^{y,k} \\ \hline
TC & \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(f,y), \beta(y,d)\}^{y} \\ \hline
\text{id} & \{\overline{a}(c,f)\}^{\varnothing}, \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(c,f), TC(a)(f,y), \beta(y,d)\}^{y} \\ \hline
\text{inst} & \{\overline{a}(c,f)\}^{\varnothing}, \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(c,z), TC(a)(z,y), \beta(y,d)\}^{y,z} \;[f/z] \\ \hline
\text{wk} & \{\overline{a}(c,f)\}^{\varnothing}, \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{a(c,y), \beta(y,d)\}^{y}, \{a(c,z), TC(a)(z,y), \beta(y,d)\}^{y,z,} \\ \hline
TC & \{\overline{a}(c,f)\}^{\varnothing}, \{\overline{a}(f,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\ \hdashline
\forall,\vee & \mathbf{P} \qquad \{\overline{aa}(c,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\ \hline
\cup & \{\overline{aa}(c,e), \overline{aba}(c,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\ \hdashline
\wedge & \mathbf{R'} \qquad \{\overline{\alpha}(c,e)\}^{\varnothing}, {\color{blue}\{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\ \hline
\cup & \mathbf{R} \qquad \{\overline{\alpha}(c,e)\}^{\varnothing}, {\color{blue}\{\overline{\alpha}(c,d), \overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(,y), \beta(y,d)\}^{y} \\ \hline
\cup & \{\overline{\alpha}(c,d), \overline{\alpha}(c,e)\}^{\varnothing}, {\color{blue}\{\overline{\alpha}(c,d), \overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\ \hline
\overline{TC} & {\color{blue}\{\overline{TC}(\overline{\alpha})(c,d)\}^{\varnothing}}, \{TC(a)(c,y), \beta(y,d)\}^{y} \;\bullet \\ \hdashline
\vee,\exists,\wedge & {\color{blue}\{\overline{\mathsf{ST}((aa \cup aba)^{+})(c,d)} \vee \mathsf{ST}(a^{+};((ba^{+})^{+} \cup a))(c,d)\}^{\varnothing}}
\end{array}
$$

where the only infinite branch displayed has progressing hypertrace indicated in blue. **R**, **R**$'$ and **P** above are the following hypersequents:

$$\begin{aligned}
\mathbf{R} &= \{\overline{\alpha}(c,d)\}^{\varnothing}, \{\overline{\alpha}(c,d), \overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\
\mathbf{R}' &= \{\overline{\alpha}(c,d)\}^{\varnothing}, \{\overline{\alpha}(c,d)\}^{\varnothing}, \{TC(a)(c,y), \beta(y,d)\}^{y} \\
\mathbf{P} &= \{\overline{aba}(c,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), \beta(y,d)\}^{y}.
\end{aligned}$$

R, **R**$'$ have finitary proofs, while **P** has a cyclic proof, shown below, which makes use of Example 4.8 above. We use the following abbreviations:

$$\begin{aligned}
\alpha(c,d) &= \mathsf{ST}(aa \cup aba)(c,d) \\
\beta(c,d) &= \mathsf{ST}((ba^{+})^{+} \cup a)(c,d) \\
\gamma(c,d) &= \mathsf{ST}(ba^{+})(c,d)
\end{aligned}$$

$$\begin{array}{rc}
 & \text{Ex. } 4.8 \\ \hline\hline
 & \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(e,d)\}^{\varnothing}, \{TC(\gamma)(e,d)\}^{\varnothing}, \{TC(a)(e,z), TC(\gamma)(z,d)\}^{z} \\ \hline
\text{inst,id} & \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{a(e_2,j), TC(a)(j,d)\}^{j}, \{a(e_2,z), TC(\gamma)(z,d)\}^{z}, \{a(e_2,w), TC(a)(w,z), TC(\gamma)(z,d)\}^{z,w} \\ \hline
TC,wk & \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(e_2,d)\}^{\varnothing}, \{a(e_2,z), TC(\gamma)(z,d)\}^{z}, \{a(e_2,w), TC(a)(w,z), TC(\gamma)(z,d)\}^{z,w} \\ \hline
TC & \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(e_2,d)\}^{\varnothing}, \{TC(a)(e_2,z), TC(\gamma)(z,d)\}^{z} \\ \hline
\text{inst,id} & \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{b(e_1,k), TC(a)(k,d)\}^{k}, \{b(e_1,z_1), TC(a)(z_1,z), TC(\gamma)(z,d)\}^{z,z_1} \\ \hdashline
\exists,\wedge & \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{\mathsf{ST}(ba^{+})(e_1,d)\}^{\varnothing}, \{\mathsf{ST}(ba^{+})(e_1,z), TC(\gamma)(z,d)\}^{z} \\ \hline
TC & \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(\gamma)(e_1,d)\}^{\varnothing} \\ \hline
\text{inst,id} & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{a(c,y), TC(\gamma)(y,d)\}^{y} \\ \hline
TC,\text{wk} & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{b}(e_1,e_2)\}^{\varnothing}, \{\overline{a}(e_2,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hdashline
\forall,\vee & \{\overline{a}(c,e_1)\}^{\varnothing}, \{\overline{ba}(e_1,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hdashline
\forall,\vee & \{\overline{aba}(c,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y} \\ \hline
\text{wk} & \{\overline{aba}(c,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), TC(\gamma)(y,d)\}^{y}, \{TC(a)(c,y), a(y,d)\}^{y} \\ \hdashline
\vee & \{\overline{aba}(c,e)\}^{\varnothing}, \{\overline{TC}(\overline{\alpha})(e,d)\}^{\varnothing}, \{TC(a)(c,y), \beta(y,d)\}^{y}
\end{array}$$

4.4 On Cyclic-Proof Checking

In usual cyclic systems, checking that a regular preproof is progressing is decidable by straightforward reduction to the universality of nondeterministic ω-word-automata, with runs ‘guessing’ a progressing thread along an infinite branch. Our notion of progress exhibits an extra quantifier alternation: we must *guess* an infinite hypertrace in which *every* trace is progressing. Nonetheless, by appealing to determinisation or alternation, we can still decide our progressing condition:

Proposition 4.10 *Checking whether a cyclic* HTC *preproof is a proof is decidable.*

Proof Sketch The result is proved using using automata-theoretic techniques. Fix a cyclic HTC preproof $\mathcal{D}$. First, using standard methods from cyclic proof theory, it is routine to construct a nondeterministic Büchi automaton recognising non-progressing hypertraces of $\mathcal{D}$. The construction is similar to that recognising progressing branches in cyclic sequent calculi, e.g. as found in [11, 14, 26], since we are asking that there *exists* a non-progressing trace within a hypertrace. By Büchi’s complementation theorem and McNaughton’s determinisation theorem (see, e.g., [30] for details), we can thus construct a deterministic parity automaton $\mathcal{P}_H$ recognising progressing hypertraces.[2]

Now we can construct a nondeterministic parity automaton $\mathcal{P}$ recognising progressing branches of $\mathcal{D}$ similarly to the previous construction, but further keeping track of states in $\mathcal{P}_H$:

- $\mathcal{P}$ essentially guesses a progressing hypertrace along the branch input;
- at the same time, $\mathcal{P}$ runs the hypertrace-in-construction along $\mathcal{P}_H$ and keeps track of the state therein;
- acceptance for $\mathcal{P}$ is inherited directly from $\mathcal{P}_H$, i.e. a run is accepting just if the hypertrace guessed along it is accepted by $\mathcal{P}_H$.

Now it is clear that $\mathcal{P}$ accepts a branch of $\mathcal{D}$ if and only if it is progressing. Assuming that $\mathcal{P}$ also accepts any ω-words over the underlying alphabet that are not branches of $\mathcal{D}$ (by adding junk states), we have that $\mathcal{D}$ is a proof (i.e. each of its infinite branches is progressing) if and only if $\mathcal{P}$ is universal. For additional material and results on infinite word automata refer to [4, 30].

4.5 Simulating Cohen–Rowe

As we mentioned earlier, cyclic proofs of HTC indeed are at least as expressive as those of Cohen and Rowe's system by a routine local simulation of rules:

Theorem 4.11 *If* $\mathsf{TC}_G \vdash_{cyc} A$ *then* $\mathsf{HTC} \vdash_{cyc} A$.

Proof Sketch Let $\mathcal{D}$ be a TC_G cyclic proof. We can convert it to a HTC cyclic proof by simply replacing each sequent $A_1, \ldots, A_n$ by the hypersequent $\{A_1\}^{\varnothing}, \ldots, \{A_n\}^{\varnothing}$ and applying some local corrections. In what follows, if $\Gamma = A_1, \ldots, A_n$, let us simply write $\boldsymbol{S}_\Gamma$ for $\{A_1\}^{\varnothing}, \ldots, \{A_n\}^{\varnothing}$.

- Any id step of $\mathcal{D}$ must be amended as follows:

$$\mathsf{id}\,\frac{}{\Gamma, p(t), \bar{p}(t)} \quad \rightsquigarrow \quad \mathsf{wk}\,\dfrac{\mathsf{id}\,\dfrac{\mathsf{init}\,\dfrac{}{\{\}^{\varnothing}}}{\{p(t)\}^{\varnothing}, \{\bar{p}(t)\}^{\varnothing}}}{\boldsymbol{S}_\Gamma, \{p(t)\}^{\varnothing}, \{\bar{p}(t)\}^{\varnothing}}$$

- Any $\vee$ step of $\mathcal{D}$ becomes a correct $\vee$ step of HTC or $\mathsf{HTC}^{=}$.
- Any $\wedge$ step of $\mathcal{D}$ must be amended as follows:

$$\wedge\,\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B} \quad \rightsquigarrow \quad \wedge\,\dfrac{\cup\,\dfrac{\boldsymbol{S}_\Gamma, \{A\}^{\varnothing} \quad \boldsymbol{S}_\Gamma, \{B\}^{\varnothing}}{\boldsymbol{S}_\Gamma, \{A, B\}^{\varnothing}}}{\boldsymbol{S}_\Gamma, \{A \wedge B\}^{\varnothing}}$$

- Any $\exists$ step of $\mathcal{D}$ must be amended as follows:

$$\exists\,\frac{\Gamma, A(t)}{\Gamma, \exists x A(x)} \quad \rightsquigarrow \quad \exists\,\dfrac{\mathsf{inst}\,\dfrac{\boldsymbol{S}_\Gamma, \{A(t)\}^{\varnothing}}{\boldsymbol{S}_\Gamma, \{A(x)\}^{x}}}{\boldsymbol{S}_\Gamma, \{\exists x A(x)\}^{\varnothing}}$$

---

[2] This is perhaps an overkill step, but allows us to easily deal with the issue of alternation in the progressing condition.

- Any $\forall$ step of $\mathcal{D}$ becomes a correct $\forall$ step of HTC or HTC$^{=}$.
- Any TC_0 step of $\mathcal{D}$ becomes a correct TC_0 step of HTC.
- Any TC_1 step of $\mathcal{D}$ must be amended as follows:

$$\dfrac{\Gamma, A(s,r) \quad \Gamma, TC(A)(r,t)}{\Gamma, TC(A)(s,t)}\,TC_1 \quad \rightsquigarrow \quad TC_1\,\dfrac{\text{inst}\,\dfrac{\cup\,\dfrac{\boldsymbol{S}_\Gamma, \{A(s,r)\}^{\varnothing} \quad \boldsymbol{S}_\Gamma, \{TC(A)(r,t)\}^{\varnothing}}{\boldsymbol{S}_\Gamma, \{A(s,r), TC(A)(r,t)\}^{\varnothing}}}{\boldsymbol{S}_\Gamma, \{A(s,x), TC(A)(x,t)\}^{x}}}{\boldsymbol{S}_\Gamma, \{TC(A)(s,t)\}^{\varnothing}}$$

- Any $\overline{TC}$ step of $\mathcal{D}$ must be amended as follows:

$$\overline{TC}\,\dfrac{\Gamma, A(s,t) \quad \Gamma, A(s,c), \overline{TC}(A)(c,t)}{\Gamma, \overline{TC}(A)(s,t)}$$

$$\rightsquigarrow \quad \overline{TC}\,\dfrac{2\cup\,\dfrac{\boldsymbol{S}_\Gamma, \{A(s,t)\}^{\varnothing} \quad \boldsymbol{S}_\Gamma, \{A(s,c)\}^{\varnothing}, \{\overline{TC}(A)(c,t)\}^{\varnothing}}{\boldsymbol{S}_\Gamma, \{A(s,t), A(s,c)\}^{\varnothing}, \{A(s,t), \overline{TC}(A)(c,t)\}^{\varnothing}}}{\boldsymbol{S}_\Gamma, \{\overline{TC}(A)(s,t)\}^{\varnothing}}$$

Particular inspection of the $\overline{TC}$ case shows that progressing traces of TC$_G$ induce progressing hypertraces of HTC. Also, since each of the cases above maps an inference step of TC$_G$ to a fixed finite gadget in HTC, regularity is preserved too.

5 Soundness of HTC

This section is devoted to the proof of the first of our main results:

Theorem 5.1 *(Soundness) If* HTC $\vdash_{nwf}$ $\boldsymbol{S}$ *then* $\models \boldsymbol{S}$.

The argument is quite technical due to the alternating nature of our progress condition. In particular the treatment of traces within hypertraces requires a more fine grained argument than usual, bespoke to our hypersequential structure.

5.1 Some Conventions on (Pre)proofs and Semantics

First, we work with proofs without substitution, in order to control the various symbols occurring in a proof.

Throughout this section, we shall fix a HTC preproof D of a hypersequent $\boldsymbol{S}$. We start by introducing some additional definitions and propositions.

Proposition 5.2 *If* HTC $\vdash_{nwf}$ $\boldsymbol{S}$ *then there is also a* HTC *proof of* $\boldsymbol{S}$ *that does not use the substitution rule.*

Proof Sketch We appeal to a coinductive argument, applying a meta-level substitution operation on proofs to admit each substitution step. Productivity of the translation is guaranteed by the progressing condition: each infinite branch must, at the very least, have infinitely many $\overline{TC}$ steps.

The utility of this is that we can now carefully control the occurrences of eigenfunctions in a proof so that, bottom-up, they are never ‘re-introduced’, thus facilitating the definition of interpretations on them.

Throughout this section, we shall allow interpretations to be only partially defined, i.e. they are now *partial* maps from the set of function symbols of our language to appropriately typed functions in the structure at hand. Typically our interpretations will indeed interpret the function symbols in the context in which they appear, but as we consider further function symbols it will be convenient to extend an interpretation 'on the fly'. This idea is formalised in the following definition:

Definition 5.3 *(Interpretation extension)* Let $\mathcal{M}$ be a structure and ρ, ρ' be two (partial) interpretations over $|\mathcal{M}|$. We say that ρ' is an *extension* of ρ, written $\rho \subseteq \rho'$, if $\rho'(f) = \rho(f)$, for all f in the domain of ρ.

Finally, we assume that the free and bound variables occurring in a hypersequent at the root a (pre)proof are all pairwise distinct, and that whenever we apply rule $\exists$ or rule TC (resp. rule $\forall$ or rule $\overline{TC}$) to a hypersequent $\boldsymbol{S}$ occurring in a branch, the rule introduces in the premiss a variable (resp. a function symbol) that does not appear in any hypersequent in the branch from the root up to $\boldsymbol{S}$, included. This strong freshness requirement guarantees that each function and variable symbol is uniquely interpreted in the countermodel that we are going to construct.

5.2 Constructing a 'Countermodel' Branch

Recall that we have fixed at the beginning of this section a HTC preproof $\mathcal{D}$ of a hypersequent $\boldsymbol{S}$. Let us fix some structure $\mathcal{M}^\times$ and an interpretation ρ_0 such that $\rho_0 \not\models \boldsymbol{S}$ (within $\mathcal{M}^\times$). As we shall prove in the following Lemma, since each rule is locally sound, by contraposition we can continually choose 'false premisses' to construct an infinite 'false branch':

Lemma 5.4 (Countermodel branch) *There is a branch $\mathcal{B}^\times = (\boldsymbol{S}_i)_{i<\omega}$ of $\mathcal{D}$ and an interpretation $\rho^\times$ such that, with respect to $\mathcal{M}^\times$:*

1. *$\rho^\times \not\models \boldsymbol{S}_i$, for all $i < \omega$;*
2. *Suppose that $\boldsymbol{S}_i$ concludes a $\overline{TC}$ step, as typeset in Fig. 4, and $\rho^\times \models TC(\bar{A})(s,t)\,[\boldsymbol{d}/\boldsymbol{x}]$. If n is minimal such that $\rho^\times \models \bar{A}(d_i, d_{i+1})$ for all $i < n$, $\rho^\times(s) = d_0$ and $\rho^\times(t) = d_n$, and $n > 1$, then $\rho^\times(f)(\boldsymbol{d}) = d_1$,[3] so that $\rho^\times \models \bar{A}(s, f(\boldsymbol{x}))[\boldsymbol{d}/\boldsymbol{x}]$, and $\rho^\times \models TC(\bar{A})(f(\boldsymbol{x}), t)[\boldsymbol{d}/\boldsymbol{x}]$.*

Intuitively, our interpretation $\rho^\times$ is going to be defined as the limit of a chain of 'partial' interpretations $(\rho_i)_{i<\omega}$, with each $\rho_i \not\models \boldsymbol{S}_i$ (with respect to $\mathcal{M}^\times$). Referring to Item 2, whenever some $\overline{TC}$-formula is principal, we shall always choose ρ_{i+1} to assign to it a falsifying path of minimal length (if one exists at all), with respect to an assignment $\mathbf{d}$ to variables $\mathbf{x}$ in the annotation of its cedent. It is crucial at this point that our definition of $\rho^\times$ is parametrised by such assignments.

Proof of Lemma 5.4 We construct $\mathcal{B}^\times$ and $\rho^\times$ simultaneously. In fact we shall define a chain of interpretations $\rho_0 \subseteq \rho_1 \subseteq \rho_2 \subseteq \cdots$ such that, for each i, $\mathcal{M}^\times, \rho_i \models \boldsymbol{S}_i$. We will define $\rho^\times$ as the limit of this chain. We distinguish cases according to the rule r_i that $\boldsymbol{S}_i$ concludes. For the case of weakening, $\boldsymbol{S}_{i+1}$ is the unique premiss of the rule, and $\rho_{i+1} = \rho_i$. We now give all the other cases:

$\triangleright$ **Case** ($\cup$)

$$\cup\,\frac{\boldsymbol{S}^1 = \mathbf{Q}, \{\Gamma_1\}^{\mathbf{x}_1} \qquad \boldsymbol{S}^2 = \mathbf{Q}, \{\Gamma_2\}^{\mathbf{x}_2}}{\boldsymbol{S}_i = \mathbf{Q}, \{\Gamma_1, \Gamma_2\}^{\mathbf{x}_1, \mathbf{x}_2}}$$

---

[3] To be clear, we here choose an arbitrary such minimal '$\bar{A}$-path' to set d_1.

By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x}_1 \forall \mathbf{x}_2 (\bigvee \overline{\Gamma}_1 \vee \bigvee \overline{\Gamma}_2)$. Set $\rho_{i+1} = \rho_i$. By the truth condition for $\forall$, we have that for all m-tuples $\mathbf{d}_1 \in |\mathcal{M}^{\times}|$ and n-tuples $\mathbf{d}_2 \in |\mathcal{M}^{\times}|$, for $n = | \mathbf{x}_1 |$, $m = | \mathbf{x}_2 |$:

$$\rho_{i+1} \models \bigvee \overline{\Gamma}_1 \vee \bigvee \overline{\Gamma}_2 \quad [\mathbf{d}_1/\mathbf{x}_1][\mathbf{d}_2/\mathbf{x}_2].$$

By the truth condition associated to $\vee$ we can conclude that, for all $\mathbf{d}_1$, $\mathbf{d}_2$, either:

$$\rho_{i+1} \models \bigvee \overline{\Gamma}_1 \quad [\mathbf{d}_1/\mathbf{x}_1][\mathbf{d}_2/\mathbf{x}_2] \quad \text{or} \quad \rho_{i+1} \models \bigvee \overline{\Gamma}_2 \quad [\mathbf{d}_1/V_1][\mathbf{d}_2/V_2].$$

Since $\mathbf{x}_1 \cap \mathsf{fv}(\Gamma_2) = \varnothing$ and $\mathbf{x}_2 \cap \mathsf{fv}(\Gamma_1) = \varnothing$, the above is equivalent to:

$$\rho_{i+1} \models \bigvee \overline{\Gamma}_1 \quad [\mathbf{d}_1/\mathbf{x}_1] \quad \text{or} \quad \rho_{i+1} \models \bigvee \overline{\Gamma}_2 \quad [\mathbf{d}_2/\mathbf{x}_2].$$

And, since this holds for all choices of $\mathbf{d}_1$ and $\mathbf{d}_2$, we can conclude that:

$$\rho_{i+1} \models \forall \mathbf{x}_1 (\bigvee \overline{\Gamma}_1) \quad \text{or} \quad \rho_{i+1} \models \forall \mathbf{x}_2 (\bigvee \overline{\Gamma}_2).$$

Take S_{i+1} to be the S^k such that $\rho_{i+1} \models \forall \mathbf{x}_k (\bigvee \overline{\Gamma}_k)$, for $k = \{1, 2\}$.

For all the remaining cases, S_{i+1} is the unique premiss of the rule r_i. Moreover, for $\mathbf{x}$ the unique (possibly empty) annotation explicitly indicated in the remaining rules, let $n = | \mathbf{x} |$ and $\mathbf{d} \in |\mathcal{M}^{\times}|^n$.

▹ **Cases** ($\wedge$), ($\vee$), ($\exists$), (id) **and** (TC)

$$\wedge \frac{S_{i+1} = \mathbf{Q}, \{\Gamma, A, B\}^{\mathbf{x}}}{S_i = \mathbf{Q}, \{\Gamma, A \wedge B\}^{\mathbf{x}}} \quad \vee \frac{S_{i+1} = \mathbf{Q}, \{\Gamma, A\}^{\mathbf{x}}, \{\Gamma, B\}^{\mathbf{x}}}{S_i = \mathbf{Q}, \{\Gamma, A \vee B\}^{\mathbf{x}}} \quad \exists \frac{S_{i+1} = \mathbf{Q}, \{\Gamma, A(y)\}^{\mathbf{x}, y}}{S_i = \mathbf{Q}, \{\Gamma, \exists x (A(x))\}^{\mathbf{x}}}$$

$$\mathsf{id} \frac{S_{i+1} = \mathbf{Q}, \{\Gamma\}^{\mathbf{x}}}{S_i = \mathbf{Q}, \{\Gamma, A\}^{\mathbf{x}}, \{\overline{A}\}^{\varnothing}} \quad TC \frac{S_{i+1} = \mathbf{Q}, \{\Gamma, A(s,t)\}^{\mathbf{x}}, \{\Gamma, A(s,z), TC(A)(z,t)\}^{\mathbf{x}, z}}{S_i = \mathbf{Q}, \{\Gamma, TC(A)(s,t)\}^{\mathbf{x}}}$$

For all these cases set $\rho_{i+1} = \rho_i$. The formula interpretation of the conclusion logically implies the formula interpretation of the premiss. Thus, from $\mathcal{M}^{\times}, \rho_i \not\models S_i$ we have that $\mathcal{M}^{\times}, \rho_{i+1} \not\models S_{i+1}$. Let us justify this explicitly for the cases ($\exists$), (id) and (TC). ($\exists$) By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x} (\bigvee \overline{\Gamma} \vee \forall x (\overline{A}(x)))$. We have $\rho_{i+1} \not\models \mathbf{Q}$.

By prenexing the quantifier and variable renaming we obtain $\rho_{i+1} \models \forall \mathbf{x} \forall y (\bigvee \overline{\Gamma} \vee \overline{A}(y))$.

(id) By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x} (\bigvee \overline{\Gamma} \vee \overline{A})$ and $\rho_i \models A$. By the truth condition for $\forall$ we have that, for all choices of $\mathbf{d}$, it holds that:

$$\rho_{i+1} \models \bigvee \overline{\Gamma} \vee \overline{A} \quad [\mathbf{d}/\mathbf{x}].$$

By the truth condition for $\vee$, for every choice of $\mathbf{d}$:

$$\rho_{i+1} \models \bigvee \overline{\Gamma} \quad [\mathbf{d}/\mathbf{x}] \quad \text{or} \quad \rho_{i+1} \models \overline{A} \quad [\mathbf{d}/\mathbf{x}].$$

Since $\mathsf{fv}(A) \cap \mathbf{x} = \varnothing$, the above is equivalent to:

$$\rho_{i+1} \models \bigvee \overline{\Gamma} \quad [\mathbf{d}/\mathbf{x}] \quad \text{or} \quad \rho_{i+1} \models \overline{A}.$$

By assumption, $\rho_{i+1} \models A$. Thus, the second disjunct cannot hold, and we have that $\rho_{i+1} \models \bigvee \overline{\Gamma} \quad [\mathbf{d}/\mathbf{x}]$. Since this holds for all choices of $\mathbf{d}$, we conclude that $\rho_{i+1} \models \forall \mathbf{x} (\bigvee \overline{\Gamma})$.

(TC) By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x} (\bigvee \overline{\Gamma} \vee \overline{TC(A)(s,t))})$. Recall that $\overline{TC(A)(s,t))} := \overline{TC}(\overline{A})(s,t)$. We reason as follows:

$$\rho_{i+1} \models \forall \mathbf{x} (\bigvee \overline{\Gamma} \vee \overline{TC}(\overline{A})(s,t))$$

$$\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee (\overline{A}(s,t) \wedge \forall z(\overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t)))) \qquad (\star)$$
$$\rho_{i+1} \models \forall \mathbf{x}((\bigvee \overline{\Gamma} \vee \overline{A}(s,t)) \wedge (\bigvee \overline{\Gamma} \vee \forall z(\overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t))))$$
$$\rho_{i+1} \models \forall \mathbf{x}((\bigvee \overline{\Gamma} \vee \overline{A}(s,t)) \wedge \forall z(\bigvee \overline{\Gamma} \vee \overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t)))$$
$$\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{A}(s,t)) \wedge \forall \mathbf{x}\forall z(\bigvee \overline{\Gamma} \vee \overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t)) \qquad (*)$$
$$\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{A}(s,t)) \text{ and } \rho_{i+1} \models \forall \mathbf{x}\forall z(\bigvee \overline{\Gamma} \vee \overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t)).$$

In the above, step $(\star)$ follows from the inductive definition of $\overline{TC}$, and step $(*)$ is obtained by distributing $\forall$ over $\wedge$, i.e., by means of the classical theorem $\forall x(A \wedge B) \to (\forall x(A) \wedge \forall x(B))$. The other steps are either standard theorems or follow from the truth conditions of the logical operators.

For the three remaining cases of (inst), ($\forall$) and ($\overline{TC}$), ρ_{i+1} extends ρ_i by adequately interpreting the new function symbols introduced, bottom-up:

▷ **Case** (inst)

$$\text{inst}\,\frac{\mathbf{S}_{i+1} \;=\; \mathbf{Q}, \{\Gamma(t)\}^{\mathbf{x}}}{\mathbf{S}_i \;=\; \mathbf{Q}, \{\Gamma(y)\}^{\mathbf{x},y}}$$

By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x}\,\forall x(\bigvee \overline{\Gamma}(x))$. Thus, for all choices of $\mathbf{d}$, we have that $\rho_i \models \forall x(\bigvee \overline{\Gamma}(x))\ [\mathbf{d}/\mathbf{x}]$. By the truth condition for $\forall$, this means that, for all $d \in |\mathcal{M}^{\times}|$, $\rho_i \models \bigvee \overline{\Gamma}(x)\ [\mathbf{d}/\mathbf{x}][d/x]$. Take ρ_{i+1} to be any extension of ρ_i that is defined on the language of $\mathbf{S}_{i+1}$. That is, if f is a function symbol in t to which ρ_i already assigns a map, then ρ_{i+1} assigns to it that same map. Otherwise, ρ_{i+1} assigns an arbitrary map to f.

It follows that $\rho_{i+1} \not\models fm(\mathbf{Q})$ and $\rho_{i+1} \models \bigvee \overline{\Gamma}(t)[\mathbf{d}/\mathbf{x}]$ and, since this holds for all $\mathbf{d}$, we have that $\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma}(t))$. Thus $\rho_{i+1} \not\models \mathbf{S}_{i+1}$.

▷ **Case** ($\forall$)

$$\forall\,\frac{\mathbf{S}_{i+1} \;=\; \mathbf{Q}, \{\Gamma, A(f(\mathbf{x}))\}^{\mathbf{x}}}{\mathbf{S}_i \;=\; \mathbf{Q}, \{\Gamma, \forall x(A(x))\}^{\mathbf{x}}}$$

By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \exists x(\overline{A}(x))$. By the truth condition for $\forall$ and $\vee$, for all choices of $\mathbf{d}$ we have:

$$\rho_i \models \bigvee \overline{\Gamma}\quad [\mathbf{d}/\mathbf{x}] \quad \text{or} \quad \rho_i \models \exists x(\overline{A}(x))\quad [\mathbf{d}/\mathbf{x}].$$

We define ρ_{i+1} to extend ρ_i by defining $\rho_{i+1}(f)$ as follows. Let $\mathbf{d} \subseteq |\mathcal{M}^{\times}|$. If $\rho_i \models \bigvee \overline{\Gamma}\,[\mathbf{d}/\mathbf{x}]$ then we may set $\rho_{i+1}(f)(\mathbf{d})$ to be arbitrary. We still have $\rho_{i+1} \models \bigvee \overline{\Gamma}\,[\mathbf{d}/\mathbf{x}]$, as required. Otherwise $\rho_i \models \exists x(\overline{A}(x))\,[\mathbf{d}/\mathbf{x}]$. By the truth condition for $\exists$, there is a $d \in |\mathcal{M}^{\times}|$ such that $\rho_i \models \overline{A}(x)\,[\mathbf{d}/\mathbf{x}][d/x]$. In this case, we define $\rho_{i+1}(f)(\mathbf{d}) = d$, so that $\rho_{i+1} \models \overline{A}(f(\mathbf{x}))\,[\mathbf{d}/\mathbf{x}]$. So, for all $\mathbf{d}$, we have that $\rho_{i+1} \models \bigvee \overline{\Gamma} \vee \overline{A}(f(\mathbf{x}))\,[\mathbf{d}/\mathbf{x}]$, and so $\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{A}(f(\mathbf{x})))$. Thus, $\rho_{i+1} \not\models \mathbf{S}_{i+1}$, as required.

▷ **Case** ($\overline{TC}$)

$$\overline{TC}\,\frac{\mathbf{S}_{i+1} \;=\; \mathbf{Q}, \{\Gamma, A(s,t), A(s, f(\mathbf{x}))\}^{\mathbf{x}}, \{\Gamma, A(s,t), \overline{TC}(A)(f(\mathbf{x}), t)\}^{\mathbf{x}}}{\mathbf{S}_i \;=\; \mathbf{Q}, \{\Gamma, \overline{TC}(A)(s,t)\}^{\mathbf{x}}}$$

By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{\overline{TC}(A)(s,t)})$ which, by definition of duality, means $\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee TC(\overline{A})(s,t))$. By the truth conditions for $\vee$ we have, for all $\mathbf{d}$:

$$1)\quad \rho_i \models \bigvee \overline{\Gamma}\,[\mathbf{d}/\mathbf{x}] \quad \text{or} \quad 2)\quad \rho_i \models TC(\bar{A})(s,t)\,[\mathbf{d}/\mathbf{x}].$$

We define ρ_{i+1} to extend ρ_i by defining $\rho_{i+1}(f)$ as follows. Let $\mathbf{d} \subseteq \mathcal{M}^\times$. If 1) holds, then we may set $\rho_{i+1}(f)(\mathbf{d})$ to be an arbitrary element of $|\mathcal{M}^\times|$. Otherwise, 2) must hold, so by the truth conditions for TC there is a $\bar{A}$-path between $\rho_i(s)$ and $\rho_i(t)$ of length greater or equal than 1, i.e. there are elements $d_0, \ldots, d_n$, with $n > 0$ and $\rho_i(s) = d_0$ and $\rho_i(t) = d_n$, such that $\rho_i \models \bar{A}(d_i, d_{i+1})$ for all $i < n$. We select a *shortest* such path, i.e. one with smallest possible $n > 0$. There are two cases:

(i) if $n = 1$, then already $\rho_i \models \bar{A}(s,t)[\mathbf{d}/\mathbf{x}]$, so we may set $\rho_{i+1}(f)(\mathbf{d})$ to be arbitrary;
(ii) otherwise $n > 1$ and we set $\rho_{i+1}(f)(\mathbf{d}) = d_1$, so that $\rho_{i+1} \models \bar{A}(s, f(\mathbf{x}))[\mathbf{d}/\mathbf{x}]$ and $\rho_{i+1} \models TC(\bar{A})(f(\mathbf{x}), t)[\mathbf{d}/\mathbf{x}]$.

We have considered all the rules, so the construction of $\mathcal{B}^\times$ and the all ρ_i's is complete. From here, note that we have $\rho_i \subseteq \rho_{i+1}$, for all $i < \omega$. Thus we can construct the limit $\rho^\times = \bigcup_{i<\omega} \rho_i$.

5.3 Canonical Assignments Along Countermodel Branches

Let us now fix $\mathcal{B}^\times$ and $\rho^\times$ as provided by Lemma 5.4 above. Moreover, let us henceforth assume that $\mathcal{D}$ is a proof, i.e. it is progressing, and fix a progressing hypertrace $\mathcal{H} = (\{\Gamma_i\}^{\mathbf{x}_i})_{i<\omega}$ along $\mathcal{B}^\times$. In order to carry out an infinite descent argument, we will need to define a particular trace along this hypertrace that 'preserves' falsity, bottom-up. This is delicate since the truth values of formulas in a trace depend on the assignment of elements to variables in the annotations. A particular issue here is the instantiation rule inst, which requires us to 'revise' whatever assignment of y we may have defined until that point. Thankfully, our earlier convention on substitution-freeness and freshness of the variables introduced by quantifiers and transitive closure rules in $\mathcal{D}$ facilitates the convergence of this process to a canonical such assignment:

Definition 5.5 *(Assignment)* We define $\delta_{\mathcal{H}} : \bigcup\limits_{i<\omega} \mathbf{x}_i \to |\mathcal{M}^\times|$ by $\delta_{\mathcal{H}}(x) := \rho(t)$ if x is instantiated by t in $\mathcal{H}$; otherwise $\delta_{\mathcal{H}}(x)$ is some arbitrary $d \in |\mathcal{M}^\times|$.

Note that $\delta_{\mathcal{H}}$ is indeed well-defined, thanks to the convention that each quantifier and transitive closure rule introduces only variables that do not appear in previous sequents in the derivation branch. In particular we have that each variable x is instantiated at most once along a hypertrace. Henceforth we shall simply write $\rho, \delta_{\mathcal{H}} \models A(\mathbf{x})$ instead of $\rho \models A(\delta_{\mathcal{H}}(\mathbf{x}))$. Working with such an assignment ensures that false formulas along $\mathcal{H}$ always have a false immediate ancestor:

Lemma 5.6 (Falsity through $\mathcal{H}$) *If $\rho^\times, \delta_{\mathcal{H}} \not\models F$ for some $F \in \Gamma_i$, then F has an immediate ancestor $F' \in \Gamma_{i+1}$ with $\rho^\times, \delta_{\mathcal{H}} \not\models F'$.*

Proof Suppose that F is a formula occurring in some cedent $\{\Gamma_i\}^{\mathbf{x}_i}$ in $\boldsymbol{S}_i$, such that $\rho^\times, \delta_{\mathcal{H}} \not\models F$. We show how to choose a F' satisfying the conditions of the Lemma. We distinguish cases according to the rule r_i that $\{\Gamma_i\}^{\mathbf{x}_i}$ concludes. Propositional cases are routine as well as $\cup$, since the failed branch has been chosen during the construction of $\mathcal{B}^\times$. For the weakening rule, observe that we could not have chosen a hypertrace going through the structure which gets weakened, as by assumption the hypertrace is infinite. We show the remaining cases. It can be easily checked that, given a formula F such that $\rho^\times, \delta_{\mathcal{H}} \not\models F$, the formula F' is an immediate ancestor of F.

▷ **Case** ($\exists$). Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma, \exists x(A(x))\}^{\mathbf{x}}$ and $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma, A(y)\}^{\mathbf{x},y}$. By assumption, $\rho^\times \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \forall x(\overline{A}(x)))$. By the truth condition for $\forall$ we obtain that, for all n-tuples $\mathbf{d}$ of elements of $|\mathcal{M}^\times|$, for $n = |\mathbf{x}|$, it holds that:

$$\rho^\times \models \bigvee \overline{\Gamma} \vee \forall x(\overline{A}(x)) \quad [\mathbf{d}/\mathbf{x}]. \tag{5}$$

By definition, $\delta_{\mathcal{H}}$ assigns a value in the domain to all the variables occurring in annotations along $\mathcal{H}$. From (5) and the truth condition for $\vee$ it follows that:

$$\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma} \quad \text{or} \quad \rho^\times, \delta_{\mathcal{H}} \models \forall x(\overline{A}(x)).$$

If $F \in \Gamma$, by hypothesis we have that $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$ and we set $F' = F$. Otherwise, $F = \exists x(A(x))$ and $\rho^\times, \delta_{\mathcal{H}} \models \forall x(\overline{A}(x))$. By the truth condition for $\forall$ we have $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(y)$, so we set $F' = A(y)$.

▷ **Case** (inst). Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma(y)\}^{\mathbf{x},y}$ and $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma(t)\}^{\mathbf{x}}$. By construction, $\rho^\times \models \forall \mathbf{x} \forall y(\bigvee \overline{\Gamma}(y))$. Reasoning as in the previous case, from the truth condition for $\forall$ it follows that $\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma}(y)$. If F does not contain y, then $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$, and we set $F' = F$. Otherwise, if F contains y, then $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}(y)$. By Lemma 5.4, $\rho^\times$ assigns a value to t, and by Definition 5.5, since y is instantiated with t along $\mathcal{H}$, $\delta_{\mathcal{H}}(y) = \rho^\times(t)$. Therefore we have $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}(t)$ so we set $F' = F(t)$.

▷ **Case** (id). Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma, A\}^{\mathbf{x}}$ and $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma\}^{\mathbf{x}}$. Observe that the hypertrace $\mathcal{H}$ could not have gone through the structure $\{\overline{A}\}^{\varnothing}$ occurring in the conclusion of the rule, because by assumption $\mathcal{H}$ is infinite. Moreover, by construction the formula interpretation of all the cedents along $\mathcal{B}^\times$ is not valid, and thus $\rho^\times \not\models \{\overline{A}\}^{\varnothing}$. This implies that $\rho^\times \models A$ and so:

$$\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma}.$$

So we have that $F \in \Gamma$ and $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$. Set $F' = F$.

▷ **Case** (TC). Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma, TC(A)(s,t)\}^{\mathbf{x}}$. By assumption, $\rho^\times \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{TC(A)(s,t)})$. Thus, we have that:

$$\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma} \text{ or } \rho^\times, \delta_{\mathcal{H}} \models \overline{TC}(\overline{A})(s,t).$$

From the inductive definition of $\overline{TC}$ and the truth condition for $\wedge$, the second disjunct is equivalent to:

$$\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s,t) \quad \text{and} \quad \rho^\times, \delta_{\mathcal{H}} \models \forall z(\overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t)). \tag{6}$$

There are two cases to consider, since the premiss of the rule has two cedents that the hypertrace $\mathcal{H}$ could follow:

(i) $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma, A(s,t)\}^{\mathbf{x}}$. By construction, $\rho^\times \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{A}(s,t))$, and thus $\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma} \vee \overline{A}(s,t)$. If $F \in \Gamma$, then $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$. Set $F = F'$. Otherwise, $F = TC(A)(s,t)$ and $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$. By (6) we have that $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s,t)$. Set $F' = A(s,t)$.

(ii) $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma, A(s,z), TC(A)(z,t)\}^{\mathbf{x},z}$. By construction, we have that $\rho^\times \models \forall \mathbf{x} \forall z(\bigvee \overline{\Gamma} \vee (\overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t)))$. Since z does not occur free in Γ, this is equivalent to $\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma} \vee \forall z(\overline{A}(s,z) \vee \overline{TC}(\overline{A})(z,t))$. If $F \in \Gamma$ and $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$, set $F' = F$. Suppose $F = TC(A)(s,t)$ and $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$. From (6) we have:

$$\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s,z) \vee \overline{TC}(A)(z,t).$$

If $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s, z)$, set $F' = A(s, z)$. Otherwise, if $\rho^\times, \delta_{\mathcal{H}} \models \overline{TC}(A)(z, t)$, set $F' = TC(A)(z, t)$.

▷ **Case** (∀). Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma, \forall x(A(x))\}^{\mathbf{x}}$ and $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma, A(f(\mathbf{x}))\}^{\mathbf{x}}$. By assumption, $\rho^\times \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \exists x(\overline{A}(x)))$ and, from the truth conditions for ∀ and ∨:

$$\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma} \quad \text{or} \quad \rho^\times, \delta_{\mathcal{H}} \models \exists x(\overline{A}(x)).$$

If $F \in \Gamma$ and $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$, set $F' = F$, since x does not occur in Γ. Otherwise, $F = \forall x(A(x))$ and $\rho^\times, \delta_{\mathcal{H}} \models \exists x(\overline{A}(x))$. By definition of $\rho^\times$ and $\delta_{\mathcal{H}}$ we have $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(f(\mathbf{x}))$, so we set $F' = A(f(\mathbf{x}))$.
▷ **Case** ($\overline{TC}$). Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma, \overline{TC}(A)(s, t)\}^{\mathbf{x}}$. By assumption, $\rho^\times \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee \overline{\overline{TC}}(A)(s, t))$, that is, $\rho^\times \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee TC(\overline{A})(s, t))$. Thus, we have that:

$$\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma} \quad \text{or} \quad \rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(s, t).$$

We need to consider two cases, depending on which cedent the hypertrace $\mathcal{H}$ follows:

(i) $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma, A(s, t), A(s, f(\mathbf{x}))\}^{\mathbf{x}}$. If $F \in \Gamma$ set $F = F'$. By assumption it follows that $\rho^\times, \delta_{\mathcal{H}} \models \overline{F}$. Otherwise, $F = \overline{TC}(A)(s, t)$ and by assumption $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(s, t)$. By the inductive definition of TC and the truth condition for ∨, this is equivalent to:

$$\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s, t) \quad \text{or} \quad \rho^\times, \delta_{\mathcal{H}} \models \exists z(\overline{A}(s, z) \wedge TC(\overline{A})(z, t)).$$

If $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s, t)$, set $F' = A(s, t)$. Otherwise, $\rho^\times, \delta_{\mathcal{H}} \models \exists z(\overline{A}(s, z) \wedge TC(\overline{A})(z, t))$. Let $\delta_{\mathcal{H}}(\mathbf{x}) = \mathbf{d}$. According to the definition of $\rho^\times$ at the ($\overline{TC}$) step, since $\rho^\times, \delta_{\mathcal{H}} \not\models \bigvee \overline{\Gamma}$ and $\rho^\times, \delta_{\mathcal{H}} \not\models \overline{A}(s, t)$, then $\rho^\times(f)(\mathbf{d})$ is defined as in case 2), subcase (ii). Thus, $\rho^\times(f)(\mathbf{d})$ is an element $d \in |\mathcal{M}^\times|$ such that $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s, f(\mathbf{x}))$ and $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(f(\mathbf{x}), t)$. Set $F' = A(s, f(\mathbf{x}))$.

(ii) $\{\Gamma_{i+1}\}^{\mathbf{x}_{i+1}} = \{\Gamma, A(s, t), \overline{TC}(A)(f(\mathbf{x}), t)\}^{\mathbf{x}}$. The proof proceeds exactly as in the previous case except for the very last step, where F' is set to be $\overline{TC}(A)(f(\mathbf{x}), t)$.

5.4 Putting It All Together

Note how rule inst of Fig. 4 is handled in the proof above: if $F \in \Gamma(y)$ then we can choose $F' = F[t/y]$ which, by definition of $\delta_{\mathcal{H}}$, has the same truth value. By repeatedly applying Lemma 5.6 we obtain:

Proposition 5.7 (False trace) *There exists an infinite trace $\tau^\times = (F_i)_{i<\omega}$ through $\mathcal{H}$ such that, for all i, it holds that $\mathcal{M}^\times, \rho^\times, \delta_{\mathcal{H}} \not\models F_i$.*

Proof Sketch We inductively define $\tau^\times$ as follows. By assumption, $\mathcal{M}^\times, \rho^\times \not\models \boldsymbol{S}_0$, and thus in particular $\rho^\times \models \forall \mathbf{x}_0(\bigvee \overline{\Gamma}_0)$. Thus, for all n-tuples $\mathbf{d}$ of elements of $|\mathcal{M}^\times|$, for $n = |\mathbf{x}_0|$, we have that $\rho^\times \models \bigvee \overline{\Gamma}_0[\mathbf{d}/\mathbf{x}_0]$. Since $\delta_{\mathcal{H}}$ assigns a value to all variables occurring in annotations of cedents in $\mathcal{H}$, $\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma}_0$. Take $F_0 \in \Gamma_0$ such that $\mathcal{M}^\times, \rho^\times, \delta_{\mathcal{H}} \not\models \overline{F}_0$. From here we define each successive F_i by appealing repeatedly to Lemma 5.6 above.

We are now ready to prove our main soundness result.

Proof of Theorem 5.1 Fix the infinite trace $\tau^\times = (F_i)_{i<\omega}$ through $\mathcal{H}$ obtained by Proposition 5.7. Since $\tau^\times$ is infinite, by definition of HTC proofs, it needs to be progressing, i.e., it is infinitely often $\overline{TC}$-principal and there is some $k \in \mathbb{N}$ s.t. for $i > k$ we have that $F_i = \overline{TC}(A)(s_i, t_i)$ for some terms s_i, t_i.

To each F_i, for $i > k$, we associate the natural number n_i measuring the '$\bar{A}$-distance between s_i and t_i'. Formally, $n_i \in \mathbb{N}$ is least s.t. there are $d_0, \dots, d_{n_i} \in |\mathcal{M}^\times|$ with $\rho^\times(s_i) = d_0, \rho^\times(t_i) = d_{n_i}$ and, for all $i < n_i$, $\rho^\times, \delta_{\mathcal{H}} \models \bar{A}(d_i, d_{i+1})$. Our aim is to show that $(n_i)_{i>k}$ has no minimal element, contradicting wellfoundness of $\mathbb{N}$. For this, we establish the following two local properties:

1. $(n_i)_{i>k}$ is *monotone decreasing*, i.e., for all $i > k$, we have $n_{i+1} \leq n_i$;
2. Whenever F_i is principal, we have $n_{i+1} < n_i$.

We prove item 1 and item 2 by inspection on HTC rules. We start with item 2. Suppose $F_i = \overline{TC}(A)(s, t)$ is the principal formula in an occurrence of $\overline{\mathsf{tc}}$ so $F_{i+1} = \overline{TC}(A)(f(\mathbf{x}), t)$, for some $\mathbf{x}$. Moreover, by construction $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(s, t)$ and $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(f(\mathbf{x}), t)$. We have to show that n_i, the $\overline{A}$-distance between $f(\mathbf{x})$ and t, is strictly smaller than n_{i+1}, the $\overline{A}$-distance between s and t, wrt. $\rho^\times$ and $\delta_{\mathcal{H}}$.

Let $\delta_{\mathcal{H}}(\mathbf{x}) = \mathbf{d}$. By case ($\overline{TC}$) of Lemma 5.4, and since $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(f(\mathbf{x}), t)$, there is a shortest $\overline{A}$-path between $\rho^\times(s)$ and $\rho^\times(t)$, composed of $n > 1$ elements $d_0, \dots, d_n$, with $\rho^\times(s) = d_0$ and $\rho^\times(t) = d_n$, and $\rho^\times(f)(\mathbf{d}) = d_1$. Consequently, it holds that $\rho^\times, \delta_{\mathcal{H}} \models \overline{A}(s, f(\mathbf{x}))$ and $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(f(\mathbf{x}), t)$. Thus, there is an $\overline{A}$-edge between $\rho^\times(s)$ and $\rho^\times(f)(\mathbf{d})$, and $\rho^\times(f)(\mathbf{d})$ is one edge closer to $\rho^\times(t)$ in one of the shortest $\overline{A}$-paths between $\rho^\times(s)$ and $\rho^\times(t)$. We conclude that n_{i+1} is exactly $n_i - 1$, and $n_{i+1} < n_i$.

To prove item 1, suppose that F_i is not principal in the occurrence r_i of a HTC rule. Suppose r_i is inst, $F_i = \overline{TC}(A)(s, x)$, $F_{i+1} = \overline{TC}(A)(s, t)$, and x gets instantiated with t by inst. By construction, $\rho^\times, \delta_{\mathcal{H}} \models TC(\overline{A})(s, x)$. Let n_i be the distance between $\rho^\times(s)$ and $\delta_{\mathcal{H}}(x)$. By definition, $\delta_{\mathcal{H}}(x) = \rho^\times(t)$. Thus, the distance between $\rho^\times(s)$ and $\rho^\times(t)$ is $n_{i+1} = n_i$. In all the other cases, $F_i = \overline{TC}(A)(s, t) = F_{i+1}$, and thus $n_{i+1} = n_i$.

So $(n_i)_{i>k}$ is monotone decreasing by item 1 but cannot converge by item 2, and by definition of progressing trace. Thus $(n_i)_{k<i}$ has no minimal element, yielding the required contradiction.

6 Completeness for PDL$^+$, Over the Standard Translation

In this section we give our next main result:

Theorem 6.1 (Completeness for PDL$^+$) *For a PDL$^+$ formula A, if $\models A$ then* HTC $\vdash_{cyc}$ ST$(A)(c)$.

The proof is by a direct simulation of a cut-free cyclic system for PDL$^+$ that is complete. We shall briefly sketch this system below.

6.1 Cyclic System for PDL$^+$

The system LPD$^+$, shown in Fig. 5, is the natural extension of the usual sequent calculus for basic multimodal logic K by rules for programs. In Fig. 5, Γ, Δ etc. range over sets of PDL$^+$ formulas, and we write $\langle a \rangle \Gamma$ as a shorthand for $\{\langle a \rangle B : B \in \Gamma\}$. (Regular) preproofs for this system are defined just like for HTC or TC$_G$.

The notion of ancestry for formulas is colour-coded in Fig. 5 as before: a formula C in a premiss is an immediate ancestor of a formula C' in the conclusion if they have the same

$$\mathsf{id}\,\frac{}{p,\bar{p}} \qquad \mathsf{wk}\,\frac{\Gamma}{\Gamma, A} \qquad \mathsf{k}_a\,\frac{\Gamma, A}{\langle a\rangle\Gamma, [a]A} \qquad \wedge\,\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B} \qquad \vee_0\,\frac{\Gamma, A_0}{\Gamma, A_0 \vee A_1} \qquad \vee_1\,\frac{\Gamma, A_1}{\Gamma, A_0 \vee A_1}$$

$$\langle\rangle\,\frac{\Gamma, \langle\alpha\rangle\langle\beta\rangle A}{\Gamma, \langle\alpha\beta\rangle A} \qquad \langle\cup\rangle_0\,\frac{\Gamma, \langle\alpha_0\rangle A}{\Gamma, \langle\alpha_0 \cup \alpha_1\rangle A} \qquad \langle\cup\rangle_1\,\frac{\Gamma, \langle\alpha_1\rangle A}{\Gamma, \langle\alpha_0 \cup \alpha_1\rangle A} \qquad [\cup]\,\frac{\Gamma, [\alpha]A \quad \Gamma, [\beta]A}{\Gamma, [\alpha \cup \beta]A}$$

$$[\,]\,\frac{\Gamma, [\alpha][\beta]A}{\Gamma, [\alpha\beta]A} \qquad \langle+\rangle_0\,\frac{\Gamma, \langle\alpha\rangle A}{\Gamma, \langle\alpha^+\rangle A} \qquad \langle+\rangle_1\,\frac{\Gamma, \langle\alpha\rangle\langle\alpha^+\rangle A}{\Gamma, \langle\alpha^+\rangle A} \qquad [+]\,\frac{\Gamma, [\alpha]A \quad \Gamma, [\alpha][\alpha^+]A}{\Gamma, [\alpha^+]A}$$

Fig. 5 Rules of LPD$^+$

colour; if $C, C' \in \Gamma$ then we furthermore require $C = C'$. More formally (and without relying on colours):

Definition 6.2 *(Immediate ancestry)* Fix a preproof $\mathcal{D}$. We say that a formula occurrence C is an *immediate ancestor* of a formula occurrence D in $\mathcal{D}$ if C and D occur in a premiss and the conclusion, respectively, of an inference step r of $\mathcal{D}$ and:

- If r is a k_a step then, as typeset in Fig. 5:
  - D is $\langle a\rangle B$ for some $B \in \Gamma$ and C is B; or,
  - D is $[a]A$ and C is A.
- If r is not a k-step then:
  - C and D are occurrences of the same formula; or,
  - D is principal and C is auxiliary in r, i.e. as typeset in Fig. 5, C and D are the (uniquely) distinguished formulas in a premiss and conclusion, respectively;

Definition 6.3 *(Non-wellfounded proofs)* Fix a preproof $\mathcal{D}$ of a sequent Γ. A *thread* is a maximal path in its graph of immediate ancestry. We say a thread is *progressing* if it has a smallest infinitely often principal formula of the form $[\alpha^+]A$.

$\mathcal{D}$ is a *proof* if every infinite branch has a progressing thread. If $\mathcal{D}$ is regular, we call it a *cyclic proof* and we may write $\mathsf{LPD}^+ \vdash_{cyc} \Gamma$.

Soundness of cyclic-LPD$^+$ is established by a standard infinite descent argument, but is also implied by the soundness of cyclic-HTC (Theorem 5.1) and the simulation we are about to give (Theorem 6.1), though this is somewhat overkill. Completeness may be established by the game theoretic approach of Niwinskí and Walukiewicz [24], as carried out by Lange in [21], or by the purely proof theoretic techniques of Studer [27]. Either way, both results follow immediately by a standard embedding of PDL$^+$ into the (guarded) μ-calculus and its known completeness results [24, 27], by way of a standard 'proof reflection' argument: μ-calculus proofs of the embedding are 'just' step-wise embeddings of LPD$^+$ proofs.

Theorem 6.4 (Soundness and completeness, essentially from [21]) *Let A be a PDL$^+$ formula. $\models A$ iff* $\mathsf{LPD}^+ \vdash_{cyc} A$.

6.2 Examples of Cyclic Proofs in LPD$^+$

Before giving our main simulation result, let us first see some examples of proofs in LPD$^+$, in particular addressing the 'counterexample' from Sect. 3.3.

Example 6.5 Below is a cyclic LPD^+ proof of the LPD^+ sequent:

$$[(aa \cup aba)^+]\overline{p},\ \langle a^+\rangle p,\ \langle (ba^+)^+\rangle p,\ \langle a^+\rangle\langle (ba^+)^+\rangle p.$$

We use the following abbreviations: $\alpha = (aa \cup aba)^+$ and $\beta = (ba^+)^+$. Moreover, we sometimes use rule $\langle + \rangle$, which is derivable from rules $\langle + \rangle_0$ and $\langle + \rangle_1$, keeping in mind that LPD^+ sequents are *sets* of formulas:

$$\langle + \rangle \frac{\Gamma,\ \langle \alpha\rangle A,\ \langle \alpha\rangle\langle \alpha^+\rangle A}{\Gamma,\ \langle \alpha^+\rangle A}$$

Similarly for rule $\vee$. There are uncountably many infinite branches, but each such branch is supported by the infinite thread induced by the blue-coloured formulas. Γ is sequent $[aa \cup aba]\overline{p}$, $\langle a^+\rangle p$, $\langle \beta\rangle p$, $\langle a^+\rangle\langle \beta\rangle p$, derivable by means of a finite derivation (which we do not show).

$$
\begin{prooftree}
\AxiomC{Γ}
\AxiomC{$\vdots$}
\RightLabel{$\circ$}
\UnaryInfC{$[\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a\rangle\langle a^+\rangle p,\ \langle a\rangle\langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle_1$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle a\rangle\langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][a][\alpha]\overline{p},\ \langle a\rangle\langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle_1$}
\UnaryInfC{$[a][a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$[;]$}
\UnaryInfC{$[aa][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\AxiomC{$\vdots$}
\RightLabel{$\circ$}
\UnaryInfC{$[\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a\rangle\langle a^+\rangle p,\ \langle a\rangle\langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle_1$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle a\rangle\langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{k_b}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle b\rangle\langle a^+\rangle p,\ \langle b\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$2\langle ; \rangle$}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle ba^+\rangle p,\ \langle ba^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle$}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle \beta\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][b][a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle_0$}
\UnaryInfC{$[a][b][a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$2[;]$}
\UnaryInfC{$[aba][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$[\cup]$}
\BinaryInfC{$[aa \cup aba][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$[+]$}
\RightLabel{$\circ$}
\BinaryInfC{$[\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\end{prooftree}
$$

Example 6.6 Below we give a cyclic LPD^+ proof of formula (2), which witnesses the incompleteness of TC_G without cut:

$$\langle (aa \cup aba)^+\rangle p \ \rightarrow\ \langle a^+((ba^+)^+ \cup a)\rangle p.$$

We employ the same shorthands as in the previous example, i.e., $\alpha = (aa \cup aba)^+$ and $\beta = (ba^+)^+$. The progressing thread along the unique infinite branch displayed (not including those from Example 6.5), looping on $\bullet$, is coloured blue.

$$
\begin{prooftree}
\AxiomC{Γ'}
\AxiomC{$\vdots$}
\RightLabel{$\bullet$}
\UnaryInfC{$[\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a\rangle\langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$\langle + \rangle_1$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][a][\alpha]\overline{p},\ \langle a\rangle\langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$\langle + \rangle_1$}
\UnaryInfC{$[a][a][\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$[;]$}
\UnaryInfC{$[aa][\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\AxiomC{Ex. 6.5}
\doubleLine
\UnaryInfC{$[\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle \beta\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{k_b}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a\rangle\langle a^+\rangle p,\ \langle a\rangle\langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle_1$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle a\rangle\langle \beta\rangle p,\ \langle a\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle$}
\UnaryInfC{$[a][\alpha]\overline{p},\ \langle a^+\rangle p,\ \langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{k_b}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle b\rangle\langle a^+\rangle p,\ \langle b\rangle\langle a^+\rangle\langle \beta\rangle p$}
\LeftLabel{$2\langle ; \rangle$}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle ba^+\rangle p,\ \langle ba^+\rangle\langle \beta\rangle p$}
\LeftLabel{$\langle + \rangle$}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle \beta\rangle p$}
\LeftLabel{$\langle \cup \rangle_0$}
\UnaryInfC{$[b][a][\alpha]\overline{p},\ \langle \beta \cup a\rangle p$}
\LeftLabel{k_a}
\UnaryInfC{$[a][b][a][\alpha]\overline{p},\ \langle a\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$\langle + \rangle_0$}
\UnaryInfC{$[a][b][a][\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$2[;]$}
\UnaryInfC{$[aba][\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$[\cup]$}
\BinaryInfC{$[aa \cup aba][\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$[+]$}
\RightLabel{$\bullet$}
\BinaryInfC{$[\alpha]\overline{p},\ \langle a^+\rangle\langle \beta \cup a\rangle p$}
\LeftLabel{$\langle ; \rangle$}
\UnaryInfC{$[\alpha]\overline{p},\ \langle a^+(\beta \cup a)\rangle p$}
\LeftLabel{$\vee$}
\UnaryInfC{$[\alpha]\overline{p} \vee \langle a^+(\beta \cup a)\rangle p$}
\end{prooftree}
$$

Here Γ' is the sequent $[aa \cup aba]\overline{p}, \langle a^+\rangle\langle\beta \cup a\rangle p$, which has a finite derivation:

$$
[\cup]\,\dfrac{
[;]\,\dfrac{
\langle + \rangle_0\,\dfrac{
k_a\,\dfrac{
\langle\cup\rangle\,\dfrac{
k_a\,\dfrac{
\mathsf{init}\,\dfrac{}{\overline{p}, p}
}{[a]\overline{p}, \langle a\rangle p}
}{[a]\overline{p}, \langle\beta \cup a\rangle p}
}{[a][a]\overline{p}, \langle a\rangle\langle\beta \cup a\rangle p}
}{[a][a]\overline{p}, \langle a^+\rangle\langle\beta \cup a\rangle p}
}{[aa]\overline{p}, \langle a^+\rangle\langle\beta \cup a\rangle p}
\qquad
2[;]\,\dfrac{
\langle + \rangle_0\,\dfrac{
k_a\,\dfrac{
\langle\cup\rangle\,\dfrac{
\langle + \rangle_0\,\dfrac{
\langle ; \rangle\,\dfrac{
k_b\,\dfrac{
\langle + \rangle_0\,\dfrac{
k_a\,\dfrac{
\mathsf{init}\,\dfrac{}{\overline{p}, p}
}{[a]\overline{p}, \langle a\rangle p}
}{[a]\overline{p}, \langle a^+\rangle p}
}{[b][a]\overline{p}, \langle b\rangle\langle a^+\rangle p}
}{[b][a]\overline{p}, \langle ba^+\rangle p}
}{[b][a]\overline{p}, \langle\beta\rangle p}
}{[b][a]\overline{p}, \langle\beta \cup a\rangle p}
}{[a][b][a]\overline{p}, \langle a\rangle\langle\beta \cup a\rangle p}
}{[a][b][a]\overline{p}, \langle a^+\rangle\langle\beta \cup a\rangle p}
}{[aba]\overline{p}, \langle a^+\rangle\langle\beta \cup a\rangle p}
}{[aa \cup aba]\overline{p}, \langle a^+\rangle\langle\beta \cup a\rangle p}
$$

6.3 A 'Local' Simulation of LPD$^+$ by HTC

In this subsection we show that LPD$^+$-preproofs can be stepwise transformed into HTC-preproofs, with respect to the standard translation. In order to produce our local simulation, we need a refined version of the standard translation, incorporating the structural elements of hypersequents.

In the definitions below we are often using arbitrarily chosen variables (e.g. "fresh variables") and constant symbols. We assume these choices do not break our previous assumptions on variables and constants occurring in hypersequents (refer to Sect. 5.1).

Definition 6.7 *(Hypersequent translation for formulas)* For x a variable and A a PDL$^+$ formula, we define the *hypersequent translation* of A, denoted by $\mathsf{HT}(A)(c)$, by induction on the complexity of A as follows, for $\circ \in \{\wedge, \vee\}$, with d always a fresh constant symbol and y always a fresh variable:

$$
\begin{aligned}
A = p \text{ or } A = B \circ C, &\quad \mathsf{HT}(A)(x) := \quad \{\mathsf{ST}(A)(x)\}^{\varnothing}\\
A = [\alpha]B, &\quad \mathsf{HT}(A)(x) := \quad \{\overline{\mathsf{ST}(\alpha)(x, d)}\}^{\varnothing}, \mathsf{HT}(B)(d)\\
A = \langle\alpha\rangle B, &\quad \mathsf{HT}(A)(x) := \quad \{\mathsf{ST}(\alpha)(x, y), \mathsf{CT}(B)(y)\}^{\mathbf{x}_B, y}
\end{aligned}
$$

where the *cedent translation* $\mathsf{CT}(B)(t)$ and $\mathbf{x}_B$ are defined as follows:

- if $B = p$ or $B = C \circ D$ or $B = [\beta]C$, then:

$$\mathsf{CT}(B)(t) := \mathsf{ST}(B)(y) \text{ and } \quad \mathbf{x}_B = \varnothing.$$

- if $B = \langle\beta\rangle C$ then, for z a fresh variable:

$$\mathsf{CT}(B)(y) := \mathsf{ST}(\beta)(y, z), \mathsf{CT}(C)(z) \quad \text{and} \quad \mathbf{x}_B = \mathbf{x}_C, z.$$

Proposition 6.8 *For $\boldsymbol{S}$ a hypersequent, and A a PDL$^+$ formula:*

1. *There is a finite* HTC*-derivation from $\boldsymbol{S}$, $\mathsf{HT}(A)(c)$ to $\boldsymbol{S}$, $\{\mathsf{CT}(A)(c)\}^{\mathbf{x}_A}$; and,*
2. *There is a finite* HTC*-derivation from $\boldsymbol{S}$, $\{\mathsf{CT}(A)(c)\}^{\mathbf{x}_A}$ to $\boldsymbol{S}$, $\{\mathsf{ST}(A)(c)\}^{\varnothing}$.*

Proof By induction on the complexity of A. If A is atomic, a conjunction or a disjunction, then $\mathsf{HT}(A)(c) = \{\mathsf{CT}(A)(c)\}^{\mathbf{x}_A} = \{\mathsf{ST}(A)(c)\}^{\varnothing}$. If $A = [\alpha]B$, then $\{\mathsf{CT}(A)(c)\}^{\mathbf{x}_A} = \{\mathsf{ST}(A)(c)\}^{\varnothing}$. If $A = \langle\alpha_1\rangle \dots \langle\alpha_n\rangle B$, for $1 \leq n$, then $\{\mathsf{CT}(A)(y)\}^{\mathbf{x}_A} = \mathsf{HT}(A)(c)$, and $\mathbf{x}_A = \{y_1, \dots, y_n\}$, for $y_1, \dots, y_n$ variables that do not occur in B. The remaining cases are shown below. The double line in the derivation below denotes $n-1$ occurrences of rules $\exists$, $\wedge$, and IH indicates appeal to the inductive hypothesis.

$$\begin{array}{rc}
 & \boldsymbol{S}, \mathsf{HT}([\alpha]B)(c) \\ \hdashline
 & \boldsymbol{S}, \{\overline{\mathsf{ST}(\alpha)(c,d)}\}^{\varnothing}, \mathsf{HT}(B)(d) \\ \hline
\mathsf{IH} & \boldsymbol{S}, \{\overline{\mathsf{ST}(\alpha)(c,d)}\}^{\varnothing}, \{\mathsf{ST}(B)(d)\}^{\varnothing} \\ \hline
\vee & \boldsymbol{S}, \{\overline{\mathsf{ST}(\alpha)(c,d)} \vee \mathsf{ST}(B)(d)\}^{\varnothing} \\ \hline
\forall & \boldsymbol{S}, \{\forall y(\overline{\mathsf{ST}(\alpha)(c,y)} \vee \mathsf{ST}(B)(y))\}^{\varnothing} \\ \hdashline
 & \boldsymbol{S}, \{\mathsf{CT}([\alpha]B)(c)\}^{\varnothing}
\end{array}$$

$$\begin{array}{rc}
 & \boldsymbol{S}, \{\mathsf{CT}(\langle\alpha_1\rangle \dots \langle\alpha_n\rangle B)(y_n)\}^{y_1,\dots,y_n} \\ \hdashline
 & \boldsymbol{S}, \{\mathsf{ST}(\alpha_1)(c,y_1), \dots, \mathsf{ST}(\alpha_n)(y_{n-1},y_n), \mathsf{ST}(B)(y_n)\}^{y_1,\dots,y_n} \\ \hline\hline
\exists,\wedge & \boldsymbol{S}, \{\mathsf{ST}(\alpha)(c,y_1), \mathsf{ST}(\langle\alpha_2\rangle \dots \langle\alpha_n\rangle B)(y_1)\}^{y_1} \\ \hline
\wedge & \boldsymbol{S}, \{\mathsf{ST}(\alpha)(c,y_1) \wedge \mathsf{ST}(\langle\alpha_2\rangle \dots \langle\alpha_n\rangle B)(y_1)\}^{y_1} \\ \hline
\exists & \boldsymbol{S}, \{\exists y_1(\mathsf{ST}(\alpha_1)(c,y_1) \wedge \mathsf{ST}(\langle\alpha_2\rangle \dots \langle\alpha_n\rangle B)(y_1))\}^{y_1} \\ \hdashline
 & \boldsymbol{S}, \{\mathsf{ST}(\langle\alpha_1\rangle \dots \langle\alpha_n\rangle B)(c)\}^{\varnothing}
\end{array}$$

Definition 6.9 (HT-*translation*) Let $\mathcal{D}$ be a PDL^+ preproof. We shall define a HTC preproof $\mathsf{HT}(\mathcal{D})(c)$ of the hypersequent $\mathsf{HT}(A)(c)$ by a local translation of inference steps. Formally speaking, the well-definedness of $\mathsf{HT}(\mathcal{D})(c)$ is guaranteed by coinduction: each rule of $\mathcal{D}$ is translated into a (nonempty) derivation.

$$\mathsf{id}\,\frac{}{\Gamma, p, \overline{p}} \quad\rightsquigarrow\quad \begin{array}{rc}
\mathsf{init} & \\ \hline
 & \{\,\}^{\varnothing} \\ \hline
\mathsf{id} & \{p(c)\}^{\varnothing}, \{\overline{p}(c)\}^{\varnothing} \\ \hline
\mathsf{wk} & \mathsf{HT}(\Gamma)(c), \{p(c)\}^{\varnothing}, \{\overline{p}(c)\}^{\varnothing} \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(p)(c), \mathsf{HT}(\overline{p})(c)
\end{array}$$

$$\mathsf{wk}\,\frac{\Gamma}{\Gamma, A} \quad\rightsquigarrow\quad \mathsf{wk}\,\frac{\mathsf{HT}(\Gamma)(c)}{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(A)(c)}$$

$$\vee_i\,\frac{\Gamma, A_i}{\Gamma, A_0 \vee A_1}\, i \in \{0,1\} \quad\rightsquigarrow\quad \begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(A_i)(c) \\ \hline\hline
\mathrm{Pr.}6.8 & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(A_i)(c)\}^{\varnothing} \\ \hline
\vee_i & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(A_0)(c) \vee \mathsf{ST}(A_1)(c)\}^{\varnothing} \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(A_0 \vee A_1)(c)
\end{array}$$

$$\wedge\,\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \wedge B} \quad\rightsquigarrow\quad \begin{array}{rc}
 & \mathrm{Pr.}6.8\,\dfrac{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(A)(c)}{\mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(A)(c)\}^{\varnothing}} \quad \mathrm{Pr.}6.8\,\dfrac{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(B)(c)}{\mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(B)(c)\}^{\varnothing}} \\ \hline
\cup & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(A)(c), \mathsf{ST}(B)(c)\}^{\varnothing} \\ \hline
\wedge & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(A)(c) \wedge \mathsf{ST}(B)(c)\}^{\varnothing} \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(A \wedge B)(c)
\end{array}$$

$$\mathsf{k}_a\,\dfrac{B_1, \dots, B_k, A}{\langle a\rangle B_1, \dots, \langle a\rangle B_k, [a]A}\ \rightsquigarrow$$

$$\begin{array}{rc}
 & \mathsf{HT}(B_1)(c), \dots, \mathsf{HT}(B_k)(c), \mathsf{HT}(A)(c) \\ \hline
[d/c] & \mathsf{HT}(B_1)(d), \dots, \mathsf{HT}(B_k)(d), \mathsf{HT}(A)(d) \\ \hline\hline
\vee,\forall & \{\mathsf{CT}(B_1)(d)\}^{\mathbf{x}_{B_1}}, \dots, \{\mathsf{CT}(B_k)(d)\}^{\mathbf{x}_{B_k}}, \mathsf{HT}(A)(d) \\ \hline
\mathsf{wk} & \{\mathsf{CT}(B_1)(d)\}^{\mathbf{x}_{B_1}}, \dots, \{\mathsf{CT}(B_k)(d)\}^{\mathbf{x}_{B_k}}, \{\overline{\mathsf{ST}(a)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hline\hline
\cup & \{\mathsf{CT}(B_1)(d)\}^{\mathbf{x}_{B_1}}, \dots, \{\mathsf{ST}(a)(c, d), \mathsf{CT}(B_k)(d)\}^{\mathbf{x}_{B_k}}, \{\overline{\mathsf{ST}(a)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hline\hline
\mathsf{inst} & \{\mathsf{ST}(a)(c, y), \mathsf{CT}(B_1)(y)\}^{\mathbf{x}_{B_1}, y}, \dots, \{\mathsf{ST}(a)(c, y), \mathsf{CT}(B_k)(y)\}^{\mathbf{x}_{B_k}, y}, \{\overline{\mathsf{ST}(a)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hdashline
 & \mathsf{HT}(\langle a\rangle B_1)(c), \dots, \mathsf{HT}(\langle a\rangle B_k)(c), \mathsf{HT}([a]A)(c)
\end{array}$$

where (omitted) left-premisses of $\cup$ steps are simply proved by wk, id, init.

$$[\cup]\,\dfrac{\Gamma, [\alpha]A \quad \Gamma, [\beta]A}{\Gamma, [\alpha \cup \beta]A}\ \rightsquigarrow$$

$$\begin{array}{rcc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha]A)(c) & \mathsf{HT}(\Gamma)(c), \mathsf{HT}([\beta]A)(c) \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) & \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\beta)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hline
\cup & \multicolumn{2}{c}{\mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c, d)}, \overline{\mathsf{ST}(\beta)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d)} \\ \hline
\wedge & \multicolumn{2}{c}{\mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c, d)} \wedge \overline{\mathsf{ST}(\beta)(c, d)}\}^{\varnothing}, \mathsf{HT}(A)(d)} \\ \hdashline
 & \multicolumn{2}{c}{\mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha \cup \beta]A)(c)}
\end{array}$$

$$\langle\cup\rangle_i\,\dfrac{\Gamma, \langle\alpha_i\rangle A}{\Gamma, \langle\alpha_0 \cup \alpha_1\rangle A}\ i \in \{0, 1\}\ \rightsquigarrow\ \begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle\alpha_i\rangle A)(c) \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha_i)(c, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_B, y} \\ \hline
\vee_i & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha_0)(c, y) \vee \mathsf{ST}(\alpha_1)(c, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y} \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle\alpha_0 \cup \alpha_1\rangle A)(c)
\end{array}$$

$$[\,]\,\dfrac{\Gamma, [\alpha][\beta]A}{\Gamma, [\alpha\beta]A}\ \rightsquigarrow\ \begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha][\beta]A)(c) \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c, e)}\}^{\varnothing}, \{\overline{\mathsf{ST}(\beta)(e, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hline
\vee & \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c, e)} \vee \overline{\mathsf{ST}(\beta)(e, d)}\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hline
\forall & \mathsf{HT}(\Gamma)(c), \{\forall z(\overline{\mathsf{ST}(\alpha)(c, z)} \vee \overline{\mathsf{ST}(\beta)(z, d)})\}^{\varnothing}, \mathsf{HT}(A)(d) \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha; \beta]A)(c)
\end{array}$$

$$\langle\rangle\,\dfrac{\Gamma, \langle\alpha\rangle\langle\beta\rangle A}{\Gamma, \langle\alpha\beta\rangle A}\ \rightsquigarrow\ \begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle\alpha\rangle\langle\beta\rangle A)(c) \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha)(c, z), \mathsf{ST}(\alpha)(z, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y, z} \\ \hline
\wedge & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha)(c, z) \wedge \mathsf{ST}(\alpha)(z, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y, z} \\ \hline
\exists & \mathsf{HT}(\Gamma)(c), \{\exists z(\mathsf{ST}(\alpha)(c, z) \wedge \mathsf{ST}(\alpha)(z, y)), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y} \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle\alpha; \beta\rangle A)(c)
\end{array}$$

$$\langle+\rangle_0\,\dfrac{\Gamma, \langle\alpha\rangle A}{\Gamma, \langle\alpha^{+}\rangle A}\ \rightsquigarrow$$

$$\begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle\alpha\rangle A)(c) \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha)(c, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y} \\ \hline
\mathsf{wk} & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha)(c, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y}, \{\mathsf{ST}(\alpha)(c, z), TC(\mathsf{ST}(\alpha))(z, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y, z} \\ \hline
\mathsf{tc} & \mathsf{HT}(\Gamma)(c), \{TC(\mathsf{ST}(\alpha))(c, y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A, y} \\ \hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle\alpha^{+}\rangle A)(c)
\end{array}$$

$$\langle + \rangle_1 \frac{\Gamma, \langle \alpha \rangle \langle \alpha^+ \rangle A}{\Gamma, \langle \alpha^+ \rangle A} \quad \rightsquigarrow$$

$$\begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle \alpha \rangle \langle \alpha^+ \rangle A)(c) \\
\hdashline
 & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha)(c,z), TC(\mathsf{ST}(\alpha))(z,y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A,y,z} \\
\mathsf{wk} & \\
\hline
 & \mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(\alpha)(c,y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A,y}, \{\mathsf{ST}(\alpha)(c,z), TC(\mathsf{ST}(\alpha))(z,y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A,y,z} \\
\mathsf{tc} & \\
\hline
 & \mathsf{HT}(\Gamma)(c), \{TC(\mathsf{ST}(\alpha))(c,y), \mathsf{CT}(A)(y)\}^{\mathbf{x}_A,y} \\
\hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}(\langle \alpha^+ \rangle A)(c)
\end{array}$$

$$[+] \frac{\Gamma, [\alpha]A \qquad \Gamma, [\alpha][\alpha^+]A}{\Gamma, [\alpha^+]A} \quad \rightsquigarrow$$

$$\begin{array}{rc}
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha][\alpha^+]A)(c) \\
\hdashline
 & \mathcal{E}' \quad \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c,f)}\}^{\varnothing}, \{\overline{TC}(\overline{\mathsf{ST}(\alpha)})(f,d)\}^{\varnothing}, \mathsf{HT}(A)(d) \\
\cup & \\
\hline
 & \mathcal{E} \quad \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c,f)}\}^{\varnothing}, \{\mathsf{ST}(\alpha)(c,d), \overline{TC}(\overline{\mathsf{ST}(\alpha)})(f,d)\}^{\varnothing}, \mathsf{HT}(A)(d) \\
\cup & \\
\hline
 & \mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(\alpha)(c,d)}, \overline{\mathsf{ST}(\alpha)(c,f)}\}^{\varnothing}, \{\overline{\mathsf{ST}(\alpha)(c,d)}, \overline{TC}(\overline{\mathsf{ST}(\alpha)})(f,d)\}^{\varnothing}, \mathsf{HT}(A)(d) \\
\overline{TC} & \\
\hline
 & \mathsf{HT}(\Gamma)(c), \{\overline{TC}(\overline{\mathsf{ST}(\alpha)})(c,d)\}^{\varnothing}, \mathsf{HT}(A)(d) \\
\hdashline
 & \mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha^+]A)(c)
\end{array}$$

where $\mathcal{E}, \mathcal{E}'$ derive $\mathsf{HT}(\Gamma)(c), \mathsf{HT}([\alpha]A)(c)$ using wk-steps.

By applying the above translation to each rule of a PDL$^+$ cyclic proof $\mathcal{D}$ of A, we obtain a preproof $\mathsf{HT}(\mathcal{D})(c)$ of $\mathsf{HT}(A)(c)$. The last step, from $\mathsf{HT}(A)(c)$ to $\mathsf{ST}(A)(c)$, follows from Prop. 6.8.

Remark 6.10 (Deeper inference) Observe that HTC can also simulate 'deeper' program rules than are available in LPD$^+$.

For example a rule $\frac{\Gamma, \langle \alpha \rangle \langle \beta_i \rangle A}{\Gamma, \langle \alpha \rangle \langle \beta_0 \cup \beta_1 \rangle A}$ may be simulated too (similarly for [ ]). Thus $\langle a^+ \rangle \langle b \rangle p \to \langle a^+ \rangle \langle b \cup c \rangle p$ admits a *finite* proof in HTC (under ST), rather than a necessarily infinite (but cyclic) one in LPD$^+$.

6.4 Justifying Regularity and Progress

Proposition 6.11 *If $\mathcal{D}$ is regular, then so is* $\mathsf{HT}(\mathcal{D})(c)$.

Proof sketch Notice that each step in $\mathcal{D}$ is translated to a finite derivation in $\mathsf{HT}(\mathcal{D})(c)$. Thus, if $\mathcal{D}$ has only finitely many distinct subproofs, then also $\mathsf{HT}(\mathcal{D})(c)$ has only finitely many distinct subproofs, modulo applications of the substitution rule.

Proposition 6.12 *If $\mathcal{D}$ is progressing, then so is* $\mathsf{HT}(\mathcal{D})(c)$.

Proof sketch We need to show that every infinite branch of $\mathsf{HT}(\mathcal{D})(c)$ has a progressing hypertrace. Since the HT translation is defined stepwise on the individual steps of $\mathcal{D}$, we can associate to each infinite branch $\mathcal{B}$ of $\mathsf{HT}(\mathcal{D})(c)$ a unique infinite branch $\mathcal{B}'$ of $\mathcal{D}$. Since $\mathcal{D}$ is progressing, let $\tau = (F_i)_{i<\omega}$ be a progressing thread along $\mathcal{B}'$. By inspecting the rules of LPD$^+$ (and by definition of progressing thread), for some $k \in \mathbb{N}$, each F_i for $i > k$ has the form: $[\alpha_{i,1}] \cdots [\alpha_{i,n_i}][\alpha^+]A$, for some $n_i \geq 0$. So, for $i > k$, $\mathsf{HT}(F_i)(d_i)$ has the form:

$$\{\overline{\mathsf{ST}(\alpha_{i,1})(c, d_{i,1})}\}^{\varnothing}, \ldots, \{\overline{\mathsf{ST}(\alpha_{i,n_i})(d_{i,n_i-1}, d_{i,n_i})}\}^{\varnothing}, \{\overline{TC}(\overline{\mathsf{ST}(\alpha)})(d_{i,n_i}, d_i)\}^{\varnothing}, \mathsf{HT}(A)(d_i).$$

By inspection of the HT-translation (Definition 6.9) whenever F_{i+1} is an immediate ancestor of F_i in $\mathcal{B}'$, there is a path from the cedent $\{\overline{TC}(\overline{\mathsf{ST}(\alpha)})(d_{i+1,n_{i+1}}, d_{i+1})\}^{\varnothing}$ to the cedent $\{\overline{TC}(\overline{\mathsf{ST}(\alpha)})(d_{i,n_i}, d_i)\}^{\varnothing}$ in the graph of immediate ancestry along $\mathcal{B}$. Thus, since $\tau = (F_i)_{i<\omega}$ is a trace along $\mathcal{B}'$, we have a (infinite) hypertrace of the form $\mathcal{H}_\tau := (\{\Delta_i, \overline{TC}(\overline{\mathsf{ST}(\alpha)})(d_{i,n_i}, d_i)\}^{\varnothing})_{i>k'}$ along $\mathcal{B}$.

By construction $\Delta_i = \varnothing$ for infinitely many $i > k'$, and so $\mathcal{H}_\tau$ has just one infinite trace. Moreover, by inspection of the $[+]$ step in Definition 6.9, this trace progresses in $\mathcal{B}$ every time τ does in $\mathcal{B}'$, and so progresses infinitely often. Thus, $\mathcal{H}$ is a progressing hypertrace. Since the choice of the branch $\mathcal{B}$ of $\mathcal{D}$ was arbitrary, we are done.

6.5 Putting It All Together

We can now finally conclude our main simulation theorem:

Proof of Theorem 6.1 Let A be a PDL$^+$ formula s.t. $\models A$. By the completeness result for LPD$^+$, Theorem 6.4, we have that LPD$^+$ $\vdash_{cyc} A$, say by a cyclic proof $\mathcal{D}$. From here we construct the HTC preproof $\mathsf{HT}(\mathcal{D})(c)$ which, by Propositions 6.11 and 6.12, is in fact a cyclic proof of $\mathsf{HT}(A)(c)$. Finally, we apply Proposition 6.8 to obtain a cyclic HTC proof of $\mathsf{ST}(A)(c)$.

7 Extension by Equality: Simulating Full PDL

We now briefly explain how our main results are extended to the ‘reflexive’ version of TCL, which we denote by TCL$^=$, whose language extends that of TCL by adding atomic formulas of the form $s = t$ and $s \neq t$.

7.1 Hypersequential System with Equality

The calculus HTC$^=$ extends HTC by the rules:

$$= \frac{\boldsymbol{S}, \{\Gamma\}^{\mathbf{x}}}{\boldsymbol{S}, \{t = t, \Gamma\}^{\mathbf{x}}} \qquad \neq \frac{\boldsymbol{S}, \{\Gamma(s), \Delta(s)\}^{\mathbf{x}}}{\boldsymbol{S}, \{\Gamma(s), s \neq t\}^{\mathbf{x}}, \{\Delta(t)\}^{\mathbf{x}}} \tag{7}$$

The notion of immediate ancestry for formulas and cedents is colour-coded in (7) just as we did for HTC in Sect. 4.2. Formally:

Definition 7.1 (*Ancestry for cedents,* HTC$^=$) Let r be a HTC$^=$ inference step, as typeset in Fig. 4 or (7). We say that a cedent S in a premiss of r is an *immediate ancestor* of a cedent S' in the conclusion of r if any of the conditions in Definition 4.2 applies (where in 3. we further ask that r is not $\neq$), or the following holds:

5. r is $\neq$ and S is the (unique) cedent distinguished in the premiss of $\neq$ and S' is the cedent $\{\Gamma(s), s \neq t\}^{\mathbf{x}}$ distinguished in the conclusion of $\neq$.

Definition 7.2 (*Ancestry for formulas,* HTC$^=$) Let r be a HTC$^=$ inference step, as typeset in Fig. 4 or (7). We say that a formula F in a premiss of r is an *immediate ancestor* of a formula F' in the conclusion of r if any of the conditions from Definition 4.3 applies, or the following holds:

(e) r is $\neq$ and $F \in \Delta(s)$ and F' is $s \neq t$.

The resulting notions of (pre)proof, (hyper)trace and progress are as in Definition 4.6. Specifically, we have that in rule $=$ as typeset in (7), no infinite trace can include formula $t = t$. Moreover, in rule $\neq$ no infinite trace can include formulas in $\Delta(t)$, while *all* formulas occurring in $\Delta(s)$ belong to a trace where $s \neq t$ belongs.

Remark 7.3 The simulation of TC_G by HTC also extends to their reflexive system RTC_G, in the presence of equality. For this we must define their operator $RTC(\lambda x, y.A)(s,t) := TC(\lambda x, y.(x = y \vee A))(s,t)$. Semantically, it is correct to set $RTC(A)(s,t)$ to be $s = t \vee TC(A)(s,t)$, but this encoding does not lift to the Cohen-Rowe rules for RTC.

7.2 Extending the Soundness Argument

Understanding that structures interpret $=$ as true equality, a modular adaptation of the soundness argument for HTC, cf. Sect. 5, yields:

Theorem 7.4 (Soundness of $\mathsf{HTC}^=$) *If* $\mathsf{HTC}^= \vdash_{nwf} \boldsymbol{S}$ *then* $\models \boldsymbol{S}$.

Proof Sketch In the soundness argument for HTC, in Lemma 5.4, we must further consider cases for equality as follows:

▷ **Case** $(=)$

$$= \frac{\boldsymbol{S}_{i+1} \; = \; \mathbf{Q}, \{\Gamma\}^{\mathbf{x}}}{\boldsymbol{S}_i \; = \; \mathbf{Q}, \{t = t, \Gamma\}^{\mathbf{x}}}$$

By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \not\models \{\Gamma, t = t\}^{\mathbf{x}}$, i.e., $\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma} \vee t \neq t)$. Since $\mathsf{fv}(t) \cap \mathbf{x} = \varnothing$, this is equivalent to $\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma})$. We set $\rho_{i+1} = \rho_i$ and conclude that $\rho_{i+1} \not\models \{\Gamma\}^{\mathbf{x}}$.

▷ **Case** $(\neq)$

$$\neq \frac{\boldsymbol{S} \; = \; \mathbf{Q}, \{\Gamma(s), \Delta(s)\}^{\mathbf{x}}}{\boldsymbol{S}_i \; = \; \mathbf{Q}, \{\Gamma(s), s \neq t\}^{\mathbf{x}}, \{\Delta(t)\}^{\mathbf{x}}}$$

By assumption, $\rho_i \not\models \mathbf{Q}$ and $\rho_i \not\models \{\Gamma(s), s \neq t\}^{\mathbf{x}}$ and $\rho_i \not\models \{\Delta(t)\}^{\mathbf{x}}$. Thus,

$$\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma}(s) \vee s = t) \quad \text{and} \quad \rho_i \models \forall \mathbf{x}(\bigvee \overline{\Delta}(t))$$

Set $\rho_{i+1} = \rho_i$. If $\rho_i \models \forall \mathbf{x}(\bigvee \overline{\Gamma}(s))$, then $\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma}(s) \vee \bigvee \overline{\Delta}(s))$, and thus $\rho_{i+1} \not\models \{\Gamma(s) \vee \Delta(s)\}^{\mathbf{x}}$. Otherwise, if $\rho_i \models \forall \mathbf{x}(s = t)$, we can safely substitute term t with term s in the second conjunct, obtaining $\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Delta}(s))$. Thus, $\rho_{i+1} \models \forall \mathbf{x}(\bigvee \overline{\Gamma}(s) \vee \bigvee \overline{\Delta}(s))$, and $\rho_{i+1} \not\models \{\Gamma(s) \vee \Delta(s)\}^{\mathbf{x}}$.

For the construction of the 'false trace' in Lemma 5.6 we add the following cases for equality:

▷ **Case** $(=)$. Suppose $\{\Gamma_i\}^{\mathbf{x}_i} = \{t = t, \Gamma\}^{\mathbf{x}}$. By assumption, $\rho^\times \models \forall \mathbf{x}(t \neq t \vee \bigvee \overline{\Gamma})$ and thus $\rho^\times, \delta_{\mathcal{H}} \models t \neq t$ or $\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma}$. Since the first disjunct cannot hold, the trace cannot follow the formula $t = t$. Thus, if $F = C$ for some C occurring in Γ and $\rho^\times, \delta_{\mathcal{H}} \models \overline{C}$, set $F' = F$ and conclude that $\rho^\times, \delta_{\mathcal{H}} \models \overline{C}$.

▷ **Case** $(\neq)$. The hypertrace $\mathcal{H}$ cannot go through the structure $\{\Delta(t)\}^{\mathbf{x}}$, because by hypothesis $\mathcal{H}$ is infinite. Thus, $\{\Gamma_i\}^{\mathbf{x}_i} = \{\Gamma(s), s \neq t\}^{\mathbf{x}}$. By assumption, we have that either:

$$\rho^\times, \delta_{\mathcal{H}} \models \bigvee \overline{\Gamma(s)} \quad \text{or} \quad \rho^\times, \delta_{\mathcal{H}} \models s = t.$$

If $F = C(s)$ for some $C(s)$ in $\Gamma(s)$ and $\rho^{\times}, \delta_{\mathcal{H}} \models \overline{F}$, set $F' = F$. Otherwise, if $\rho^{\times}, \delta_{\mathcal{H}} \not\models \bigvee \overline{\Gamma(s)}$, then $\rho^{\times}, \delta_{\mathcal{H}} \models s = t$ and F is $s \neq t$. At every occurrence of rule $\neq$ all the formulas occurring in $\Delta(s)$ are immediate ancestors of $s \neq t$. Moreover, by assumption $\rho^{\times} \models \forall \mathbf{x}(\bigvee \overline{\Delta}(t))$. By the truth condition associated to $\forall$ and since $\mathbf{x}$ is contained in the domain of $\delta_{\mathcal{H}}$, we have that $\rho^{\times}, \delta_{\mathcal{H}} \models \bigvee \Delta(t)$. Since $\rho^{\times}, \delta_{\mathcal{H}} \models s = t$, we conclude that $\rho^{\times}, \delta_{\mathcal{H}} \models \bigvee \Delta(s)$. Thus, there exists a $D(s) \in \Delta(s)$ such that $\rho^{\times}, \delta_{\mathcal{H}} \models \bigvee D(s)$. Set $F' = D(s)$.

7.3 Completeness for PDL (with Tests)

Turning to the modal setting, PDL may be defined as the extension of PDL^{+} by including a program $A?$ for each formula A. Semantically, we have $A?^{\mathcal{M}} = \{(v, v) : \mathcal{M}, v \models A\}$. From here we may define $\varepsilon := \top?$ and $\alpha^{*} := (\varepsilon \cup \alpha)^{+}$. Again, while it is semantically correct to set $\alpha^{*} = \varepsilon \cup \alpha^{+}$, this encoding does not lift to the standard sequent rules for $*$.

The system LPD is obtained from LPD^{+} by including the rules:

$$\langle ? \rangle\,\frac{\Gamma, A \qquad \Gamma, B}{\Gamma, \langle A? \rangle B} \qquad\qquad [?]\,\frac{\Gamma, \bar{A}, B}{\Gamma, [A?]B}$$

The notion of ancestry for formulas is defined as for LPD^{+} (Definition 6.2) and colour-coded in the rules. The resulting notions of (pre)proof, thread and progress are as in Definition 6.3. We write LPD $\vdash_{cyc} A$ if there is a cyclic proof of A in LPD. Just like for LPD^{+}, a standard encoding of LPD into the μ-calculus yields its soundness and completeness, thanks to known sequent systems for the latter, cf. [21, 24, 27].

Theorem 7.5 (Soundness and completeness, essentially from [21]) *Let A be a PDL formula. $\models A$ iff* LPD $\vdash_{cyc} A$.

Again, a modular adaptation of the simulation of LPD^{+} by HTC, cf. Sect. 6, yields:

Theorem 7.6 (Completeness for *PDL*) *Let A be a PDL formula. If $\models A$ then* HTC$^{=}$ $\vdash_{cyc}$ ST$(A)(c)$.

Proof sketch To show completeness, we must add the the cases for the test rules to Definition 6.9:

$$\langle ? \rangle\,\frac{\Gamma, A \quad \Delta, B}{\Gamma, \Delta, \langle A? \rangle B} \quad\rightsquigarrow\quad \dfrac{\wedge\,\dfrac{\cup\,\dfrac{\text{Pr.}6.8\,\dfrac{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(A)(c)}{\mathsf{HT}(\Gamma)(c), \{\mathsf{ST}(A)(c)\}^{\varnothing}} \qquad \text{inst}\,\dfrac{=\,\dfrac{\text{Pr.}6.8\,\dfrac{\mathsf{HT}(\Delta)(c), \mathsf{HT}(B)(c)}{\mathsf{HT}(\Delta)(c), \{\mathsf{CT}(B)(c)\}^{\mathbf{x}_B}}}{\mathsf{HT}(\Delta)(c), \{c = c, \mathsf{CT}(B)(x)\}^{\mathbf{x}_B}}}{\mathsf{HT}(\Delta)(c), \{c = y, \mathsf{CT}(y)(B)\}^{\mathbf{x}_B, y}}}{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(\Delta)(c), \{c = y, \mathsf{ST}(A)(c), \mathsf{CT}(B)(y)\}^{\mathbf{x}_B, y}}}{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(\Delta)(c), \{c = y \wedge \mathsf{ST}(A)(c), \mathsf{CT}(B)(y)\}^{\mathbf{x}_B, y}}}{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(\Delta)(c), \mathsf{HT}(\langle A? \rangle B)(c)}$$

$$[?]_0\,\frac{\Gamma, \overline{A}}{\Gamma, [A?]B} \quad\rightsquigarrow\quad \dfrac{\vee\,\dfrac{\text{wk}\,\dfrac{\text{Pr.}6.8\,\dfrac{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(\overline{A})(c)}{\mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(A)(c)}\}^{\varnothing}}}{\mathsf{HT}(\Gamma)(c), \{\overline{\mathsf{ST}(A)(c)}\}^{\varnothing}, \mathsf{HT}(B)(d)}}{\mathsf{HT}(\Gamma)(c), \{c \neq d \vee \overline{\mathsf{ST}(A)(c)}\}^{\varnothing}, \mathsf{HT}(B)(d)}}{\mathsf{HT}(\Gamma)(c), \mathsf{HT}([A?]B)(c)}$$

$$[?]_1\ \frac{\Gamma, B}{\Gamma, [A?]B} \quad\rightsquigarrow\quad \begin{array}{c} \vee\,\dfrac{\neq\,\dfrac{\mathsf{HT}(\Gamma)(c), \mathsf{HT}(B)(c)}{\mathsf{HT}(\Gamma)(c), \{c \neq d\}^{\varnothing}, \mathsf{HT}(B)(d)}}{\mathsf{HT}(\Gamma)(c), \{c \neq d \vee \overline{\mathsf{ST}(A)(c)}\}^{\varnothing}, \mathsf{HT}(B)(d)} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathsf{HT}(\Gamma)(c), \mathsf{HT}([A?]B)(c) \end{array}$$

8 Conclusions

In this work we proposed a novel cyclic system HTC for Transitive Closure Logic (TCL) based on a form of hypersequents. We showed a soundness theorem for standard semantics, requiring an argument bespoke to our hypersequents. Our system is cut-free, rendering it suitable for automated reasoning via proof search. We showcased its expressivity by demonstrating completeness for PDL, over the standard translation. In particular, we demonstrated formally that such expressivity is not available in the previously proposed system TC_G of Cohen and Rowe (Theorem 3.7). Our system HTC locally simulates TC_G too (Theorem 4.11).

As far as we know, HTC is the first cyclic system employing a form of *deep inference* resembling *alternation* in automata theory, e.g. wrt. proof checking, cf. Proposition 4.10. It would be interesting to investigate the structural proof theory that emerges from our notion of hypersequent. As hinted at in Example 4.7, HTC exhibits more liberal rule permutations than usual sequents, so we expect their *focussing* and *cut-elimination* behaviours to similarly be richer, cf. [22, 23].

Finally, our work bridges the cyclic proof theories of (identity-free) PDL and (reflexive) TCL. With increasing interest in both modal and predicate cyclic proof theory, it would be interesting to further develop such correspondences.

Acknowledgements This work was supported by a UKRI Future Leaders Fellowship, 'Structure vs Invariants in Proofs', project reference MR/S035540/1, and by the HORIZON-MSCA-2021-PF-01-01 project 'CYclic DErivations for Recursive operators', project ID 101064105.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Berardi, S., Tatsuta, M.: Classical system of Martin-Lof's inductive definitions is not equivalent to cyclic proofs. CoRR. http://arXiv.org/abs/1712.09603 (2017)
2. Blackburn, P., van Benthem, J.: Modal logic: a semantic perspective. In: Blackburn, P., van Benthem, J.F.A.K., Wolter, F. (eds.) Handbook of Modal Logic. Studies in Logic and Practical Reasoning, vol. 3, pp. 1–84. North-Holland, Amsterdam (2007)
3. Blackburn, P., De Rijke, M., Venema, Y.: Modal Logic, vol. 53. Cambridge University Press, Cambridge (2002)
4. Bojańczyk, M., Czerwiński, W.: An Automata Toolbox. Lecture Notes. https://www.mimuw.edu.pl/bojan/paper/automata-toolbox-book (2018)
5. Brotherston, J., Simpson, A.: Sequent calculi for induction and infinite descent. J. Log. Comput. **21**(6), 1177–1216 (2011)

6. Brotherston, J., Distefano, D., Petersen, R.L.: Automated cyclic entailment proofs in separation logic. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) Automated Deduction—CADE-23—23rd International Conference on Automated Deduction, Wroclaw, Poland, July 31–August 5, 2011. Proceedings. Lecture Notes in Computer Science, Vol. 6803, pp. 131–146. Springer, Berlin (2011). https://doi.org/10.1007/978-3-642-22438-6_12
7. Brotherston, J., Gorogiannis, N., Petersen, R.L.: A generic cyclic theorem prover. In: Jhala, R., Igarashi, A. (eds.) Programming Languages and Systems—10th Asian Symposium, APLAS 2012, Kyoto, Japan, December 11–13, 2012. Proceedings. Lecture Notes in Computer Science, Vol. 7705, pp. 350–367. Springer, Berlin (2012). https://doi.org/10.1007/978-3-642-35182-2_25
8. Cohen, L., Rowe, R.N.S.: Uniform inductive reasoning in transitive closure logic via infinite descent. In: Ghica, D.R., Jung, A. (eds.) 27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4–7, 2018, Birmingham, UK. LIPIcs, Vol. 119, pp. 17–11716. Schloss Dagstuhl, Leibniz-Zentrum für Informatik (2018). https://doi.org/10.4230/LIPIcs.CSL.2018.17
9. Cohen, L., Rowe, R.N.S.: Integrating induction and coinduction via closure operators and proof cycles. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Automated Reasoning—10th International Joint Conference, IJCAR 2020, Paris, France, July 1–4, 2020, Proceedings, Part I. Lecture Notes in Computer Science, Vol. 12166, pp. 375–394. Springer, Berlin (2020). https://doi.org/10.1007/978-3-030-51074-9_21
10. Cohen, L., Rowe, R.N.: Non-well-founded proof theory of transitive closure logic. ACM Trans. Comput. Logic **21**(4), 1–31 (2020)
11. Das, A.: On the logical complexity of cyclic arithmetic. Log. Methods Comput. Sci. **16**, 1 (2020). https://doi.org/10.23638/LMCS-16(1:1)2020
12. Das, A., Girlando, M.: Cyclic proofs, hypersequents, and transitive closure logic. In: Blanchette, J., Kovács, L., Pattinson, D. (eds.) Automated Reasoning, pp. 509–528. Springer, Cham (2022)
13. Das, A., Pous, D.: A cut-free cyclic proof system for kleene algebra. In: Automated Reasoning with Analytic Tableaux and Related Methods: 26th International Conference, TABLEAUX 2017, Brasília, Brazil, September 25–28, 2017, Proceedings, pp. 261–277. Springer (2017)
14. Dax, C., Hofmann, M., Lange, M.: A Proof System for the Linear Time μ-Calculus. vol. 4337, pp. 273–284 (2006). https://doi.org/10.1007/11944836_26
15. Grädel, E.: On transitive closure logic. In: Börger, E., Jäger, G., Büning, H.K., Richter, M.M. (eds.) Computer Science Logic, 5th Workshop, CSL '91, Berne, Switzerland, October 7–11, 1991, Proceedings. Lecture Notes in Computer Science, vol. 626, pp. 149–163. Springer, Berlin (1991). https://doi.org/10.1007/BFb0023764
16. Gurevich, Y.: Logic and the Challenge of Computer Science, pp. 1–57. Computer Science Press, Rockville (1988)
17. Immerman, N.: Languages that capture complexity classes. SIAM J. Comput. **16**(4), 760–778 (1987). https://doi.org/10.1137/0216051
18. Kamareddine, F.D. (ed.): Transitive Closure and the Mechanization of Mathematics, pp. 149–171. Springer, Dordrecht (2003)
19. Kozen, D.: A completeness theorem for kleene algebras and the algebra of regular events. In: Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15–18, 1991, pp. 214–225. IEEE Computer Society, Washington, DC (1991)
20. Krob, D.: Complete systems of b-rational identities. Theor. Comput. Sci. **89**(2), 207–343 (1991). https://doi.org/10.1016/0304-3975(91)90395-I
21. Lange, M.: Games for modal and temporal logics. PhD thesis (2003)
22. Marin, S., Miller, D., Volpe, M.: A focused framework for emulating modal proof systems. In: Beklemishev, L.D., Demri, S., Maté, A. (eds.) Advances in Modal Logic 11, Proceedings of the 11th Conference on "Advances in Modal Logic," Held in Budapest, Hungary, August 30–September 2, 2016, pp. 469–488. College Publications, England and Wales. http://www.aiml.net/volumes/volume11/Marin-Miller-Volpe.pdf (2016)
23. Miller, D., Volpe, M.: Focused labeled proof systems for modal logic. In: Logic for Programming, Artificial Intelligence, and Reasoning, pp. 266–280. Springer (2015)
24. Niwiński, D., Walukiewicz, I.: Games for the mu-calculus. Theoret. Comput. Sci. **163**(1), 99–116 (1996). https://doi.org/10.1016/0304-3975(95)00136-0
25. Rowe, R.N.S., Brotherston, J.: Automatic cyclic termination proofs for recursive procedures in separation logic. In: Bertot, Y., Vafeiadis, V. (eds.) Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16–17, 2017, pp. 53–65. ACM, New York (2017). https://doi.org/10.1145/3018610.3018623
26. Simpson, A.: Cyclic arithmetic is equivalent to peano arithmetic. In: Esparza, J., Murawski, A.S. (eds.) Foundations of Software Science and Computation Structures—20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017,

Uppsala, Sweden, April 22–29, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10203, pp. 283–300 (2017). https://doi.org/10.1007/978-3-662-54458-7_17
27. Studer, T.: On the proof theory of the modal mu-calculus. Stud. Logica. **89**(3), 343–363 (2008)
28. Tellez, G., Brotherston, J.: Automatically verifying temporal properties of pointer programs with cyclic proof. In: de Moura, L. (ed.) Automated Deduction—CADE 26—26th International Conference on Automated Deduction, Gothenburg, Sweden, August 6–11, 2017, Proceedings. Lecture Notes in Computer Science, vol. 10395, pp. 491–508. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-63046-5_30
29. Tellez, G., Brotherston, J.: Automatically verifying temporal properties of pointer programs with cyclic proof. J. Autom. Reason. **64**(3), 555–578 (2020)
30. Thomas, W.: Languages, automata, and logic. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 389–455. Springer, Berlin (1997)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.