A Contraction-free and Cut-free Sequent Calculus for Propositional Dynamic Logic
— **Source link** 🔗

Brian Hill, Francesca Poggiolesi

Institutions: HEC Paris, Vrije Universiteit Brussel

Related papers:

- A Contraction-free and Cut-free Sequent Calculus for Propositional Dynamic Logic

- A cut-free simple sequent calculus for modal logic S5

- A purely syntactic and cut-free sequent calculus for the modal logic of provability

- A Sequent Calculus for Propositional Dynamic Logic for Agents

- A coding method for a sequent calculus of propositional logic

A Contraction-free and Cut-free Sequent Calculus for Propositional Dynamic Logic

B Hill, F Poggiolesi

B. Hill

F. Poggiolesi

A Contraction-free and Cut-free Sequent Calculus for Propositional Dynamic Logic

Abstract.   In this paper we present a sequent calculus for propositional dynamic logic built using an enriched version of the tree-hypersequent method and including an infinitary rule for the iteration operator. We prove that this sequent calculus is theoremwise equivalent to the corresponding Hilbert-style system, and that it is contraction-free and cut-free. All results are proved in a purely syntactic way.

Keywords: Contraction-free; Cut-free; Propositional Dynamic Logic; Tree-hypersequent; Proof theory.

1.   Introduction

Propositional dynamic logic, or PDL for short, is a (modal) logic, first studied at the end of the 60's by Engeler [3], Hoare [5] and Yanov [14], that is based on the idea of associating with each program term a of a programming language a modality $[a]$. This means that in PDL we still deal with boxed formulas as we do in modal logic, but the box is no longer empty but filled with program terms.

One could naturally ask: what kind of programs[1] can fill the box of modal logic? Well, we have atomic programs (a_0, a_1, a_2, ...), but also more complex programs that can be constructed by means of the following program operators: the union operator, $\alpha \cup \beta$, that should be interpreted as: "do α or β non-deterministically;" the composition operator, $\alpha \otimes \beta$,[2] that should be interpreted as: "first do α and then do β;" the test operator, $A?$, that should be interpreted as: "verify that A is true;" and finally the iteration operator, α^*, that should be interpreted as: "repeat α a finite number of times."

In PDL, we thus deal with formulas of the following form: $[a]A$, $[\alpha \cup \beta]A$, $[\alpha \otimes \beta]A$, $[B?]A$, $[\alpha^*]A$, each of which should be read as: "A is true after every terminating execution of the program that is in the box."

---

[1] Without risk of confusion, we shall use "programs" and "program terms" interchangeably.

[2] Note that standardly the composition operator is indicated by a semicolon. However since the semicolon plays a central role in the tree-hypersequent method used below, in order to avoid any confusion, we prefer to use the symbol $\otimes$ for the composition operator.

From the point of view of Hilbert systems, propositional dynamic logic is well-defined. Indeed, there are several equivalent axiomatisations of PDL (see for example [4, 7]), each of which is obtained by adding to classical propositional logic: (i) the distribution axiom schema, that now has the form: $[\alpha](A \rightarrow B) \rightarrow ([\alpha]A \rightarrow [\alpha]B)$, for each program α; (ii) modus ponens and the rule of necessitation; and (iii) at least one axiom schema or inference rule for each program operator. What about Gentzen calculi for propositional dynamic logic? In this case the situation is not so positive. As far as we know only two sequent calculi have been proposed; namely, the calculus of Nishimura [8] and the calculus of Wansing [13]. The first calculus exploits classical sequents, treats the iteration operator with a finitary rule and it is proved not to be cut-free. By contrast, Wansing's calculus exploits display sequents, it is cut-free but it does not treat the program operator $*$. Given this situation, a question seems to naturally arise: what happens if we want a sequent calculus which is cut-free and has rules for the iteration operator? In this article we provide an answer to this question. We exploit the tree-hypersequent method, introduced in [9], in order to build a cut-free tree-hypersequent calculus for the full system of PDL. As is often the case, to get this result, there is a price to pay, which is the finiteness of the calculus. Indeed the rule that introduces the program operator $*$ on the right side of the sequent has infinitely many premises. On reflection, this fact may turn out to be unsurprising. Although there is an axiomatisation of PDL which does not contain infinitary rules, from the semantic point of view the $*$ operator is potentially infinitary. But the tree-hypersequent method, though a purely syntactic method since it does not use any explicit semantic elements, fully exploits Kripke semantics and therefore in such a framework the infinitary aspect of the program operator $*$ emerges quite naturally. On the other hand, the tree-hypersequent method has revealed itself to be useful in the case of modal logic, enabling a syntactic proof of the cut-elimination theorem. Even in the application to PDL, it does not disappoint: there is a quite straightforward, syntactic proof of cut-elimination.

We will proceed in the following way: in Section 2 we will explain how the tree-hypersequent method can be adapted to the case of propositional dynamic logic and we will introduce the calculus $CSPDL$; in Section 3 we will show which structural rules are (height-preserving) admissible in $CSPDL$; in Section 4 we will prove that the calculus $CSPDL$ is valid and complete with respect to the Hilbert system $HPDL$; finally, in Section 5, we will prove the cut-elimination theorem for $CSPDL$.

2.   The calculus CSPDL

The tree-hypersequent method is a generalisation of the classical sequent calculus originally built in order to generate sequent calculi for the main systems of modal propositional logic. Let us briefly see how this method can be naturally enriched to taking account of the propositional dynamic case.

The intuition behind the tree-hypersequent calculus is that of internalising in the framework of the Gentzen calculus the structure of the tree-frames of Kripke semantics.[3] In order to understand how this internalisation works, let us consider the following simple tree:

$$
\bullet \qquad \bullet \qquad \bullet
$$
$$
\nwarrow^{a_1} \uparrow^{a_2} \nearrow^{a_3}
$$
$$
\bullet
$$
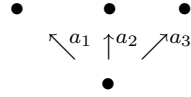
We internalise the structure of this tree-frame in the following way (of course the same technique can be applied to any other tree-frame). The place of the worlds is taken by classical sequents, i.e. in this case we have the four sequents $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4$ that stand for the root of the tree and the three worlds at distance one, respectively. The accessibility relation is simply rendered by using a slash in the following way: $\Gamma_1/\Gamma_2\Gamma_3\Gamma_4$. The separation between worlds that are at the same distance is rendered with a semicolon, i.e. we have the more precise: $\Gamma_1/\Gamma_2; \Gamma_3; \Gamma_4$. Tree-hypersequents were introduced in [10], [11], where they were used to develop sequent calculi for a large family of modal logics including modal logics corresponding to frames with different properties, such as reflexivity, transitivity and symmetry. For the case of PDL we need to internalise the programs a_1, a_2, a_3 associated with the accessibility relation. This is done as follows: $\Gamma_1/a_1\colon \Gamma_2; a_2\colon \Gamma_3; a_3\colon \Gamma_4$.

Enough for the intuitive level. Let us now introduce important notations and definitions (for more details see [4]). The language of dynamic propositional language $\mathcal{L}_{\mathcal{PDL}}$ contains:

A set Φ_0 of propositional atoms

A set Π_0 of atomic programs

Propositional Operators: $\wedge, \neg$

Program Operators: $\otimes, \cup, *$

Mix operators: $?, []$

---

[3]Note that the restriction to tree-frames is not limitative thanks to the unraveling result [2, pp. 62–63]; for more details see [9].

The other connectives as well as the mix operator $<>$ are defined as usual. We follow the standard syntactic conventions: atomic formulas are denoted $p, q, \ldots$, formulas are denoted $A, B, \ldots$, atomic programs are denoted $a, b, \ldots$, and programs are denoted $\alpha, \beta, \ldots$. The set Φ of formulas and the set Π of programs of $\mathcal{L}_{\mathcal{PDL}}$ are defined to be the smallest sets such that:

- $\Phi_0 \subseteq \Phi$

- $\Pi_0 \subseteq \Pi$

- if $A, B \in \Phi$, then $A \wedge B$ and $\neg A \in \Phi$

- if $\alpha, \beta \in \Pi$, then $\alpha \otimes \beta, \alpha \cup \beta$ and $\alpha^* \in \Pi$

- if $A \in \Phi$, then $A? \in \Pi$

- if $\alpha \in \Pi$ and $A \in \Phi$, then $[\alpha]A \in \Phi$.

One axiomatisation of PDL, let us call it $HPDL$, consists of the following axioms:

1. Axioms of propositional logic
2. $[\alpha](A \rightarrow B) \rightarrow ([\alpha]A \rightarrow [\alpha]B)$ (distribution axiom)
3. $[\alpha \cup \beta]A \leftrightarrow [\alpha]A \wedge [\beta]A$
4. $[\alpha \otimes \beta]A \leftrightarrow [\alpha][\beta]A$
5. $[B?]A \leftrightarrow (B \rightarrow A)$
6. $A \wedge [\alpha][\alpha^*]A \leftrightarrow [\alpha^*]A$ (mix axiom)
7. $A \wedge [\alpha^*](A \rightarrow [\alpha]A) \rightarrow [\alpha^*]A$ (induction axiom)

and the following rules of inference:

(MP) From A, $A \rightarrow B$, infer B

(Nec) From A, infer $[\alpha]A$.

In order to introduce tree-hypersequents we adopt the following syntactic conventions: multisets of formulas are denoted $M, N, \ldots$, sequents are denoted $\Gamma, \Delta, \ldots$, and tree-hypersequents are denoted $G, H, \ldots$. For the sake of brevity we will use the following notation: given $\Gamma \equiv M \Rightarrow N$ and $\Pi \equiv P \Rightarrow Q$, we will write:

- B, Γ, A instead of $B, M \Rightarrow N, A$,
- $\Gamma \,\textbf{.}\, \Pi$ instead of $M, P \Rightarrow N, Q$,

- $B, \Gamma . \Pi, A$ instead of $B, M, P \Rightarrow N, Q, A$.

DEFINITION 2.1. The set of sequents (SEQ) is defined as standard. The set of tree-hypersequents (THS) is inductively defined in the following way:

- if $\Gamma \in$ SEQ, then $\Gamma \in$ THS,

- if $\Gamma \in$ SEQ, $a_1, ..., a_n$ are atomic programs, and $G_1, ..., G_n$ are tree-hypersequents, then $\Gamma / a_1 : G_1; ...; a_n : G_n \in$ THS.

Note that instead of writing $a_1 : G_1; ...; a_n : G_n$ we will often adopt the shorter notation $\underline{X}$.

DEFINITION 2.2. The intended interpretation of a tree-hypersequent is inductively defined in the following way:

- $(M \Rightarrow N)^\tau := \bigwedge M \to \bigvee N$

- $(\Gamma / a_1 : G_1; ...; a_n : G_n)^\tau := \Gamma^\tau \vee [a_1] G_1^\tau \vee ... \vee [a_n] G_n^\tau$

In order to display the rules of the calculi, we will use the notation $G[*]$ defined as follows:

DEFINITION 2.3. The set of zoom tree-hypersequents (ZTHS) is inductively defined in the following way:

- $[-] \in$ ZTHS,

- if $G_1, ..., G_n \in$ THS, $a_1, ..., a_n$ are atomic programs, then $[-]/a_1 : G_1; ...; a_n : G_n \in$ ZTHS,

- if $\Gamma \in$ SEQ, $G_2, ..., G_n \in$ THS, $a_1, ..., a_n$ are atomic programs and $G_1[-] \in$ ZTHS, then $\Gamma / a_1 : G_1[-]; ...; a_n : G_n \in$ ZTHS.

DEFINITION 2.4. For any zoom tree-hypersequent $G[-]$, and tree-hypersequent H, we define $G[H]$, the result of substituting H into $G[-]$, as follows:

- if $G[-] = [-]$, then $G[H] = H$

- if $G[-] = [-]/a_1 : G_1; ...; a_n : G_n$ and $H = \Delta / b_1 : J_1; ...; b_m : J_m$, then $G[H] = \Delta / a_1 : G_1; ...; a_n : G_n; b_1 : J_1; ...; b_m : J_m$

- if $G[-] = \Gamma / a_1 : G_1[-], ..., a_n : G_n$, then $G[H] = \Gamma / a_1 : G_1[H], ..., a_n : G_n$

Note that a sequent is a tree-hypersequent so that Definition 2.4 also applies to the case of substituting a sequent into a zoom tree-hypersequent.

Given what we have said up to now, one might wonder: (i) what is the intuitive meaning of the last two definitions? (ii) how are we going to use them? Let us start by answering the first question. Intuitively $G[-]$ can be thought of as a tree-hypersequent G together with one hole $[-]$, where the hole should be understood, metaphorically, as a zoom by means of which we can focus attention on a particular part, $-$, of G. The operation of substitution fills the hole with a sequent or a tree-hypersequent, and therefore allows us to make explicit the particular part in the tree-hypersequent that we want to concentrate our attention on. As concerns the second question, a brief inspection of the calculus $CSPDL$ makes clear the importance of Definitions 2.3 and 2.4. The postulates of the calculus $CSPDL$ are:

Initial Tree-hypersequents

$$G\ [p,\,\Gamma,\,p]$$

Propositional Rules

$$\frac{G[\Gamma, A]}{G[\neg A, \Gamma]}\ \neg A \qquad\qquad \frac{G[A, \Gamma]}{G[\Gamma, \neg A]}\ \neg K$$

$$\frac{G[A, B, \Gamma]}{G[A \wedge B, \Gamma]}\ \wedge A \qquad\qquad \frac{G[\Gamma, A] \qquad G[\Gamma, B]}{G[\Gamma, A \wedge B]}\ \wedge K$$

Modal Rules

$$\frac{G[[b]\, A, \Gamma / (b\colon A, \Sigma / \underline{X})]}{G[[b]\, A, \Gamma / (b\colon \Sigma / \underline{X})]}\ \Box A \qquad\qquad \frac{G[\Gamma / b\colon \Rightarrow A]}{G[\Gamma, [b]\, A]}\ \Box K$$

Program Rules

$$\frac{G[[\beta]\, A, [\gamma]\, A, \Gamma]}{G[[\beta \cup \gamma]\, A, \Gamma]}\ \cup A \qquad\qquad \frac{G[\Gamma, [\beta]\, A] \qquad G[\Gamma, [\gamma]\, A]}{G[\Gamma, [\beta \cup \gamma]\, A]}\ \cup K$$

$$\frac{G[[\beta]\, [\gamma]\, A, \Gamma]}{G[[\beta \otimes \gamma]\, A, \Gamma]}\ \otimes A \qquad\qquad \frac{G[\Gamma, [\beta]\, [\gamma]\, A]}{G[\Gamma, [\beta \otimes \gamma]\, A]}\ \otimes K$$

$$\frac{G[\Gamma, A] \qquad G[B, \Gamma]}{G[[A?]B, \Gamma]}\ ?A \qquad\qquad \frac{G[A, \Gamma, B]}{G[\Gamma, [A?]\, B]}\ ?K$$

$$\frac{G[[\beta^*]\,A, [\beta]^n\,A, \Gamma]}{G[[\beta^*]\,A, \Gamma]}\;{*A} \qquad \frac{G[\Gamma, [\beta]^n\,A]\;\;for\;each\;n < \omega}{G[\Gamma, [\beta^*]\,A]}\;{*K}$$

Cut Rule

$$\frac{G[\Gamma, A] \qquad G[A, \Gamma]}{G[\Gamma]}\;Cut_A$$

Note that in both the rules $*A$ and $*K$, we use the notation $[\beta]^n\,A$, which is inductively defined in the following way:

- $[\alpha]^0\,A := A$
- $[\alpha]^{k+1}\,A := [\alpha]\,[\alpha]^k\,A$

Therefore $[\alpha]^n\,A \equiv \overbrace{[\alpha]\ldots[\alpha]}^{n}\,A$.

Let us make two remarks. The first one concerns the modal rules. Note that these rules only apply to boxed formulas in which the program that occurs in the box is atomic. The second remark concerns the $*K$ rule. Note that this rule has ω-many premisses.

3. Admissibility of the Structural Rules

In this section we will show which structural rules are admissible in the calculus $CSPDL$. Moreover, in order to show that the two rules of contraction are height-preserving admissible, we will show that all the logical, modal, and program rules are height-preserving invertible. In Section 5 it will be proved that the cut-rule is admissible.

DEFINITION 3.1. We define the *complexity* of a formula A in the following inductive way:

- $cmp(p) = 1$,
- $cmp(\neg A) = cmp([a]A) = cmp(A) + 1$,
- $cmp(A \wedge B) = cmp([A?]B) = max(cmp(A), cmp(B)) + 1$,
- $cmp([\alpha \cup \beta]A) = max(cmp([\alpha]A), cmp([\beta]A)) + 1$,
- $cmp([\alpha \otimes \beta]A) = cmp([\alpha][\beta]A) + 1$,
- $cmp([\alpha^*]A) = cmp([\alpha]A) + \omega$.

DEFINITION 3.2. We associate to each proof d in $CSPDL$ an ordinal $h(d)$ — the height — in the standard way. That is, h is inductively defined as follows:

$$d = G[p, \Gamma, p] \,:\, h(d) = 0,$$

$$d = \cfrac{\vdots\ d_i}{\underline{\ldots G'[\Gamma']\ldots}} \quad \text{with } i \in I\colon h(d) = \sup_{i \in I}(h(d_i) + 1).$$

Note that I can generally have 1, 2 or ω elements (see rules p. 5).

DEFINITION 3.3. For any ordinal κ, we write $\vdash^{\langle \kappa \rangle} G$ (respectively, $\vdash^{\langle < \kappa \rangle} G$) or just $^{\langle \kappa \rangle}G$ (resp. $^{\langle < \kappa \rangle}G$), for: "there exists a proof d of G such that $h(d) \leq \kappa$ (resp. $h(d) < \kappa$)."

DEFINITION 3.4. If $\mathcal{R}$ is the rule that allows us to infer G from G', then call the *inverse* of the rule $\mathcal{R}$, written as $\widehat{\mathcal{R}}$, the rule that allows us to infer G' from G.

In the sequent calculus for classical logic, we usually say that a (some) formula(s) is (are) auxiliary in the premise(s) of a rule when the rule operates on that (those) formula(s). In a similar way, we will say that a (some) sequent(s) is (are) auxiliary in the premise(s) of a rule, when the rule concerns that (those) sequent(s). More precisely we will consider as auxiliary those sequents that are displayed in the premise(s) of the rules of the tree-hypersequent calculi.

In the following proofs of the (height-preserving) admissibility of the structural rules and invertibility of the logical, modal and program rules,[4] we will only take into account those cases in which the last applied rule operates on the auxiliary sequent(s) of the rule that we want to show to be admissible or invertible. All the other cases are dealt with easily, as shown in Lemmas 3.13 and 3.14, which are proved at the end of the current section.

LEMMA 3.5. *Tree-hypersequents of the form $G[A, \Gamma, A]$, with A an arbitrary formula, are derivable in $CSPDL$.*

PROOF. By induction on the complexity of A.   ∎

LEMMA 3.6. (***Admissibility of the Structural Rules***) *In $CSPDL$ the following rules are height-preserving admissible:*

---

[4]For a precise definition of these notions, see [12, pp. 65–68].

(i) *the necessitation rule:*

$$\frac{G}{\Rightarrow /a\colon G}\ rn$$

(ii) *the weakening rules:*

$$\frac{G[\Gamma]}{G[A,\Gamma]}\ WA \qquad \frac{G[\Gamma]}{G[\Gamma,A]}\ WK$$

(iii) *the external weakening rule:*

$$\frac{G[\Gamma]}{G[\Gamma/b\colon \Sigma]}\ EA$$

(iv) *the merge rule:*

$$\frac{G[\Delta/(b\colon \Gamma/\underline{X});(b\colon \Pi/\underline{X}')]}{G[\Delta/(b\colon \Gamma\bullet\Pi/\underline{X};\underline{X}')]}\ merge$$

PROOF. By straightforward induction on the height of the derivation of the premise. ∎

LEMMA 3.7. *The logical rules of $CSPDL$ are height-preserving invertible.*

PROOF. The proof, by induction on the height of the derivation of the premise of the rule considered, can be developed in the classical way. Indeed, the only differences — the fact that we are dealing with tree-hypersequents, and the cases where the rule before the logical rule is a modal or a program rule — are dealt with easily. ∎

LEMMA 3.8. *The rules $\Box A$ and $*A$ of $CSPDL$ are height-preserving invertible.*

PROOF. Thanks to the height-preserving admissibility of the weakening rules. ∎

LEMMA 3.9. *The rules $\cup A$, $\cup K$ and $?A$, $?K$ of $CSPDL$ are height-preserving invertible.*

PROOF. The proof, by induction on the height of the derivation of the premise of the rule considered, is analogous to the classical one for the connectives $\vee$ and $\rightarrow$, respectively. ∎

LEMMA 3.10. *The rules $\otimes A$ and $\otimes K$ are height-preserving invertible.*

PROOF. By induction on the height of the derivation of the premise of the rule considered. We only consider the invertibility of the $\otimes K$ rule. The proof of the invertibility of the $\otimes A$ rule is analogous.

If $G[\Gamma, [\beta \otimes \gamma] A]$ is an initial tree-hypersequent, then so is $G[\Gamma, [\beta] [\gamma] A]$. If $G[\Gamma, [\beta \otimes \gamma] A]$ is preceded by a logical rule $\mathcal{R}$, we apply the inductive hypothesis on the premise(s), $G[\Gamma', [\beta \otimes \gamma] A]$ $(G[\Gamma'', [\beta \otimes \gamma] A])$ and we obtain derivation(s), of height less than κ, of $G[\Gamma', [\beta] [\gamma] A]$ $(G[\Gamma'', [\beta] [\gamma] A])$. By applying the rule $\mathcal{R}$, we obtain a derivation of height at most κ of $G[\Gamma, [\beta] [\gamma] A]$.

If $G[\Gamma, [\beta \otimes \gamma] A]$ is of the form $G[\Gamma', [\beta \otimes \gamma] A, [b] B]$ and is concluded by the modal rule $\Box K$ (for the modal rule $\Box A$ the procedure is analogous), we apply the inductive hypothesis on $G[\Gamma', [\beta \otimes \gamma] A/b{:}\Rightarrow B]$ and we obtain a derivation of height less than κ of $G[\Gamma', [\beta] [\gamma] A/b{:}\Rightarrow B]$. By applying the rule $\Box K$, we obtain a derivation of height at most κ of $G[\Gamma', [\beta] [\gamma] A, [b] B]$.

If $G[\Gamma, [\beta \otimes \gamma] A]$ is concluded by one of the program rules, including the $\otimes K$ rule without $[\beta \otimes \gamma] A$ as principal formula, then the procedure is analogous to the one for the logical rules.

Finally, if $G[\Gamma, [\beta \otimes \gamma] A]$ is preceded by the program rule $\otimes K$ and $[\beta \otimes \gamma] A$ is the principal formula, the premise of the last step gives the conclusion. ■

LEMMA 3.11. *The rules $\Box K$ and $*K$ are height-preserving invertible.*

PROOF. By induction on the height of the derivation of the premise of the rule considered. We only consider the invertibility of the $*K$ rule. The proof of the invertibility of the $\Box K$ rule is analogous.

If $G[\Gamma, [\beta^*] A]$ is an initial tree-hypersequent, then so are the premises: $G[\Gamma, [\beta]^n A]$ for all $n \geqslant 0$. If $G[\Gamma, [\beta^*] A]$ is preceded by a logical rule $\mathcal{R}$, we apply the inductive hypothesis on the premise(s), $G[\Gamma', [\beta^*] A]$ $(G[\Gamma'', [\beta^*] A])$ and we obtain derivations, of height less than κ, of $G[\Gamma', [\beta]^n A]$, for all $n \geqslant 0$, $(G[\Gamma'', [\beta]^n A]$, for all $n \geqslant 0)$. By applying the rule $\mathcal{R}$, we obtain derivations of height at most κ of $G[\Gamma, [\beta]^n A]$, for all $n \geqslant 0$.

If $G[\Gamma, [\beta^*] A]$ is of the form $G[\Gamma', [\beta^*] A, [b] B]$ and is concluded by the modal rule $\Box K$ (for the modal rule $\Box A$ the procedure is analogous), we apply the inductive hypothesis on $G[\Gamma', [\beta^*] A/b{:}\Rightarrow B]$ and we obtain derivations of height less than κ of $G[\Gamma', [\beta]^n A/b{:}\Rightarrow B]$, for all $n \geqslant 0$. By applying the rule $\Box K$, we obtain derivations of height at most κ of $G[\Gamma', [\beta]^n A, [b] B]$, for all $n \geqslant 0$.

If $G[\Gamma, [\beta^*] A]$ is concluded by one of the program rules, including the $*K$ rule without the formula $[\beta^*] A$ as principal formula, then the procedure is analogous to the one for the logical rules.

Finally, if $G[\Gamma, [\beta^*]\, A]$ is preceded by the program rule $*K$ and $[\beta^*]\, A$ is the principal formula, the premises of the last step give the conclusion. $\blacksquare$

LEMMA 3.12. *The rules of contraction:*

$$\frac{G[A, A, \Gamma]}{G[A, \Gamma]}\; CA \qquad \frac{G[\Gamma, A, A]}{G[\Gamma, A]}\; CK$$

are height-preserving admissible in $CSPDL$.

PROOF. By induction on the derivation of the premise $G[\Gamma, A, A]$. We only analyse the case of the rule CK. The case of the rule CA is similar.

If $G[\Gamma, A, A]$ is an initial tree-hypersequent, so is $G[\Gamma, A]$. If $G[\Gamma, A, A]$ is preceded by a rule $\mathcal{R}$ which does not have any of the two occurrences of the formula A as principal, we apply the inductive hypothesis on the premise(s) $G[\Gamma', A, A]$ ($G[\Gamma'', A, A]$ or the infinite premises of the $*K$ rule), obtaining derivation(s) of height less than κ of $G[\Gamma', A]$ ($G[\Gamma'', A]$ or the infinite premises of the $*K$ rule). By applying the rule $\mathcal{R}$ we obtain a derivation of height at most κ of $G[\Gamma, A]$.

Now we consider the case where $G[\Gamma, A, A]$ is preceded by a logical or modal or program rule and one of the two occurrences of the formula A is principal. Hence the rule which concludes $G[\Gamma, A, A]$ is a K-rule and we have to analyse the following cases: $\neg K, \wedge K, \Box K, \cup K, \otimes K, ?K, *K$. Since the procedure in all these cases is similar, we will only deal with the most significant ones.

$[\wedge K]$:

$$\frac{{}^{\langle <\kappa\rangle}G[\Gamma, B, B \wedge C] \quad {}^{\langle <\kappa\rangle}G[\Gamma, C, B \wedge C]}{{}^{\langle \kappa\rangle}G[\Gamma, B \wedge C, B \wedge C]}\; \wedge K \qquad \dashrightarrow^{5}$$

$$\frac{\dfrac{{}^{\langle <\kappa\rangle}G[\Gamma, B, B]}{{}^{\langle <\kappa\rangle}G[\Gamma, B]}\; i.h. \quad \dfrac{{}^{\langle <\kappa\rangle}G[\Gamma, C, C]}{{}^{\langle <\kappa\rangle}G[\Gamma, C]}\; i.h.}{{}^{\langle \kappa\rangle}G[\Gamma, B \wedge C]}\; \wedge K$$

---

[5] The symbol $\dashrightarrow$ means: the premise of the right side is preceded by application of one of the lemmas 3.7 - 3.11 on the premise of the left side.

$[\Box K]$:

$$\frac{{}^{\langle <\kappa \rangle}G[\Gamma, [b]\, B/b{:}\Rightarrow B]}{{}^{\langle \kappa \rangle}G[\Gamma, [b]\, B, [b]\, B]} \;\Box K \qquad \dashrightarrow$$

$$\frac{\dfrac{\dfrac{{}^{\langle <\kappa \rangle}G[\Gamma/b{:}\Rightarrow B; b{:}\Rightarrow B]}{{}^{\langle <\kappa \rangle}G[\Gamma/b{:}\Rightarrow B, B]}\; merge}{{}^{\langle <\kappa \rangle}G[\Gamma/b{:}\Rightarrow B]}\; i.h.}{{}^{\langle \kappa \rangle}G[\Gamma, [b]\, B]}\;\Box K$$

$[*K]$:

$$\frac{{}^{\langle <\kappa \rangle}\;\vdots\;\; G[\Gamma, [\beta^*]\, B, [\beta]^n\, B]\;\vdots}{{}^{\langle \kappa \rangle}\qquad G[\Gamma, [\beta^*]\, B, [\beta^*]\, B]}\;*K \qquad \dashrightarrow$$

$$\vdots\;\frac{\dfrac{{}^{\langle <\kappa \rangle}G[\Gamma, [\beta]^n\, B, [\beta]^n\, B]}{{}^{\langle <\kappa \rangle}G[\Gamma, [\beta]^n\, B]}\; i.h.}{{}^{\langle \kappa \rangle}G[\Gamma, [\beta^*]\, B]}\;\vdots\;*K$$

where i.h. stands for inductive hypothesis.   ■

LEMMA 3.13. *Let $G[H]$ be any tree-hypersequent of the calculus $CSPDL$ together with an occurrence of a tree-hypersequent in it, and $G'[H]$ the result of the application of one of the height-preserving admissible rules - rn, WA, WK, EW, merge, CA, CK - on $G[H]$. If for a rule $\mathcal{R}$ we have:*

$$\frac{G[H']}{G[H]}\;\mathcal{R}$$

then it holds that:

$$\frac{G'[H']}{G'[H]}\;\mathcal{R}$$

PROOF. By induction on the form of the tree-hypersequent $G[H]$.   ■

LEMMA 3.14. *Let $G[H]$ be any tree-hypersequent of the calculus $CSPDL$ together with an occurrence of a tree-hypersequent in it, and $G[H']$ the result*

of the application of one of the logical, modal, or program rules on $G[H]$. If for a rule $\mathcal{R}$ we have:

$$\frac{G^{'}[H^{'}]}{G[H^{'}]} \, \mathcal{R}$$

then it holds that:

$$\frac{G^{'}[H]}{G[H]} \, \mathcal{R}$$

PROOF. By induction on the form of the tree-hypersequent $G[H^{'}]$. ∎

4. The Adequacy Theorem

In this section we prove that our calculus *CSPDL* proves exactly the same formulas as its corresponding Hilbert-style system *HPDL*.

We begin with the proof of soundness. This proof is quite straightforward except for the case of the rule $*K$. In order to deal with this case, we introduce the following definition and lemma.

DEFINITION 4.1. Let $\mathcal{F}$ be the set of propositional functions such that:

- $\mathcal{F}_0 = \{-\}$
- $\mathcal{F}_{i+1} = \{B \vee [b]C \mid B \in \Phi, \ b \in \Pi_0, \ C \in \mathcal{F}_i\}$
- $\mathcal{F} = \bigcup_{i < \omega} \mathcal{F}_i$

For a propositional function $f \in \mathcal{F}$ and a formula $A \in \Phi$, $f(A)$ is the formula obtained by substituting A for the dash.

Intuitively the set $\mathcal{F}$ can be thought of as the equivalent, in $\mathcal{L}_{\mathcal{PDL}}$, of the set of zoom tree-hypersequents ZTHS: in fact the translation of any zoom tree-hypersequent is an element of $\mathcal{F}$.

LEMMA 4.2. *The rule:*

$$\frac{\vdash f(B \to [\alpha]^n A) \text{ for each } n < \omega}{\vdash f(B \to [\alpha^*]A)}$$

is derivable in $HPDL$.

PROOF. The proof uses the completeness of PDL with respect to the standard semantics (see [4], for example). In fact, we prove that, for any f, A, B, α, for any state i in any model $\mathfrak{M}$, if $i \vDash_{\mathfrak{M}} f(B \to [\alpha]^n A)$ for all

$n < \omega$, then $i \vDash_{\mathfrak{M}} f(B \to [\alpha^*]A)$. The proof operates by induction on the construction of f. By the interpretation of the $*$ operator, we have the base case: for any state i in any model $\mathfrak{M}$, we have that, if $i \vDash_{\mathfrak{M}} B \to [\alpha]^n A$ for all $n < \omega$, then $i \vDash_{\mathfrak{M}} B \to [\alpha^*]A$. Now suppose that the inductive hypothesis holds for f, and consider $C \vee [b]f(B \to [\alpha]^n A)$. If $i \vDash_{\mathfrak{M}} C \vee [b]f(B \to [\alpha]^n A)$ for every $n < \omega$, then $i \vDash_{\mathfrak{M}} C$ or, for every state j related to i via the accessibility relation for b, $j \vDash_{\mathfrak{M}} f(B \to [\alpha]^n A)$, for all $n < \omega$. Hence, by the inductive hypothesis, either $i \vDash_{\mathfrak{M}} C$ or, for every state j related to i via the accessibility relation for b, $j \vDash_{\mathfrak{M}} f(B \to [\alpha^*]A)$, so $i \vDash_{\mathfrak{M}} C \vee [b]f(B \to [\alpha^*]A)$ as required.

Since, if $i \vDash_{\mathfrak{M}} f(B \to [\alpha]^n A)$ for all $n < \omega$, then $i \vDash_{\mathfrak{M}} f(B \to [\alpha^*]A)$, we have that, if $\vDash_{\mathfrak{M}} f(B \to [\alpha]^n A)$ for all $n < \omega$, then $\vDash_{\mathfrak{M}} f(B \to [\alpha^*]A)$, whence, by completeness, if $\vdash f(B \to [\alpha]^n A)$ for all $n < \omega$, then $\vdash f(B \to [\alpha^*]A)$, as required. ∎

THEOREM 4.3. *If $\vdash G$ in $CSPDL$, then $\vdash (G)^\tau$ in $HPDL$.*

PROOF. By induction on the height of derivations in $CSPDL$. The cases of the finitary rules are easily dealt with. The technique for each consists of the following two steps: first of all, the sequent(s) affected by the rule should be isolated and the corresponding implication proved, then the implication should be transported up all along the tree so that, by modus ponens, the desired result is immediately achieved. Lemma 4.2 deals with the case of the infinitary rule $*K$. ∎

In order to simplify the quite complex proof of completeness, we will firstly prove the following two lemmas.

LEMMA 4.4. *Let $\mathscr{A}, \mathscr{B}, \dots$ denote sequences of program modalities. Then the following two rules:*

$$\frac{G[\mathscr{A}[\alpha]^n A, \Gamma]}{G[\mathscr{A}[\alpha^*]A, \Gamma]} \; PA \qquad \frac{\vdots \quad G[\Gamma, \mathscr{A}[\alpha]^n A] \quad \vdots}{G[\Gamma, \mathscr{A}[\alpha^*]A]} \; PK$$

are admissible in $CSPDL$.

PROOF. By induction on the height of the derivation of the premises. We only analyse the case of the rule PK. The case of the rule PA is similar.

If $\mathscr{A}$ is empty, then the lemma is trivial. Let us consider the case where $\mathscr{A}$ is not empty. We distinguish cases by the last rules applied on the premises of the rule PK.

`Case 1.` For all $i \geqslant 0$, $G[\Gamma, \mathscr{A}[\alpha]^i A]$ are initial tree-hypersequents. In this case the conclusion is also an initial tree-hypersequent.

`Case 2.` Each $G[\Gamma, \mathscr{A}[\alpha]^i A]$ is inferred by a rule in which $\mathscr{A}[\alpha]^i A$ is principal. Therefore the rules are the same. Since $\mathscr{A}$ denotes a sequence of program modalities, the last applied rule with $\mathscr{A}[\alpha]^i A$ as principal formula, can only be a program rule or a modal rule. We distinguish cases.

(2a) Suppose the first program of the sequence $\mathscr{A}$ is an atomic program b and the last applied rule is $\Box K$, then we have the following situation:

$$\cdots \quad \frac{G[\Gamma/b{:}\Rightarrow \mathscr{A}'[\alpha]^i A]}{G[\Gamma, [b]\mathscr{A}'[\alpha]^i A]} \Box K \quad \cdots \quad \rightsquigarrow$$

$$\frac{G[\Gamma/b{:}\Rightarrow \mathscr{A}'[\alpha^*] A]}{G[\Gamma, [b]\mathscr{A}'[\alpha^*] A]} \Box K$$

(2b) Suppose the first program of the sequence $\mathscr{A}$ is the test program and the last applied rule is $?K$, then we have the following situation:

$$\cdots \quad \frac{G[B, \Gamma, \mathscr{A}'[\alpha]^i A]}{G[\Gamma, [B?]\mathscr{A}'[\alpha]^i A]} \, ?K \quad \cdots \quad \rightsquigarrow$$

$$\frac{G[B, \Gamma, \mathscr{A}'[\alpha^*] A]}{G[\Gamma, [B?]\mathscr{A}'[\alpha^*] A]} \, ?K$$

(2c) Suppose the first program of the sequence $\mathscr{A}$ is the composition program (for the union program the procedure is analogous) and the last applied rule is $\otimes K$, then we have the following situation:

$$\cdots \quad \frac{G[\Gamma, [\beta][\gamma]\mathscr{A}'[\alpha]^i A]}{G[\Gamma, [\beta \otimes \gamma]\mathscr{A}'[\alpha]^i A]} \otimes K \cdots \qquad \rightsquigarrow$$

$$\frac{G[\Gamma, [\beta][\gamma]\mathscr{A}'[\alpha^*] A]}{G[\Gamma, [\beta \otimes \gamma]\mathscr{A}'[\alpha^*] A]} \otimes K$$

(2d) Suppose the first program of the sequence $\mathscr{A}$ is the iteration program and the last applied rule is $*K$, then we have the following situation:

$$
\cdots \quad \frac{\vdots \quad G[\Gamma, [\beta]^k \mathscr{A}'[\alpha]^i A] \quad \vdots}{G[\Gamma, [\beta^*]\mathscr{A}'[\alpha]^i A]} \, {}_{*K} \quad \cdots \quad \rightsquigarrow
$$

$$
\frac{\vdots \quad G[\Gamma, [\beta]^k \mathscr{A}'[\alpha^*] A] \quad \vdots}{G[\Gamma, [\beta^*]\mathscr{A}'[\alpha^*] A]} \, {}_{*K}
$$

`Case 3.` None of the $\mathscr{A}[\alpha]^i A$ are principal, but the same rules are applied to the same formulas of the same sequents in the premises. This case is straightforward.

`Case 4.` The rules are not all the same, or they have not all been applied to the same formula of the same sequent in the premises. Proceed in the following way.

Define the relation $\sim$ on the natural numbers as follows. $i \sim j$ iff the last rule applied on $G[\Gamma, \mathscr{A}[\alpha]^i A]$ is the same as the last rule applied on $G[\Gamma, \mathscr{A}[\alpha]^j A]$ and the rules have been applied to the same formula of the same sequent. Note that $\sim$ is an equivalence relation.

Let $S_1, ..., S_m$ be the equivalence classes under $\sim$. Note that since there is a finite number of rules and the tree-hypersequents are finite objects, there is a finite number of equivalence classes. Note also that to each S_k, $1 \leq k \leq m$, is naturally associated a rule and a formula to which the rule has been applied. Let $\mathcal{R}_k$ denote the rule associated with S_k.

For each S_k and for each $i \in S_k$, apply the inverses of the rules $\mathcal{R}_l$, for all $l \neq k$, to the tree-hypersequent $G[\Gamma, \mathscr{A}[\alpha]^i A]$, i.e. the tree-hypersequent associated to the natural number i. Note that thanks to Lemmas 3.10-3.14, the height of the derivations of each tree-hypersequent is preserved.

Now all premises have the same form with their derivations having the same height as previously. Apply the inductive hypothesis on these premises and then apply the rules $\mathcal{R}_1, ..., \mathcal{R}_k$ to obtain a derivation of $G[\Gamma, \mathscr{A}[\alpha^*] A]$. ∎

LEMMA 4.5. *The following rule:*

$$
\frac{A_1, ..., A_n \Rightarrow A}{[\alpha]A_1, ..., [\alpha]A_n \Rightarrow [\alpha]A} \, {}^{RN}
$$

is admissible in CSPDL.

PROOF. By induction on the complexity of the formulas.   ∎

THEOREM 4.6. *If $\vdash \alpha$ in $HPDL$, then $\vdash \Rightarrow \alpha$ in $CSPDL$.*

PROOF. By primary induction on the complexity of the formula α and secondary induction on the height of the proof. The classical axioms and the modus ponens rule are proved as usual, we present the proof of: (i) the distribution axiom; (ii) axioms for programs; (iii) necessitation rules.

(i) Distribution axiom. We distinguish cases depending on the program α that occurs in the box. If α is an atomic program a, then the proof is the following:

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{
          \cfrac{
            [a](A \to B), [a]A \Rightarrow /a\colon A \Rightarrow A \qquad [a](A \to B), [a]A \Rightarrow /a\colon B \Rightarrow B
          }{[a](A \to B), [a]A \Rightarrow /a\colon A, A \to B \Rightarrow B} {\scriptstyle \to A}
        }{[a](A \to B), [a]A \Rightarrow /a\colon A, \Rightarrow B} {\scriptstyle \Box A}
      }{[a](A \to B), [a]A \Rightarrow /a\colon \Rightarrow B} {\scriptstyle \Box A}
    }{[a](A \to B), [a]A \Rightarrow [a]B} {\scriptstyle \Box K}
  }{[a](A \to B) \Rightarrow [a]A \to [a]B} {\scriptstyle \to K}
}{\Rightarrow [a](A \to B) \to ([a]A \to [a]B)} {\scriptstyle \to K}
$$

If α is the test program $B?$, then the proof is the following:

$$
\cfrac{
  \cfrac{
    \cfrac{
      C, [C?](A \to B) \Rightarrow B, C \qquad
      \cfrac{
        A, C \Rightarrow B, C \qquad
        \cfrac{
          A, C \Rightarrow B, A \quad B, A, C \Rightarrow B
        }{A \to B, A, C \Rightarrow B} {\scriptstyle \to A}
      }{A, C, [C?](A \to B) \Rightarrow B} {\scriptstyle ?A}
    }{C, [C?](A \to B), [C?]A \Rightarrow B} {\scriptstyle ?A}
  }{[C?](A \to B), [C?]A \Rightarrow [C?]B} {\scriptstyle ?K}
}{\cfrac{[C?](A \to B) \Rightarrow [C?]A \to [C?]B}{\Rightarrow [C?](A \to B) \to ([C?]A \to [C?]B)} {\scriptstyle \to K}} {\scriptstyle \to K}
$$

If α is the union, composition or iteration program, then we have to use the inductive hypothesis. Let us consider the case of the composition program (the procedure for the union and iteration programs is analogous). So suppose that $\alpha \equiv \beta \otimes \gamma$, we have:

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{
        \cfrac{
          \cfrac{
            \cfrac{
              \Rightarrow [\beta][\gamma](A \to B) \to ([\beta][\gamma]A \to [\beta][\gamma]B)
            }{[\beta][\gamma](A \to B) \Rightarrow [\beta][\gamma]A \to [\beta][\gamma]B} {\scriptstyle \widehat{\to K}}
          }{[\beta][\gamma](A \to B), [\beta][\gamma]A \Rightarrow [\beta][\gamma]B} {\scriptstyle \widehat{\to K}}
        }{[\beta \otimes \gamma](A \to B), [\beta][\gamma]A \Rightarrow [\beta][\gamma]B} {\scriptstyle \otimes A}
      }{[\beta \otimes \gamma](A \to B), [\beta \otimes \gamma]A \Rightarrow [\beta][\gamma]B} {\scriptstyle \otimes A}
    }{[\beta \otimes \gamma](A \to B), [\beta \otimes \gamma]A \Rightarrow [\beta \otimes \gamma]B} {\scriptstyle \otimes K}
  }{[\beta \otimes \gamma](A \to B) \Rightarrow [\beta \otimes \gamma]A \to [\beta \otimes \gamma]B} {\scriptstyle \to K}
}{\Rightarrow [\beta \otimes \gamma](A \to B) \to ([\beta \otimes \gamma]A \to [\beta \otimes \gamma]B)} {\scriptstyle \to K}
$$

Note that in the last two inferences, reading the proof bottom up, we have used the inverse of the rule $\rightarrow K$. The last tree-hypersequent, still reading the proof bottom up, is provable by the inductive hypothesis.

(ii) The proof of the following axioms, $[\alpha \cup \beta]A \leftrightarrow [\alpha]A \wedge [\beta]A$, $[\alpha \cup \beta]A \leftrightarrow [\alpha]A \wedge [\beta]A$ and $[A?]B \leftrightarrow A \rightarrow B$ is trivial. We are going to show the proofs of the mix axiom and of the induction axiom. In these proofs Lemma 4.4 and Lemma 4.5, respectively, will play an important role.

(iia) mix axiom:

$$
\cfrac{
\cfrac{A \Rightarrow A}{[\alpha^*]A \Rightarrow A} \; *A
\qquad
\cfrac{
\cfrac{
\cfrac{[\alpha]^{n+1}A \Rightarrow [\alpha][\alpha]^n A}{[\alpha^*]A \Rightarrow [\alpha][\alpha]^n A} \; *A
}{[\alpha^*]A \Rightarrow [\alpha][\alpha^*]A} \; PK
}
{[\alpha^*]A \Rightarrow A \wedge [\alpha][\alpha^*]A} \; \wedge K
}{\Rightarrow [\alpha^*]A \rightarrow A \wedge [\alpha][\alpha^*]A} \; \rightarrow K
$$

(iib) induction axiom. We start the proof in the following way:

$$
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{A, A \rightarrow [\alpha]A, [\alpha](A \rightarrow [\alpha]A), ..., [\alpha]^{n-1}(A \rightarrow [\alpha]A) \Rightarrow [\alpha]^n A}{A, A \rightarrow [\alpha]A, [\alpha](A \rightarrow [\alpha]A), ..., [\alpha]^{n-1}(A \rightarrow [\alpha]A), [\alpha^*](A \rightarrow [\alpha]A) \Rightarrow [\alpha]^n A} \; WA
}{\vdots} \; *A
}{
\cfrac{
\cfrac{A, [\alpha^*](A \rightarrow [\alpha]A) \Rightarrow [\alpha]^n A}{A, [\alpha^*](A \rightarrow [\alpha]A) \Rightarrow [\alpha^*]A} \; *A
}{\cfrac{A \wedge [\alpha^*](A \rightarrow [\alpha]A) \Rightarrow [\alpha^*]A}{\Rightarrow A \wedge [\alpha^*](A \rightarrow [\alpha]A) \rightarrow [\alpha^*]A} \; \rightarrow K} \; \wedge A
}
} \; *K
$$

Note that we have reached the second last sequent, reading the proof bottom up, by repeated applications of the rule $*A$; this is what the dots stand for. In order to continue the proof we distinguish cases depending on the program α that occurs in the box.

- Let us start by assuming that α is an atomic program a. Then by applying the rule $\rightarrow A$ on $A, A \rightarrow [a]A, [a](A \rightarrow [a]A), ..., [a]^{n-1}(A \rightarrow [a]A) \Rightarrow [a]^n A$, we obtain the axiom $A \Rightarrow A$ and the tree-hypersequent: $[a]A, [a](A \rightarrow [a]A), ..., [a]^{n-1}(A \rightarrow [a]A) \Rightarrow [a]^n A$. We continue the proof as follows:

$$\cfrac{\cfrac{\Rightarrow /a\colon A \Rightarrow A \quad [a]A, [a](A \to [a]A), ..., [a]^{n-1}(A \to [a]A) \Rightarrow /a\colon A, [a]A \Rightarrow [a]^{n-1}A}{\cfrac{[a]A, [a](A \to [a]A), ..., [a]^{n-1}(A \to [a]A) \Rightarrow /a\colon A, A \to [a]A \Rightarrow [a]^{n-1}A}{\cfrac{[a]A, [a](A \to [a]A), ..., [a]^{n-1}(A \to [a]A) \Rightarrow /a\colon A \Rightarrow [a]^{n-1}A}{\cfrac{[a]A, [a](A \to [a]A), ..., [a]^{n-1}(A \to [a]A) \Rightarrow /a\colon \Rightarrow [a]^{n-1}A}{[a]A, [a](A \to [a]A), ..., [a]^{n-1}(A \to [a]A) \Rightarrow [a]^n A} \,\Box K} \,\Box A} \,\Box A} \,\to A$$

By repeated applications (n-times) of passages analogous to the ones above, we reach the axiom:

$$[a]A, ..., [a]^{n-1}(A \to [a]A) \Rightarrow \overbrace{/a\colon A, ..., [a]^{n-2}(A \to [a]A) \Rightarrow /.../a\colon A \Rightarrow A}^{n}$$

- Let us assume that α is a test program $B?$, then we have to prove the tree-hypersequent: $A, A \to [B?]A, ..., [B?]^{n-1}(A \to [B?]A) \Rightarrow [B?]^n A$. We prove it by induction on n. If $n = 1$, then simply:

$$\cfrac{A \Rightarrow A \quad [B?]A \Rightarrow [B?]A}{A, A \to [B?]A \Rightarrow [B?]A} \,\to A$$

Let us suppose that the proof holds for n. We have to show that it holds for $n+1$. We have:

$$\cfrac{\cfrac{\cfrac{A, A \to [B?]A, ..., [B?]^{n-1}(A \to [B?]A) \Rightarrow [B?]^n A}{A, A \to [B?]A, ..., [B?]^{n-1}(A \to [B?]A), [B?]^n(A \to [B?]A) \Rightarrow [B?]^n A} \,WA}{B, A, A \to [B?]A, ..., [B?]^n(A \to [B?]A) \Rightarrow [B?]^n A} \,WA}{A, A \to [B?]A, ..., [B?]^n(A \to [B?]A) \Rightarrow [B?]^{n+1} A} \,?K$$

- Let us assume that α is a composition program $\beta \otimes \gamma$ (for the union and iteration programs the procedure is analogous), then we have to prove the tree-hypersequent: $A, A \to [\beta \otimes \gamma]A, ..., [\beta \otimes \gamma]^{n-1}(A \to [\beta \otimes \gamma]A) \Rightarrow [\beta \otimes \gamma]^n A$. We prove it by induction on n. If $n = 1$, then simply:

$$\cfrac{A \Rightarrow A \quad [\beta \otimes \gamma]A \Rightarrow [\beta \otimes \gamma]A}{A, A \to [\beta \otimes \gamma]A \Rightarrow [\beta \otimes \gamma]A} \,\to A$$

Let us suppose that the proof holds for n. We have to show that it holds for $n+1$. We have:

$$\cfrac{A \Rightarrow A \quad \cfrac{\vdots}{\cfrac{\cfrac{\cfrac{\cfrac{A, A \to [\beta \otimes \gamma]A, ..., [\beta \otimes \gamma]^{n-1}(A \to [\beta \otimes \gamma]A) \Rightarrow [\beta \otimes \gamma]^n A}{[\gamma]A, [\gamma](A \to [\beta \otimes \gamma]A), ..., [\gamma][\beta \otimes \gamma]^{n-1}(A \to [\beta \otimes \gamma]A) \Rightarrow [\gamma][\beta \otimes \gamma]^n A} \,RN}{[\beta][\gamma]A, [\beta][\gamma](A \to [\beta \otimes \gamma]A), ..., [\beta][\gamma][\beta \otimes \gamma]^{n-1}(A \to [\beta \otimes \gamma]A) \Rightarrow [\beta][\gamma][\beta \otimes \gamma]^n A} \,RN}{[\beta][\gamma]A, [\beta][\gamma](A \to [\beta \otimes \gamma]A), ..., [\beta][\gamma][\beta \otimes \gamma]^{n-1}(A \to [\beta \otimes \gamma]A) \Rightarrow [\beta \otimes \gamma]^{n+1} A} \,\otimes K}{[\beta \otimes \gamma]A, [\beta \otimes \gamma](A \to [\beta \otimes \gamma]A), ..., [\beta \otimes \gamma]^n(A \to [\beta \otimes \gamma]A) \Rightarrow [\beta \otimes \gamma]^{n+1} A} \,\otimes A}}{A, A \to [\beta \otimes \gamma]A, ..., [\beta \otimes \gamma]^n(A \to [\beta \otimes \gamma]A) \Rightarrow [\beta \otimes \gamma]^{n+1} A} \,\to A$$

where the dots stand for repeated applications of the rule $\otimes A$.

(iii) rule of necessitation. We distinguish cases depending on the program α that occurs in the box. If α is an atomic program a, then the proof is the following:

$$\frac{\dfrac{\Rightarrow A}{\Rightarrow /a\!:\Rightarrow A}\ rn}{\Rightarrow [a]A}\ \Box K$$

If α is the test program $B?$, then the proof is the following:

$$\frac{\dfrac{\Rightarrow A}{B \Rightarrow A}\ WA}{\Rightarrow [B?]A}\ ?A$$

If α is a union, composition or iteration program, then we have to use the inductive hypothesis. Let us consider the case of the composition program (the procedure for the union and iteration programs is analogous). So suppose that $\alpha = \beta \otimes \gamma$, we have:

$$\frac{\dfrac{\dfrac{\Rightarrow A}{\Rightarrow [\gamma]A}\ i.h.}{\Rightarrow [\beta][\gamma]A}\ i.h.}{\Rightarrow [\beta \otimes \gamma]A}\ \otimes K$$

$\blacksquare$

5. Cut-elimination Theorem

In this section we prove that the cut-rule is admissible in calculus $CSPDL$, as the following theorem states.

THEOREM 5.1. *Let $G[\Gamma, A]$ and $G[A, \Gamma]$ be two tree-hypersequents. If:*

$$\frac{\begin{array}{cc} \vdots\ {\scriptstyle d_1} & \vdots\ {\scriptstyle d_2} \\ G[\Gamma, A] & G[A, \Gamma] \end{array}}{G[\Gamma]}\ cut_A$$

and d_1 and d_2 do not contain any other application of the cut rule, then we can construct a proof of $G[\Gamma]$ without any application of the cut rule.

PROOF. The proof is developed by induction on the complexity of the cut formula (see Definition 3.1), with subinduction on the natural (or Hessenberg) sum of the heights of the derivations of the premises of cut (for a definition of the natural sum of ordinals see, e.g., [12]). We will distinguish cases by the last rule applied on the left premise.

Case 1. $G[\Gamma, A]$ is an initial tree-hypersequent. Then either the conclusion is also an initial tree-hypersequent or it can be obtained by an application of the contraction rule on the right premise.

Case 2. $G[\Gamma, A]$ is inferred by a rule $\mathcal{R}$ in which A is not principal. Then we can have the following situation:[6]

$$
\dfrac{\dfrac{G[\Gamma', A]}{G[\Gamma, A]} \; \mathcal{R} \qquad \vdots \atop G[A, \Gamma]}{G[\Gamma]} \; cut_A
$$

We apply the inverse of the $\mathcal{R}$ rule on $G[A, \Gamma]$ and we obtain $G[A, \Gamma']$. Note that the height is not modified since the rules of the calculus $CSPDL$ are *height-preserving* invertible. We can therefore proceed in the following way:

$$
\dfrac{\dfrac{G[\Gamma', A] \quad G[A, \Gamma']}{G[\Gamma']} \; cut_A}{G[\Gamma]} \; \mathcal{R}
$$

where this cut is eliminable by induction on the sum of the heights of the derivations of the premises of cut.

Case 3. $G[\Gamma, A]$ is inferred by a rule $\mathcal{R}$ in which A is principal. We distinguish three subcases: (3.1.) $\mathcal{R}$ is a logical rule; (3.2.) $\mathcal{R}$ is a modal rule; (3.3.) $\mathcal{R}$ is a program rule. We analyse each of these cases.

Case 3.1. We consider, as an example, the case where the rule before $G[\Gamma, A]$ is $\neg K$. We have:

$$
\dfrac{\dfrac{G[B, \Gamma]}{G[\Gamma, \neg B]} \; \neg K \qquad \vdots \atop G[\neg B, \Gamma]}{G[\Gamma]} \; cut_{\neg B}
$$

---

[6]Note that the rule $\mathcal{R}$ can also have been applied to some sequents Σ, different from Γ, and belonging to the zoom tree-hypersequent $G[*]$. The procedure is analogous to the one considered here.

By applying the inverse of the rule $\neg A$ on $G[\neg B, \Gamma]$, we obtain $G[\Gamma, B]$. We replace the previous cut with the following one which is eliminable by induction on the complexity of the cut formula:

$$\frac{G[\Gamma, B] \quad G[B, \Gamma]}{G[\Gamma]} \; cut_B$$

Case 3.2. $\mathcal{R}$ is $\Box K$ and $A = [b] \, B$. We have the following situation:

$$\frac{\dfrac{G[\Gamma/b{:}\Rightarrow B]}{G[\Gamma, [b] \, B]} \; \Box K \qquad \vdots \quad G[[b] \, B, \Gamma]}{G[\Gamma]} \; cut_{[b]B}$$

We have to consider the last rule $\mathcal{R}'$ of d_2. If there is no rule $\mathcal{R}'$ which introduces $G[[b] \, B, \Gamma]$ because $G[[b] \, B, \Gamma]$ is an initial tree-hypersequent, then we can solve the case as in 1. If $\mathcal{R}'$ is a rule in which $[b] \, B$ is not the principal formula, then we solve the case as in 2. The only problematic case is the case where $\mathcal{R}'$ is $\Box A$ and $[b] \, B$ is the principal formula. We analyse it:[7]

$$\frac{\dfrac{G'[\Gamma/b{:}\Rightarrow B; (b\colon \Delta/\underline{X})]}{G'[\Gamma, [b] \, B/(b\colon \Delta/\underline{X})]} \; \Box K \qquad \dfrac{G'[[b] \, B, \Gamma/(b\colon B, \Delta/\underline{X})]}{G'[[b] \, B, \Gamma/(b\colon \Delta/\underline{X})]} \; \Box A}{G'[\Gamma/(b\colon \Delta/\underline{X})]} \; cut_{[b]B}$$

We reduce to:

$$\frac{\dfrac{G'[\Gamma, [b] \, B/(b\colon \Delta/\underline{X})]}{G'[\Gamma, [b] \, B/(b\colon B, \Delta/\underline{X})]} \; WA \qquad G'[[b] \, B, \Gamma/(b\colon B, \Delta/\underline{X})]}{G'[\Gamma/(b\colon B, \Delta/\underline{X})]} \; cut_{[b]B}$$

$$\frac{\dfrac{G'[\Gamma/b{:}\Rightarrow B; (b\colon \Delta/\underline{X})]}{G'[\Gamma/(b\colon \Delta, B/\underline{X})]} \; merge \qquad G'[\Gamma/(b\colon B, \Delta/\underline{X})]}{G'[\Gamma/(b\colon \Delta/\underline{X})]} \; cut_B$$

---

[7]Note that $G'[\Gamma, [b] \, B/(b\colon \Delta/\underline{X})]$ $(G'[[b] \, B, \Gamma/(b\colon \Delta/\underline{X})])$ is just another way of writing $G[\Gamma, [b] \, B]$ $(G[[b] \, B, \Gamma])$.

where the first cut is eliminable by induction on the sum of the heights of the derivations of the premises of cut and the second cut is eliminable by induction on the complexity of the cut formula.

`Case 3.3.` In this subcase we can have several situations depending on the program that appears in the box. We analyse each of these situations by supposing that the right premise has also been introduced by a rule that has A has principal formula. The other cases can be developed as in point 3.2.

$\cup$:

$$\cfrac{\cfrac{G[\Gamma,[\beta]\,B] \quad G[\Gamma,[\gamma]\,B]}{G[\Gamma,[\beta\cup\gamma]\,B]}\ \cup K \quad \cfrac{G[[\beta]\,B,[\gamma]\,B,\Gamma]}{G[[\beta\cup\gamma]\,B,\Gamma]}\ \cup A}{G[\Gamma]}\ cut_{[\beta\cup\gamma]B}$$

We reduce to:

$$\cfrac{G[\Gamma,[\beta]\,B] \quad \cfrac{\cfrac{\cfrac{G[\Gamma,[\gamma]\,B]}{G[[\beta]\,B,\Gamma,[\gamma]\,B]}\ WA \quad G[[\beta]\,B,[\gamma]\,B,\Gamma]}{G[[\beta]\,B,\Gamma]}\ cut_{[\gamma]B}}{}}{G[\Gamma]}\ cut_{[\beta]B}$$

where both cuts are eliminable by induction on the complexity of the cut formula.

$\otimes$:

$$\cfrac{\cfrac{G[\Gamma,[\beta]\,[\gamma]\,B]}{G[\Gamma,[\beta\otimes\gamma]\,B]}\ \otimes K \quad \cfrac{G[[\beta]\,[\gamma]\,B,\Gamma]}{G[[\beta\otimes\gamma]\,B,\Gamma]}\ \otimes A}{G[\Gamma]}\ cut_{[\beta\otimes\gamma]B}$$

We reduce to:

$$\cfrac{G[\Gamma,[\beta]\,[\gamma]\,B] \quad G[[\beta]\,[\gamma]\,B,\Gamma]}{G[\Gamma]}\ cut_{[\beta][\gamma]B}$$

where this cut is eliminable by induction on the complexity of the cut formula.

?:

$$\cfrac{\cfrac{G[C,\Gamma,B]}{G[\Gamma,[C?]\,B]}\ ?A \quad \cfrac{G[\Gamma,C] \quad G[B,\Gamma]}{G[[C?]\,B,\Gamma]}\ ?K}{G[\Gamma]}\ cut_{[C?]B}$$

We reduce to:

$$
\cfrac{
  G[\Gamma, C] \qquad
  \cfrac{
    G[C, \Gamma, B] \qquad
    \cfrac{G[B, \Gamma]}{G[C, B, \Gamma]} \; WA
  }{G[C, \Gamma]} \; cut_B
}{G[\Gamma]} \; cut_C
$$

where both cuts are eliminable by induction on the complexity of the cut formula.

$*$:

$$
\cfrac{
  \cfrac{\vdots \quad G[\Gamma, [\beta]^n B] \quad \vdots}{G[\Gamma, [\beta^*] B]} \; *K
  \qquad
  \cfrac{G[[\beta^*] B, [\beta]^n B, \Gamma]}{G[[\beta^*] B, \Gamma]} \; *A
}{G[\Gamma]} \; cut_{[\beta^*]B}
$$

We reduce to:

$$
\cfrac{
  G[\Gamma, [\beta]^n B] \qquad
  \cfrac{
    \cfrac{G[\Gamma, [\beta^*] B]}{G[[\beta]^n B, \Gamma, [\beta^*] B]} \; WA
    \qquad
    G[[\beta^*] B, [\beta]^n B, \Gamma]
  }{G[[\beta]^n B, \Gamma]} \; cut_{[\beta^*]B}
}{G[\Gamma]} \; cut_{[\beta]^n B}
$$

where the first cut is eliminable by induction on the sum of the heights of the derivations of the premises of cut and the second cut is eliminable by induction on the complexity of cut formula. ∎

6. Conclusion

We have presented a sequent calculus for propositional dynamic logic. This calculus enjoys many attractive properties: all the structural rules, including the contraction and cut rules, are (height-preserving) admissible, the logical, modal and program rules are height-preserving invertible, and the cut-elimination proof exploits the standard syntactic procedure. On the other hand the calculus is infinitary: the rule that introduces the program operator $*$ on the right side of the sequent has infinitely many premises.

Given this situation, the first future task should be to find a sequent calculus that enjoys the properties of $CSPDL$ whilst being finitary. This task is far from trivial, as attested by the existing literature on this problem and similar problems. On the one hand, there is Nishimura's attempt to find

a finitary calculus for PDL: his calculus is not cut-free [8]. On the other hand, there is the growing literature on sequent calculi for common knowledge, which is informative insofar as the common knowledge operator is quite similar to the iteration operator of propositional dynamic logic, both semantically and axiomatically. There the results are very mitigated. The best calculus which has been achieved with a finitary rule mimicking the finitary Hilbert axiomatisation (the common knowledge equivalent of $HPDL$) has a partial cut-elimination theorem, which can only be proved using semantic methods, but no full, syntactically proven cut-elimination theorem [6]. Moreover, the only finitary calculus with cut-elimination employs a variant of our infinitary rule $*K$, with the set of premises limited to those with n less than some finite bound which depends on the conclusion of the rule [1]; once again the proof of cut-elimination is based on semantic completeness. It therefore would not be surprising if similar types of limitation also held for the program operator $*$ of dynamic logic.

These reflections just serve to emphasise the depth of a possible and important direction of research.

References

[1] ALBERUCCI, LUCA, and GERHARD JÄGER, 'About cut elimination for logics of common knowledge', *Annals of Pure and Applied Logic*, 133 (2005), 1–3, 73–99.

[2] BLACKBURN, PATRICK, MAARTEN DE RIJKE, and YDE VENEMA, *Modal Logic*, Cambridge University Press, Cambridge, 2001.

[3] ENGELER, ERWIN, 'Algorithmic properties of structures', *Mathematical Systems Theory*, 1 (1967), 183–195.

[4] HAREL, DAVID, DEXTER KOZEN, and JERZY TIURYN, *Dynamic Logic*, MIT Press, Cambridge, 2000.

[5] HOARE, CHARLES ANTHONY RICHARD, 'An axiomatic basis for computer programming', *Communications of the ACM*, 12 (1969), 576–580.

[6] JÄGER, GERHARD, MATHIS KRETZ, and THOMAS STUDER, 'Cut-free common knowledge', *Journal of Applied Logic*, 5 (2007), 4, 681–689.

[7] KNIJNENBURG, PETER M. W., and JAN VAN LEEUWEN, 'On models for propositional dynamic logic', *Theoretical Computer Science*, 91 (1991), 181–203.

[8] NISHIMURA, HIROKAZU, 'Sequential method in propositional dynamic logic', *Acta Informatica*, 12 (1979), 377–400.

[9] POGGIOLESI, FRANCESCA, *Sequent Calculi for Modal Logic*, Ph.D Thesis, Florence, 2008.

[10] POGGIOLESI, FRANCESCA, 'The method of tree-hypersequent for modal propositional logic', in D. Makinson, J. Malinowski, and H. Wansing, (eds.), *Trends in logic: Towards mathematical philosophy*, Springer, 2009, pp. 31–51.

[11] POGGIOLESI, FRANCESCA, *Reflecting the semantic features of S5 at the syntactic level*, SILFS conference proceedings, Forthcoming, 2010.

[12] TROELSTRA, ANNE SJERP, and HELMUT SCHWICHTENBERG, *Basic Proof Theory*, Cambridge University Press, Cambridge, 1996.

[13] WANSING, HEINRICH, *Displaying Modal Logic*, Kluwer Academic Publisher, Dordrecht/Boston/London, 1998.

[14] YANOV, JOSEPH, 'On equivalence of operator schemes', *Problems of Cybernetic*, 1 (1959), 1–100.

FRANCESCA POGGIOLESI
Center of Logic and Philosophy of Science (CLFW),
Vrije Universiteit Brussel, Pleinaan 2,
1050, Brussels, Belgium
poggiolesi@gmail.com

BRIAN HILL
HEC Paris and IHPST (CNRS / Paris 1 / ENS)
1 rue de la Libération
78351 Jouy-en-Josas, France
brian@brian-hill.org