



HAL
open science

Sequential composition of propositional Krom logic programs

Christian Antic

► **To cite this version:**

| Christian Antic. Sequential composition of propositional Krom logic programs. 2023. hal-04023753v1

HAL Id: hal-04023753

<https://hal.science/hal-04023753v1>

Preprint submitted on 10 Mar 2023 (v1), last revised 21 Oct 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SEQUENTIAL COMPOSITION OF PROPOSITIONAL KROM LOGIC PROGRAMS

CHRISTIAN ANTIĆ
CHRISTIAN.ANTIC@ICLOUD.COM
VIENNA UNIVERSITY OF TECHNOLOGY
VIENNA, AUSTRIA

ABSTRACT. The author has recently introduced the sequential composition of propositional logic programs. This paper studies composition in the Krom fragment from an algebraic point of view. In a broader sense, this paper is a further step towards an algebraic theory of logic programming.

1. INTRODUCTION

Rule-based reasoning is an essential part of human intelligence prominently formalized in artificial intelligence research via logic programs which are formal constructs written in a rule-like sublanguage of predicate logic (cf. Apt, 1990, 1997; Hodges, 1994; Lloyd, 1987; Makowsky, 1987; Sterling & Shapiro, 1994).

The sequential composition of propositional logic programs has been recently introduced by Antić (2021c) and extended to answer set programming by Antić (2021b).

In this paper, we study the sequential composition operation in the Krom fragment (Krom, 1967) consisting only of facts, and rules of the simple form $a \leftarrow b$. Although the class of Krom programs is very restricted, the algebraic theory of composition turns out to be still non-trivial in that setting. This is due to the fact that we can identify every such program with a graph where each rule $a \leftarrow b$ corresponds to an edge from b to a which shows that propositional Krom logic programs are, despite their simplicity, non-trivial objects.

A characteristic feature of the Krom fragment is that sequential composition distributes over union, which simplifies matters considerably; this is not the case for arbitrary programs (cf. Antić, 2021c, Example 8). Moreover, the proper rules operator selecting the non-facts of a program is a homomorphism with respect to composition in the Krom fragment which again makes things easier.

In a broader sense, this paper is a further step towards an algebraic theory of logic programming (Antić, 2021c, 2021b, 2023b, 2021a, 2023a).

2. PROPOSITIONAL KROM LOGIC PROGRAMS

This section recalls propositional Krom logic programs by mainly following the lines of Antić (2021c).

Let A be a finite alphabet of propositional atoms. A (*propositional Krom logic*) *program* over A is a finite set of *rules* of two forms, *facts* of the form $a \in A$, and *proper rules* of the form $a \leftarrow b$, for $a, b \in A$. We denote the set of all Krom programs over A by \mathbb{K}_A or simply by \mathbb{K} in case A is understood from the context.

It will be convenient to define $h(a \leftarrow b) := \{a\}$ and $h(a) := \{a\}$ (*head*), and $b(a \leftarrow b) := \{b\}$ and $b(a) := \emptyset$ (*body*), extended to programs by $h(K) := \bigcup_{r \in K} h(r)$ and $b(K) := \bigcup_{r \in K} b(r)$. Moreover, we define the *dual* of K by

$$K^d := f(K) \cup \{b \leftarrow a \mid a \leftarrow b \in K\}.$$

An *interpretation* is any subset of A . We define the *entailment relation*, for every interpretation I , inductively as follows: (i) for an atom a , $I \models a$ iff $a \in I$; (ii) for a proper rule, $I \models a \leftarrow b$ iff $I \models b$ implies $I \models a$; and, finally, (iii) for a propositional Krom program K , $I \models K$ iff $I \models r$ holds for each rule $r \in K$. In case $I \models K$, we call I a *model* of K . The set of all models of K has a least element with respect to set inclusion called the *least model* of K and denoted by $\mathcal{LM}(K)$.

3. SEQUENTIAL COMPOSITION

This section recalls the sequential composition of propositional Krom logic programs and adds some new observations.

In the rest of the paper, K, L, M, N denote propositional Krom logic programs. The (*sequential*) *composition* (Antić, 2021c) of K and L is given by

$$K \circ L := f(K) \cup \{a \in A \mid a \leftarrow b \in K, b \in L\} \cup \{a \leftarrow b \mid a \leftarrow c \in K, c \leftarrow b \in L\}.$$

We simply write KL in case the composition operation is understood.

Define the *unit program* over A by

$$1_A := \{a \leftarrow a \mid a \in A\}.$$

As usual, we often omit A and write 1 .

We have

$$h(K) = KA \quad \text{and} \quad b(K) = p(K)^d A.$$

Moreover, we have

$$(1) \quad IK = I, \quad \text{for every interpretation } I,$$

that is, every interpretation is a *left zero* in \mathbb{K} .

Every permutation $\pi : A \rightarrow A$ corresponds to the *permutation program*

$$\pi := \{\pi(a) \leftarrow a \mid a \in A\}.$$

The space of all permutation programs Π forms a subgroup of \mathbb{K} , where every π has the *inverse* π^d , that is,

$$\pi\pi^d = \pi^d\pi = 1.$$

We have the following identities (the identities (3)–(6) are shown in Antić (2021c, Theorem 12)):

$$\begin{aligned}
 (2) \quad & K = f(K) \cup p(K) \\
 (3) \quad & K(LM) = (KL)M \\
 (4) \quad & K1 = 1K = K \\
 (5) \quad & K(L \cup M) = KL \cup KM \\
 (6) \quad & (L \cup M)K = LK \cup MK \\
 (7) \quad & \emptyset K = \emptyset \\
 (8) \quad & K\emptyset = f(K) \\
 (9) \quad & p(K \cup L) = p(K) \cup p(L) \\
 (10) \quad & p(KL) = p(K)p(L) \\
 (11) \quad & p(1) = 1 \\
 (12) \quad & p(\emptyset) = \emptyset \\
 (13) \quad & f(K \cup L) = f(K) \cup f(L) \\
 (14) \quad & f(KL) = f(K) \cup p(K)f(L) \\
 (15) \quad & f(1) = \emptyset \\
 (16) \quad & f(\emptyset) = \emptyset.
 \end{aligned}$$

Equations (3)–(8) show that the set of all proper programs together with the operations \cup and \circ and constants \emptyset and 1 forms a *semiring*, and $K\emptyset \neq \emptyset$ is the only reason why the space of all programs (possibly containing facts) fails to be a semiring. Moreover, equations (3)–(12) show that the space of all programs together with \cup , \circ , p and constants \emptyset , 1 form a *quemiring* (Elgot, 1976). Finally, equations (9)–(12) show that the proper rule operator p is a *semiring homomorphism* which is essential in the rest of the paper.¹

The next result provides a concise formula for the computation of the facts of a sequence of program compositions which will be often used below.

Theorem 1. *Given programs $K_1, \dots, K_n \in \mathbb{K}$, $n \geq 1$, we have*

$$(17) \quad f(K_1 \dots K_n) = f(K_1) \cup \bigcup_{i=1}^{n-1} (p(K_1 \dots K_i) f(K_{i+1})).$$

¹This is one of the distinguished features of the Krom fragment since for arbitrary propositional programs P, R we usually have $p(PR) \neq p(P)p(R)$ (cf. Antić, 2021c, §4.4).

Proof. By induction on n . The induction base $n = 1$ holds trivially. For the induction step, we compute

$$\begin{aligned}
f(K_1 \dots K_{n+1}) &\stackrel{(14)}{=} f(K_1) \cup p(K_1)f(K_2 \dots K_{n+1}) \\
&\stackrel{IH}{=} f(K_1) \cup p(K_1) \left(f(K_2) \cup \bigcup_{i=2}^n (p(K_2 \dots K_i)f(K_{i+1})) \right) \\
&\stackrel{(5)}{=} f(K_1) \cup p(K_1)f(K_2) \cup \bigcup_{i=2}^n (p(K_1)p(K_2 \dots K_i)f(K_{i+1})) \\
&\stackrel{(10)}{=} f(K_1) \cup p(K_1)f(K_2) \cup \bigcup_{i=2}^n (p(K_1 K_2 \dots K_i)f(K_{i+1})) \\
&= f(K_1) \cup \bigcup_{i=1}^{n-1} (p(K_1 \dots K_i)f(K_{i+1})).
\end{aligned}$$

□

We say that K and L *commute* iff $KL = LK$ which can be characterized by

$$\begin{aligned}
f(KL) = f(LK) &\quad \text{iff} \quad f(K) \cup p(K)f(L) = f(L) \cup p(L)f(K), \\
p(KL) = p(LK) &\quad \text{iff} \quad p(K)p(L) = p(L)p(K).
\end{aligned}$$

We call K *idempotent* iff $KK = K$, which is characterized by

$$f(K) = f(KK) \stackrel{(14)}{=} f(K) \cup p(K)f(K) \quad \text{and} \quad p(K) = p(KK) \stackrel{(10)}{=} p(K)p(K),$$

where the first identity holds iff

$$p(K)f(K) \subseteq f(K).$$

We call L a *quasi-inverse* of K iff $K = KLK$, in which case we call K *quasi-invertible*.² Notice that in case L is a quasi-inverse of K , the product KL is idempotent, that is,

$$(KL)^2 = KLKL = KL.$$

We can characterize quasi-invertible programs as follows. By (10) and (14), we have

$$\begin{aligned}
K = KLK &\quad \text{iff} \quad f(K) = f(KLK) = f(K) \cup p(K)f(L) \cup p(K)p(L)f(K), \\
p(K) &= p(KLK) = p(K)p(L)p(K).
\end{aligned}$$

The first line is equivalent to

$$p(K)f(L) \subseteq f(K) \quad \text{and} \quad p(K)p(L)f(K) \subseteq f(K).$$

4. ALGEBRAIC SEMANTICS

In this section, we study the semantics of propositional Krom logic programs from an algebraic point of view analogous to Antić (2021c, §6).

Define the *Kleene star* and *plus* of K by

$$K^* := \bigcup_{n \geq 0} K^n \quad \text{and} \quad K^+ := KK^* = K^*K.$$

²In the semigroup literature, this property is usually called “regular” (cf. Howie, 2003).

Finally, define the *omega* operation by

$$K^\omega := f(K^+) \stackrel{(8)}{=} K^+ \emptyset.$$

We have by Antić's (2021c, Theorem 40)

$$\mathcal{LM}(K) = K^\omega.$$

We now wish to compute K^ω . Notice that computing the omega of an idempotent program K satisfying $K^2 = K$ is trivial since

$$(18) \quad K^\omega = f(K).$$

In general, we compute

$$\begin{aligned} K^\omega &= f(K^+) \\ &= \left(\bigcup_{n \geq 1} K^n \right) \emptyset \\ &= \bigcup_{n \geq 1} (K^n \emptyset) \\ &= \bigcup_{n \geq 1} f(K^n) \\ &\stackrel{(17)}{=} \bigcup_{n \geq 1} \left(f(K) \cup \bigcup_{i=1}^{n-1} (p(K^i) f(K)) \right) \\ &\stackrel{(10)}{=} f(K) \cup \bigcup_{n \geq 1} \bigcup_{i=1}^{n-1} (p(K)^i f(K)) \\ &= \bigcup_{n \geq 1} \bigcup_{i=0}^{n-1} (p(K)^i f(K)). \end{aligned}$$

The same formula has been derived in Antić (2023a) by other means.

We can compute the omega of the composition of two programs by

$$\begin{aligned} (KL)^\omega &= \bigcup_{n \geq 1} f((KL)^n) \\ &\stackrel{(17)}{=} \bigcup_{n \geq 1} \left[f(KL) \cup \bigcup_{i=1}^{n-1} (p(KL)^i f(KL)) \right] \\ &\stackrel{(14)}{=} \bigcup_{n \geq 1} \left[f(KL) \cup \bigcup_{i=1}^{n-1} (p((KL)^i) (f(K) \cup p(K) f(L))) \right] \\ &\stackrel{(5)}{=} \bigcup_{n \geq 1} \left[f(KL) \cup \bigcup_{i=1}^{n-1} (p((KL)^i) f(K) \cup p((KL)^i K) f(L)) \right] \\ &= \bigcup_{n \geq 1} \bigcup_{i=0}^{n-1} (p((KL)^i) f(K) \cup p((KL)^i K) f(L)), \end{aligned}$$

where the last identity follows from

$$p((KL)^0)f(K) \cup p((KL)^0K)f(L) \stackrel{(10),(11),(14)}{=} f(KL).$$

Notice that in case L is a quasi-inverse of K , we have $K = KLK$ and thus

$$(KL)^\omega \stackrel{(18)}{=} f(KL) \stackrel{(14)}{=} f(K) \cup p(K)f(L).$$

5. FUTURE WORK

A major line of future research is to lift the results of this paper from the Krom fragment to all propositional logic programs which is non-trivial given that the composition of the latter is not associative and does not distribute over union (cf. Antić, 2021c, Example 10); even more difficult is the extension to the class of answer set programs (Gelfond & Lifschitz, 1991) containing negation as failure in rule bodies since the composition of answer set programs is rather complicated (Antić, 2021b).

Since the space of all propositional Krom logic programs forms a monoid with respect to sequential composition, and almost³ a semiring with respect to composition and union, we can ask all kinds of algebraic questions. A particularly interesting one seems to be: What are “prime” propositional Krom logic programs? Another one is how Green’s (1951) relations \mathcal{L} , \mathcal{R} , \mathcal{J} are characterized in the Krom fragment (cf. Antić, 2023b).

6. CONCLUSION

This paper studied the sequential composition of propositional Krom logic programs. We showed that the space of all such programs has nice algebraic properties. In a broader sense, this paper is a further step towards an algebraic theory of logic programming.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The manuscript has no data associated.

REFERENCES

- Antić, C. (2021a). On syntactically similar logic programs and sequential decompositions. <https://arxiv.org/pdf/2109.05300.pdf>.
- Antić, C. (2021b). Sequential composition of answer set programs. <https://arxiv.org/pdf/2104.12156v2.pdf>.
- Antić, C. (2021c). Sequential composition of propositional logic programs. <https://arxiv.org/pdf/2009.05774v4.pdf>.
- Antić, C. (2023a). Algebraic characterizations of least model and uniform equivalence of propositional Krom logic programs. <https://arxiv.org/pdf/2302.04664.pdf>.
- Antić, C. (2023b). Sequential decomposition of propositional logic programs. to appear.
- Apt, K. R. (1990). Logic programming. In van Leeuwen, J. (Ed.), *Handbook of Theoretical Computer Science*, Vol. B, pp. 493–574. Elsevier, Amsterdam.

³Recall that the only reason for the space of all propositional Krom programs to form a semiring is that in case K contains facts, $K\emptyset = f(K) \neq \emptyset$.

- Apt, K. R. (1997). *From Logic Programming to Prolog*. C.A.R. Hoare Series. Prentice Hall, Prentice Hall, Englewood Cliffs, NJ.
- Elgot, C. C. (1976). Matrical theories. *Journal of Algebra*, 42(2), 391–422.
- Gelfond, M., & Lifschitz, V. (1991). Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3-4), 365–385.
- Green, J. A. (1951). On the structure of semigroups. *Annals of Mathematics*, 54(1), 163–172.
- Hodges, W. (1994). Logical features of Horn clauses. In Gabbay, D. M., Hogger, C. J., & Robinson, J. A. (Eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 1, pp. 449–503. Clarendon Press, Oxford/New York.
- Howie, J. M. (2003). *Fundamentals of Semigroup Theory*. London Mathematical Society Monographs New Series. Oxford University Press, Oxford.
- Krom, M. R. (1967). The decision problem for a class of first-order formulas in which all disjunctions are binary. *Mathematical Logic Quarterly*, 13(1-2), 15–20.
- Lloyd, J. W. (1987). *Foundations of Logic Programming* (2 edition). Springer-Verlag, Berlin, Heidelberg.
- Makowsky, J. A. (1987). Why Horn formulas matter in computer science: initial structures and generic examples. *Journal of Computer and System Sciences*, 34(2-3), 266–292.
- Sterling, L., & Shapiro, E. (1994). *The Art of Prolog: Advanced Programming Techniques* (2 edition). The MIT Press, Cambridge MA.