



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-04023753v2

HAL Id: hal-04023753

<https://hal.science/hal-04023753v2>

Preprint submitted on 21 Oct 2023

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.

1. INTRODUCTION

The sequential composition of propositional logic programs has been recently introduced by Antić (2023b) and extended to answer set programming by Antić (2023a).

The **purpose of this paper** is to study the sequential composition operation in the Krom fragment (Krom, 1967) consisting only of facts, and rules of the simple form $a \leftarrow b$.¹ 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, 2023b, 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.

2. PROPOSITIONAL KROM LOGIC PROGRAMS

This section recalls the syntax and semantics of propositional Krom logic programs.

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 $a \in A$,
- *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 the *head* and *body* of a rule, respectively, by

$$h(a) := \{a\} \quad \text{and} \quad h(a \leftarrow b) := \{a\}, \quad b(a) := \emptyset \quad \text{and} \quad b(a \leftarrow b) := \{b\},$$

extended to programs by

$$h(K) := \bigcup_{r \in K} h(r) \quad \text{and} \quad b(K) := \bigcup_{r \in K} b(r).$$

¹This class of programs is often called “binary” in the literature.

We define the *dual* of K by

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

An *interpretation* is any subset of A . We define the *entailment relation*, for every interpretation I , inductively as follows:

- for an atom a , $I \models a$ iff $a \in I$;
- for a proper rule, $I \models a \leftarrow b$ iff $I \models b$ implies $I \models a$;
- 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 $LM(K)$.

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, 2023b) 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

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

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 forms a subgroup of \mathbb{K} , where every π has the *inverse* $d(\pi)$, that is,

$$\pi d(\pi) = d(\pi)\pi = 1.$$

We have the following identities (the identities (3)–(6) were shown in Antić (2023b, 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 forms a *quemiring* (Elgot, 1976). Finally, equations (9)–(12) show that the proper rule operator p is a homomorphism which is essential in the rest of the paper.²

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*

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

²This 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, 2023b, §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

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

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*.³ 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

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

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 (2023b, §6).

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

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

³In 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 (2023b, Theorem 40)

$$LM(K) = K^\omega.$$

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

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

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

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

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. LEAST MODEL EQUIVALENCE

Recall that two programs K, L are equivalent with respect to the least model semantics iff $K^\omega = L^\omega$. We therefore wish to compute K^ω . We have

$$K^\omega = f(K^+) = K^+\emptyset = \left(\bigcup_{n \geq 1} K^n \right) \emptyset = \bigcup_{n \geq 1} (K^n \emptyset) = \bigcup_{n \geq 1} f(K^n).$$

Let us compute K^n . We have

$$\begin{aligned} K^2 &= (f(K) \cup p(K))(f(K) \cup p(K)) \\ &= f(K)^2 \cup p(K)f(K) \cup f(K)p(K) \cup p(K)^2 \\ &\stackrel{(1)}{=} f(K) \cup p(K)f(K) \cup p(K), \end{aligned}$$

and

$$K^3 = K^2K = f(K) \cup p(K)f(K) \cup p(K)^2K = f(K) \cup p(K)f(K) \cup p(K)^2f(K) \cup p(K)^3,$$

and

$$\begin{aligned} K^4 &= K^3K \\ &= f(K) \cup p(K)f(K) \cup p(K)^2f(K) \cup p(K)^3K \\ &= f(K) \cup p(K)f(K) \cup p(K)^2f(K) \cup p(K)^3f(K) \cup p(K)^4. \end{aligned}$$

A simple proof by induction shows the general formula for arbitrary $n \geq 1$ given by

$$K^n = f(K) \cup p(K)^n \cup \bigcup_{i=1}^{n-1} (p(K)^i f(K)).$$

This implies

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

where the fifth identity follows from $f(f(K)) = f(K)$ and $f(p(K)^n) = \emptyset$, and the last identity follows from $p(K)^0 = 1$.

Hence, we obtain an algebraic characterization of equivalence with respect to the least model semantics given by

$$K \equiv L \Leftrightarrow \bigcup_{n \geq 1} \bigcup_{i=0}^{n-1} p(K)^i f(K) = \bigcup_{n \geq 1} \bigcup_{i=0}^{n-1} p(L)^i f(L).$$

6. SUBSUMPTION EQUIVALENCE

Recall that two Krom programs K, L are called *subsumption equivalent* (cf. Maher, 1988) — in symbols, $K \equiv_{ss} L$ — iff $KI = LI$ holds for each interpretation I .⁴

Our first observation is that in case K and L are subsumption equivalent, we must have

$$K\emptyset = L\emptyset$$

which by (8) is equivalent to

$$f(K) = f(L).$$

Now let a be an atom in the head of K which is not a fact of K , that is,

$$a \in h(K) \quad \text{and} \quad a \notin f(K).$$

Then there must be a rule $a \leftarrow b \in K$, for some $b \in A$. Let $I := \{b\}$. Since

$$KI = LI \quad \text{and} \quad a \in KI \quad \text{implies} \quad a \in LI,$$

we must have $a \leftarrow b \in L$. This immediately implies $K \subseteq L$. An analogous argument shows $L \subseteq K$, which leads to the trivial characterization of subsumption equivalence given by

$$K \equiv_{ss} L \quad \text{iff} \quad K = L.$$

7. UNIFORM EQUIVALENCE

Recall that two Krom programs K, L are called *uniformly equivalent* (Maher, 1988; Sagiv, 1988; Eiter & Fink, 2003) — in symbols, $K \equiv_u L$ — iff $K \cup I \equiv L \cup I$ holds for every interpretation I . The condition $K \cup I \equiv L \cup I$ can be rephrased in terms of the omega operation as $(K \cup I)^\omega = (L \cup I)^\omega$. This means that in order to understand uniform equivalence, we need to understand $(K \cup I)^\omega$.

First, we have

$$(K \cup I)^2 = (K \cup I)(K \cup I) \stackrel{(6),(5)}{=} K^2 \cup KI \cup IK \cup I^2 \stackrel{(1)}{=} K^2 \cup KI \cup I$$

and

$$(K \cup I)^3 = (K^2 \cup KI \cup I)(K \cup I) = K^3 \cup KIK \cup IK \cup K^2I \cup KI^2 \cup I^2 \stackrel{(1)}{=} K^3 \cup K^2I \cup KI \cup I.$$

A straightforward induction proof shows that the general pattern for arbitrary $n \geq 1$ is

$$(K \cup I)^n = K^n \cup K^{n-1}I \cup \dots \cup KI \cup I.$$

Hence, we have

$$(K \cup I)^+ = \bigcup_{n \geq 1} (K \cup I)^n = \bigcup_{n \geq 1} (K^n \cup K^{n-1}I \cup \dots \cup KI \cup I) = K^+ \cup K^*I,$$

⁴Subsumption equivalence is usually defined in terms of the van Emden Kowalski operator.

and therefore

$$(K \cup I)^\omega = f((K \cup I)^+) = f(K^+ \cup K^*I) = f(K^+) \cup f(K^*I) = f(K^+) \cup K^*I = K^\omega \cup K^*I.$$

That is, we can compute the least model of $K \cup I$ in terms of the least model of K . This implies the characterization of uniform equivalence given by

$$K \equiv_u L \Leftrightarrow K^\omega \cup K^*I = L^\omega \cup L^*I.$$

This means that in order to understand uniform equivalence, we need to understand the omega and star operators.

We wish to understand K^*I . We therefore compute

$$\begin{aligned} K^*I &= \left(\bigcup_{n \geq 0} K^n \right) I \\ &= \bigcup_{n \geq 0} (K^n I) \\ &= \bigcup_{n \geq 0} \left(f(K)I \cup p(K)^n I \cup \bigcup_{i=1}^{n-1} p(K)^i f(K)I \right) \\ &= f(K) \cup \bigcup_{n \geq 0} \left(p(K)^n I \cup \bigcup_{i=1}^{n-1} p(K)^i f(K) \right) \\ &= f(K) \cup \bigcup_{n \geq 0} p(K)^n I \cup \bigcup_{n \geq 0} \bigcup_{i=1}^{n-1} p(K)^i f(K) \\ &= K^\omega \cup \bigcup_{n \geq 0} p(K)^n I \\ &= K^\omega \cup p(K)^*I. \end{aligned}$$

In total, we therefore have

$$(K \cup I)^\omega = K^\omega \cup K^*I = K^\omega \cup p(K)^*I.$$

Hence, we have finally arrived at an algebraic characterization of uniform equivalence of propositional Krom logic programs given by

$$K \equiv_u L \Leftrightarrow K^\omega \cup p(K)^*I = L^\omega \cup p(L)^*I, \quad \text{for every interpretation } I.$$

8. 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, 2023b, 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, 2023a).

Since the space of all propositional Krom logic programs forms a monoid with respect to sequential composition, and almost⁵ a semiring with respect to composition and union, we can ask all kinds of

⁵Recall that the only reason for the space of all propositional Krom programs to form a semiring is that in case K contains facts, $K\emptyset = f(K) \neq \emptyset$.

algebraic questions. A particularly interesting one seems to be: What are “prime” propositional Krom logic programs?

9. 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. We provided algebraic characterizations of least model and uniform equivalence of propositional Krom logic programs in terms of the sequential composition of programs.

The major line for future research is to lift the results of this paper from Krom to arbitrary propositional, first-order, and answer set programs containing negation as failure (Clark, 1978). The former is challenging since the sequential composition of arbitrary propositional programs is in general not associative and does not distribute over union (cf. Antić, 2023b) which means that most of the results of this paper are not directly transferable. The latter is difficult since the sequential composition of answer set programs is rather complicated (cf. Antić, 2023a).

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. (2023a). Sequential composition of answer set programs. *Artificial Intelligence*, under review. <https://arxiv.org/pdf/2104.12156.pdf>.
- Antić, C. (2023b). Sequential composition of propositional logic programs. *Annals of Mathematics and Artificial Intelligence*, under review. <https://arxiv.org/pdf/2009.05774.pdf>.
- Clark, K. L. (1978). Negation as failure. In Gallaire, H., & Minker, J. (Eds.), *Logic and Data Bases*, pp. 293–322. Plenum Press, New York.
- Eiter, T., & Fink, M. (2003). Uniform equivalence of logic programs under the stable model semantics. In *ICLP 2003*, LNCS 2916, pp. 224–238. Springer-Verlag.
- 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.
- 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.
- Maher, M. J. (1988). Equivalences of logic programs. In Minker, J. (Ed.), *Foundations of Deductive Databases and Logic Programming*, chap. 16, pp. 627–658. Morgan Kaufmann Publishers.
- Sagiv, Y. (1988). Optimizing datalog programs. In Minker, J. (Ed.), *Foundations of Deductive Databases and Logic Programming*, chap. 17, pp. 659–698. Morgan Kaufmann Publishers.