From Farkas lemma to linear programming: an exercise in diagrammatic algebra

Filippo Bonchi
University of Pisa, Italy

Alessandro Di Giorgio
University of Pisa, Italy

Fabio Zanasi
University College London, UK

Abstract

Farkas lemma is a celebrated result on the solutions of systems of linear inequalities, which finds application pervasively in mathematics and computer science. In this work we show how to formulate and prove Farkas lemma in diagrammatic polyhedral algebra, a sound and complete graphical calculus for polyhedra. Furthermore, we show how linear programs can be modeled within the calculus and how some famous duality results can be proved.

2012 ACM Subject Classification Theory of computation → Categorical semantics

Keywords and phrases String diagrams, Farkas Lemma, Duality, Linear Programming

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Farkas lemma is a classical result on the solutions of systems of linear inequalities, which appears ubiquitously across various fields of Mathematics and Computer Science; more than a century after its introduction in [16, 17], it continues to receive attention and generate new lines of research [3, 10, 15, 22, 30, 25, 31, 4, 24, 28, 1]. Throughout the decades, different proofs have been given, and many variations have been proposed. The most established formulation asserts that, given a $m \times n$ matrix A, a vector $b \in \mathbb{R}^m$ and their transposes A^T and b^T, exactly one of the following two propositions is true.

$$\text{(a) } \exists x \in \mathbb{R}^n \text{ s.t. } x \geq 0 \text{ and } Ax = b \qquad \text{(b) } \exists y \in \mathbb{R}^m \text{ s.t. } A^T y \geq 0 \text{ and } b^T y < 0$$

Farkas lemma finds application in a number of different scenarios, ranging from non-linear optimisation [28, 4] to the algebraic semantics of non-deterministic and probabilistic systems [23]. Most computer scientists first meet Farkas lemma when studying duality theory in linear programming. A gentle introduction to this theory is provided by the *farmer problem*.

A farmer grows wheat and barley on a land of size l, with a provision f of fertilizer and p of pesticide. To grow one unit of wheat the farmer needs one unit of land, f_1 units of fertilizer and p_1 units of pesticide. Analogously, one unit of barley requires one unit of land, f_2 of fertilizer and p_2 of pesticide. The sell prices for wheat and barley are, respectively, s_1 and s_2. By fixing x_1 to be the units of wheat and x_2 those of barley to be produced, the farmer should solve the following linear program to maximize the profit out of the production.

$$max\{c \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mid x_1, x_2 \geq 0,\ A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq b\} \text{ where } c = \begin{pmatrix} s_1 & s_2 \end{pmatrix},\ A = \begin{pmatrix} 1 & 1 \\ f_1 & f_2 \\ p_1 & p_2 \end{pmatrix},\ b = \begin{pmatrix} l \\ f \\ p \end{pmatrix}$$

Now assume that a planning board needs to establish prices for land, fertilizer and pesticide. The board's job is to minimize the cost of production while assuring some profit to the farmer. To do so, it is sufficient to solve the following program where A, b and c are as above.

$$min\{b^T \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \mid y_1, y_2, y_3 \geq 0,\ A^T \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \leq c^T\}$$

The problem of the farmer and the one of the board are a typical example of a pair of dual problems. A result in duality theory (which makes the relevance of Farkas lemma apparent) is that, if a problem has unbounded solution, then its dual has no solution. Most importantly, when a problem and its dual have finite solutions, then these solutions coincide. In the example above, the minimum cost of the production and the maximum profit of the farmer should then be equal.

In this paper we revisit Farkas lemma and duality results in linear programming through the lens of *string diagrams*.

String diagrams came to the fore as a graphical syntax for representing arrows of symmetric monoidal categories [33]. In recent years, increasingly they have been adopted as a formal language to study component-based systems across different fields of science [12, 2, 18, 19, 21, 29, 32] using the compositional methods that are typical of programming language semantics. One striking property of this approach is that, even though string diagrams have an appealing graphical representation, they are completely formal syntactic objects. Furthermore, they may receive semantics interpretation in some mathematical domain (such as functions, relations, matrices, subspaces, etc.) and many results have been provided on how equational theories of string diagrams are able to *axiomatise* semantic equality over these domains, see e.g. [6, 8, 36, 37, 2, 7]. Such a *complete* equational theory yields a powerful pictorial calculus to reason algebraically about system behaviour, for instance in concurrency [6, 11], control [9, 2] and quantum theory [13].

The core of the calculus that we exploit in this paper is the theory of Interacting Hopf Algebras [36, 8, 2], originally introduced to reason about the behaviour of signal flow graphs [34]. Such theory has been extended first in [7] to study non-passive electrical network and concurrent connectors [11], and then in [5], for studying continuous Petri nets [14]. The latter extension, called *diagrammatic polyhedral algebra*, provides a sound and complete calculus which is able to express exactly *polyhedra*. We claim this is the proper string diagrammatic setting to express Farkas lemma and duality in linear programming.

In diagrammatic polyhedral algebra, recalled in Section 2, different entities of traditional algebra, like vectors, matrices and subsets $C \subseteq \mathbb{R}^n$ are all regarded as relations amongst vectors spaces. Starting from few primitive relations (depicted as wires and gates of circuits), one can syntactically construct all polyhedra by means of relational composition and cartesian product (graphically rendered as horizontal and vertical juxtaposition). It is exactly this linguistic aspect the main novelty of our proof of Farkas lemma: statements about existence of solutions, like (a) and (b) above, translate into equations amongst terms of the string diagrammatic syntax; proofs are symbolic manipulation of diagrams, whose soundness is guaranteed by the axiomatisation. Moreover, compositionality allows to break complex notions into simple inductive definitions on the sets of primitive relations. For instance, the *polar operator* which is given inductively in Section 3, captures the notions of polar and dual cone that are defined in the traditional language by mean of universal quantifications.

In the context of diagrammatic polyhedral algebra, the proof of Farkas lemma becomes straightforward: using a basic observation, named the *lemma of alternatives* in Section 4, the proof –in Section 5– reduces to compute the polar operator over a certain string diagram.

The final part of our work (Section 6) is dedicated to duality in linear programming. Interestingly, diagrammatic polyhedral algebra allows to prove various duality theorems in a rather different way than those found in traditional textbooks (see e.g. [35]). In the classical approach, one first needs to massage the dual problems to bring them into an appropriate shape, and then prove, in sequence, a weak and a strong duality theorems. Our proof method instead is based on a general principle (Theorem 23) that, independently from the shape of the problem at hand, allows to prove all the results at once. Curiously, our proof does not rely on Farkas lemma: rather both the duality theorems and Farkas lemma stem from general results encoded in the axiomatisation of diagrammatic polyhedral algebra.

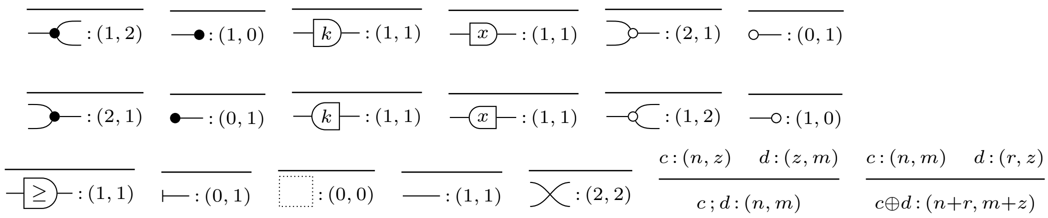
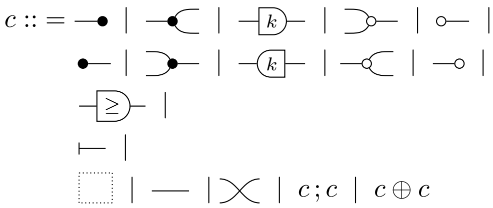


Figure 1 Sort inference rules.

2 Diagrammatic Polyhedral Algebra

This section presents a calculus of string diagrams for reasoning about polyhedra, which we will later use to prove Farkas Lemma and the duality theorems for linear programming. The calculus was first introduced in [5], to which we refer for a more detailed exposition.

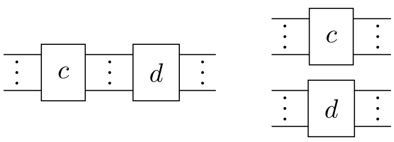
We fix an *ordered field* k, i.e. a field equipped with a total order $\geq$ such that for all $i, j, k \in \mathsf{k}$: (a) if $i \geq j$, then $i + k \geq j + k$; (b) if $i \geq 0$ and $j \geq 0$, then $i \cdot j \geq 0$. The syntax of the calculus is given by the following context free grammar, where k ranges over k.



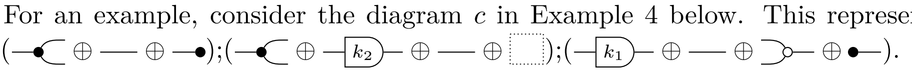
(1)
(2)
(3)
(4)
(5)

We shall consider only terms that are *sortable*, i.e. that one may associate with a pair $(n,\ m)$ of natural numbers $n, m \in \mathbb{N}$ using the rules in Figure 1.

The above syntax specification purposefully uses a graphical rendering of the components. As customary for string diagrams, we will render composition via ; and $\oplus$ graphically by horizontal and vertical juxtaposition of boxes, respectively.



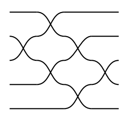
For an example, consider the diagram c in Example 4 below. This represents the term



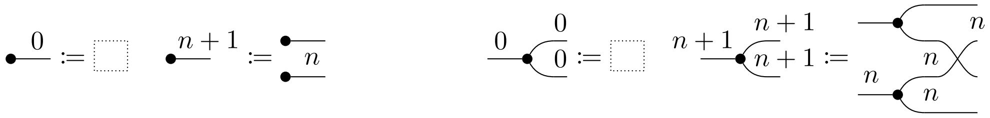
Note that one-dimensional syntax coincides with diagrammatic notation only modulo certain structural rules (e.g. associativity of composition), which amount to the equations of *symmetric monoidal categories* [33] (SMCs). It turns out that structurally equivalent terms have the same meaning in the semantic model we will consider below. Thus, henceforth we shall exclusively focus on string diagrams as our notation for syntax.

It is worth to also recall the categorical viewpoint on diagrammatic syntax. Equivalently to the presentation given above, one may formalise string diagrams as the morphisms of a *prop* (product and permutation category [27, 26]), i.e. a strict SMC with objects the natural numbers, where $\oplus$ on objects is by addition. We introduce the prop for our syntax below.

▶ **Definition 1.** *The prop freely generated by* (1), (2), (3) *and* (4) *is denoted by* PDiag. *In other words,* PDiag *is the prop where arrows* $n \to m$ *are terms of sort* (n, m) *quotiented by the axioms of symmetric monoidal categories. Composition* ; *and monoidal product* $\oplus$ *of diagrams are given by the syntax operations in* (5). *The identities are* $id_0 :=$ *and* $id_{n+1} := id_n \oplus$ —. *The symmetries* $\sigma_{n,m} \colon n + m \to m + n$ *are defined in the obvious way starting from* $\sigma_{1,1} :=$. *For instance,* $\sigma_{2,3}$ *is the diagram below.*



We will depict id_n as and $\sigma_{n,m}$ as . Using these diagrams one can define for each $n \in \mathbb{N}$ the n-version of each of the generator in (1), (2), (3) and (4). For instance, $: 0 \to n$ and $: n \to n + n$ are inductively defined as



When clear from the context, we will omit the n. A semantic interpretation for string diagrams of PDiag will be provided by morphisms in another prop, which we present below.

▶ **Definition 2.** Rel_k *is the prop where arrows* $n \to m$ *are relations* $R \subseteq \mathsf{k}^n \times \mathsf{k}^m$.

- *Composition is relational: given* $R \colon n \to m$ *and* $S \colon m \to o$,

$$R \,;\, S = \{\, (u, v) \in \mathsf{k}^n \times \mathsf{k}^o \mid \exists w \in \mathsf{k}^m.\ (u, w) \in S \wedge (w, v) \in R \,\}$$

- *The monoidal product is cartesian product: given* $R \colon n \to m$ *and* $S \colon o \to p$,

$$R \oplus S = \{\, (\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}) \in \mathsf{k}^{n+o} \times \mathsf{k}^{m+p} \mid (u_1, v_1) \in R \wedge (v_1, v_2) \in S \,\}$$

- *The symmetries* $\sigma_{n,m} \colon n + m \to m + n$ *are the relations*

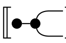
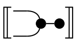
$$\{\, (\begin{pmatrix} u \\ v \end{pmatrix}, \begin{pmatrix} v \\ u \end{pmatrix}) \mid u \in \mathsf{k}^n, v \in \mathsf{k}^m \,\}$$

We can now formally define the semantic interpretation as a *prop morphism* (an identity-on-objects symmetric monoidal functor) $[\![\cdot]\!] : \mathsf{PDiag} \to \mathsf{Rel}_\mathsf{k}$. For the generators in (1), $[\![\cdot]\!]$ is

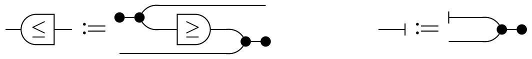
$$\begin{array}{llll} [\![\;]\!] = \{(x, \begin{pmatrix} x \\ x \end{pmatrix}) \mid x \in \mathsf{k}\} & [\![\;]\!] = \{(\begin{pmatrix} x \\ y \end{pmatrix}, x + y) \mid x, y \in \mathsf{k}\} & \\ [\![\;]\!] = \{(x, \bullet) \mid x \in \mathsf{k}\} & [\![\;]\!] = \{(\bullet, 0)\} & [\![\; k \;]\!] = \{(x, k \cdot x) \mid x \in \mathsf{k}\} \end{array} \quad (6)$$

and, symmetrically, for the generators in (2). For instance, $[\![\; k \;]\!] = \{(k \cdot x, x) \mid x \in \mathsf{k}\}$. For the generators in (3) and (4), the semantics is defined, respectively, as $[\![\; \geq \;]\!] = \{\, (x, y) \mid x, y \in \mathsf{k}, x \geq y \,\}$ and $[\![\;]\!] = \{(\bullet, 1)\}$. The semantics of the identities, symmetries and compositions – in (5)– is given by the *functoriality* of $[\![\cdot]\!]$, e.g., $[\![c\,; d]\!] = [\![c]\!] \,;\, [\![d]\!]$ and $[\![\;]\!] = [\![id_0]\!] = \{(\bullet, \bullet)\}$. Above we used $\bullet$ for the unique element of the vector space k^0.

▶ **Example 3.** Two string diagrams will play a special role in our exposition: and . By definition of $[\![\cdot]\!]$, note that their semantics forces the two ports on the right (resp. left) to carry the same value, thus acting as a left (right) feedback.

$[\![$  $]\!] = \{(\bullet, \begin{pmatrix} x \\ x \end{pmatrix}) \mid x \in \mathsf{k}\}$ $\qquad$ $[\![$  $]\!] = \{(\begin{pmatrix} x \\ x \end{pmatrix}, \bullet) \mid x \in \mathsf{k}\}$

We can use these feedback diagrams to arbitrarily move wires from left to right. For instance



As expected, $[\![$ $\boxed{\leq}$ $]\!] = \{(y, x) \mid x, y \in \mathsf{k},\ x \geq y\}$ and $[\![$ $]\!] = \{(1, \bullet)\}$.

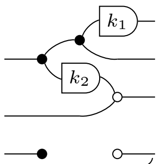
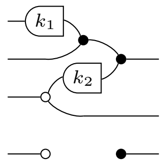
In [5] it is shown that diagrams of PDiag can express, amongst all the relations $R \subseteq \mathsf{k}^n \times \mathsf{k}^m$, exactly all those that are *polyhedra*, *cf.* Example 7 below. Moreover, it is worth recalling that fragments of PDiag also characterise well-known classes of relational objects, as indicated in the table below (see [36, 5] for an overview of these results).

prop	**syntax**	**semantics**
$\mathsf{MD\overrightarrow{iag}}$	(1), (5)	matrices
$\mathsf{MD\overleftarrow{iag}}$	(2), (5)	reversed matrices
LDiag	(1), (2), (5)	linear relations (sub-spaces)
PCDiag	(1), (2), (3), (5)	polyhedral cones
PDiag	(1), (2), (3), (4), (5)	polyhedra

(7)

For instance, the arrows of PDiag, which are only built from the components in (1) and (5), form a sub-prop of PDiag, denoted by $\mathsf{MD\overrightarrow{iag}}$, and characterise k-matrices — in terms of the semantics functor $[\![\cdot]\!] : \mathsf{PDiag} \to \mathsf{Rel}_{\mathsf{k}}$, they denote precisely the relations of the form $\{(x, Ax) \mid x \in \mathsf{k}^p\}$ for some matrix A. Similarly $\mathsf{MD\overleftarrow{iag}}$, LDiag and PCDiag are the sub-props of PDiag of arrows built from the generators specified in (7). Hereafter we illustrate some examples of these fragments, and the corresponding semantic characterisation.

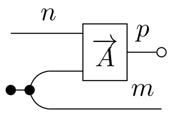
▶ **Example 4** ((Reversed) Matrices)**.** As mentioned, diagrams $c\colon n \to m$ in $\mathsf{MD\overrightarrow{iag}}$ denote precisely the $m \times n$ matrices (see [36] for all details). Consider for instance, the diagram $c\colon 3 \to 4$ and its representation as a 4×3 matrix. Note that $A_{ij} = k$ whenever k is the scalar encountered on the path from the ith port to the jth port. If there is no path, then $A_{ij} = 0$. It is easy to check that $[\![c]\!] = \{(x, y) \in \mathsf{k}^3 \times \mathsf{k}^4 \mid y = Ax\}$.

$c =$  $\qquad A = \begin{pmatrix} k_1 & 0 & 0 \\ 1 & 0 & 0 \\ k_2 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad d =$ 

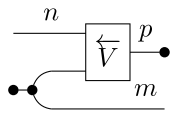
Dually, diagrams in $\mathsf{MD\overleftarrow{iag}}$ are "reversed" matrices: inputs on the right and outputs on the left. For instance $d\colon 4 \to 3$ again encodes A, but its semantics is $[\![d]\!] = \{(y, x) \in \mathsf{k}^4 \times \mathsf{k}^3 \mid y = Ax\}$.

Hereafter we will use $\overset{n}{—}\boxed{\overrightarrow{A}}\overset{m}{—}$ and $\overset{m}{—}\boxed{\overleftarrow{A}}\overset{n}{—}$ for some diagrams of $\mathsf{MD\overrightarrow{iag}}$ and, respectively $\mathsf{MD\overleftarrow{iag}}$, corresponding to some $m \times n$ matrix A. For matrices of type $m \times 1$ and $1 \times n$ we will use lower case letters, usually b and c respectively. It is worth to remark that while $m \times 1$ matrices and vectors in k^m have the same representation in the traditional notation, in PDiag, they are presented as $—\boxed{\overrightarrow{b}}\overset{m}{—}$ and $\vdash\boxed{\overrightarrow{b}}\overset{m}{—}$. Indeed, the semantics of the former is $\{(k, bk) \in \mathsf{k}^1 \times \mathsf{k}^m \mid k \in \mathsf{k}\}$, while the semantics of the latter is $\{(\bullet, b) \in \mathsf{k}^0 \times \mathsf{k}^m\}$.

▶ **Example 5** (Linear Relations)**.** Consider the following diagrams in LDiag.



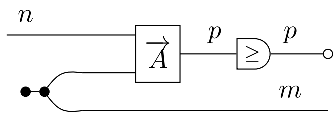
(8)



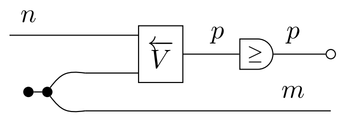
(9)

It easy to check that the semantics of (8) is the set $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} = 0\}$, that is the set of solutions of some system of linear equations. Such system has p rows in $n+m$ variables: n variables stand on the left and m variables on the right. This means that $[\![(8)]\!]$ is a sub-vector space of $\mathsf{k}^n \times \mathsf{k}^m$, namely a *linear relation*. The semantics of (9) is $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid \exists z \in \mathsf{k}^p \text{ s.t. } \begin{pmatrix} x \\ y \end{pmatrix} = Vz\}$, that is the linear hull of the set of column vectors of the matrix V, or in other words the subspace generated by V. Recall that any subspace can be represented both in the form of a system of linear equations and in the form of a set of generating vectors. Indeed, diagrams (8) and (9) represents two normal forms for the diagrams in LDiag.

▶ **Example 6** (Polyhedral cones)**.** Consider the following diagrams in PCDiag



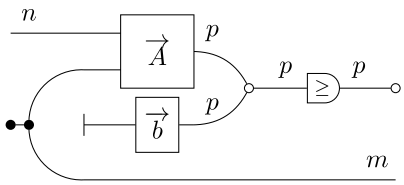
(10)



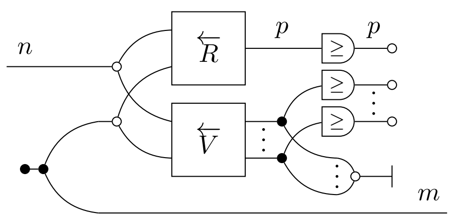
(11)

with semantics $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} \geq 0\}$ and $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid \exists z \in \mathsf{k}^p \text{ s.t. } \begin{pmatrix} x \\ y \end{pmatrix} = Vz, z \geq 0\}$, respectively. The semantics of (10) is thus the set of solutions of a systems of linear *in*equalities, namely a *polyhedral cone*, while the semantics of (11) is the conic hull of V (seen as a set of column vectors). Similarly to Example 5, diagrams in (10) and (11) can be regarded as two normal forms for diagrams in PCDiag.

▶ **Example 7** (Polyhedra)**.** Consider the following diagrams in PDiag.



(12)



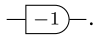
(13)

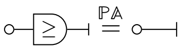
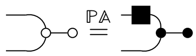
It is easy to check that the semantics of (12) is the relation $\{\, (x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} + b \geq 0 \,\}$ and thus the representation of a *polyhedron* as the set of solutions of a system of affine inequalities. The semantics of (13) is the relation $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid \exists z \in \mathsf{k}^p, w \in \mathsf{k}^o \text{ s.t. } z \geq 0, w \geq 0, \sum w_i = 1, Rz + Vw = \begin{pmatrix} x \\ y \end{pmatrix}\}$ and thus a *vertex* representation of a polyhedron. In other words, $[\![(13)]\!]$ is Minkowsky sum of the conic hull of R, $\{\begin{pmatrix} x \\ y \end{pmatrix} \mid \exists z \in \mathsf{k}^p, \text{ s.t. } z \geq 0, Rz = \begin{pmatrix} x \\ y \end{pmatrix}\}$, and of the *convex hull* of V, $\{\begin{pmatrix} x \\ y \end{pmatrix} \mid \exists w \in \mathsf{k}^o \text{ s.t. } w \geq 0, \sum w_i = 1, Vw = \begin{pmatrix} x \\ y \end{pmatrix}\}$.

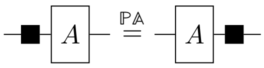
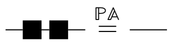
The functor $[\![\cdot]\!] : \mathsf{PDiag} \to \mathsf{Rel}_{\mathsf{k}}$ is not *faithful*: two different string diagrams may denote the same relation. However, PDiag can be equipped with a *sound and complete axiomatisation*, meaning an equational theory making two diagrams c and d equal precisely when $[\![c]\!] = [\![d]\!]$.

Such axiomatisation, called *Polyhedral Algebra* ($\mathbb{PA}$) is illustrated in Figure 2, where we write $l = r$ for the two inequalities $l \sqsubseteq r$ and $r \sqsubseteq l$. In order to state the completeness theorem, we define $\overset{\mathbb{PA}}{\sqsubseteq}$ as the smallest precongruence containing all the pairs (c, d) such that $c \sqsubseteq d$ appears in the Figure 2. In other words, $\overset{\mathbb{PA}}{\sqsubseteq}$ is the smallest relation containing $\sqsubseteq$ which is closed by reflexivity, transitivity, composition ; and monoidal product $\otimes$. Finally, we write $c \overset{\mathbb{PA}}{=} d$ iff $c \overset{\mathbb{PA}}{\sqsubseteq} d$ and $d \overset{\mathbb{PA}}{\sqsubseteq} c$.

▶ **Theorem 8** (From [5]). *For all diagrams c, d in* PDiag, *$[\![c]\!] \subseteq [\![d]\!]$ if and only if $c \overset{\mathbb{PA}}{\sqsubseteq} d$.*

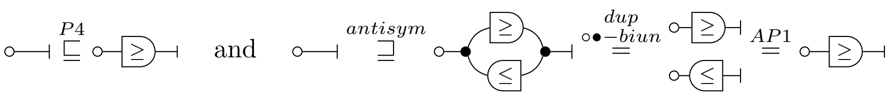
Here are some interesting consequences of the theory $\mathbb{PA}$, where we use  for .

 (14)  (15)

 for any A in LDiag (16)  (17)

Theorem 8 implies that equivalences like (14), (15) and (17) may be also proved by purely graphical means, using derivations involving the axioms of $\mathbb{PA}$, without resorting to the semantic interpretation $[\![\cdot]\!]$. The proofs of more sophisticated statements, as (16), involve axioms in combination with other proof techniques, e.g., induction.

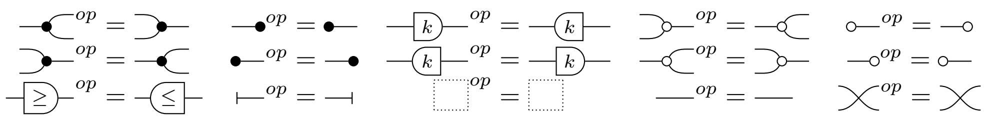
The following is an example of derivation proving (14).



(18)

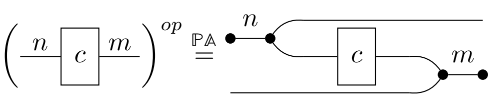
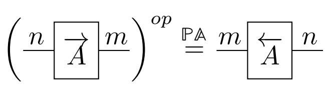
Note that in (18) we used a version of axioms $P4$, *dup* and $AP1$ where diagrams are "rotated over the y axis". We formalise such a notion, in a way that justifies this use.

▶ **Definition 9.** *The prop morphism* $\cdot^{op}\colon \mathsf{PDiag}^{op} \to \mathsf{PDiag}$ *is inductively defined as:*

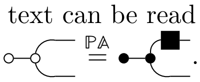


$$(c\,;d)^{op} = d^{op}\,;c^{op} \qquad (c \oplus d)^{op} = c^{op} \oplus d^{op}$$

Observe that $\cdot^{op}$ is controvariant: it maps a diagram $c\colon n \to m$ into $c^{op}\colon m \to n$ which is graphically rendered as the *mirror image* of c: for instance, referring to Example 4, $c^{op} = d$. By exploiting the inductive definition, one can prove that the following hold.

 (19)  (20)

if $c \overset{\mathbb{PA}}{\sqsubseteq} d$ then $c^{op} \overset{\mathbb{PA}}{\sqsubseteq} d^{op}$ (21) $\quad (c^{op})^{op} \overset{\mathbb{PA}}{=} c$ (22)

Equation (19) states that $[\![c^{op}]\!]$ is exactly the *opposite relation* of $[\![c]\!]$, i.e., $[\![c^{op}]\!] = \{(y, x) \in \mathsf{k}^m \times \mathsf{k}^n \mid (x, y) \in [\![c]\!]\}$. In particular, by (20), any diagram in $\mathsf{MD\overrightarrow{iag}}$ representing a matrix A is mapped into a diagram in $\mathsf{MD\overleftarrow{iag}}$ representing the same matrix (see Example 4). Thanks to (21) and (22), one has that $c \overset{\mathbb{PA}}{=} d$ iff $c^{op} \overset{\mathbb{PA}}{=} d^{op}$. Therefore, each of the axioms in Figure 2 and each of the laws that we prove in this text can be read both as $c \overset{\mathbb{PA}}{=} d$ and as $c^{op} \overset{\mathbb{PA}}{=} d^{op}$. For instance, by (15) we also know that . Like in (18), in our derivations we will always use this property implicitly.

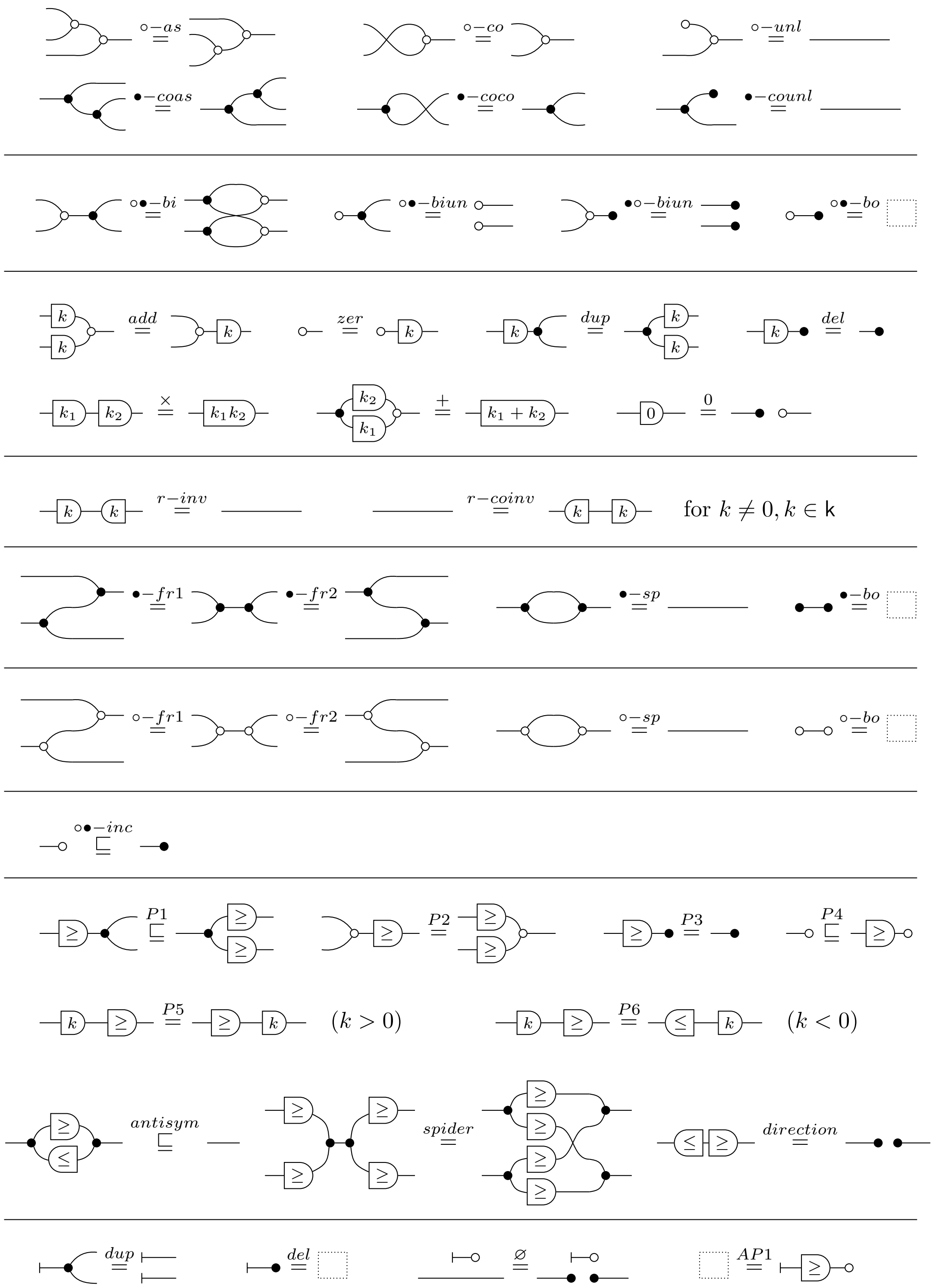


Figure 2 Axioms of $\mathbb{PA}_{\mathsf{k}}$

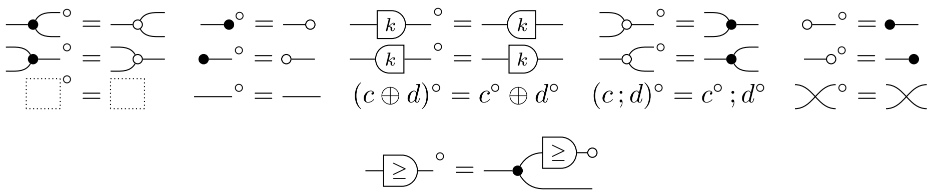
3 The polar operator

When reasoning about cones $C \subseteq \mathsf{k}^n$ in convex algebra, an important role is played by the notions of *polar* and *dual* cone:

$$polar(C) = \{b \in \mathsf{k}^n \mid \forall x \in C,\ b^T x \leq 0\} \qquad dual(C) = \{b \in \mathsf{k}^n \mid \forall x \in C,\ b^T x \geq 0\}$$

As these concepts will also be relevant to our developments, we now study how they are expressible in PCDiag. The fundamental ingredient is the *polar operator* from [5]:

▶ **Definition 10.** *The prop morphism* $\cdot^\circ \colon \mathsf{PCDiag} \to \mathsf{PCDiag}$ *is inductively defined as:*



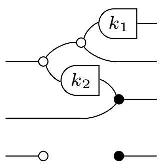
$(c \oplus d)^\circ = c^\circ \oplus d^\circ \qquad (c\,;d)^\circ = c^\circ\,;d^\circ$

The polar operator subsumes both the concept of dual and polar cone. This can be made precise via the following proposition, whose proof we defer to the end of the next section.

▶ **Proposition 11.** *Let* $C \subseteq \mathsf{k}^n$ *be a polyhedral cone. Let* $c\colon 0 \to n$ *and* $d\colon n \to 0$ *be such that* $[\![c]\!] = \{(\bullet, x) \mid x \in C\}$ *and* $[\![d]\!] = \{(x, \bullet) \mid x \in C\}$. *Then* $[\![c^\circ]\!] = \{(\bullet, b) \mid b \in polar(C)\}$ *and* $[\![d^\circ]\!] = \{(b, \bullet) \mid b \in dual(C)\}$.

Note that Proposition 11 uses two representations of the cone C as a string diagram, one of type $0 \to n$ and the other of type $n \to 0$. Depending on which one we pick, one obtains the polar or the dual of C. Another interesting departure from the traditional approaches is that the polar/dual cone is now specified *inductively* on the structure of the string diagram, following Definition 10. We now provide some properties and examples of the polar operator. First we observe how it behaves on string diagrams representing matrices.

▶ **Example 12.** Consider the matrix A in Example 4 and its encoding as the diagram $c\colon 3 \to 4$ in $\mathsf{MD\overrightarrow{iag}}$. Applying the polar operator on c yields the following diagram in $\mathsf{MD\overleftarrow{iag}}$

$c^\circ =$ 

$$A^T = \begin{pmatrix} k_1 & 1 & k_2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

representing the transpose A^T of the matrix A. Indeed, $[\![c^\circ]\!] = \{(x, y) \in \mathsf{k}^3 \times \mathsf{k}^4 \mid A^T y = x\}$. This is an instance of a more general phenomenon: when applied to matrices (i.e. string diagrams of $\mathsf{MD\overrightarrow{iag}}$), the polar operator yields their transpose matrix, represented by a string diagram in $\mathsf{MD\overleftarrow{iag}}$ (and thus to be read "right-to-left').

▶ **Lemma 13** (From [36]). *For all* $-\boxed{\overrightarrow{A}}-\colon n \to m$ *of* $\mathsf{MD\overrightarrow{iag}}$*, it holds that* $-\boxed{\overrightarrow{A}}-^{\circ} \stackrel{\mathbb{PA}}{=} -\boxed{\overleftarrow{A^T}}-$

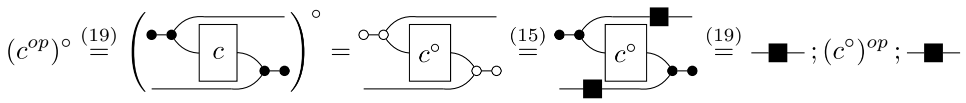
▶ **Proposition 14** (From [5]). *For all diagrams* $c, d\colon n \to m$ *in* PCDiag*, it holds that*

1. *if* $c \stackrel{\mathbb{PA}}{\sqsubseteq} d$ *then* $(d)^\circ \stackrel{\mathbb{PA}}{\sqsubseteq} (c)^\circ$*;*
2. $(c^\circ)^\circ \stackrel{\mathbb{PA}}{=} c$.

The first item of the above proposition informs us that if $c \stackrel{\mathbb{PA}}{=} d$ then one can safely conclude that $c^\circ \stackrel{\mathbb{PA}}{=} d^\circ$. Viceversa, if $c^\circ \stackrel{\mathbb{PA}}{=} d^\circ$, by the second item, $c \stackrel{\mathbb{PA}}{=} d$. The next lemma illustrates the interaction of the polar operator with $\cdot^{op}$ (see Definition 9).

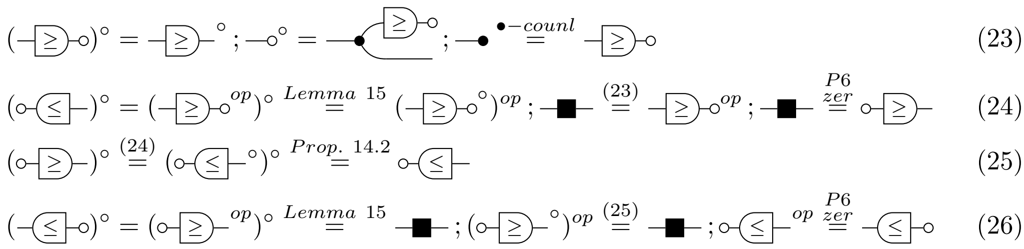
▶ **Lemma 15.** *For all $c\colon n \to m$ in* PCDiag*, it holds that* $(c^{op})^{\circ} \stackrel{\mathbb{PA}}{=} {}_{m}\blacksquare_{m} \,;\, (c^{\circ})^{op} \,;\, {}_{n}\blacksquare_{n}$.

Proof.



◀

▶ **Example 16.** The diagrams $: 0 \to 1$ and $: 1 \to 0$, denoting the relations $\{(\bullet, x) \mid x \geq 0\} \subseteq \mathsf{k}^0 \times \mathsf{k}^1$ and $\{(x, \bullet) \mid x \geq 0\} \subseteq \mathsf{k}^1 \times \mathsf{k}^0$, are two different representations for the same object in traditional algebra: the polyhedral cone $\{x \in \mathsf{k} \mid x \geq 0\} \subseteq \mathsf{k}^1$. Interestingly enough, applying the polar operator to them yields two different results. Analogous considerations hold for $\{x \in \mathsf{k} \mid x \leq 0\} \subseteq \mathsf{k}^1$.



Observe that for the two diagrams above of type $1 \to 0$, $\cdot^{\circ}$ act as identity, while for those of type $0 \to 1$, it reverses the sign. This behaviour is justified by Proposition 11.

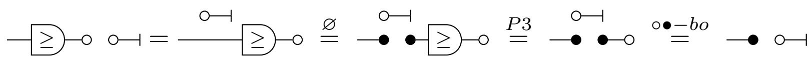
There is a number of other observations about the polar operator, which may be proven with graphical reasoning taking advantage of the inductive definition, the complete axiomatisation and the laws illustrated so far. While this material is not essential to our developments, we conclude this section with two simple "exercises" of that kind, which are left to the interested reader.

- **(Exercise 1)** Prove that, for all c in the form of (8), there exists some d in the form of (9) such that $c^{\circ} \stackrel{\mathbb{PA}}{=} d$. *Hint:* use Lemma 13 and (15).
- **(Exercise 2)** Prove that, for all c in the form of (10), there exists some d in the form of (11) such that $c^{\circ} \stackrel{\mathbb{PA}}{=} d$. *Hint:* use (23).

4 Lemma of the alternatives

This section is devoted to the diagrammatic formulation of a lemma of *alternatives*, asserting that exactly one of two systems of linear inequalities (i.e. polyhedra) has a solution.

To approach the lemma, an important question is how to model "does a system have a solution?" in our graphical calculus. We focus attention on two morphisms of PDiag of type $0 \to 0$: the empty diagram and the diagram ○—|. Intuitively, ○—| asserts that "0 = 1"; its denotational semantics is the composition of the relations $\{(\bullet, 0)\}$ and $\{(1, \bullet)\}$, which gives the empty relation $\emptyset$. Since for any relation R in $\mathsf{Rel}_{\mathsf{k}}$, $R \oplus \emptyset = \emptyset = \emptyset \oplus R$, the behaviour of ○—| resembles that of a logical false. From the viewpoint of the equational theory, ○—| introduces an inconsistency; in particular, by means of the axiom $\varnothing$ we are able to prove that ○—| $\oplus\, c \stackrel{\mathbb{PA}}{=}$ ○—| $\oplus\, d$ for any $c, d\colon n \to m$ in PDiag. As an example, consider the following equation:



(27)

In an analogous way, the behaviour of the diagram $\boxed{}$ can be regarded as a logical true. In particular, its semantics is the relation $id_0 = \{(\bullet, \bullet)\}$ which for any R in $\mathsf{Rel_k}$ is such that $R \oplus id_0 = R = id_0 \oplus R$.

Finally, note that in $\mathsf{Rel_k}$ the only possible morphisms of type $0 \to 0$ are exactly $\emptyset$ and id_0. Thus the following lemma holds.

▶ **Lemma 17** (From [5]). *For any diagram $c\colon 0 \to 0$ of* PDiag, *either* $c \stackrel{\mathbb{PA}}{=} \boxed{}$ *or* $c \stackrel{\mathbb{PA}}{=} \circ\!\!-\!\!\dashv$

▶ **Lemma 18** (Lemma of the alternatives). *Let $c\colon 0 \to 1$ be a diagram in* PCDiag. *Then exactly one of the following two equations holds:*

$$(a)\quad c\,;\!\dashv \stackrel{\mathbb{PA}}{=} \boxed{} \qquad\qquad (b)\quad c^{\circ}\,;\!\dashv \stackrel{\mathbb{PA}}{=} \boxed{}.$$

Proof. Since c is in PCDiag, then $[\![c]\!] \subseteq \mathsf{k}^0 \times \mathsf{k}^1$ is a polyhedral cone. Thus $[\![c]\!]$ must be one of the following:

$$\{(\bullet, k) \mid k \in \mathsf{k}\} \quad \{(\bullet, k) \mid k \geq 0\} \quad \{(\bullet, k) \mid k \leq 0\} \quad \{(\bullet, k) \mid k = 0\}$$

By Theorem 8, it holds[1] that either

$$c \stackrel{\mathbb{PA}}{=} \bullet\!- \quad \text{or} \quad c \stackrel{\mathbb{PA}}{=} \circ\!\!-\!\!(\leq)\!- \quad \text{or} \quad c \stackrel{\mathbb{PA}}{=} \circ\!\!-\!\!(\geq)\!- \quad \text{or} \quad c \stackrel{\mathbb{PA}}{=} \circ\!-.$$

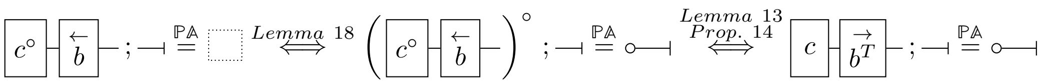
By Proposition 14.1, we can thus consider only these four cases:

- If $c \stackrel{\mathbb{PA}}{=} \bullet\!-$ then $c\,;\!\dashv \stackrel{del}{=} \boxed{}$ and $c^{\circ}\,;\!\dashv \stackrel{Prop.\ 14}{=} (\bullet\!-)^{\circ}\,;\!\dashv\, = \circ\!-\,;\!\dashv\, = \circ\!\!-\!\!\dashv$.
- If $c \stackrel{\mathbb{PA}}{=} \circ\!\!-\!\!(\leq)\!-$ then $c\,;\!\dashv \stackrel{AP1}{=} \boxed{}$ and $c^{\circ}\,;\!\dashv \stackrel{Prop.\ 14}{=} (\circ\!\!-\!\!(\leq)\!-)^{\circ}\,;\!\dashv \stackrel{(24)}{=} \circ\!\!-\!\!(\geq)\!-\,;\!\dashv \stackrel{(14)}{=} \circ\!\!-\!\!\dashv$.
- If $c \stackrel{\mathbb{PA}}{=} \circ\!\!-\!\!(\geq)\!-$ then $c\,;\!\dashv \stackrel{(14)}{=} \circ\!\!-\!\!\dashv$ and $c^{\circ}\,;\!\dashv \stackrel{Prop.\ 14}{=} (\circ\!\!-\!\!(\geq)\!-)^{\circ}\,;\!\dashv \stackrel{(25)}{=} \circ\!\!-\!\!(\leq)\!-\,;\!\dashv \stackrel{AP1}{=} \boxed{}$.
- If $c \stackrel{\mathbb{PA}}{=} \circ\!-$ then $c\,;\!\dashv \stackrel{\mathbb{PA}}{=} \circ\!\!-\!\!\dashv$ and $c^{\circ}\,;\!\dashv \stackrel{Prop.\ 14}{=} (\circ\!-)^{\circ}\,;\!\dashv\, = \bullet\!-\,;\!\dashv \stackrel{del}{=} \boxed{}$.

◀

The lemma of alternatives yields as a corollary a proof of Proposition 11.

Proof of Proposition 11. Observe that



By definition of $[\![\cdot]\!]$, the former equation holds iff $(\bullet, b) \in [\![c^{\circ}]\!]$, while the latter holds iff $\forall (\bullet, x) \in [\![c]\!]\,,\ b^T x \neq 1$. That is $[\![c^{\circ}]\!]$ is the relation $\{(\bullet, b) \mid \forall x \in C,\ b^T x \neq 1\}$ which is readily seen to be equal to $\{(\bullet, b) \mid b \in polar(C)\}$.

For d, note that $d \stackrel{\mathbb{PA}}{=} c^{op}$. Thus, by Lemma 15, $d^{\circ} \stackrel{\mathbb{PA}}{=} {}_n\!\blacksquare_n\,; (c^{\circ})^{op}$. Thus $(b, \bullet) \in [\![d^{\circ}]\!]$ iff $(\bullet, -b) \in [\![c^{\circ}]\!]$ iff $-b \in polar(C)$ iff $b \in dual(C)$. That is $[\![d^{\circ}]\!] = \{(b, \bullet) \mid b \in dual(C)\}$. ◀

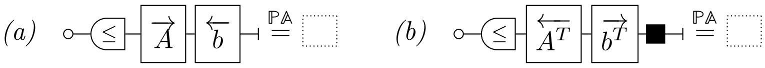
▶ Remark 19. Interestingly, the lemma of alternatives does not hold for diagrams $c\colon 1 \to 0$ (when taking $\vdash\!;c$ and $\vdash\!;c^{\circ}$ in place of $c\,;\!\dashv$ and $c^{\circ}\,;\!\dashv$): it is easy to see this with (23) and (26). In order to obtain a lemma of alternatives for diagrams of type $c\colon 1 \to 0$, one should replace $\cdot^{\circ}$ by a novel operator $\cdot^{\bullet}$ defined as $-(\geq)\!-^{\bullet} = -\!\bullet\!\!\!<\,;(-(\leq)\!-\!\circ \oplus -\!\!-)$ and as $c^{\bullet} = c^{\circ}$ for all the other generators c. Such operator behaves as the dual for diagrams $c\colon 0 \to n$ and as the polar for diagrams $d\colon n \to 0$.

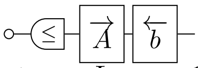
[1] See Appendix A for a purely equational proof that does not invoke completeness.

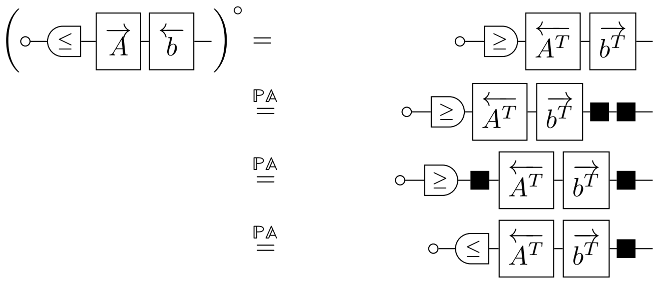
5 A string diagrammatic proof of Farkas Lemma

The lemma of alternatives provides a direct route to a diagrammatic proof of Farkas lemma.

▶ **Lemma 20** (Farkas lemma). *Let* $\overrightarrow{A}\colon n \to m$ *be a diagram in* $\mathsf{MDiag}^{\rightarrow}$ *and* $\overleftarrow{b}: m \to 1$ *in* $\mathsf{MDiag}^{\leftarrow}$, *then exactly one of the following two equations holds:*



Proof. Observe that  is a diagram $c\colon 0 \to 1$ in PCDiag. In order to conclude, it is therefore enough to use Lemma 18 and observe that



(Lemma 13 and (24))

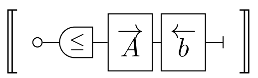
((17))

((16))

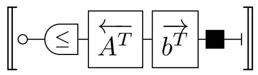
(Axioms $P6$ and *del*)

◀

It is instructive to make explicit in which sense Lemma 20 amounts to the well known result of Farkas. By using the inductive definition of $[\![\cdot]\!]$, one may compute the semantics on the left hand sides of the equations (a) and (b):



$$= \begin{cases} \{(\bullet,\bullet)\} & \text{if } \exists x \in \mathsf{k}^n \text{ s.t. } x \geq 0 \text{ and } Ax = b \\ \emptyset & \text{otherwise} \end{cases}$$



$$= \begin{cases} \{(\bullet,\bullet)\} & \text{if } \exists y \in \mathsf{k}^m \text{ s.t. } A^T y \geq 0 \text{ and } b^T y = -1 \\ \emptyset & \text{otherwise} \end{cases}$$

Therefore equation (a) holds if and only if $\exists x \in \mathsf{k}^n$ s.t. $x \geq 0$ and $Ax = b$ while equation (b) if and only if $\exists y \in \mathsf{k}^m$ s.t. $A^T y \geq 0$ and $b^T y = -1$. In the usual presentation of the Farkas lemma, e.g. [20], the former condition is exactly the same, while the second one is often expressed by the equivalent condition $\exists y \in \mathsf{k}^m$ s.t. $A^T y \geq 0$ and $b^T y < 0$.[2]

6 Duality in linear programming

Farkas lemma is closely related to *linear programming*, as it is one of the main tools to prove duality results in this area. In this section, we explore such duality theorems in the context of diagrammatic polyhedral algebra; it turns out that our formulation does not require the direct application of Farkas lemma, but rather relies on more general principle (Theorem 23) which allows to prove all the results at once.

[2] By mean of Proposition 11 one can also translate our proof in traditional algebraic language: first observe that for all one-dimensional polyhedral cones C either 1 belongs to C or 1 belong to the polar of C (Lemma 18); then prove that the polar of $\{z \in \mathsf{k} \mid \exists x \in \mathsf{k}^n \text{ s.t. } x \geq 0 \text{ and } Ax = bz\}$ is exactly $\{z \in \mathsf{k} \mid \exists y \in \mathsf{k}^m \text{ s.t. } A^T y \geq 0 \text{ and } b^T y = -z\}$ (proof of Lemma 20). We could not find the same proof in literature, but it is hard to claim that it does not exist.

Duality in linear programming studies pairs of problems of the following shape

$$(P) \coloneqq \quad max\{cx \mid Ax \leq b, x \geq 0\} \qquad\qquad (D) \coloneqq \quad min\{b^T y \mid A^T y \geq c^T \text{ and } y \geq 0\}$$

where A, b and c are matrices of type $m \times n$, $1 \times m$ and $n \times 1$, respectively. The *primal problem* (P) requires to maximise cx subject to the condition that $Ax \leq b$ and $x \geq 0$. Its *dual problem* (D) requires to minimise $b^T y$ subject to the condition that $A^T y \geq c^T$ and $y \geq 0$. The farmer problem and the one of the board from the Introduction are instances of (P) and (D). The primal problem has three possible outcomes: (P) may be *unfeasible*, in the sense that there exists no $x \geq 0$ such that $Ax \leq b$; it can be *unbounded*, when the latter inequality holds for some non-negative vectors $x \in \mathsf{k}^n$, but there exists no maximum $k \in \mathsf{k}$ for cx; or it can be *bounded*, if such k exists. The same possibilities apply to (D).

Duality theory in linear programming establishes a series of possibilities between these possible outcomes: in particular, if (P) is unbounded then (D) is unfeasible and, viceversa, if (D) is unbounded then (P) is unfeasible. Moreover, (P) is bounded if and only if (D) is bounded. The following table summarises such results.

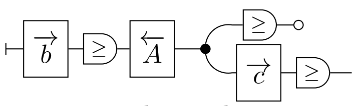
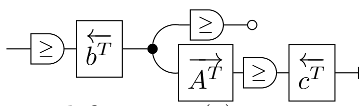
(P) \ (D)	bounded	unbounded	unfeasible
bounded	✓		
unbounded			✓
unfeasible		✓	✓

(28)

The most useful fact is that when (P) and (D) are bounded, they have the same result, i.e.,

$$max\{cx \mid Ax \leq b, x \geq 0\} = min\{b^T y \mid A^T y \geq c^T \text{ and } y \geq 0\}. \tag{29}$$

We now turn to the question of modelling the primal problem (P) and the dual problem (D) in PDiag. Let us fix $\overleftarrow{A} : m \to n$ in $\mathsf{MD\overleftarrow{iag}}$, $\overrightarrow{b} : 1 \to m$ and $\overrightarrow{c} : n \to 1$ in $\mathsf{MD\overrightarrow{iag}}$, and consider the following diagrams in PDiag

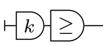
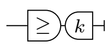
$P \coloneqq$  $\qquad D \coloneqq$ 

Their semantics can be easily computed with the inductive defintion in (6):

$$[\![P]\!] = \{(\bullet, z) \in \mathsf{k}^0 \times \mathsf{k}^1 \mid z \leq cx, x \geq 0, Ax \leq b\}$$

$$[\![D]\!] = \{(z, \bullet) \in \mathsf{k}^1 \times \mathsf{k}^0 \mid z \geq b^T y, y \geq 0, A^T y \geq c^T\}$$

As expected, P models the primal problem (P) and D its dual (D). Indeed, (P) is bounded if and only if $[\![P]\!] = \{(\bullet, z) \mid z \leq k\}$, where k is exactly $max\{cx \mid Ax \leq b, x \geq 0\}$. Also, (P) is unbounded if and only if $[\![P]\!] = \{(\bullet, z) \mid z \in \mathsf{k}\}$ and (P) is unfeasible if and only if $[\![P]\!] = \emptyset$. Analogous considerations hold for (D). The three possibilities can then be expressed in equational terms as follows.

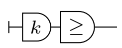
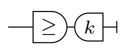
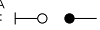
$P \stackrel{\mathbb{PA}}{=}$  iff $k = max\{cx \mid Ax \leq b, x \geq 0\}$ | $D \stackrel{\mathbb{PA}}{=}$  iff $k = min\{b^T y \mid A^T y \leq c^T, y \geq 0\}$

$P \stackrel{\mathbb{PA}}{=}$ iff (P) is unbounded | $D \stackrel{\mathbb{PA}}{=}$ iff (D) is unbounded

$P \stackrel{\mathbb{PA}}{=}$ iff (P) is unfeasible | $D \stackrel{\mathbb{PA}}{=}$ iff (D) is unfeasible

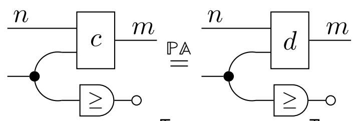
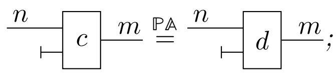
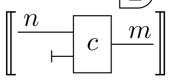
In light of this analysis, the results in (28) and (29) amount to the following theorem.

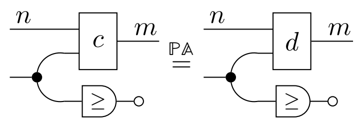
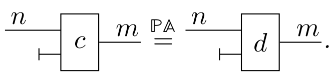
▶ **Theorem 21** (Duality). *The following hold:*

1. *For all $k \in \mathsf{k}$, $P \overset{\mathbb{PA}}{=}$*  *if and only if $D \overset{\mathbb{PA}}{=}$* 
2. *If $P \overset{\mathbb{PA}}{=}$* *, then $D \overset{\mathbb{PA}}{=}$*
3. *If $D \overset{\mathbb{PA}}{=}$* *, then $P \overset{\mathbb{PA}}{=}$* 

In order to prove the above theorem, we exploit homogenisation, a traditional technique to transform polyhedra into cones. The *homogenisation* of polyhedron $P = \{x \in \mathsf{k}^n \mid Ax+b \geq 0\}$ is the polyhedral cone $P^H = \{(x, y) \in \mathsf{k}^{n+1} \mid Ax + by \geq 0, y \geq 0\}$. It holds that $P_1^H = P_2^H$ if and only if $P_1 = P_2$ for all *non-empty* polyhedra P_1, P_2 (see e.g. Lemma 22 in [5]). By exploiting the normal forms in (12) and (10), one obtains the following useful lemma.

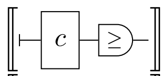
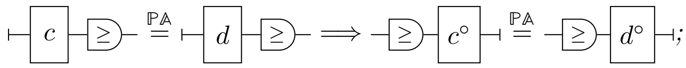
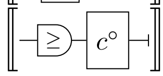
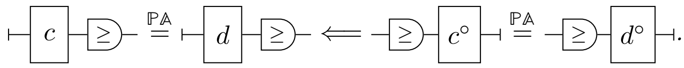
▶ **Lemma 22.** *Let $c, d \colon n + 1 \to m$ be string diagrams in* PCDiag.

1. *If*  *then* *;*
2. *Moreover when*  *$\neq \emptyset$, the other implication holds, that is:*

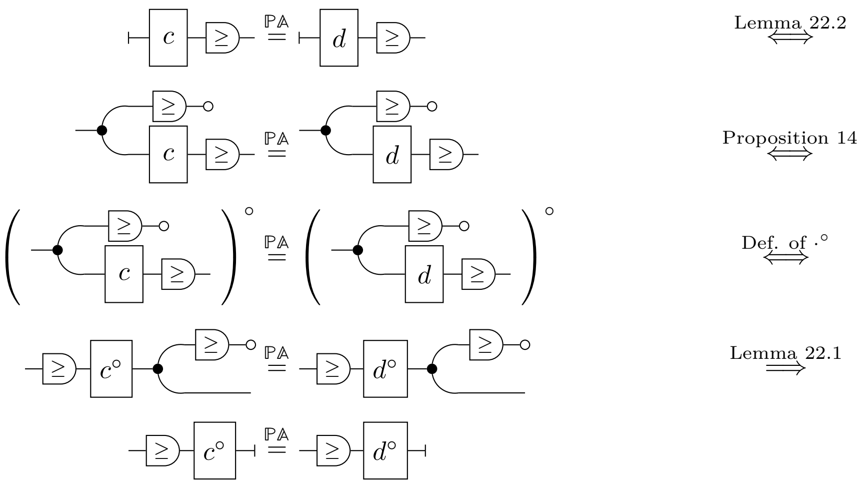
 *iff* .

By combining homogenisation with the polar operator, we obtain a general proof schema which includes as a particular cases the three points of Theorem 21.

▶ **Theorem 23.** *Let $c, d \colon 1 \to 1$ be diagrams in* PCDiag.

1. *If*  *$\neq \emptyset$, then* *;*
2. *If*  *$\neq \emptyset$, then* .

Proof. For the first statement, observe that

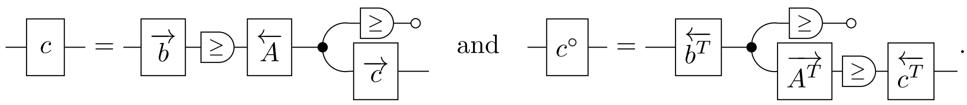


The first step uses Lemma 22.2 because, by hypothesis, the two diagrams denote a non-empty relation. Also, note that the last step uses only the first item of Lemma 22 –thus it is only an implication– because we do not know whether the string diagram denote the empty relation.

To prove the second statement, we use the derivation above, but in the first step we replace $\iff$ by $\Longleftarrow$ (Lemma 22.1) and in the last step we replace $\Longrightarrow$ by $\iff$ (Lemma 22.2). ◀

From Theorem 23 one may immediately derive the three dualities in Theorem 21.

Proof of Theorem 21. First observe that $P =$ and $D =$ where



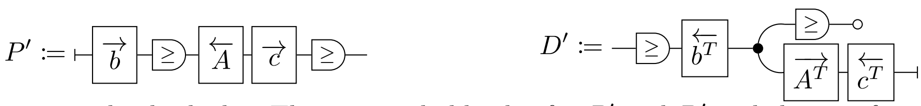
1. Since $[\![\ ]\!] \neq \emptyset$ and $[\![\ ]\!] \neq \emptyset$, then one can exploit the two implications of Theorem 23, by taking d $=$ k and observe that $(k)^\circ = k$.
2. Since $[\![\ ]\!] \neq \emptyset$, then one can use Theorem 23.1 with d $=$ and observe that $\overset{del}{\underset{P3}{=}}$ and $\geq\ ;(\ )^\circ\ ; = \geq \overset{(27)}{=}$.
3. Since $[\![\ ]\!] \neq \emptyset$, then use Theorem 23.2 with d° $=$ and proceed as in 2.

◀

▶ Remark 24. Traditional textbooks do not prove duality results for problems in the form of (P) and (D) above, but they need to first massage problems to obtain the following shape.

$$(P') \coloneqq \ max\{cx \mid Ax \leq b\} \qquad\qquad (D') \coloneqq \ min\{b^T y \mid A^T y = c^T \text{ and } y \geq 0\}$$

Thanks to Theorem 23, we do not really need to rely on a specific form. Indeed by taking



one can easily check that Theorem 21 holds also for P' and D' and the proof is the same, modulo the obvious choice of c.

7 Conclusions

This paper investigates Farkas lemma and duality in linear programming within the language of diagrammatic polyhedral algebra. Besides the elegance of the proofs, the linguistic aspect is, in our opinion, the most interesting angle. Indeed, this work can be thought as an exercise in diagrammatic algebra, illustrating its appeal in the following ways:

- by identifying the right primitive components and the appropriate ways to compose them, one is able to express exactly all the objects of interest (in this case, polyhedra) and to formally reason about them by means of a sound and complete axiomatisation ($\mathbb{PA}$);
- operations on few primitives can be extended inductively to all the objects of interest, resulting in an effective way to compute sophisticated notions, like polar and dual cones;
- equations amongst diagrams can express complex statements, like those about the existence of a solution, the maximal or the minimal solution;
- symbolic manipulation of diagrams by means of axioms and derived laws allows to prove such statements.

The last point leads us to believe that our proofs are suitable to be formalised in proof assistants, such as Coq or Agda. Finally, we think that this work may inspire further duality results in string diagrammatic languages other than diagrammatic polyhedral algebra.

References

1 David Avis and Bohdan Kaluzny. Solving inequalities and proving farkas's lemma made easy. *The American Mathematical Monthly*, 111(2):152–157, 2004.

2 John C Baez and Jason Erbele. Categories in control. *Theory and Applications of Categories*, 30(24):836–881, 2015.

3 David Bartl. A short algebraic proof of the farkas lemma. *SIAM Journal on Optimization*, 19(1):234–239, 2008.

4 Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

5 Guillaume Boisseau, Filippo Bonchi, Alessandro Di Giorgio, and Pawel Sobocinski. Diagrammatic polyhedral algebra, 2021. `arXiv:2105.10946`.

6 Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proceedings of the 46th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 3:1–28, 2019.

7 Filippo Bonchi, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Graphical affine algebra. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2019.

8 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In *Proceedings of the 25th International Conference on Concurrency Theory (CONCUR)*, pages 435–450. Springer, 2014.

9 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. The calculus of signal flow diagrams I: linear relations on streams. *Information and Computation*, 252:2–29, 2017.

10 CG Broyden. A simple algebraic proof of farkas's lemma and related theorems. *Optimization methods and software*, 8(3-4):185–199, 1998.

11 Roberto Bruni, Ivan Lanese, and Ugo Montanari. A basic algebra of stateless connectors. *Theoretical Computer Science*, 366(1–2):98–120, 2006.

12 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

13 Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes - A first course in Quantum Theory and Diagrammatic Reasoning.* Cambridge University Press, 2017.

14 René David and Hassane Alla. *Discrete, Continuous, and Hybrid Petri Nets.* Springer, Berlin, 2 edition, 2010. `doi:10.1007/978-3-642-10669-9`.

15 Achiya Dax. An elementary proof of farkas' lemma. *SIAM Review*, 39(3):503–507, 1997. URL: `http://www.jstor.org/stable/2133044`.

16 Gyula Farkas. A fourier-féle mechanikai elv alkalmazásának algebrai alapja [hungarian; on the algebraic foundation of the applications of the mechanical principle of fourier]. *Mathematikai és Physikai Lapok*, 5:49–54, 1896.

17 J. Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1902:1 – 27, 1902.

18 Brendan Fong, Paolo Rapisarda, and Paweł Sobociński. A categorical approach to open and interconnected dynamical systems. In *LICS 2016*, 2016.

19 Brendan Fong and David I. Spivak. String diagrams for regular logic (extended abstract). In John Baez and Bob Coecke, editors, *Proceedings Applied Category Theory 2019, ACT 2019, University of Oxford, UK, 15-19 July 2019*, volume 323 of *EPTCS*, pages 196–229, 2019. `doi:10.4204/EPTCS.323.14`.

20 David Gale, Harold W Kuhn, and Albert W Tucker. Linear programming and the theory of games. *Activity analysis of production and allocation*, 13:317–335, 1951.

21 Dan R Ghica and Achim Jung. Categorical semantics of digital circuits. In *Proceedings of the 16th Conference on Formal Methods in Computer-Aided Design (FMCAD)*, pages 41–48, 2016.

22 RA Good. Systems of linear relations. *SIAM Review*, 1(1):1–31, 1959.

23 Alexandre Goy and Daniela Petrisan. Combining probabilistic and non-deterministic choice via weak distributive laws. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 454–464. ACM, 2020. `doi:10.1145/3373718.3394795`.

24 Murray C Kemp and Yoshio Kimura. Introduction to mathematical economics. Technical report, Springer, 1978.

25 Vilmos Komornik. A simple proof of farkas' lemma. *The American Mathematical Monthly*, 105(10):949–950, 1998. `arXiv:https://doi.org/10.1080/00029890.1998.12004992`, `doi:10.1080/00029890.1998.12004992`.

26 Stephen Lack. Composing PROPs. *Theory and Application of Categories*, 13(9):147–163, 2004.

27 Saunders Mac Lane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71:40–106, 1965.

28 Olvi L Mangasarian. *Nonlinear programming*. SIAM, 1994.

29 Koko Muroya, Steven W. T. Cheung, and Dan R. Ghica. The geometry of computation-graph abstraction. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 749–758. ACM, 2018. `doi:10.1145/3209108.3209127`.

30 Hukukane Nikaido. *Convex structures and economic theory*. Elsevier, 2016.

31 Martin H. Pearl. A matrix proof of Farkas's Theorem. *The Quarterly Journal of Mathematics*, 18(1):193–197, 01 1967. `arXiv:https://academic.oup.com/qjmath/article-pdf/18/1/193/4498437/18-1-193.pdf`, `doi:10.1093/qmath/18.1.193`.

32 Robin Piedeleu and Fabio Zanasi. A string diagrammatic axiomatisation of finite-state automata. In *FoSSaCS 2021*, 2021.

33 Peter Selinger. A survey of graphical languages for monoidal categories. *Springer Lecture Notes in Physics*, 13(813):289–355, 2011.

34 Claude E. Shannon. The theory and design of linear differential equation machines. Technical report, National Defence Research Council, 1942.

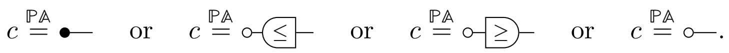
35 Robert J Vanderbei et al. *Linear programming*, volume 3. Springer, 2015.

36 Fabio Zanasi. *Interacting Hopf Algebras: the theory of linear systems*. PhD thesis, Ecole Normale Supérieure de Lyon, 2015.

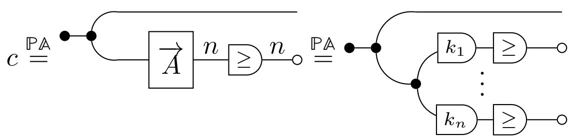
37 Fabio Zanasi. The algebra of partial equivalence relations. *Electronic Notes in Theoretical Computer Science*, 325:313–333, 2016.

A Proofs of Section 4

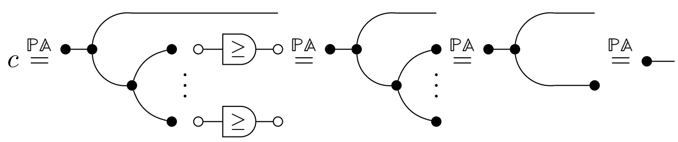
Alternative proof of Lemma 18. We prove diagrammatically, without relying on Theorem 8, that the following property holds for any $c\colon 0 \to 1$ in PCDiag



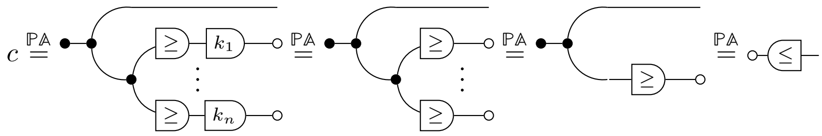
Any $n \to m$ diagram in PCDiag is equivalent to one in the form of (10). A diagram $c\colon 0 \to 1$ has the following normal form, where A is a $n \times 1$ matrix



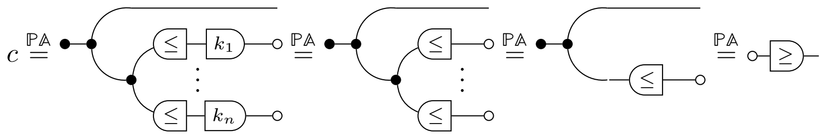
- If $k_i = 0$ for all $i = 1 \ldots n$, then



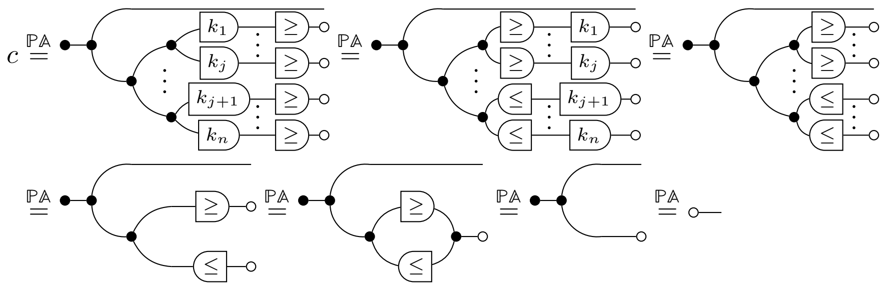
- If $k_i \geq 0$ for all $i = 1 \dots n$, then those $k_i = 0$ are just *detached* as in the first case and for the others



- If $k_i \leq 0$ for all $i = 1 \dots n$, then those $k_i = 0$ are just *detached* as in the first case and for the others



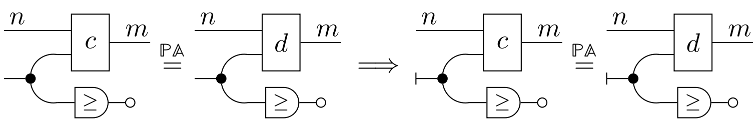
- If some $k_i \geq 0$, and some $k_i \leq 0$, then those $k_i = 0$ are just *detached* as in the first case. For the others, assume without loss of generality that the first j are positive and the remaining ones are negative.



The rest of the proof goes as the original one. ◀

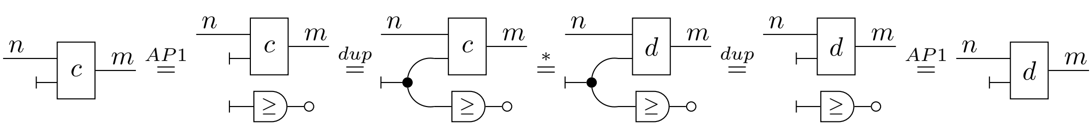
B Proofs of Section 6

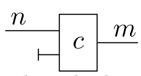
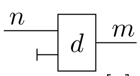
Proof of Lemma 22. To prove 1., first notice that



(*)

Then it is enough to show that



For 2., notice that  and  denote two non-empty polyhedra. Then one can conclude immediately by Lemma 22 in [5] and Theorem 8. ◀