1 Introduction

The Minkowski-Weyl Theorem is a theorem of utmost importance in the theory of polyhedra. Ziegler refers to it as the "main theorem." It is a theorem which is easy to intuitively understand, but is surprisingly difficult to prove. Going through the proof in detail, however, is a good excersice because it introduces various other imporant notions and techniques, as well as helps to mentally clarify the basic definitions relevant to the theory.

Polyhedra are "finitely generated" convex subsets of $\mathbb{R}^n$. "Finitely generated" here means that they can be described in a finite manner, in particular as a convex hull of either: a finite set of points and ray, or a finite intersection of halfspaces. The Minkowski-Weyl Theorem states that these two finite representations are "equivalent," that every polyhedra can be described either way.

For a more concrete example, consider a polygon in the plane (provide some nice pictures here).

In this paper, the theorem and relvant definitions will be stated, and a proof will be given. The proof will closely follow Ziegler's, however it is the author's intent to make this paper as accessible as possible (who is the target audience?). Therefore, some concepts will be stated in somewhat more painstaking detail than some may find necessary, however these details will have to be addressed to implement an enumeration-algorithm that will accompany the proof.

Perhaps the most important part of the paper is the diagram in figure 1. This diagram illustrates the main steps of the proof, and will serve as a "roadmap" to the implementation.

2 Definitions

2.1 Informal Notions

Here shall be stated some of the fundamental ideas involved in the theory of polyhedra (formal definitions will be given in the next section).

Convexity Convexity is an essential notion in the study of polyhedra. Here, it will be considered a property of subsets of $\mathbb{R}^n$. Let $A \subseteq \mathbb{R}^n$. A is *convex* if, given any two points of A, the line segment connecting these two points is also contained in A. It formalizes a notion of being "filled-in," and implies that (given the standard metric) the shortest path between any two points of A is included in A.

Convex Hulls Associated with the notion of convexity is the operation of *convex hull*. Given any subset $B \subseteq \mathbb{R}^n$, the *convex hull* of B is the smallest set which contains B and is convex. For two isolated points, their convex hull is the line segment with the given points as endpoints. For more complex sets, it can be thought of as in iterative procedure where the convex hull of every

two points is added to the set, repeating this operation until no more points are added. (a picture showing how the procedure must iterate).

Minkowski Sums There are a number of ways to "combine" two sets together in a meaningful way. The most common include union, intersection, and product (sometimes called direct sum). Another, less common, may be disjoint union. In a space with a binary operator $+$, you may also consider a translation of a set, given by adding an element to every member of a set. The Minkowski Sum is an aggregated translation, where instead of translating a set with one element, you translate a set with every member of one set, and then take the union of all these translations.

Projections, Lifts, Relaxations, Restrictions These terms will be will be used informally in this paper, however when one is needed in a formal context it will be specified as necessary.

Projections are very simple operations, which basically cause one to "ignore" the coordinates of a set for a given index set of coordinates. For example, $\pi : (x, y) \mapsto (x)$ is a projection from $\mathbb{R}^2$ to $\mathbb{R}$. Projections are often given a subscript to indicate which dimensions are being "projected to," or a superscript to indicate which dimensions are being "projected away." A formal notation for linear projections is provided below.

A *lift* refers to taking a set and representing it in a higher dimension. For example, one may take the unit interval $I = [0, 1] \subseteq \mathbb{R}^1$ and *lift* it to $\mathbb{R}^2$ by considering $I \times 0$.

Relaxations and Restrictions: A *relaxation* of a set will typically be a way to let a set "grow," however it will typically be in a "controlled" or "reversible" manner. These "reversals" will be referred to as *restrictions*. Take, for instance, this relaxation of the unit interval $I = [0, 1]$:

$$\bigcup_{t \geq 0} [0, t] \times t \quad = \quad \{(x, y) \mid x \geq 0\} \cap \{(x, y) \mid y - x \geq 0\}$$

From this relaxation (an "infinite triangle" of sorts) we may recover I with a respective restriction, in this case by intersecting the relaxation with $\{(x, y) \mid y = 1\}$ and projecting it to the x axis. Again, the terms *relaxation* and *restriction* will be used informally, while specific instances will be specified as needed.

2.2 Formal Definitions

Here are the formal defintions required for the statement of the theorem and its proof. In all definitions, unless otherwise stated, I represents and arbitray *finite* index set, and J is an *arbitrary* index set.

Convex Combination Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\lambda \in [0, 1]$. Then

$$\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}$$

is called a *convex combination* of $\mathbf{x}$ and $\mathbf{y}$. More generally, $\mathbf{x}_i \in \mathbb{R}^n$, $\lambda_i \in [0,1]$, and $\sum_i \lambda_i = 1$. Then

$$\sum_i \lambda_i \mathbf{x}_i$$

is called a *convex combination* of $\mathbf{x}_i$.

Convex Let $A \subseteq R^n$. A is by definition *convex* if every convex combination of its members is again a member. In symbols:

$$\mathbf{x}_i \in A \Rightarrow \sum_i \lambda_i \mathbf{x}_i \in A$$

Note: Let $(\forall j \in J)\, A_j \subseteq \mathbb{R}^n$ be convex, and $(\forall i \in I)\, \mathbf{x}_i, \in \bigcap_{j \in J} A_j$. Then

$$(\forall j \in J)\ \sum_i \lambda_i \mathbf{x}_i \in A_j \Rightarrow \sum_i \lambda_i \mathbf{x}_i \in \bigcap_{j \in J} A_j$$

This implies that $\bigcap_{i \in I} A_i$ is also convex. In other words, the property of being convex is closed under the operation of intersection.

Halfspace Let $\mathbf{y} \in \mathbb{R}^n$, $c \in \mathbb{R}$. Then the set

$$\{\mathbf{x} \mid \langle \mathbf{x}, \mathbf{y} \rangle \leq c\}$$

is called a *halfspace*. In particular, H_k shall denote the halfspace given by

$$\{\mathbf{x} \mid \langle \mathbf{x}, \mathbf{e}_k \rangle \leq 0\} = \{\mathbf{x} \mid x_k \leq 0\}$$

Note: Suppose that $\langle \mathbf{x}, \mathbf{y} \rangle \leq c$, $\langle \mathbf{z}, \mathbf{y} \rangle \leq c$, and $\lambda \in [0,1]$. Then

$$\langle \lambda \mathbf{x} + (1-\lambda)\mathbf{z}, \mathbf{y} \rangle = \lambda \langle \mathbf{x}, \mathbf{y} \rangle + (1-\lambda)\langle \mathbf{z}, \mathbf{y} \rangle \leq \lambda \cdot c + (1-\lambda) \cdot c = c$$

This means that halfspaces are *convex*.

Hyperplane Let $\mathbf{y} \in \mathbb{R}^n$, $c \in \mathbb{R}$. Then the set

$$\{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{y} \rangle = c\}$$

is called a *hyperplane*. This is the precisely the boundary of a halfspace. Note: A family of useful hyperplanes come from projections, i.e.

$$\{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{e}_i \rangle = c\} = \{\mathbf{x} \in \mathbb{R}^n \mid x_i = c\}$$

H-polyhedron An *H-polyhedron* is a finite intersection of halfspaces. By the remark given under the definition of **convexity**, *H-polyhedra are convex.* Here is a useful way of "writing down" an H-polyhedron:

$$\{\mathbf{x} \in \mathbb{R}^n \mid \forall i : \langle \mathbf{x}, \mathbf{a}_i \rangle \leq b_i\}$$

If all the $\mathbf{a}_i$ are gathered into a matrix A (where the $\mathbf{a}_i$ become the rows of A), this is sometimes also written as:

$$\{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$$

Convex Hull Let $A \subseteq \mathbb{R}^n$. B is by definition the *convex-hull* of A if

$$B = \text{conv}(A) = \{\mathbf{x} \mid \exists \mathbf{a}_i \in A, \exists \lambda_i \in [0,1] : \sum_i \lambda_i = 1,\ \mathbf{x} = \sum\nolimits_i \lambda_i \mathbf{a}_i\}$$

Note: Another useful, equivalent definition of *convex hull* can be given as:

$$B' = \text{conv}'(A) = \bigcap \{C \mid A \subseteq C,\ C \text{ is convex}\}$$

This more clearly illustrates the idea of "smallest convex set containing A." To see that these definitions are equivalent, first note that B is convex, and $A \subseteq B$, so $B' \subseteq B$ (since B will be included in the intersection). Then note that if C is convex and $A \subseteq C$, then any convex combination of members of A must (by definition) be in C, so it will be in every such C and therefore in their intersection, so $B' \subseteq B$.

Conical Hull Let $\cup_{i \in I} \mathbf{a}_i = A \subseteq \mathbb{R}^n$, then B is by definition the *conical hull* of A if:

$$B = \text{cone}(A) = \{\mathbf{x} \in \mathbb{R}^n \mid\ \exists t_i \in \mathbb{R}^n \mid\ t_i \geq 0,\ \mathbf{x} = \sum_{i \in I} t_i \mathbf{a}_i\}$$

Note: A conical hull is convex. Let K, $(\forall i \in I)\, J_i$ be finite index sets (i.e. J_i is a collection of finite index sets, indexed by the finite set I), $(\forall i \in I)(j \in J_i) : t_{i,j} \geq 0$, $\mathbf{a}_{i,j} \in A$, and, as usual, $\lambda_i \in [0,1]$, $\sum_i \lambda_i = 1$. Then

$$\sum\nolimits_i \lambda_i \left(\sum\nolimits_{j \in J_i} t_{i,j} \mathbf{a}_{i,j} \right) = \sum\nolimits_{i,j \in J_i} \lambda_i \cdot t_{i,j} \mathbf{a}_{i,j}$$

is a "conical combination" of vectors in A (because $\lambda_i \cdot t_{i,j} \geq 0$), so every convex combination from cone(A) is again in cone(A).

Minkowski Sum Let $P, Q \subseteq \mathbb{R}^n$. Then the *Minkowski Sum* of P and Q is given by:

$$P \oplus Q = \{p + q \mid p \in P,\ q \in Q\}$$

Note: If both P and Q are convex, then so is $P \oplus Q$. Indeed, let $p_i \in P$, $q_i \in Q$, then

$$\sum\nolimits_i \lambda_i (p_i + q_i) = \sum\nolimits_i \lambda_i p_i + \sum\nolimits_i \lambda_i q_i = \bar{p} + \bar{q}$$

Where $\bar{p} \in P$ and $\bar{q} \in Q$ (because they are convex), so $\bar{p} + \bar{q} \in P \oplus Q$.

Linear Transforms Let $T : \mathbb{R}^n \to \mathbb{R}^n$ satisfy:

$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \alpha \in \mathbb{R}$,
$T(\alpha \mathbf{x} + \mathbf{y}) = \alpha T\mathbf{x} + T\mathbf{y}$

Then T is called a *linear transform*. By induction:

$$T\left(\sum_i \alpha_i \mathbf{x}_i\right) = \sum_i \alpha_i T(\mathbf{x}_i)$$

Note: Let $A, B \subseteq \mathbb{R}^n$, $\alpha_i \geq 0$, $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$. Then as a consequence of this definition,

$$\begin{aligned}
T(A \oplus B) &= \{T(\mathbf{x}+\mathbf{y}) \mid \mathbf{x} \in A,\, \mathbf{y} \in B\} \\
&= \{T(\mathbf{x}) + T(\mathbf{y}) \mid \mathbf{x} \in A,\, \mathbf{y} \in B\} \\
&= T(A) \oplus T(B) \\
T(\text{cone}(A)) &= \left\{T\left(\sum_i \alpha_i \mathbf{a}_i\right) \;\middle|\; \mathbf{a}_i \in A\right\} \\
&= \left\{\sum_i \alpha_i T(\mathbf{a}_i) \;\middle|\; \mathbf{a}_i \in A\right\} \\
&= \text{cone}(T(A)) \\
T(\text{conv}(A)) &= \left\{T\left(\sum_i \lambda_i \mathbf{a}_i\right) \;\middle|\; \mathbf{a}_i \in A\right\} \\
&= \left\{\sum_i \lambda_i T(\mathbf{a}_i) \;\middle|\; \mathbf{a}_i \in A\right\} \\
&= \text{conv}(T(A))
\end{aligned}$$

Linear Projections Let $\mathbf{u} \in \mathbb{R}^n$ be a unit vector, and $I \in \mathbb{R}^{n \times n}$ be the identity matrix. Then

$$\pi^{\mathbf{u}} : \mathbb{R}^n \to \mathbb{R}^n = \mathbf{x} \mapsto I - \mathbf{u}\mathbf{u}^T\mathbf{x}$$

Is a *linear projection induced by* $\mathbf{u}$. It is so-called because it is a *linear* transform:

$$\pi^{\mathbf{u}}(\alpha\mathbf{x} + \mathbf{y}) = \alpha\pi^{\mathbf{u}}\mathbf{x} + \pi^{\mathbf{u}}\mathbf{y}$$

and a projection:

$$\pi^{\mathbf{u}} \circ \pi^{\mathbf{u}} = \pi^{\mathbf{u}}$$

When $\mathbf{u} = \mathbf{e}_k$, then $\pi^{\mathbf{e}_k}$ is written π^k. Another family of useful *linear projections* are those given by

$$\pi_k = I - \sum_{i \neq k} \mathbf{e}_i \mathbf{e}_i^T$$

Note that:

$$\langle \mathbf{e}_j, \pi_k \mathbf{x} \rangle = \delta_{jk} x_j$$

Projections are discussed in more detail at the end of the paper.

V-Polyhedron A *V-Polyhedron* is a subset $\mathcal{P}$ of $\mathbb{R}^n$ given by

$$\mathcal{P} = \text{cone}(\mathcal{U}) \oplus \text{conv}(\mathcal{V})$$

where $\mathcal{U}$ and $\mathcal{V}$ are both finite (possibly empty) subsets of $\mathbb{R}^n$. By the remark given under the definition of Minkowski Sums, *V-Polyhedra are convex*.

We are now prepared to state the Minkowski-Weyl Theorem.

Minkowski-Weyl Theorem Every V-Polyhedron is an H-Polyhedron, and every H-Polyhedron is a V-Polyhedron. This permits the use of the term "Polyhedron" unambiguously.

Note: I've read that "definitions should be hard and theorems should be easy." Perhaps there is a bit of this involved in the Minkowski-Weyl Theorem, but an unfortunate side effect is that the significance of a theorem may not be immediate. The Minkowski-Weyl Theorem is a "representation theorem," and is a consequence of a type of duality that exists between the V and H –type definitions of polyhedra.

For a quick demonstration of why a dual representation is important for these types of sets, consider the following two tasks:

- Given a point $\mathbf{x}$, determine if it is a member of the set
- Produce an arbitrary member of the set

These two fundamental tasks tell completely different stories given one representation as opposed to the other. For an H-Polyhedron, determining membership is straightforward and relatively inexpensive: for each halfspace constraint, check to see that it is fulfilled. For a V-Polyhedron, you would need to either find a combination that produces the point, or provide a "certificate" that demonstrates that it is not in the set. On the other hand, given an H-Polyhedron, producing a member of the set is not straightforward at all. Given a list of half-spaces, it isn't even obvious (or necessarily cheap to determine) that their intersection is not empty! A V-Polyhedron, on the other hand, is given by members of the set, so the task of providing a member (or determining if the set is empty or not) is already done for you. Observe that these tasks are facilitated by the qualifiers used to describe the sets (the qualifier $\forall$ is used for the H-Polyhedron, while $\exists$ is used for the V-Polyhedron).

It should be noted that guaranteeing the existence of two representations does not promise an efficient method of creating one from the other. The proof of the Minkowski-Weyl Theorem indicates an enumeration algorithm, and an implementation will be provided. This enumeration can be inefficient, however.

2.3 Equivalence of Polyhedra

The notion of equivalence is worth a small discussion. Consider the two polyhedra $I = [0, 1]$ and $I \times \{0\}$ (i.e. the unit interval in $\mathbb{R}^1$ and the unit interval in $\mathbb{R}^2$). It would be unfortunate if the theory of polyhedra didn't consider these two entities as *equivalent* in some manner. The basic operation of equivalence in the theory is the *affine transformation*. Then polyhedra are considered equivalent if there is an affine transformation from one to the other, which is bijective *when restricted to the second polyhedron*. In our setting, it is enough to consider a subset of the affine transformations: *linear projections*. For example, $\pi^1 : (x, y) \mapsto (x)$. Then we have $\pi^1(I \times \{0\}) = I$. Note that this projection, when restricted to I (and, more generally, the x–axis) is bijective. Therefore,

given our notion of equivalence, we can conclude that I and $I \times \{0\}$ are equivalent. As an anti–example, consider $I \times I$ (i.e. the unit square in $\mathbb{R}^2$). While $\pi^1(I \times I) = I$, this mapping is not bijective when restricted to I. Observe that $(0,0) \mapsto (0)$, and $(0,1) \mapsto (0)$, so the projection fails to be injective on I. To indicate why this restricted definition of equivalence may not be suitable for a more robust theory, note that the two line segments $[0,1]$ and $[0,2]$ are not equivalent (given our defintion), even though they are geometrically congruent. Considering the more general affine transformations resolves this problem, but is not necessary here (mention projective transformations and give a reference for further reading).

2.4 Notation

Here, some notation will be introduced that will simplify the proof and help to clarify the ideas involved. Since this is a section on notation, it is essentially just saying the same thing over and over again, so it may be a bit dry.

Matrix Multiplication Some "notational tricks" from the toolbox of linear programming will be needed for the proof of the Minkowski-Weyl Theorem, in particular, useful ways of manipulating expressions involving multiplying matrices with a vector. First, let $A \in \mathbb{R}^{l \times m}, B \in \mathbb{R}^{l \times n}, \mathbf{a}, \mathbf{x} \in \mathbb{R}^m, \mathbf{b}, \mathbf{y} \in \mathbb{R}^n$. Then let $(\mathbf{x}, \mathbf{y}), (\mathbf{a}, \mathbf{b}) \in \mathbb{R}^{m+n}$ be given in the obvious way, i.e:

$$\langle (\mathbf{x}, \mathbf{y}), \mathbf{e}_k \rangle = \begin{cases} x_k & 1 \leq k \leq m \\ y_{k-m} & m+1 \leq k \leq m+n \end{cases}$$

In a similar fashion, define $(A\,B)$ by:

$$(A\,B)_{i,j} = \begin{cases} A_{i,j} & 1 \leq j \leq m \\ B_{i,j-m} & m+1 \leq j \leq m+n \end{cases}$$

(Note that these are just the simple "concatenation" operations done row-wise). With these definitions, it is possible to write:

$$A\mathbf{x} + B\mathbf{y} = (A\,B) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$$

Sometimes, equations given separately will be combined into larger systems. In order to do so, we need to be able to stack matrices on top of each other. Let $A' \in \mathbb{R}^{l' \times m}$. Then the matrix $\left(\begin{smallmatrix} A \\ A' \end{smallmatrix}\right)$ is given by:

$$\begin{pmatrix} A \\ A' \end{pmatrix}_{i,j} = \begin{cases} A_{i,j} & 1 \leq i \leq l \\ A'_{i-l,j} & l+1 \leq i \leq l+l' \end{cases}$$

As an example, say $\mathbf{c} \in \mathbb{R}^l, \mathbf{c}' \in \mathbb{R}^{l'}$, then we can do the rewrite:

$$A\mathbf{x} = \mathbf{c}, A'\mathbf{x} = \mathbf{c}' \rightarrow \begin{pmatrix} A \\ A' \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{c} \\ \mathbf{c}' \end{pmatrix}$$

Another example uses the identity $\mathbf{x} = \mathbf{c} \Leftrightarrow \mathbf{x} \leq \mathbf{c} \wedge -\mathbf{x} \leq -\mathbf{c}$:

$$A\mathbf{x} = \mathbf{c} \rightarrow \begin{pmatrix} A \\ -A \end{pmatrix} \mathbf{x} \leq \begin{pmatrix} \mathbf{c} \\ -\mathbf{c} \end{pmatrix} \rightarrow \begin{pmatrix} A \\ -A \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{c} \\ -\mathbf{c} \end{pmatrix} \leq 0 \rightarrow \begin{pmatrix} -\mathbf{c} & A \\ \mathbf{c} & -A \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \leq 0$$

Dot Product Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, then

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum\nolimits_i x_i y_i$$

It is worth noting that matrix multiplication can be defined in terms of the dot product. Let $A \in \mathbb{R}^{m \times n}$, and let $\mathbf{a}_i : 1 \leq i \leq m$ be the rows of A.

$$A\mathbf{x} = \begin{pmatrix} \langle \mathbf{a}_1, \mathbf{x} \rangle \\ \langle \mathbf{a}_2, \mathbf{x} \rangle \\ \vdots \\ \langle \mathbf{a}_m, \mathbf{x} \rangle \end{pmatrix}$$

Also observe that we can "split up" dot products with our projection tranforms:

$$\begin{aligned} \langle \pi_k(\mathbf{x}), \pi_k(\mathbf{a}) \rangle &= x_k a_k \\ \langle \pi^k(\mathbf{x}), \pi^k(\mathbf{a}) \rangle &= \sum\nolimits_{i \neq k} x_i a_i \\ \langle \mathbf{x}, \mathbf{a} \rangle &= \sum\nolimits_i x_i a_i \\ &= x_k a_k + \sum\nolimits_{i \neq k} x_i a_i \\ &= \langle \pi_k(\mathbf{x}), \pi_k(\mathbf{x}) \rangle + \langle \pi^k(\mathbf{x}), \pi^k(\mathbf{a}) \rangle \end{aligned}$$

3 Proof of the Minkowski-Weyl Theorem

3.1 Overview

The proof shall proceed by first considering an H-Polyhedron $\mathcal{P}_\mathcal{H}$, and constructing from it a V-Polyhedron $\mathcal{P}_\mathcal{V}$ which represents the same set of points, then starting with a V-Polyhedron $\mathcal{P}_\mathcal{V}$ and constructing an H-Polyhedron $\mathcal{P}_\mathcal{H}$. The high-level steps are almost identical, and is illustrated in Figure 1.

First, $\mathcal{P}_\mathcal{H}$ will be relaxed to a cone $\mathcal{C}_\mathcal{H}$. While technically unnecessary, in the steps that follow it is convenient to have a cone, as opposed to a more general polyhedron. $\mathcal{C}_\mathcal{H}$ is then immediately lifted to a higher dimension so that it can be directly represented as a V-Cone $\mathcal{C}_\mathcal{V}$ (have we defined V-Cone?). It is at this point that the "real work" must be done (which is why this arrow is colored red). This step requires intersecting the V-Cone with a number of hyperplanes, and requires a process which is dual to the so-called "Fourier-Motzkin Elimination." At this point, a simple observation will allow us to restrict $\mathcal{C}_\mathcal{V}$ to get us to $\mathcal{P}_\mathcal{V}$, completing this direction of the proof. Note that this restriction is the opposite of the relaxation which occurs at the beginning of this direction of the proof, as the colors of the diagram reflect.

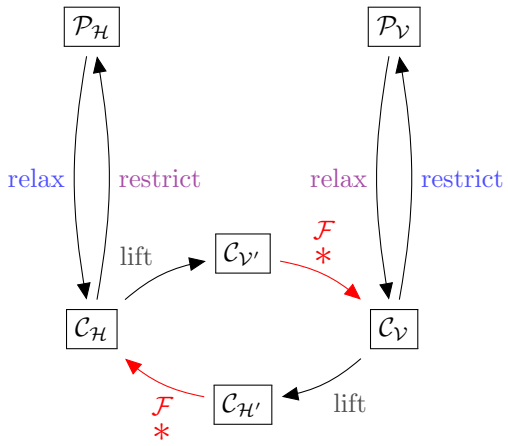

Figure 1: Diagram of the proof

For the other direction, the exact same steps are taken, however the relaxing and lifting that occur are designed to convert the V-Polyhedron into a V-Cone and H-Cone, instead of the other way around[1]. Also, the full fledged "Fourier-Motzkin Elimination" shall take place, this time because we need to project an H-Cone down a number of dimensions.

The transformation which go from $V \to H$ and $H \to V$ representations in the proof seem like a bit of trickery, but in earnest these come from the field of Linear Programming. For more information on these types of transformations, and a more systematic treatment of them, see (that book by Matusek...), for instance.

Finally, before we get started, I'd like to mention that Ziegler provides the questions:

- "Is a polyhedron intersected with a hyperplane also a polyhedron?"
- "Is the projection of a polyhedron also a polyhedron?"

as examples of the utility of the two different representations. The answer to the first question is clear for H-Polyhedra, this is merely adding another constraint to the system, while it is not so clear for a V-Polyhedra (it seems as though you must somehow "solve" something to prove this...) Similarly, the second question is clear for V-Polyhedra, you simply take the vectors that you already have and forget about some of the coordinates, while for H-Polyhedra it again seems like some system needs to somehow be solved in order to prove this statement. Before you go and try to prove it yourself, know that the "solving" of these problems is essentially Fourier-Motzkin Elimination, and is the bulk of the work in the proof that follows.

It should then be of no surprise that the actual proof is essentially just elucidating this diagram.

3.2 H-Polyhedra $\rightarrow$ V-Polyhedra

Overview As Figure 1 indicates, the proof in this direction will generally be of the following form:

$$\mathcal{P}_{\mathcal{H}} \rightarrow \mathcal{C}_{\mathcal{H}} \rightarrow \mathcal{C}_{\mathcal{V}'} \rightarrow \mathcal{C}_{\mathcal{V}} \rightarrow \mathcal{P}_{\mathcal{V}}$$

It should then be of no surprise that the actual proof is essentially just elucidating this diagram.

3.2.1 Relax: $\mathcal{P}_{\mathcal{H}} \rightarrow \mathcal{C}_{\mathcal{H}}$

This is a straightforward step. Let $\mathcal{P}_{\mathcal{H}} = \{x : A\mathbf{x} \leq \mathbf{b}\}$ for some A and $\mathbf{b}$. To achieve the form of a cone, the right hand side of the inequality needs to be 0. In order to do this, we prepend $\mathbf{b}$ to the matrix A, and introduce a new variable as follows:

$$A\mathbf{x} \leq \mathbf{b} \rightarrow [\mathbf{b}|A] \begin{pmatrix} x_0 \\ \mathbf{x} \end{pmatrix} \leq 0$$

Observe that this is in fact a relaxation to an H-Cone. The restriction back to the original H-Polyhedron is to intersect the H-Cone with the hyperplane $\{\mathbf{x} : x_0 = 1\}$, that is:

$$\mathcal{P}_{\mathcal{H}} = \left\{ [\mathbf{b}|A] \begin{pmatrix} x_0 \\ \mathbf{x} \end{pmatrix} \leq 0 \right\} \cap \{\mathbf{x} : x_0 = 1\}$$

For now, we deal with the relaxed cone $\mathcal{C}_{\mathcal{H}}$ and do this restriction as the last step in the proof of this direction. In what follows, $A' = [\mathbf{b}|A]$ and $\mathbf{x}' = \begin{pmatrix} x_0 \\ \mathbf{x} \end{pmatrix}$. This notation will simplify the expressions that follow.

3.2.2 Lift: $\mathcal{C}_{\mathcal{H}} \rightarrow \mathcal{C}_{\mathcal{V}'}$

The task is to *somehow* go from dealing with halfspaces to dealing with rays. This step is not exactly straighforward, however the technique used here is systematic from the point of view of linear programming. Let A^i denote the i-th column of A. Observe that $A\mathbf{x} = \sum_i A^i \cdot x_i$. The significance of this is that matrix multiplication can be considered in two fundamentally different ways, as either: a number of dot products $\langle \mathbf{a}_i, \mathbf{x} \rangle$, or as a linear combination of column vectors $\sum_i A^i \cdot x_i$. This important difference is key to representing the inequality above as a cone. Suppose that $A \in \mathbb{R}^{m \times n}$, i.e. that A as m rows, and consider the vector $x_i \cdot \begin{pmatrix} \mathbf{e}_i \\ A^i \end{pmatrix} \in \mathbb{R}^{n+m}$. Here, the first n rows keep track of the *value* and *position* of the contribution of x_i (a scalar) to the vector $\mathbf{x}$, while the bottom m rows correspond to the contribution of x_i to the *sum* $\sum_i A^i \cdot x_i$.

Let $\pi_{\mathbf{x}} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$ be a projection to the first n coordinates, and $\pi^{\mathbf{x}} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$ a projection to the last m coordinates. Now, consider the sum

$$\sum_i x_i \begin{pmatrix} \mathbf{e}_i \\ A^i \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ A\mathbf{x} \end{pmatrix}$$

This equality follows from the discussion that precedes it, and it is quite significant, mostly because:

$$\pi_{\mathbf{x}}\begin{pmatrix}\mathbf{x}\\ A\mathbf{x}\end{pmatrix} = \mathbf{x}, \quad \pi^{\mathbf{x}}\begin{pmatrix}\mathbf{x}\\ A\mathbf{x}\end{pmatrix} = A\mathbf{x}$$

(I think that switching the notation of "projection to" and "projection from" would be in order) Without further adieu, consider the cone:

$$\mathcal{C}_0 = \text{cone}\left(\left\{\pm\begin{pmatrix}\mathbf{e}_i\\ A^i\end{pmatrix}\right\}\right)$$

What are the members of this cone? Say $\left(\begin{smallmatrix}\mathbf{x}\\ \mathbf{z}\end{smallmatrix}\right) \in \mathcal{C}_0$, then this vector must be a positive linear combination of its generators (define generators). In other words:

$$\exists\, t_i^+, t_i^- \geq 0 \,\Big|\, \begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} = \sum\nolimits_i t_i^+ \cdot \begin{pmatrix}\mathbf{e}_i\\ A^i\end{pmatrix} + t_i^- \cdot -\begin{pmatrix}\mathbf{e}_i\\ A^i\end{pmatrix} = \sum\nolimits_i (t_i^+ - t_i^-) \cdot \begin{pmatrix}\mathbf{e}_i\\ A^i\end{pmatrix}$$

Letting $t_i = (t_i^+ - t_i^-)$ in the above equations, and noting that t_i can range over all of $\mathbb{R}$, we get:

$$\exists t_i \in \mathbb{R} \,\Big|\, \begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} = \sum\nolimits_i t_i \cdot \begin{pmatrix}\mathbf{e}_i\\ A^i\end{pmatrix} \Rightarrow \mathbf{z} = A\mathbf{x}$$

To recap what has been accomplished, we have created a V-Cone, in which the first n coordinates keep track of some values corresponding to a vector we've been calling $\mathbf{x}$, and the final m coordinates keep track of the result of the multiplication $A\mathbf{x}$. This V-Cone becomes powerful when we fix $\mathbf{z}$, that is, taking the V-Cone's intersection with a set of the form $\{\left(\begin{smallmatrix}\mathbf{x}\\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, \mathbf{z} = \mathbf{b}\}$ for some carefully chosen vector $\mathbf{b}$. Then, we take a projection of this new set formed by intersection, and recover useful values of $\mathbf{x}$, that is:

$$\pi_{\mathbf{x}}\left(\mathcal{C}_0 \cap \left\{\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} \in \mathbb{R}^{m+n} \,\Big|\, \mathbf{z} = \mathbf{b}\right\}\right) = \{\mathbf{x} \mid A\mathbf{x} = \mathbf{b}\}$$

In this way, we can recover the solutions to the equation $A\mathbf{x} = \mathbf{b}$. As previously mentioned, the projection of a V-Cone is easy to deal with, just restrict the generators to the desired coordinates, but the intersection is a bit trickier to deal with, and is the subject of the next part of the proof. There is one more loose end to tie up before we go on.

As of right now, we have an alleged way to get the solutions of $A\mathbf{x} = \mathbf{b}$ from a V-Cone, but what we really need are the solutions to $A\mathbf{x} \leq \mathbf{b}$. There is a quick and dirty way to get this from $\mathcal{C}_0$, through introducing what are known in linear programming circles as *slack variables*. This is just a form of relaxation.

Suppose that you have an $\mathbf{x}$ such that $A\mathbf{x} \leq \mathbf{b}$, and let $\mathbf{w} = \mathbf{b} - A\mathbf{x}$. This $\mathbf{w}$ can be thought of as *slack*, and is the key to getting from $\mathcal{C}_0$ to $\mathcal{C}_{\mathcal{V}'}$. The key property of this slack is $0 \leq \mathbf{w}$, so $\mathbf{w}$ can be written as

$$\mathbf{w} = \sum\nolimits_j w_j \cdot \mathbf{e}_j \Rightarrow \exists\, w_j \geq 0 \,\Big|\, \mathbf{w} = \sum\nolimits_j w_j \cdot \mathbf{e}_j$$

Now, we add these bases vectors $\mathbf{e}_j$ to the generators of $\mathcal{C}_0$ to create $\mathcal{C}_{\mathcal{V}'}$:

$$\mathcal{C}_{\mathcal{V}'} = \text{cone}\left(\left\{\pm\begin{pmatrix}\mathbf{e}_i \\ A^i\end{pmatrix}\right\} \cup \left\{\begin{pmatrix}\mathbf{0} \\ \mathbf{e}_j\end{pmatrix}\right\}\right)$$

As before, we fix the last m coordinates, carefully choosing our $\mathbf{b}$ to be $\mathbf{0}$, and again projecting to the coordinates that interest us:

$$\pi_{\mathbf{x}}\left(\mathcal{C}_{\mathcal{V}'} \cap \left\{\begin{pmatrix}\mathbf{x} \\ \mathbf{z}\end{pmatrix} \in \mathbb{R}^{m+n} \,\middle|\, \mathbf{z} = \mathbf{0}\right\}\right) = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{0}\}$$

3.2.3 Drop: $\mathcal{C}_{\mathcal{V}'} \to \mathcal{C}_{\mathcal{V}}$

Now we turn to intersecting $\mathcal{C}_{\mathcal{V}'}$ with $\{\left(\begin{smallmatrix}\mathbf{x} \\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, \mathbf{z} = \mathbf{0}\}$. This requires a pretty good idea, and is again not straightforward. Remembering that $\mathbf{z} \in \mathbb{R}^m$, and letting $M = \{1, 2, \ldots, m\}$, the first step is to note that:

$$\left\{\begin{pmatrix}\mathbf{x} \\ \mathbf{z}\end{pmatrix} \in \mathbb{R}^{m+n} \,\middle|\, \mathbf{z} = \mathbf{0}\right\} = \bigcap_{k \in M} \left\{\begin{pmatrix}\mathbf{x} \\ \mathbf{z}\end{pmatrix} \in \mathbb{R}^{m+n} \,\middle|\, z_k = 0\right\}$$

This suggests that, instead of taking $\mathcal{C}_{\mathcal{V}'}$ and intersecting it with $\{\left(\begin{smallmatrix}\mathbf{x} \\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, \mathbf{z} = \mathbf{0}\}$ all at once, we intersect it with $\{\left(\begin{smallmatrix}\mathbf{x} \\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, z_k = 0\}$ for each $k \in M$ one at a time. This greatly simplifies our task of creating a cone to represent $\mathcal{C}_{\mathcal{V}'} \cap \{\left(\begin{smallmatrix}\mathbf{x} \\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, \mathbf{z} = \mathbf{0}\}$ to finding a cone that represents $\mathcal{C}_{\mathcal{V}'} \cap \{\left(\begin{smallmatrix}\mathbf{x} \\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, z_k = 0\}$, and applying this construction inductively.

The basic idea is as follows. Say we are given to numbers x and y, and the equation: $\alpha x - \beta y = 0$, where we are to choose α and β so as to satisfy it. One obvious choice would be to pick $\alpha = \beta = 0$. This choice is pretty trivial, and doesn't really say anything about x or y. Another, slightly more clever choice, would be $\alpha = y$ and $\beta = x$.

Now say we have two vectors $\mathbf{x}$ and $\mathbf{y}$, such that $x_k > 0$ and $y_k < 0$, and are given the expression $\alpha\mathbf{x} + \beta\mathbf{y}$, where we are to choose $\alpha \geq 0$ and $\beta \geq 0$ such that the resulting vector is 0 in the k-th coordinate. Again, $\alpha = \beta = 0$ works, but it isn't too helpful. A more useful choice would be $\alpha = -y_k$ and $\beta = x_k$.

Now to intersect our cone $\mathcal{C}_{\mathcal{V}'}$ with $\{\left(\begin{smallmatrix}\mathbf{x} \\ \mathbf{z}\end{smallmatrix}\right) \in \mathbb{R}^{m+n} \,|\, z_k = 0\}$. The first step is to partition the generators of $\mathcal{C}_{\mathcal{V}'}$. Say $\mathcal{C}_{\mathcal{V}'}$ is given by $\mathcal{C}_{\mathcal{V}'} = \text{cone}(\mathcal{U})$. We want to partition $\mathcal{U}$ into three sets, depending on each vector's value at z_k. In particular, let

$$\begin{aligned}\mathcal{Z} &= \{\mathbf{u} \in \mathcal{U} \mid u_k = 0\} \\ \mathcal{P} &= \{\mathbf{u} \in \mathcal{U} \mid u_k > 0\} \\ \mathcal{N} &= \{\mathbf{u} \in \mathcal{U} \mid u_k < 0\}\end{aligned}$$

(Here there is a good chance to use Minkowski sums, and they may make some of the reasoning more clear and concise) These sets *partition* $\mathcal{U}$, i.e. $\mathcal{U} = \mathcal{Z} \cup \mathcal{P} \cup \mathcal{N}$, and each $\mathcal{Z}, \mathcal{P}, \mathcal{N}$ are pairwise disjoint. This means, for any vector $\mathbf{v} \in \text{cone}(\mathcal{U})$, we have:

$$\mathbf{v} = \sum\nolimits_{l \in \mathcal{Z}} t_l \mathbf{u}^l + \sum\nolimits_{i \in \mathcal{P}} t_i \mathbf{u}^i + \sum\nolimits_{j \in \mathcal{N}} t_j \mathbf{u}^j$$

Note that the superscript notation $\mathbf{u}^i$ is merely meant to index the vectors. This makes it easier to refer to the k-th coordinate of the i-th vector from $\mathcal{P}$ as u^i_k. What we would like to have is $\mathbf{v}$ expressed in terms of vectors whose k-th coordinate is already 0, so we need to construct these vectors from $\mathcal{Z}$, $\mathcal{P}$, and $\mathcal{N}$. First note that the vectors in $\mathcal{Z}$ already have this property, so we can forget about them for a second. Next, suppose that $v_k = 0$, and consider the contribution from $\sum_{i\in\mathcal{P}} t_i\mathbf{u}^i + \sum_{j\in\mathcal{N}} t_j\mathbf{u}^j$ to the k-th coordinate. We must have that:

$$\sum\nolimits_{i\in\mathcal{P}} t_i u^i_k + \sum\nolimits_{j\in\mathcal{N}} t_j u^j_k = 0 \quad \Rightarrow \quad \sum\nolimits_{i\in\mathcal{P}} t_i u^i_k = \sum\nolimits_{j\in\mathcal{N}} -t_j u^j_k$$

This final equality will be the key to our new set of generators, and the sum will be denoted σ. The next step is more easily expressed as equations than in words, so:

$$\sum\nolimits_{i\in\mathcal{P}} t_i\mathbf{u}^i + \sum\nolimits_{j\in\mathcal{N}} t_j\mathbf{u}^j \quad =$$

$$\frac{1}{\sigma}\left(\sum\nolimits_{j\in\mathcal{N}} -t_j u^j_k\right)\sum\nolimits_{i\in\mathcal{P}} t_i\mathbf{u}^i + \frac{1}{\sigma}\left(\sum\nolimits_{i\in\mathcal{P}} t_i u^i_k\right)\sum\nolimits_{j\in\mathcal{N}} t_j\mathbf{u}^j \quad =$$

$$\frac{1}{\sigma}\sum_{\substack{i\in\mathcal{P}\\ j\in\mathcal{N}}} -(t_it_j)u^j_k\mathbf{u}^i + \frac{1}{\sigma}\sum_{\substack{i\in\mathcal{P}\\ j\in\mathcal{N}}} (t_it_j)u^i_k\mathbf{u}^j \quad =$$

$$\frac{1}{\sigma}\sum_{\substack{i\in\mathcal{P}\\ j\in\mathcal{N}}} (t_it_j)(u^i_k\mathbf{u}^j - u^j_k\mathbf{u}^i)$$

All we've done here is multiply by 1, separate, and combine terms, but the result is pretty neat. Since $u^i_k\mathbf{u}^j - u^j_k\mathbf{u}^i$ has 0 in the k-th coordinate, and $t_it_j > 0$, what we have is the orginal contribution of vectors from $\mathcal{P}$ and $\mathcal{N}$ expressed as a positive combination of vectors with 0 in the k-th coordinate, and this is precisely the property that we sought in the beginning. Reconsidering our vector $\mathbf{v}$,

$$\begin{aligned}\mathbf{v} &= \sum\nolimits_{l\in\mathcal{Z}} t_l\mathbf{u}^l + \sum\nolimits_{i\in\mathcal{P}} t_i\mathbf{u}^i + \sum\nolimits_{j\in\mathcal{N}} t_j\mathbf{u}^j = \\ &= \sum\nolimits_{l\in\mathcal{Z}} t_l\mathbf{u}^l + \frac{1}{\sigma}\sum_{\substack{i\in\mathcal{P}\\ j\in\mathcal{N}}} (t_it_j)(u^i_k\mathbf{u}^j - u^j_k\mathbf{u}^i)\end{aligned}$$

If we let:

$$\mathcal{P} \overset{\mathcal{F}_k}{*} \mathcal{N} = \{u^i_k\mathbf{u}^j - u^j_k\mathbf{u}^i \mid \mathbf{u}^i \in \mathcal{P},\, \mathbf{u}^j \in \mathcal{N}\}$$

and quickly set $H_k = \{\binom{\mathbf{x}}{\mathbf{z}} \in \mathbb{R}^{m+n} \mid z_k = 0\}$, then we see:

$$\text{cone}(\mathcal{U}) \cap H_k = \text{cone}(\mathcal{Z}\cup\mathcal{P}\cup\mathcal{N}) \cap H_k = \text{cone}(\mathcal{Z}\cup\mathcal{P}\overset{\mathcal{F}_k}{*}\mathcal{N})$$

As was previously discussed, we can iterate this procedure over every $k \in M$, and in doing so take the set $\mathcal{C}_{\mathcal{V}'} \cap \{\binom{\mathbf{x}}{\mathbf{z}} \in \mathbb{R}^{m+n} \,|\, \mathbf{z} = \mathbf{0}\}$ to a new cone expressed without intersection. After we have this cone, we can project away the coordinates that are always 0 with $\pi^{\mathbf{x}}$ to get a new cone $\mathcal{C}_{\mathcal{V}}$. (It may be worthwhile to give this procedure more explicitly, creating a new operator $\mathcal{F}$ that takes a set, and, for every k, partitions it on k then $\overset{\mathcal{F}_k}{*}$'s it).

3.2.4 Restrict: $\mathcal{C}_{\mathcal{V}} \to \mathcal{P}_{\mathcal{V}}$

There's one last loose end to tie up to finish this direction of the proof. When we relaxed $\mathcal{P}_{\mathcal{H}}$, we introduced a new coordinate, x_0, and now we need to intersect $\mathcal{C}_{\mathcal{V}}$ with $H_1 = \{\binom{x_0}{\mathbf{x}} | x_0 = 1\}$. (I may have forgotten about x_0, in particular when saying things like $\cdots \in \mathbb{R}^{n+m}$). As before, suppose that $\mathcal{C}_{\mathcal{V}} = \text{cone}(\mathcal{U}')$, and partition $\mathcal{U}'$ into $\mathcal{Z}, \mathcal{N}$, and $\mathcal{P}$. Denote $\sum_{\mathbf{u} \in \mathcal{P}} u_0 = \sigma_p$. It will be convenient to consider a "normalized" version of $\mathcal{P}$:

$$\mathcal{V}' = \left\{ \frac{\mathbf{u}^i}{u_0^i} \mid \mathbf{u}^i \in \mathcal{P} \right\}$$

The choice of term "normalized" will be clear shortly. In order to intersect our cone with H_1, we will first recognize that vectors from $\mathcal{Z}$ won't help us reach our hyperplane. Likewise, any contribution from vectors of $\mathcal{N}$ will "cancel out" some contribution from $\mathcal{P}$. With this in mind, consider again a vector $\mathbf{v} \in \mathcal{C}_{\mathcal{V}} \cap H_1$,

$$\begin{aligned}
\mathbf{v} &= \sum\nolimits_{l \in \mathcal{Z}} t_l \mathbf{u}^l + \sum\nolimits_{i \in \mathcal{P}} t_i \mathbf{u}^i + \sum\nolimits_{j \in \mathcal{N}} t_j \mathbf{u}^j \\
1 &= \sum\nolimits_{i \in \mathcal{P}} t_i u_k^i + \sum\nolimits_{j \in \mathcal{N}} t_j u_k^j \;\Rightarrow \\
\sigma_p &= \sum\nolimits_{i \in \mathcal{P}} t_i u_k^i = 1 - \sum\nolimits_{j \in \mathcal{N}} t_j u_k^j
\end{aligned}$$

Similarly as before, we'll try to "simplify" the expression involving vectors from $\mathcal{N}$.

$$\begin{aligned}
&\sum\nolimits_{i \in \mathcal{P}} t_i \mathbf{u}^i + \sum\nolimits_{j \in \mathcal{N}} t_j \mathbf{u}^j \quad = \\
&\frac{1}{\sigma_p} \left(1 - \sum\nolimits_{j \in \mathcal{N}} t_j u_0^j\right) \sum\nolimits_{i \in \mathcal{P}} t_i \mathbf{u}^i + \frac{1}{\sigma_p} \left(\sum\nolimits_{i \in \mathcal{P}} t_i u_0^i\right) \sum\nolimits_{j \in \mathcal{N}} t_j \mathbf{u}^j \quad = \\
&\frac{1}{\sigma_p} \sum_{i \in \mathcal{P}} t_i \mathbf{u}^i + \frac{1}{\sigma_p} \sum_{\substack{i \in \mathcal{P} \\ j \in \mathcal{N}}} -(t_i t_j) u_0^j \mathbf{u}^i + \frac{1}{\sigma_p} \sum_{\substack{i \in \mathcal{P} \\ j \in \mathcal{N}}} (t_i t_j) u_0^i \mathbf{u}^j \quad = \\
&\sum_{i \in \mathcal{P}} \left(\frac{t_i u_0^i}{\sigma_p}\right) \frac{\mathbf{u}^i}{u_0^i} + \frac{1}{\sigma_p} \sum_{\substack{i \in \mathcal{P} \\ j \in \mathcal{N}}} (t_i t_j)(u_0^i \mathbf{u}^j - u_0^j \mathbf{u}^i)
\end{aligned}$$

First, note that the vectors in the second term all belong to $\mathcal{P} \overset{\mathcal{F}_0}{*} \mathcal{N}$ and contribute nothing to the 0-th coordinate. Now consider the sum $\sum_{i \in \mathcal{P}} \frac{t_i u_0^i}{\sigma_p}$ from the first term. Because the 0-th coordinate must sum to 1, the second term contributes nothing, and every vector of the form $\frac{\mathbf{u}^i}{u_0^i}$ has value 1 in the 0-th position, it must be that $\sum_{i \in \mathcal{P}} \frac{t_i u_0^i}{\sigma_p} = 1$. Furthermore, since $t_i, u_0^i, \sigma_p \geq 0$, $\sum_{i \in \mathcal{P}} \frac{t_i u_0^i}{\sigma_p} \frac{\mathbf{u}^i}{u_0^i}$ is actually a *convex combination* of vectors from $\mathcal{V}'$! Reconsidering the $\mathbf{v}$:

$$\begin{aligned}
\mathbf{v} &= \sum\nolimits_{l \in \mathcal{Z}} t_l \mathbf{u}^l + \sum\nolimits_{i \in \mathcal{P}} t_i \mathbf{u}^i + \sum\nolimits_{j \in \mathcal{N}} t_j \mathbf{u}^j \\
&= \sum_{i \in \mathcal{P}} \left(\frac{t_i u_0^i}{\sigma_p} \right) \frac{\mathbf{u}^i}{u_0^i} + \sum\nolimits_{l \in \mathcal{Z}} t_l \mathbf{u}^l + \frac{1}{\sigma_p} \sum_{\substack{i \in \mathcal{P} \\ j \in \mathcal{N}}} (t_i t_j)(u_0^i \mathbf{u}^j - u_0^j \mathbf{u}^i)
\end{aligned}$$

What this shows is that any vector from $\mathcal{C}_\mathcal{V} \cap H_1$ can be written as the sum of vectors from the convex hull of $\mathcal{V}'$ and the conical hull of $\mathcal{U}'' = \mathcal{Z} \cup \overset{\mathcal{F}_\mathcal{P}}{*} \mathcal{N}$. In symbols:

$$\mathcal{C}_\mathcal{V} \cap H_1 = \text{conv}(\mathcal{V}') \oplus \text{cone}(\mathcal{U}'')$$

The final step is to rid ourselves of the 0-th coordinate with π^0:

$$\begin{aligned}
\mathcal{U} &= \pi^0(\mathcal{U}'') \\
\mathcal{V} &= \pi^0(\mathcal{V}') \\
\mathcal{P}_\mathcal{V} &= \text{conv}(\mathcal{V}) \oplus \text{cone}(\mathcal{U})
\end{aligned}$$

And we finally have our V-Polyhedron constructed from $\mathcal{P}_\mathcal{H}$ from the beginning of the proof.

3.2.5 Summary

Here will be presented a terse summary of the proof we've just gone through (see Figure 2). There will be liberal abuse of notation. First, we relax an H-Polyhedron $\{A\mathbf{x} \leq \mathbf{b}\}$ to an H-Cone by adding an extra variable $\mathbf{x}_0$ that "scales" the constraint $\mathbf{b}$. This H-Cone is then lifted to an intersection of many hyperplanes of the form $\{\mathbf{x}_k = 0\}$. This set is then expanded with the operator $\overset{\mathcal{F}_k}{*}$ and projected down with π^k, one extraneous dimension at a time. At the end, the cone $\mathcal{C}_\mathcal{V}$ is restricted back to the original Polyhedron with one last, modified application of $\overset{\mathcal{F}_0}{*}$ and π^0. The final result is a Minkowski Sum of a cone and convex hull $\text{cone}(U) \oplus \text{conv}(V)$. Calling the polyhedra $\mathcal{P}_\mathcal{H}$ and $\mathcal{P}_\mathcal{V}$ equivalent is justified because of the notion of polyhedra equivalence, and the fact that the projections used maintained a bijective equivalence with the original set.

3.3 V-Polyhedra → H-Polyhedra

Overview This part of the paper, much like the last, will bring the diagram to life, but in the other direction. Many of the steps are remarkably similar to

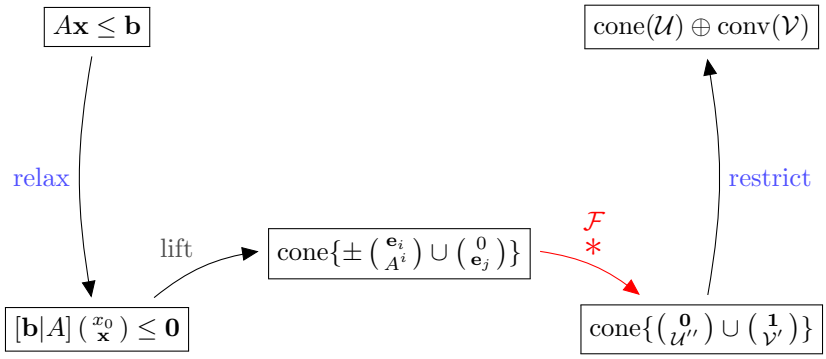

Figure 2: Diagram of the proof ($\mathcal{P}_\mathcal{H} \to \mathcal{C}_\mathcal{H} \to \mathcal{C}_{\mathcal{V}'} \to \mathcal{C}_\mathcal{V} \to \mathcal{P}_\mathcal{H}$)

the other direction (and they *will* get remarks later), so, while all details are provided, they may be provided more directly.

3.3.1 Relax: $\mathcal{P}_\mathcal{V} \to \mathcal{C}_\mathcal{V}$

Given $\mathcal{P}_\mathcal{V} = \mathrm{cone}(\mathcal{U}) \oplus \mathrm{conv}(\mathcal{V})$, we need to relax this object to a cone. We take each $\mathbf{u}_i \in \mathcal{U} \to \binom{0}{\mathbf{u}_i}$, and each $\mathbf{v}_j \in \mathcal{V} \to \binom{1}{\mathbf{v}_j}$. Consider now:

$$\mathcal{C}_\mathcal{V} = \mathrm{cone}\left(\left\{\binom{0}{\mathbf{u}_i}\right\} \cup \left\{\binom{1}{\mathbf{v}_j}\right\}\right)$$

Then $\mathcal{P}_\mathcal{V} = \pi^0(\mathcal{C}_\mathcal{V} \cap \{\binom{x_0}{\mathbf{x}} : x_0 = 1\})$. That is, $\mathcal{P}_\mathcal{V}$ can be recovered by first intersecting $\mathcal{C}_\mathcal{V}$ with the hyperplane $\{\binom{x_0}{\mathbf{x}} : x_0 = 1\}$, then projecting to the original dimensions. To prove this, note that

$$\mathbf{u} \in \mathrm{cone}(\mathcal{U}),\ \mathbf{v} \in \mathrm{conv}(\mathcal{V}) \Rightarrow$$
$$\binom{1}{\mathbf{u}+\mathbf{v}} \in \mathcal{C}_\mathcal{V} \cap \left\{\binom{x_0}{\mathbf{x}} \middle| x_0 = 1\right\}$$

So every element of $\mathcal{P}_\mathcal{V}$ has a representative in $\mathcal{C}_\mathcal{V} \cap \{\binom{x_0}{\mathbf{x}} : x_0 = 1\}$. Furthermore,

$$\binom{1}{\mathbf{z}} \in \mathcal{C}_\mathcal{V} \Rightarrow$$
$$\exists \alpha_i, \beta_j \geq 0 \,\middle|\, \binom{1}{\mathbf{z}} = \sum\nolimits_i \alpha_i \binom{0}{\mathbf{u}_i} + \sum\nolimits_j \beta_j \binom{1}{\mathbf{v}_j} \Rightarrow$$
$$\sum\nolimits_j \beta_j = 1$$

So, every member of $\mathcal{C}_\mathcal{V} \cap \{\binom{x_0}{\mathbf{x}} : x_0 = 1\}$ is a sum of a conical combination of $\binom{\mathbf{0}}{\mathcal{U}}$ and a convex combination of $\binom{\mathbf{1}}{\mathcal{V}}$, so is represented in $\mathcal{P}_\mathcal{V}$.

3.3.2 Lift: $\mathcal{C}_{\mathcal{V}} \to \mathcal{C}_{\mathcal{H}'}$

The next task is to lift the cone $\mathcal{C}_{\mathcal{V}}$ to a represenation as an intersection of halfspaces. This is a bit simpler than when we had to lift in the other direction.

With a few rewrites of $\mathcal{C}_{\mathcal{V}}$ we get:

$$\begin{aligned}
&\mathcal{C}_{\mathcal{V}} = \text{cone}(\mathcal{V}') = \\
&\left\{\mathbf{x} \;\middle|\; \exists t_i \geq 0 : \mathbf{x} = \textstyle\sum_i t_i \mathbf{v}_i\right\} = \\
&\left\{\mathbf{x} \;\middle|\; \exists \mathbf{t} \geq 0 : \mathbf{x} = \mathcal{V}'\mathbf{t}\right\} = \\
&\pi_{\mathbf{x}}\left\{\begin{pmatrix}\mathbf{x}\\ \mathbf{t}\end{pmatrix} \;\middle|\; (\mathbf{0}\ I)\begin{pmatrix}\mathbf{x}\\ \mathbf{t}\end{pmatrix} \geq \mathbf{0},\ \begin{pmatrix}-I & \mathcal{V}'\\ I & -\mathcal{V}'\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{t}\end{pmatrix} \leq \mathbf{0}\right\} = \\
&\pi_{\mathbf{x}}\left\{\begin{pmatrix}\mathbf{x}\\ \mathbf{t}\end{pmatrix} \;\middle|\; \begin{pmatrix}\mathbf{0} & -I\\ -I & \mathcal{V}'\\ I & -\mathcal{V}'\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \mathbf{t}\end{pmatrix} \leq \mathbf{0}\right\}
\end{aligned}$$

Here, $\mathcal{V}'$ has been realized as not just a collection of column vectors, but also as a stack of row vectors, dual to what was seen in the previous lifting procedure. So, the final expression gives us our cone as a projection of an H-Cone, this H-Cone will be denoted $\mathcal{C}_{\mathcal{H}'}$, and the projections will be considered in the next section.

3.3.3 Drop: $\mathcal{C}_{\mathcal{H}'} \to \mathcal{C}_{\mathcal{H}}$

As before, $\mathcal{C}_{\mathcal{H}'}$ will be dropped one dimension at a time, so the task is to project away the dimension k from $A\mathbf{x} \leq \mathbf{0}$. Let's dwell on this for a moment. Say that $A\mathbf{x} \leq \mathbf{0}$, and $A\mathbf{y} \leq \mathbf{0}$. Furthermore, suppose that $\pi^k(\mathbf{x}) = \pi^k(\mathbf{y})$. We need to "collapse" these two vectors into one representative. Observe:

$$\begin{aligned}
\pi^k\mathcal{C}_{\mathcal{H}'} &= \pi^k\{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{0}\} \\
&= \{\pi^k\mathbf{x} \mid A\mathbf{x} \leq \mathbf{0}\} \\
&= \{\mathbf{x} : x_k = 0 \mid \exists\alpha \in \mathbb{R} :\ A(\mathbf{x} + \alpha\mathbf{e}_k)\ \leq \mathbf{0}\}
\end{aligned}$$

Consider A as a collection of row vectors $\mathbf{a}_i$. Then, this final set can be written:

$$\begin{aligned}
&\{\mathbf{x} : x_k = 0 \mid \exists\alpha \in \mathbb{R} :\ \forall i\ \langle\mathbf{a}_i, \mathbf{x} + \alpha\mathbf{e}_k\rangle \leq 0\} \\
&\{\mathbf{x} : x_k = 0 \mid \exists\alpha \in \mathbb{R} :\ \forall i\ \langle\mathbf{a}_i, \mathbf{x}\rangle \leq -\alpha\, a_i^k\}
\end{aligned}$$

Now define:

$$\begin{aligned}
\mathcal{Z} &= \{\mathbf{a}_i \in A \mid a_i^k = 0\} \\
\mathcal{P} &= \{\mathbf{a}_i \in A \mid a_i^k > 0\} \\
\mathcal{N} &= \{\mathbf{a}_i \in A \mid a_i^k < 0\}
\end{aligned}$$

In what follows, let $\mathbf{x}' \in \mathcal{C}_{\mathcal{H}'} | x_k = 0$ (so I don't have to keep writing it). We will now investigate what constraints rows from the sets $\mathcal{Z}$, $\mathcal{P}$, and $\mathcal{N}$ place on α. If all of these constraints can be simultaneously satisfied, then we will get an α that places $\mathbf{x} + \alpha \mathbf{e}_k$ in $\mathcal{C}_{\mathcal{H}'}$ and therefore $\mathbf{x} \in \pi^k(\mathcal{C}_{\mathcal{H}'})$. First suppose $\mathbf{a}_i \in \mathcal{Z}$. Then $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq -\alpha\, a_i^k = 0$ places no additional constraints on α. If $\langle \mathbf{a}_i, \mathbf{x} \rangle > 0$ then $\mathbf{x}$ is not in $\pi^k(\mathcal{C}_{\mathcal{H}'})$, end of story.

Next let $\mathbf{a}_i \in \mathcal{P}$. Then we can choose any α satisfying:

$$\alpha \leq \langle -\mathbf{a}_i / a_i^k, \mathbf{x} \rangle$$

Similarly, let $\mathbf{a}_j \in \mathcal{N}$. Then we can choose any α satisfying:

$$-\alpha \leq \langle \mathbf{a}_j / a_j^k, \mathbf{x} \rangle$$

Taken separately, either of these constraints can be satisfied with some arbitrarily large or small α. What is needed, however, is a way to take them into account at the same time. Then, given some $\mathbf{x}$ from $\pi^k(\mathcal{C}_{\mathcal{H}'})$, if we can satisfy all these constraints for all i, j, then we ought to be able to lift it back to $\mathcal{C}_{\mathcal{H}'}$. It seems natural to add these inequalities together, leading to:

$$\begin{aligned}
0 \leq \langle -\mathbf{a}_i / a_i^k, \mathbf{x} \rangle + \langle \mathbf{a}_j / a_j^k, \mathbf{x} \rangle \Leftrightarrow \\
\langle -a_j^k \mathbf{a}_i, \mathbf{x} \rangle + \langle a_i^k \mathbf{a}_j, \mathbf{x} \rangle \leq 0 \Leftrightarrow \\
\langle a_i^k \mathbf{a}_j - a_j^k \mathbf{a}_i, \mathbf{x} \rangle \leq 0 \Leftrightarrow \\
\langle a_i^k \mathbf{a}_j, \mathbf{x} \rangle \leq \langle a_j^k \mathbf{a}_i, \mathbf{x} \rangle
\end{aligned}$$

Care must be taken when multiplying through by $a_j^k \in \mathcal{N}$, making sure to reverse the direction of inequalities. Consider the last two forms. $\langle a_i^k \mathbf{a}_j - a_j^k \mathbf{a}_i, \mathbf{x} \rangle \leq 0$ is a positive linear combination of rows from $\mathcal{C}_{\mathcal{H}'}$, whose coefficient at x_k is 0. This is a constraint that we can use as we "project away" the k-th coordinate. The second form $\langle a_i^k \mathbf{a}_j, \mathbf{x} \rangle \leq \langle a_j^k \mathbf{a}_i, \mathbf{x} \rangle$ places a constraint on the α we may choose to "lift" our x back to $\mathcal{C}_{\mathcal{H}'}$ after we apply $\pi^k(\mathcal{C}_{\mathcal{H}'})$.

The same $\overset{\mathcal{F}_k}{*}$ operator will be used as before.

$$\mathcal{P} \overset{\mathcal{F}_k}{*} \mathcal{N} = \{ a_k^i \mathbf{a}_j - a_k^j \mathbf{a}_i \mid \mathbf{a}_i \in \mathcal{P},\, \mathbf{a}_j \in \mathcal{N} \}$$

We'd like to let $A' = \pi^k(\mathcal{Z} \cup \mathcal{P} \overset{\mathcal{F}_k}{*} \mathcal{N})$, and claim that $\pi^k(\mathcal{C}_{\mathcal{H}'}) = \{ \mathbf{x} \mid A'\mathbf{x} \leq 0 \}$, but we need to make sure that

$$\begin{aligned}
A\mathbf{x} \leq \mathbf{0} \Rightarrow A'(\mathbf{x} - x_k \mathbf{e}_k) \leq \mathbf{0} \\
A'\mathbf{x}' \leq \mathbf{0} \Rightarrow \exists \alpha \mid A(\mathbf{x}' + \alpha \mathbf{e}_k) \leq \mathbf{0}
\end{aligned}$$

The first requirement states that, given a member of $\mathcal{C}_{\mathcal{H}'}$, it projects down to $\pi^k(\mathcal{C}_{\mathcal{H}'})$. The second states that, given a member of $\pi^k(\mathcal{C}_{\mathcal{H}'})$, we can lift it back to $\mathcal{C}_{\mathcal{H}'}$.

The first requirement is easily satisfied.

$$\begin{aligned}
& A\mathbf{x} \leq \mathbf{0} \Rightarrow \\
(\forall \mathbf{a}_l \in \mathcal{Z}) \quad & \langle \mathbf{a}_l, \mathbf{x} \rangle \leq 0 \\
(\forall \mathbf{a}_i \in \mathcal{P}, \forall \mathbf{a}_j \in \mathcal{N}) \quad & \langle \mathbf{a}_i, \mathbf{x} \rangle \leq 0,\ \langle \mathbf{a}_j, \mathbf{x} \rangle \leq 0 \Rightarrow \\
(\forall \mathbf{a}_i \in \mathcal{P}, \forall \mathbf{a}_j \in \mathcal{N}) \quad & \langle a_i^k \mathbf{a}_j - a_j^k \mathbf{a}_i, \mathbf{x} \rangle \leq 0 \Rightarrow \\
(\forall \mathbf{a}_i \in \mathcal{Z} \cup \mathcal{P} \overset{\mathcal{F}_k}{*} \mathcal{N}) \quad & \langle \mathbf{a}_i, \mathbf{x} \rangle \leq 0 \Rightarrow \\
& \pi^k(\mathbf{x}) \in \pi^k(\mathcal{C}_{\mathcal{H}'})
\end{aligned}$$

The other requirement is a bit trickier, as we must calculate an α to meet our requirements. Suppose that $A'\mathbf{x}' \leq \mathbf{0}$, $x'_k = 0$, and let $\mathbf{a}_i \in \mathcal{Z}$:

$$\begin{aligned}
\langle \mathbf{a}_i, \mathbf{x}' + \alpha \mathbf{e}_k \rangle &= \langle \mathbf{a}_i, \mathbf{x}' \rangle + \alpha \langle \mathbf{a}_i, \mathbf{e}_k \rangle \\
&= \langle \mathbf{a}_i, \mathbf{x}' \rangle + 0 \\
&= \langle \mathbf{a}_i, \mathbf{x}' \rangle \leq 0
\end{aligned}$$

As previously mentioned, the rows from $\mathcal{Z}$ do not constrain the value of α. Next, consider $\mathcal{P}$ and $\mathcal{N}$.

$$\begin{aligned}
& (\forall \mathbf{a}_i \in \mathcal{P}, \forall \mathbf{a}_j \in \mathcal{N}) \\
& \quad \langle a_i^k \mathbf{a}_j - a_j^k \mathbf{a}_i, \mathbf{x}' \rangle \leq 0 \Rightarrow \\
& \qquad \langle a_i^k \mathbf{a}_j, \mathbf{x}' \rangle \leq \langle a_j^k \mathbf{a}_i, \mathbf{x}' \rangle \Rightarrow \\
& \qquad \langle \mathbf{a}_i / a_i^k, \mathbf{x}' \rangle \leq \langle \mathbf{a}_j / a_j^k, \mathbf{x}' \rangle \Rightarrow \\
& \quad \exists \alpha \mid \langle \mathbf{a}_i / a_i^k, \mathbf{x}' \rangle \leq \alpha \leq \langle \mathbf{a}_j / a_j^k, \mathbf{x}' \rangle
\end{aligned}$$

So, we take an α that satisfies all pairs of the above constraints. (Note: it is possible that no such α exists for any $\mathbf{x}'$, then our polyhedron is empty. This is elegantly summarized in what is known as the Farkas Lemma, and will be discussed later). Then, we plug α back into our original constraints:

$$\begin{aligned}
\langle \mathbf{a}_i, \mathbf{x}' - \alpha \mathbf{e}_k \rangle &= \langle \mathbf{a}_i, \mathbf{x}' \rangle - \alpha \langle \mathbf{a}_i, \mathbf{e}_k \rangle \\
&= \langle \mathbf{a}_i, \mathbf{x}' \rangle - \alpha\, a_i^k \\
&= a_i^k (\langle \mathbf{a}_i / a_i^k, \mathbf{x}' \rangle - \alpha) \leq 0 \\
\langle \mathbf{a}_j, \mathbf{x}' - \alpha \mathbf{e}_k \rangle &= \langle \mathbf{a}_j, \mathbf{x}' \rangle - \alpha \langle \mathbf{a}_j, \mathbf{e}_k \rangle \\
&= \langle \mathbf{a}_j, \mathbf{x}' \rangle - \alpha\, a_j^k \\
&= a_j^k (\langle \mathbf{a}_j / a_j^k, \mathbf{x}' \rangle - \alpha) \leq 0
\end{aligned}$$

The inequalities follow from the fact that $a_j^k < 0 < a_i^k$. So the conclusion is that, given $A'\mathbf{x}' \leq \mathbf{0}$, we can find an α such that $A(\mathbf{x}' - \alpha \mathbf{e}_k) \leq \mathbf{0}$, and that $\pi^k(\mathcal{C}_{\mathcal{H}'}) = \{\mathbf{x}' \mid x'_k = 0,\ A'\mathbf{x}' \leq \mathbf{0}\}$.

Since we can project away one dimension, we can succesively form $\pi_{\mathbf{x}}(\mathcal{C}_{\mathcal{H}'})$. Then, we denote $\mathcal{C}_{\mathcal{H}} = \pi_{\mathbf{x}}(\mathcal{C}_{\mathcal{H}'})$.

3.3.4 Restrict: $\mathcal{C}_\mathcal{H} \to \mathcal{P}_\mathcal{H}$

The last loose end to get back to our original polyhedron is to intersect $\mathcal{C}_\mathcal{H}$ with the hyperplane $\{\mathbf{x} \mid x_k = 0\}$. This is simply:

$$A\mathbf{x} \leq \mathbf{0},\ x_0 = 1 \to \begin{pmatrix} 1 & \mathbf{0} \\ -1 & \mathbf{0} \\ A & \end{pmatrix} \mathbf{x} \leq \begin{pmatrix} 1 \\ -1 \\ \mathbf{0} \end{pmatrix}$$

This H-Polyhedron has been denoted $\mathcal{P}_\mathcal{H}$, and the construction is done.

3.3.5 Summary

4 Program

5 Discussion

Projections (reconcile the remarks about notation earlier). In general, a projection can be considered any function such that:

$$f^2 = f$$

Say the domain of f is D, then the image is $f(D)$, and the above equation states that f restricted to D is the identity, i.e.

$$\forall x \in f(D) :\ f(x) = x$$

A class of particularly useful projections are those that map $\mathbb{R}^n$ to a linear subspace. A simple example is to project some set A to the hyperplane $\{\mathbf{x} \mid x_k = 0\}$. Say you would like a closed form for this projection, how would you calculate it? Well, any point $\mathbf{x}$ must be mapped to $\mathbf{x} - x_k\mathbf{e}_k$, or:

$$\mathbf{x} \mapsto \mathbf{x} - x_k\mathbf{e}_k$$

After staring at this for awhile, you may realize that there is a nice form for this transformation, namely:

$$T : \mathbb{R}^n \to \mathbb{R}^n = I - \mathbf{e}_k\mathbf{e}_k^T$$

In more generality, let $\mathbf{u}$ be a unit vector. Then the transformation given by

$$T : \mathbb{R}^n \to \mathbb{R}^n = I - \mathbf{u}\mathbf{u}^T$$

is a projection to the linear subspace given by $\{\langle \mathbf{u}, \mathbf{x} \rangle = 0\}$. (mention householder matrices) By way of proof, first note that

$$\begin{aligned} T^2 &= (I - \mathbf{u}\mathbf{u}^T)(I - \mathbf{u}\mathbf{u}^T) \\ &= I - \mathbf{u}\mathbf{u}^T - \mathbf{u}\mathbf{u}^T + (\mathbf{u}\mathbf{u}^T)(\mathbf{u}\mathbf{u}^T) \\ &= I - 2\mathbf{u}\mathbf{u}^T + \mathbf{u}\mathbf{u}^T \\ &= I - \mathbf{u}\mathbf{u}^T \\ &= T \end{aligned}$$

Here both $\mathbf{u}^T\mathbf{u} = 1$ and the fact that matrix multiplication is associative has been used. Also observe that $\langle \mathbf{u}, \mathbf{x} - \mathbf{u}\mathbf{u}^T\mathbf{x} \rangle = \mathbf{u}^T\mathbf{x} - \mathbf{u}^T\mathbf{u}\mathbf{u}^T\mathbf{x} = \mathbf{u}^T\mathbf{x} - \mathbf{u}^T\mathbf{x} = 0$. The projections given by $I - \mathbf{e}_k\mathbf{e}_k^T$ are important, and are here denoted π^k. These can be thought of as "projecting away" the k-th component of all vectors in $\mathbb{R}^n$. Similarly, $\pi_k = I - \sum_{j \neq k} \mathbf{e}_j\mathbf{e}_j^T$. This is the identity matrix with all but the k-th 1 entry set to 0. This transformation sets all entries of $\mathbf{x}$ to zero except for x_k. It is "projecting to" the k-axis.

The way projections are defined here, they do not map vectors to lower dimensional "ambient" spaces. This is mostly to make notation consistent, and to avoid difficulties associated with trying to keep track of which vectors are in which dimension, particularly important when doing matrix multiplication and taking dot products. After the discussion of polyhedral equivalence, this shouldn't be too much of an issue, but here is a detailed (and overwhelmingly boring) example. Let $\Pi^{n+1} : \mathbb{R}^{n+1} \to \mathbb{R}^n$ be given by the tranformation:

$$(x_1, x_2, \ldots, x_n, x_{n+1}) \mapsto (x_1, x_2, \ldots, x_n)$$

We would like to say that Π^{n+1} is "equivalent" to π^{n+1} in a precise way. For this, we need a bijection from $\Pi^{n+1}(\mathbb{R}^{n+1})$ to $\pi^{n+1}(\mathbb{R}^{n+1})$. The most obvious choice is:

$$(x_1, x_2, \ldots, x_n) \leftrightarrow (x_1, x_2, \ldots, x_n, 0)$$

There is no good idea here, just confirming the obvious to make sure that our symbolic manipulations are justified.

Now take a polyhedron: $\mathcal{P}_\mathcal{V} = \text{cone}(\mathcal{U}) \oplus \text{conv}(\mathcal{V})$. What is the projection of this polyhedron? Say $\mathbf{x} \in \mathcal{P}_\mathcal{V}$, then $\mathbf{x} = \sum_i \alpha_i \mathbf{u}_i + \sum_j \lambda_j \mathbf{v}_j$. Then

$$\begin{aligned}
\pi^k(\mathbf{x}) &= (I - \mathbf{e}_k\mathbf{e}_k^T)(\sum\nolimits_i \alpha_i \mathbf{u}_i + \sum\nolimits_j \lambda_j \mathbf{v}_j) \\
&= \sum\nolimits_i \alpha_i (I - \mathbf{e}_k\mathbf{e}_k^T)\mathbf{u}_i + \sum_j \lambda_j (I - \mathbf{e}_k\mathbf{e}_k^T)\mathbf{v}_j \\
&= \sum\nolimits_i \alpha_i \pi^k(\mathbf{u}_i) + \sum_j \lambda_j \pi^k(\mathbf{v}_j)
\end{aligned}$$

What this shows is that

$$\pi^k(\text{cone}(\mathcal{U}) \oplus \text{conv}(\mathcal{V})) = \text{cone}(\pi^k(\mathcal{U})) \oplus \text{conv}(\pi^k(\mathcal{V}))$$

What this *means* is that taking the projection of a V-Polyhedron is essentially trivial, it can be found by projecting each of its generators.

Intersections and Equivalence Classes (I started writing this, then sort of got lost. I'm trying to demonstrate why taking projections/ intersections is harder/easier for the different types of representations, but now I'm not sure I have a useful point here... Maybe this should just be removed or thought about

more thoroughly before attempting to present it) An equivalence relation is a binary relation $\sim$ that is reflexive, symmetric, and transitive. In symbols:

$$\forall a, b, c$$
$$a \sim a$$
$$a \sim b \Rightarrow b \sim a$$
$$a \sim b \wedge b \sim c \Rightarrow a \sim c$$

Every function f from one set to another induces an equivalence relation, sometimes written $\overset{f}{\sim}$. Let $f : A \to B$, then the relation given by $a \overset{f}{\sim} b \Leftrightarrow f(a) = f(b)$ is trivially an equivalence relation. The classes of this equivalence are the given by $f^{-1}[b] = \{a \in A \mid f(a) = b\}$.

Consider the projection π_k, a projection to the k-th axis, and a set $A \subseteq \mathbb{R}^n$. Then $\pi_k(A)$ is all of the equivalence classes given by $\overset{\pi^k}{\sim}$. What does one of these classes look like? Fix a value $r \in \mathbb{R}$, then the set $\{\mathbf{a} \in A \mid \pi_k(\mathbf{a}) = a_k = r\}$ is one such class. This is precisely the intersection of A with the hyperplane $\{\mathbf{x} \mid x_k = r\}$.