Note: This document is a working document and isn't not fully fleshed out (or as formal as desired)

1 General Sets

1.1 Dimension-awareness

[1] [2] [3]

Definition 1 (Dimensional Aware Set Definition) *Let $\mathcal{D}_V : \nu_i \to i$ be a dictionary that relates the key ν_i to each of the vector fields $\mathbb{V}_i \subset \mathbb{V}$, $\forall_{i \in \mathbb{N}}$.*
Let $X = (X_1, X_2, \dots, X_n) \subseteq (\mathbb{V}_1, \mathbb{V}_2, \dots, \mathbb{V}_n)$ and $\mathcal{D}_X = \{\nu_1 \to 1, \dots, \nu_n \to n\} \subseteq \mathcal{D}_V$. The set and the dictionary $(X, \mathcal{D}_X) \subseteq (V, \mathcal{D}_V)$ is then considered dimensional-aware as the dictionary relates the key ν_i with the associated set $X_i \subseteq \mathbb{V}_i$.
Denote $(\mathbb{V}, \mathcal{D}_V)$ as a space of all dimensionally-aware sets.

do we want to use "dictionary" or some "indexed family" type definition?

Definition 2 (Dimensional Aware Relations) *A dimensionally-aware relation, $\overset{*}{\mathcal{R}}$, between the two dimensionally-aware sets $(X, \mathcal{D}_X), (Y, \mathcal{D}_Y) \subseteq (V, \mathcal{D}_V)$ is defined differently depending on the keys in each associated dictionary:*

1. *For unique keys, ν, within each dictionary $\mathcal{D}_X$ and $\mathcal{D}_Y$, the associated sets, X_i and Y_j are kept and associated within $\mathcal{D}_C$ to the appropriate index in the new collection as C_k.*

2. *For keys that are shared between the two dictionaries $\nu_i \in \mathcal{D}_X = \nu_j \in \mathcal{D}_Y$ the relation is then applied to the corresponding sets $C_k = X_i \; \mathcal{R} \; Y_j$ and associated within $\mathcal{D}_C$ to the appropriate index k in the new collection.*

look at the best way to notate this... do we need to make ν within a set? does this make sense as is?

(i.e)

$$\begin{aligned}(X, \mathcal{D}_X) \overset{*}{\mathcal{R}} (Y, \mathcal{D}_Y) &= (C, \mathcal{D}_C) \\ &= ((X_1, \dots, X_{n_a}), \mathcal{D}_X) \overset{*}{\mathcal{R}} ((Y_1, \dots, Y_{n_b}), \mathcal{D}_Y) \\ &= (([X_i]_{\forall_{i \in I_x}}, [Y_j]_{\forall j \in J_y}, [X_i \; \mathcal{R} \; Y_j]_{\forall i \in I_{xy}, j \in J_{xy}}), \mathcal{D}_C)\end{aligned} \tag{1}$$

where $I_x = \{i \mid \nu_i^x \notin \mathcal{D}_Y\}$, $J_y = \{j \mid \nu_j^y \notin \mathcal{D}_X\}$, $I_{xy} = \{i \mid \nu_i^x \in \mathcal{D}_Y\}$, $J_{xy} = \{j \mid \nu_j^y \in \mathcal{D}_X\}$, and $\mathcal{D}_C : \nu_k^c \to k$ relates the shared keys to the appropriate sets in collection C.

look at notation...

Remark 1 (Dimension-aware set operations) *Since set operations are in essence just relations, any set operations between two dimension-aware sets can be implemented as described with the appropriate relation applied upon the appropriate dimension.*

Remark 2 (Additional Remark on Dimension-aware Set Operations) *Under dimension-aware operations, any dimensions that are unique (i.e. not defined in the other dictionary) then either $\emptyset$ (for cartisian product, union, minkowski sum...) or $\mathbb{V}$ (intersection, ...) may instead be associated with the specific key, ν, and the operation being applied between the appropriate sets would result in the same operation.*

[1] The specific-base variable-type of any set of dimensions is not restricted to any particular $\mathbb{R}$,$\mathbb{Z}$,$\mathbb{C}$ (or in MALTAB class()=double,sym,int,optimvar,etc.) however if the notation of $\mathbb{R}^n$ is used this should still be true...

[2] Should we restrict to just a collection of 1-D dimensions or keep it general to any dimensional spaces being labeled together??? I'm thinking the later as I'm not restricting it to n-dimensional fields anyway...

[3] So I think we could restrict this to a banach space or less to a field, but technically I think any vector space should work... And if we restrict to a space consisting of 1-D (scaler?) for each dimension that could also work...

1.2 Memory-based ("Lazy")-set Definition

[4]

The premise of memory-preservation that this general-notation will be pursuing will consist of having origin sets who are placed through transformation and arbitrary relations (i.e. set-operations) that will be saved as constraints upon the yet to be evaluated preserved-memory set.
Let $X_1, \dots, X_{n_x} \subseteq \mathbb{V}_X$ and $Y_1, \dots, Y_{n_y} \subset \mathbb{V}_Y$ be sets with associated dictionaries, $\mathcal{D}_X : \nu_i^X \to i$ and $\mathcal{D}_Y : \nu_j^Y$, associating the keys with the origin sets.
Consider a generic affine transformation $(f(x) = Mx + b)$ and a general transformation $g_k(x, y) : V_X \times V_Y \to V_{g(x,y)}$.
Additionally, suppose/assume that there are relations $\mathcal{R}$, (such as set operations) so that within $(X_i, Y_j, X_i \mathcal{R} Y_j)$, $X_i \mathcal{R} Y_j$ can be defined with constraints dependent on X_i and Y_j.

1.2.1 Affine transformation:

For memory-encoded sets $([X_i]_{i=1,\dots,n_x}, \mathcal{D}_X)$ being placed through the affine transformation f, the new sets can be transformed with no adjustment to the dictionary but the individual sets will now be endowed with an affine mapping. (i.e.)

$$([X_i]_{i=1,\dots,n_x}, \mathcal{D}_X) \overset{f(x)}{\to} (f([X_i]_{i=1,\dots,n_x}), \mathcal{D}_X) = ([f(X_i)]_{i=1,\dots,n_x}, \mathcal{D}_X) = ([MX_i + b]_{i=1,\dots,n_x}, \mathcal{D}_X)$$

Within our case of memory, the previous sets could also be maintained (although technically not necessary for this operation if not doing a successor set) resulting in the following:

$$([X_i]_{i=1,\dots,n_x}, \mathcal{D}_X) \overset{f(x)}{\to} (([X_i]_{i=1,\dots,n_x}, [MX_i + b]_{i=1,\dots,n_x}), \mathcal{D}_X)$$

Note that the dictionary $\mathcal{D}_X$ continues to indicate that the X_i sets are the same/related/from the same origin set.

1.2.2 General transformation

For memory-encoded sets $([X_i]_{i=1,\dots,m}, \mathcal{D}_X)$ and $([Y_j]_{j=1,\dots,m}, \mathcal{D}_Y)$ $(m < n_x, n_y)$ being placed through the general transformation of $g_k(x, y)$ (of which k is distinct for each pair of i and j), the memory-preservation transformation (just a successor set???) results in

$$([X_i]_{i=1,\dots,m}, [Y_j]_{j=1,\dots,m}, [(X_i, Y_j) \overset{g_k(x,y)}{\to} g_k(X_i, Y_j)]_{\forall_{i=j=k=1,\dots,m}}, (\mathcal{D}_X, \mathcal{D}_Y))$$

Note again that the dictionary still exists and connects the origination sets to the transformed results

2 Zonotope-based Set Implementation

Zonotopes are built upon the basic principles of a Minkowski Sum to represent closed and bounded sets within an n-dimensional vector space.

Continue with this and implement the relations... that involves adding constraints that relate specific ‘"dimensions"

[4]The previous attempt at "Lazy"-set memory-based operation is not as well-structured or as general as the dimension-aware version, thus I'm going to have to attempt it differently.

Standard zonotopes are represented strictly as the minkowski sum of line-segments, often described in an affine form constructed of a center-vector ($c \in \mathbb{V}^n$) and a generator matrix ($G \in \mathbb{V}^{n \times n_g}$) with factors ($\xi_i \in [0,1]$) acting upon individual generators$g_i \in \mathbb{V}^n$:

$$Z = \{G, c\} = \left\{c + G\xi \mid \forall_{\xi \in \mathbb{R}^{n_g}} \|\xi\|_\infty \leq 1\right\}$$

In order to represent non-symetric convex polytopes, additional constraints can be introduced upon the factors using constraint matrix $X \in \mathbb{R}^{n_c \times n_g}$ and vector $b \in \mathbb{R}^{n_c}$, ($X\xi = b$), to break symmetry and represent more complicated shapes:

$$Z = \{G, c, A, b\} = \left\{c + G\xi \mid \forall_{\xi \in \mathbb{R}^{n_g}} \|\xi\|_\infty \leq 1 \wedge A\xi = b\right\}$$

Another extension involves the introduction of a restriction upon certain factors to become discrete factors (i.e. $\xi_i \in \{-1, 1\}$). This "hybrid-zonotope" can be represented as two separate generator and constraint matrices (as is common in other literature) [5] or alternatively an additional constraint can be placed upon certain generators that requires them to be discrete. This can be easily described with an additional vector $\kappa \in \{0,1\}^{n_g}$, in which the boolean true ($\kappa_i = 1$) indicates that the generator is discrete. This restriction can either be ensured with distinct constraints of $|\xi_i| = 1\forall_{i \mid \kappa_i = 1}$ or through an element-wise 1-norm constraint of $\|\kappa \odot \xi\|_1 = \|\kappa\|_1$.

$$\begin{aligned} Z &= \{G, c, A, b, \kappa\} \\ &= \left\{c + G\xi, \forall_{\xi \in \mathbb{R}^{n_g}} \|\xi\|_\infty \leq 1 \wedge \|\kappa \odot \xi\|_1 = \|\kappa\|_1 \wedge A\xi = b\right\} \end{aligned}$$

2.1 Dimension-awareness in Zonotope-based Sets

Given the structure of the affine and generator-based construction of a zonotope within an n-dimensional space, it makes sense to label each of the dimensions of the center and generator matrices instead of maintaining the ordered collection (family?) of vector-spaces for each set. Thus, for zonotope-based set $Z = \{G_Z, c_Z, A_Z, b_Z, \kappa_Z\}$, each of the keys, ${}_d\nu_i^Z \in {}_d\mathcal{D}_Z$, will be associated with a row index i of generator matrix G_Z and center vector c_Z.

2.2 Memory-preservation within Zonotope-based sets

Under all the defined memory-perserving set-operations (as defined above/below?), the information of volume origin is solely contained within the generators and any associated origin constraints, while the relationships between sets imparted by set-operations appear as additional constraints. This allows for memory of origin to be maintained through labeling and appropriate bookkeeping of factors as the columns of the generator and constraint matrices (and κ indices) with ${}_g\nu_i^Z \in {}_g\mathcal{D}_Z$, while the memory of both origin and relationship constraints can be maintained by keeping track of the rows of the constraint matrix and vector with ${}_c\nu_i^Z \in {}_c\mathcal{D}_Z$.

2.3 Dimension-awareness and Memory-preservation within zonotope-based sets

To preserve both dimension-awareness and memory-preservation, the dictionaries associated with dimensions, factors, and constraints are maintained. Since each of the keys has a one-to-one

[5] I don't want to use that notation... it's too confusing to me when comparing to the full zonotope framework

correspondence(bi-jective?), each dictionary can represented as an ordered sets [6] where the index within the set (or array) directly corresponds with the index of the dimensions, factors, or constraints.

Definition 3 (Dimension-aware and Memory-preserving Zonotope-based Set Definition) *Let $G \in \mathbb{V}^{n \times n_g}$, $c \in \mathbb{V}^n$, $A \in \mathbb{R}^{n_c \times n_g}$, $b \in \mathbb{R}^{n_c}$, and $\kappa \in \{0,1\}^{n_g}$.*
A zonotope-based set Z is defined as

$$\begin{aligned} Z &= \{G, c, A, b, \kappa\} \\ &= \left\{c + G\xi, \forall_{\xi \in \mathbb{R}^{n_g}} \|\xi\|_\infty \leq 1 \wedge \|\kappa \odot \xi\|_1 = \|\kappa\|_1 \wedge A\xi = b\right\} \end{aligned} \tag{2}$$

Let $\mathcal{D}_Z = \{{}_d\mathcal{D}_Z, {}_g\mathcal{D}_Z, {}_c\mathcal{D}_Z\}$ be the dictionaries for the dimensions, factors, and constraints respectively.
$(Z, \mathcal{D}_Z)$ is now considered to be a dimension-aware and memory-preserving set.

should we do $(Z, \mathcal{D}_Z)$ or $Z = \{G, c, A, b, \kappa$

The following are explicit definitions for the opperations but need not be fully written out (or in seperate definitions) for each to provide all the info

Definition 4 (Demension-aware and Memory-preserving Minkowski Sum) *Let $(X, \mathcal{D}_X)$ and $(Y, \mathcal{D}_Y)$ be demension aware and memory preserving sets. The demension-aware and memory-preserving minkowski-sum [7] $(X, \mathcal{D}_X) \overset{*}{\oplus} (Y, \mathcal{D}_Y)$ results in the following:*

$$(X, \mathcal{D}_X) \overset{*}{\oplus} (Y, \mathcal{D}_Y) \tag{3}$$

how do we best notate this... this relates to the function combine(X,Y)

Definition 5 (Demension-aware and Memory-preserving Intersection) *Let $(X, \mathcal{D}_X)$ and $(Y, \mathcal{D}_Y)$ be demension aware and memory preserving sets. The demension-aware and memory-preserving intersection [8] $(X, \mathcal{D}_X) \overset{*}{\cap} (Y, \mathcal{D}_Y)$ results in the following:*

$$(X, \mathcal{D}_X) \overset{*}{\cap} (Y, \mathcal{D}_Y) = \left(\begin{array}{r} \left\{ \begin{bmatrix} G_X^{x,x} & G_X^{x,xy} & \mathbf{0} \\ \mathbf{0} & G_Y^{y,xy} & G_Y^{y,y} \\ G_X^{xy,x} & G_X^{xy,xy} & \mathbf{0} \end{bmatrix}, \begin{bmatrix} c_X^x \\ c_Y^y \\ c_X^{xy} \end{bmatrix}, \begin{bmatrix} A_X^{x,x} & A_X^{x,xy} & \mathbf{0} \\ \mathbf{0} & A_{XY}^{xy,xy} & \mathbf{0} \\ \mathbf{0} & A_Y^{y,xy} & A_Y^{y,y} \\ G_X^{xy,x} & G_X^{xy,xy} - G_Y^{xy,xy} & -G_Y^{xy,y} \end{bmatrix}, \begin{bmatrix} b_X^x \\ b_{XY}^{xy} \\ b_Y^y \\ c_Y^{xy} - c_X^{xy} \end{bmatrix}, \begin{bmatrix} \kappa_X^x \\ \kappa_{XY}^{xy} \\ \kappa_Y^y \end{bmatrix} \right\}, \\ \mathcal{D}_{X\cap Y} = \begin{cases} {}_d\mathcal{D}_{X\cap Y} = \left\{{}_d\nu_X^x, {}_d\nu_Y^y, {}_d\nu_{XY}^{xy}\right\} \\ {}_g\mathcal{D}_{X\cap Y} = \left\{{}_g\nu_X^x, {}_g\nu_{XY}^{xy}, {}_g\nu_Y^y\right\} \\ {}_c\mathcal{D}_{X\cap Y} = \left\{{}_c\nu_X^x, {}_c\nu_{XY}^{xy}, {}_c\nu_Y^y, {}_c\nu_{X\cap Y}^{new}\right\} \end{cases} \Bigg\} \end{array} \right) \tag{4}$$

replace X and Y w/ X and Y for clarity???

The union (and compliment) operations are super interesting and a memory-preserving and dim-aware implimentation from the hybrid-zonotope paper appears to be a natural extension... here's that hyb-zono union/compliment paper: `https://ieeexplore.ieee.org/document/9638970`

how do we best notate this... this relates to the function merge(X,Y) also, is there a generalized-intersection version? I'm not sure...

[6] string/cell-array in matlab

[7] we could likely just call this intersection and define a specific algebra instead of doing the $\overset{*}{\oplus}$ and fancy name

[8] we could likely just call this intersection and define a specific algebra instead of doing the $\overset{*}{\cap}$ and fancy name