

Modeling Choice in Co-Design

Marius Furter

May 29, 2021

Contents

1	A Brief Introduction to Co-Design	1
1.1	Resource and Functionality Preorders	2
1.2	Feasibility Relations	3
2	Choice between Resources	4
2.1	Choice Connectives	5
2.1.1	Interaction of Choice with Free Transformations	5
2.1.2	Mutual Interaction of the Two Modes of Choice	6
2.1.3	Laws for the Choice Connectives	7
2.2	The Upper-Lower Model	8
2.2.1	Formally using the Upper-Lower Model	8
2.2.2	Interpretation-Based Reconstruction	13
3	Feasibility between Choices	13
4	Choice between Transformations	13
5	Towards Modeling Dependencies	13

1 A Brief Introduction to Co-Design

Co-Design investigates *feasibility relations* between *resources* and *functionalities*. We describe these in turn.

1.1 Resource and Functionality Preorders

We model the resources and functionalities using preordered sets. The order relation $a \rightarrow b$ signifies that we can freely transform a into b . Equivalently, $a \rightarrow b$ can also mean that a is implicitly already b .

Example 1.1. (Money as Resources) The set $\mathbb{N} := \{0, 1, 2, \dots\}$ with the order relation $a \rightarrow b \Leftrightarrow a \geq b$ can be used to describe money (of a single currency, say dollars) as a resource. The element a represents having a dollars, while $a \rightarrow b$ means that if you have a dollars, you also implicitly have b dollars. For example, $3 \rightarrow 1$ signifies that if we have 3 dollars, we also possess the purchasing power 1 dollar. Observe that being able to freely transform $3 \rightarrow 1$ presupposes people are willing to accept 3 dollars in lieu of 1 dollar. This hypothesis is system relative. For example, a vending machine might only accept bills of low denomination. So in this case one might well not have $50 \rightarrow 1$.

Returning to the general theory, we observe that the preorder axioms precisely describe the relation of free transformations.

Definition 1.1. (Preorder) A *preorder* is a set with a binary relation $R \subseteq A \times A$ satisfying reflexivity (r) and transitivity (t):

- (r) $(a, a) \in R$ for all $a \in A$,
- (t) if $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$.

Usually we write the pair (a, b) using an infix symbol. In our case, $a \rightarrow b$ means (a, b) is part of the relation R .

Using our infix notation, the two axioms become

- (r) $a \rightarrow a$ for all a ,
- (t) $a \rightarrow b$ and $b \rightarrow c$ imply $a \rightarrow c$.

In other words, each resource is freely transformable into itself (do nothing). Further, if we can freely transform a into b , and b into c , then we can freely transform a into c by doing one transformation after the other.

Everything that has been said here applies equally to functionalities. In fact, what we view as resources or functionalities is relative to the process we are considering. When I purchase a candy bar from a vending machine, the dollar I spend is the resource, and the candy bar the functionality. When I eat the candy bar to gain energy, the candy bar has become the resource.

1.2 Feasibility Relations

Given a preorder of resources \mathcal{R} and a preorder of functionalities \mathcal{F} , a feasibility relation $\Phi : \mathcal{R} \rightarrow \mathcal{F}$ between them expresses which functionalities I am able to obtain from which resources. We write $\Phi(r, f) = \mathbf{true}$ to indicate that f is obtainable from r .

Example 1.2. Consider buying groceries. Our resources \mathcal{R} will be money as in Example 1.1. The possible groceries we can buy are the functionalities \mathcal{F} . The feasibility relation $\Phi : \mathcal{R} \rightarrow \mathcal{F}$ will describe what groceries we can purchase given a specific amount of cash. For example, a `carrot` might cost 1\$. Hence, $\Phi(1\$, \text{carrot}) = \mathbf{true}$. On the other hand, $\Phi(0\$, \text{carrot}) = \mathbf{false}$, since we can't get a `carrot` without paying anything. Suppose now that a `bag of carrots` costs 5\$. This means $\Phi(5\$, \text{bag of carrots}) = \mathbf{true}$. Observe that also $\Phi(5\$, \text{carrot}) = \mathbf{true}$ should hold. Either we can pick up a single carrot in the store and pay for it with 5\$, or we could buy the whole bag for 5\$ and this would also give us a single `carrot`.

This shows that feasibility interacts with the free transformations we have for the resources and functionalities. In our example $5\$ \rightarrow 1\$$ and $\Phi(1\$, \text{carrot}) = \mathbf{true}$ implied $\Phi(5\$, \text{carrot}) = \mathbf{true}$. Similarly, $\text{bag of carrots} \rightarrow \text{carrot}$ and $\Phi(5\$, \text{bag of carrots}) = \mathbf{true}$ implied $\Phi(5\$, \text{carrot}) = \mathbf{true}$.

The interaction we observed between free transformations and feasibility justifies the formal definition of a feasibility relation.

Definition 1.2 (Feasibility Relation). Given preorders \mathcal{R} and \mathcal{F} , a *feasibility relation* $\Phi : \mathcal{R} \rightarrow \mathcal{F}$ is a relation $\Phi \subseteq \mathcal{R} \times \mathcal{F}$ such that

- (i) if $r \rightarrow s$ in \mathcal{R} and $(s, f) \in \Phi$, then $(r, f) \in \Phi$,
- (ii) if $f \rightarrow g$ in \mathcal{F} and $(r, f) \in \Phi$, then $(r, g) \in \Phi$.

If $(r, f) \in \Phi$ we write $\Phi(r, f) = \mathbf{true}$ or simply say that $\Phi(r, f)$ holds.

In other words, a feasibility relation is a relation from \mathcal{R} to \mathcal{F} that is downward closed with respect to free transformations in \mathcal{R} and upward closed with respect to free transformations in \mathcal{F} . We will often draw internal diagrams of feasibility relations as in Figure 1.2. We put an arrow between an $r \in \mathcal{R}$ and $f \in \mathcal{F}$ iff $\Phi(r, f) = \mathbf{true}$. The arrows coming from Φ can be thought of the transformations that Φ enables. Observe that the closure conditions in Definition 1.2 mean that the arrows coming from Φ are transitively closed

Figure 1: Internal diagram of a feasibility relation.

with respect to the internal arrows in \mathcal{R} and \mathcal{F} : If there is a path from an $r \in \mathcal{R}$ to an $f \in \mathcal{F}$ following any type of arrow, then we also have a direct arrow between r and f coming from Φ .

We could have also defined feasibility relations as special types of monotone maps.

Lemma 1.1. *There is a one-to-one correspondence between feasibility relations $\Phi : \mathcal{R} \rightarrow \mathcal{F}$ and monotone maps $\mathcal{R}^{op} \times \mathcal{F} \rightarrow \mathbf{Bool}$, where \mathbf{Bool} denotes the preorder generated by $\{\mathbf{false} \rightarrow \mathbf{true}\}$.*

2 Choice between Resources

We will now extend what we've seen so far to encompass choice and uncertainty. What is meant by these terms is best illustrated with an example.

Example 2.1 (Menu Options). In a restaurant we might see the following menu options.

- Coffee or tea (customer choice)
- Asparagus or beetroot soup (depending on availability)

In the first case, we as customers are given the choice between coffee or tea. In the second, it is chosen for us which soup we will get, depending on what the cooks have available. From our perspective as customers, this represents an uncertainty. In both cases, however, we will end up with exactly one of the options.

We might refer to the two modes of choice presented in Example 2.1 as *internal choice* and *external choice*, respectively. Alternatively, we could call them *choice* and *uncertainty*, or *free* and *forced* choice. Regardless of the names we choose, the terms are relative to which perspective we are taking. If we look through the eyes of the restaurant, the customer choice of coffee or tea is external, while the choice of soup is internal.

2.1 Choice Connectives

To formalize the two modes of choice described above, we introduce two connectives, denoted \sqcup and \sqcap , respectively. Because the distinction between what is internal and external is relative to what perspective we are taking, we will hold off on generally fixing which symbol denotes which mode of choice. Instead, we will always indicate what the interpretation should be in each specific case.

For the moment, fix \sqcap as free choice and \sqcup as forced choice. Hence, to denote a free choice from a set A we write $\sqcap_{a \in A} a$, while for a forced choice from a set B we write $\sqcup_{b \in B} b$. We will now see how such choices should interact with the free transformations we have in a resource preorder, as well as each other.

2.1.1 Interaction of Choice with Free Transformations

Let \mathcal{R} be a preorder of resources. If we have a free choice of any $a \in A$, then we can freely transform that choice into any specific $a \in A$ by making the corresponding choice. For example, if I have a choice between coffee or tea, I implicitly have coffee, and I implicitly have tea. In other words, I can downgrade any choice I get to make to the certainty of having one of the items I get to choose from. Formally, we have

$$\sqcap_{a \in A} a \rightarrow a, \quad \forall a \in A. \quad (1)$$

Now suppose we can freely obtain all a in a set A from a fixed resource t . In that case, having t means implicitly having a free choice between any $a \in A$. For any a you end up choosing, there is a free way to get that a from t . Formally,

$$\forall a \in A : t \rightarrow a \quad \Rightarrow \quad t \rightarrow \sqcap_{a \in A} a. \quad (2)$$

Dually, if I have any $b \in B$ for certain, I can freely transform it into a forced choice among all $b \in B$. For example, having asparagus soup also implicitly means you have asparagus or beetroot soup (depending on availability). In other words, we can downgrade the certainty of having a specific $b \in B$ to the uncertainty of having some $b \in B$. Formally, this becomes

$$b \rightarrow \sqcup_{b \in B} b, \quad \forall b \in B. \quad (3)$$

Finally, suppose that for every $b \in B$ there is a free transformation to a fixed resource t . In that case, if someone chooses an arbitrary $b \in B$ for you, you can still freely obtain t using the corresponding free transformation. Hence,

$$\forall b \in B : b \rightarrow t \quad \Rightarrow \quad \bigsqcup_{b \in B} b \rightarrow t. \quad (4)$$

Mathematically, (1) and (2) mean that \sqcap is the *meet* over $a \in A$, while (3) and (4) show that \sqcup is the *join* over $b \in B$.

2.1.2 Mutual Interaction of the Two Modes of Choice

We now look at how the two connectives should interact with each other. First we consider the finite case. For simplicity we write

$$\prod_{x \in \{a, b\}} x =: a \sqcap b,$$

$$\bigsqcup_{x \in \{a, b\}} x =: a \sqcup b.$$

Suppose we have a series of choices given by $a \sqcap (b \sqcup c)$. This means we get to freely choose between getting a for certain, or between the uncertainty of getting b , or c . In this scenario, we can always guarantee a if we want, but we are uncertain whether we will get b or c , if we don't choose a . Compare this with the series of choices given by $(a \sqcap b) \sqcup (a \sqcap c)$. Here too we can always guarantee the a , but are uncertain whether we will get b or c , if we don't choose a . In fact, in terms of which resources we can guarantee, the two formulations are equivalent from our perspective. Hence, we can assume the following distributive law

$$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c).$$

Dually, we can compare $a \sqcup (b \sqcap c)$ with $(a \sqcup b) \sqcap (a \sqcup c)$. In the first expression we are uncertain whether we will get a , or a free choice between b and c . This means that if we don't get a , we can guarantee either b or c . This is also the case in the second expression. Hence, we may assume the distributive law

$$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c).$$

More generally, these equalities should also hold for the infinite case. The statement of the infinite distributive law is more complicated: For any doubly indexed family of elements $\{b_{j,k} : j \in J, k \in K_j\}$ of \mathcal{R} we have

$$\bigsqcap_{j \in J} \left(\bigsqcup_{k \in K_j} b_{j,k} \right) = \bigsqcup_{f \in F} \left(\bigsqcap_{j \in J} b_{j,f(j)} \right), \quad (5)$$

where F is the set of all choice functions choosing for each index $j \in J$ some index $f(j) \in K_j$.

Suppose we are presented with a choice of the form $\bigsqcap_{j \in J} \left(\bigsqcup_{k \in K_j} b_{j,k} \right)$. Then we can guarantee $\bigsqcup_{k \in K_j} b_{j,k}$ for any $j \in J$ we choose. That is we can guarantee uncertainty only between the $b_{j,k}$ for one specific j of our choice. Now consider $\bigsqcup_{f \in F} \left(\bigsqcap_{j \in J} b_{j,f(j)} \right)$. Here, some $f \in F$ is chosen for us. We then get to choose freely between the $b_{j,f(j)}$ for that fixed f . Hence we still have uncertainty among a set of $b_{j,k}$ for a j of our choice. Moreover, f picks precisely one $k = f(j)$ among all $k \in K_j$, so the set of $b_{j,k}$ for which there is uncertainty remains the same in both cases.

We can also formulate the dual distributive law

$$\bigsqcup_{j \in J} \left(\bigsqcap_{k \in K_j} b_{j,k} \right) = \bigsqcap_{f \in F} \left(\bigsqcup_{j \in J} b_{j,f(j)} \right), \quad (6)$$

which happens to be equivalent to (5) provided all both $\bigsqcap_{a \in A}$ and $\bigsqcup_{a \in A}$ exist for any subset $A \subseteq \mathcal{R}$.

2.1.3 Laws for the Choice Connectives

We have seen that the choice connectives \bigsqcap and \bigsqcup on a resource preorder should behave like meets and joins over arbitrary subsets of \mathcal{R} . Moreover, they should distribute over one another, even in the infinite case. Hence, any structure that we use to model choice should be a completely distributive lattice. Conversely, such a model shouldn't satisfy any other laws that are not implied by our desired laws. This leads us to consider the free distributive lattice on \mathcal{R} to model choice.

There are several notions of free distributive lattice one could use, depending on what set of maps one is interested in. We will follow [Mor04] which shows that the upper sets of lower sets of \mathcal{R} , denoted $\mathbf{U}(\mathbf{L}(\mathcal{R}))$ satisfy the following universal property:

There is an order embedding $i : \mathcal{R} \hookrightarrow \mathbf{U}(\mathbf{L}(\mathcal{R}))$, such that for any monotone function $f : \mathcal{R} \rightarrow M$ into a completely distributive lattice M , there is a unique complete homomorphism $u : \mathbf{U}(\mathbf{L}(\mathcal{R})) \rightarrow M$ (preserving arbitrary meets and joins) satisfying $u \circ i = f$.

In particular, if we put any completely distributive choice structure M on \mathcal{R} , there will be a unique map from $u : \mathbf{U}(\mathbf{L}(\mathcal{R})) \rightarrow M$, preserving choices.

2.2 The Upper-Lower Model

In this section we introduce and interpret the choice structure on a resource preorder \mathcal{R} given by $\mathbf{U}(\mathbf{L}(\mathcal{R}))$. First, we will show how to use the upper-lower construction as a purely formal tool that realizes the desired properties we derived in Sections 2.1.1 and 2.1.2. Then, we will reconstruct the upper-lower model by giving an interpretation to the sets appearing therein. This will include deriving the corresponding laws based on that interpretation. The fact that both approaches agree suggests that we have chosen a good model.

2.2.1 Formally using the Upper-Lower Model

We begin by describing $\mathbf{U}(\mathbf{L}(\mathcal{R}))$ in greater detail. First, we define the lattice of lower sets.

Definition 2.1 (Lower Sets). Let (P, \rightarrow) be a preorder. A subset $L \subseteq P$ is a lower set, iff for any $l \in L$, $x \rightarrow l$ implies $x \in L$. The set of lower sets of P , ordered by inclusion is denoted by $\mathbf{L}(P)$.

Lemma 2.1. *For any preorder P , $\mathbf{L}(P)$ is a completely distributive lattice with meets given by set intersection and joins by set union.*

Definition 2.2 (Lower Closure). Given a subset $A \subseteq P$, its lower closure is the lower set given by

$$\downarrow A := \{x \in P : x \rightarrow a \text{ for some } a \in A\}.$$

Lemma 2.2. *For any preorder P , we can embed $i : P \hookrightarrow \mathbf{L}(P)$ by taking lower closures $p \mapsto \downarrow \{p\}$. This map is injective and satisfies $p \rightarrow q \Leftrightarrow i(p) \subseteq i(q)$.*

Dually, we can make the same definitions for upper sets.

Definition 2.3 (Upper Sets). Let (P, \rightarrow) be a preorder. A subset $U \subseteq P$ is an upper set, iff for any $u \in U$, $u \rightarrow x$ implies $x \in U$. The set of upper sets of P , ordered by containment is denoted by $\mathbf{U}(P)$.

Lemma 2.3. *For any preorder P , $\mathbf{U}(P)$ is a completely distributive lattice with meets given by set union and joins by set intersection.*

Definition 2.4 (Lower Closure). Given a subset $A \subseteq P$, its lower closure is the lower set given by

$$\uparrow A := \{x \in P : a \rightarrow x \text{ for some } a \in A\}.$$

Lemma 2.4. *For any preorder P , we can embed $i : P \hookrightarrow \mathbf{U}(P)$ by taking upper closures $p \mapsto \uparrow \{p\}$. This map is injective and satisfies $p \rightarrow q \Leftrightarrow i(p) \supseteq i(q)$.*

In particular, for any preorder of resources \mathcal{R} , $\mathbf{U}(\mathbf{L}(\mathcal{R}))$ is completely distributive with meets given by set union and joins by set intersection. Moreover, we have an injective map $i : \mathcal{R} \hookrightarrow \mathbf{U}(\mathbf{L}(\mathcal{R}))$ given by $r \mapsto \uparrow \downarrow \{r\}$ that satisfies

$$r \rightarrow s \Leftrightarrow i(r) \supseteq i(s).$$

We implement our connectives \sqcap and \sqcup in $\mathbf{U}(\mathbf{L}(\mathcal{R}))$ by setting \sqcap to be the meet and \sqcup to be the join in $\mathbf{U}(\mathbf{L}(\mathcal{R}))$: For a subset $A \subseteq \mathbf{U}(\mathbf{L}(\mathcal{R}))$,

$$\prod_{a \in A} a := \bigcup_{a \in A} a,$$

$$\bigsqcup_{a \in A} a := \bigcap_{a \in A} a.$$

In particular, if we are interested choice between elements of \mathcal{R} , we first embed those elements into $\mathbf{U}(\mathbf{L}(\mathcal{R}))$ and then apply the corresponding operation. That is if $B \subseteq \mathcal{R}$,

$$\prod_{r \in B} a := \bigcup_{r \in B} i(r),$$

$$\bigsqcup_{r \in B} a := \bigcap_{r \in B} i(r).$$

Example 2.2 (Menu 2). In a restaurant we are allowed to choose between a vegetarian and a meat option. If we choose the vegetarian option, the restaurant will provide us a curry dish, or a casserole, depending on availability. If we choose the meat option we will get chicken or beef, depending on availability. Using our connectives, we can describe this as

$$(\text{curry} \sqcup \text{casserole}) \sqcap (\text{chicken} \sqcup \text{beef}).$$

We model our options as the discrete preorder $\{\text{curry}, \text{casserole}, \text{chicken}, \text{beef}\}$. Using the rules above for describing this choice in the upper-lower model, we represent

$$\begin{aligned} \text{curry} \sqcup \text{casserole} &= i(\text{curry}) \cap i(\text{casserole}) \\ &= \uparrow \{\{\text{curry}\}\} \cap \uparrow \{\{\text{casserole}\}\} \\ &= \uparrow \{\{\text{curry}, \text{casserole}\}\}. \end{aligned}$$

A similar calculation shows $\text{chicken} \sqcup \text{beef} = \uparrow \{\{\text{chicken}, \text{beef}\}\}$. Hence,

$$\begin{aligned} (\text{curry} \sqcup \text{casserole}) \sqcap (\text{chicken} \sqcup \text{beef}) &= \uparrow \{\{\text{curry}, \text{casserole}\}\} \cup \uparrow \{\{\text{chicken}, \text{beef}\}\} \\ &= \uparrow \{\{\text{curry}, \text{casserole}\}, \{\text{chicken}, \text{beef}\}\}. \end{aligned}$$

Example 2.3 (Menu 3). A restaurant offers two menus, depending on the day. As customers we can freely choose any option on the current menu. Suppose the menu items the first menu are $\{\text{curry}, \text{beef}\}$ and those on the second are $\{\text{casserole}, \text{chicken}\}$. In terms of our connectives, this choice becomes

$$(\text{curry} \sqcap \text{beef}) \sqcup (\text{casserole} \sqcap \text{chicken}).$$

In the upper-lower model we represent

$$\begin{aligned} \text{curry} \sqcap \text{beef} &= i(\text{curry}) \cup i(\text{beef}) \\ &= \uparrow \{\{\text{curry}\}\} \cup \uparrow \{\{\text{beef}\}\} \\ &= \uparrow \{\{\text{curry}\}, \{\text{beef}\}\}. \end{aligned}$$

Similarly, $\text{casserole} \sqcap \text{chicken} = \uparrow \{\{\text{casserole}\}, \{\text{chicken}\}\}$. Hence,

$$\begin{aligned} (\text{curry} \sqcap \text{beef}) \sqcup (\text{casserole} \sqcap \text{chicken}) &= \uparrow \{\{\text{curry}\}, \{\text{beef}\}\} \cap \uparrow \{\{\text{casserole}\}, \{\text{chicken}\}\} \\ &= \uparrow \{\{\text{curry}, \text{casserole}\}, \{\text{curry}, \text{chicken}\}, \\ &\quad \{\text{beef}, \text{casserole}\}, \{\text{beef}, \text{chicken}\}\}. \end{aligned}$$

The examples above show the need for some rules for calculating unions and intersections in $\mathcal{U}(\mathcal{L}(\mathcal{R}))$.

Calculation Rules

Lemma 2.5. *The following equalities hold for upper closures of singletons:*

$$(i) \uparrow \{a\} \cup \uparrow \{b\} = \uparrow \{a, b\}$$

$$(ii) \uparrow \{a\} \cap \uparrow \{b\} = \uparrow \{a \cup b\}$$

Lemma 2.6. *For arbitrary upper closures we have:*

$$(i) \uparrow A \cup \uparrow B = \uparrow \{x : x \in A \text{ or } x \in B\} = \uparrow (A \cup B)$$

$$(ii) \bigcup_{i \in I} (\uparrow A_i) = \uparrow (\bigcup_{i \in I} A_i)$$

$$(iii) \uparrow A \cap \uparrow B = \uparrow \{a \cup b : a \in A, b \in B\}$$

$$(iv) \bigcap_{i \in I} (\uparrow A_i) = \uparrow \{\bigcup a_i : a_i \in A_i \text{ for all } i\} = \{\bigcup a_i : a_i \in \uparrow A_i \text{ for all } i\}$$

Proof. It suffices to prove (ii) and (iv).

(ii) Suppose $a \in \bigcup_{i \in I} (\uparrow A_i)$. Then $a \in \uparrow A_i$ for some i . In particular, $a_i \subseteq a$ for $a_i \in A_i$. Hence $a_i \subseteq a$ for $a_i \in \bigcup_{i \in I} A_i$, showing $a \in \uparrow (\bigcup_{i \in I} A_i)$.

Conversely, if $a \in \uparrow (\bigcup_{i \in I} A_i)$, then $a_i \subseteq a$ for some $a_i \in \bigcup_{i \in I} A_i$, that is for some $a_i \in A_i$. Thus $a \in \uparrow A_i$, meaning $a \in \bigcup_{i \in I} (\uparrow A_i)$.

(iv) Suppose $a \in \bigcap_{i \in I} (\uparrow A_i)$. Then for all i there exists $a_i \in A_i$ such that $a_i \subseteq a$. Thus $\bigcup a_i \subseteq a$, meaning $a \in \uparrow \{\bigcup a_i : a_i \in A_i \text{ for all } i\}$.

Conversely, if $a \in \uparrow \{\bigcup a_i : a_i \in A_i \text{ for all } i\}$, there are i such that $\bigcup a_i \subseteq a$, where $a_i \in A_i$. Hence for all i , $a_i \subseteq \bigcup a_i \subseteq a$, meaning $a \in \bigcap_{i \in I} (\uparrow A_i)$.

The final equality assume $a \in \uparrow \{\bigcup a_i : a_i \in A_i \text{ for all } i\}$. Then there are i such that $\bigcup a_i \subseteq a$, where $a_i \in A_i$. This implies $a_i \subseteq a$ for all i , whence $a \in \uparrow A_i$ for all i . Hence $a \in \bigcap_{i \in I} (\uparrow A_i)$.

If $a \in \bigcap_{i \in I} (\uparrow A_i)$, then there are i such that $\bigcup a_i = a$, where $a_i \in \uparrow A_i$. For each i , we have $x_i \subseteq a_i$ for $x_i \in A_i$. Hence $\bigcup x_i \subseteq \bigcup a_i = a$, showing $a \in \uparrow \{\bigcup a_i : a_i \in A_i \text{ for all } i\}$. \square

2.2.2 Interpretation-Based Reconstruction

3 Feasibility between Choices

4 Choice between Transformations

5 Towards Modeling Dependencies

References

- [Mor04] Joseph M. Morris. “Augmenting types with unbounded demonic and angelic nondeterminacy”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3125 (2004), pp. 274–288. ISSN: 16113349. DOI: [10.1007/978-3-540-27764-4_15](https://doi.org/10.1007/978-3-540-27764-4_15). URL: https://link.springer.com/chapter/10.1007/978-3-540-27764-4_15.