Modeling Choice in Co-Design

Marius Furter

June 12, 2021

Abstract

Write this

Contents

1 Introduction

Living life involves making choices. Before a choice is made, a set of unrealized options lies dormant. Making a choice involves collapsing this set to a single outcome. This collapse may occur in several ways. On the one extreme, we may be in full control of which option is realized: We are able to freely choose a desired outcome. On the other extreme, we may be powerless to steer the collapse: A choice is forced upon us.

Choice is relevant in any discipline that involves a multiplicity of potential outcomes. In computer science, for example, choice is used to describe program specifications [MCR07; Mor04]. Here the choice is between possible outputs of non-deterministic code. Free choice corresponds to the code behaving in the programmers best interest, while forced choice describes an uncertainty of outcome. These two modes are personified by an angel choosing on behalf of the programmer in the first case, while a demon chooses in the latter.

Choice also arises whenever one requires coordination between agents. Alice might only be able to guarantee a range of outcomes to Bob. This is weaker than guaranteeing a specific outcome. On the other hand, Alice may be able to offer Bob a free choice between a set of outcomes. This is stronger than offering a single outcome. We see that allowing for free and forced choices expands the space of possible interactions between Alice and Bob.

In engineering too, choices abound. A team can freely choose a specific motor from a catalog of options. On the other hand, a team might only be able to guarantee that their design will fulfill one of several properties: Their robot will either be able to move fast or carry a lot of weight, but they aren't specifying which. Such uncertainty in outcome has ramifications for other teams working on the project.

Questions of coordination between teams are addressed in the Co-Design framework [Cen15]. However, the original framework does not include a way to express free and forced choices. Here we aim to extend the framework in a way that does.

1.1 Goals

This report aims to succinctly describe how to model free and forced choice within the Co-Design framework. Additionally, it will highlight avenues for future exploration. Ideally, it could serve as a starting point for anyone seeking to make further contributions to this topic.

1.2 Structure

2 A Brief Introduction to Co-Design

We begin by providing a short introduction to the Co-Design framework established in [Cen15]. This will also serve to fix the conventions we will be using throughout.

In Co-Design we consider resources that can be used to achieve functionalities. For example, we might consider different types of motor as resources, with the power output they provide as functionalities. If a resource r can be used to provide functionality f we say that the pair (r, f) is feasible. Continuing our example, if motor A is capable of ouputting 2 kW then we would mark $(A, 2\text{ kW})$ as feasible.

It can occur that certain resources or functionalities imply others. If motor A can provide 2 kW, it should also be able to provide 1 kW. We will indicate this by writing 2 kW $\rightarrow$ 1 kW. Hence, our sets of resources and functionalities posses additional structure: They are preorders. Consequently, we want our notion of feasibility to take this structure into account. This leads to the concept of *feasibility relation*. We will make these ideas precise in what follows.

2.1 Resource and Functionality Preorders

We model resources and functionalities using preordered sets, which will be formally introduced shortly. We interpret an element a of the preorder as the guarantee "I can provide a". We put an arrow $a \to b$ whenever being able to provide a implies being able to provide b. This is best illustrated with an example.

Example 2.1. (Money as Resources) The set $\mathbb{N} := \{0, 1, 2, \ldots\}$ with the order relation $a \to b \Leftrightarrow a \geq b$ can be used to describe money (of a single currency, say dollars) as a resource. The element a represents being able to provide a dollars, while $a \to b$ means that if you can provide a dollars, you can also provide b dollars. For example, $3 \to 1$ signifies that if we could pay 3 dollars, we also possess the purchasing power of 1 dollar. Observe that being able to transform $3 \to 1$ presupposes that 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 $50 \to 1$.

Returning to the general theory, we observe that the preorder axioms precisely describe how we would like the relation $a \to b$ to behave.

Definition 2.1. (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, $a \to b$ means (a, b) is part of the relation R.

Using our infix notation, the two axioms become

(r) $a \to a$ for all a,

(t) $a \to b$ and $b \to c$ imply $a \to c$.

In other words, for each resource a, being able to provide a implies being able to provide b. Further, if being able to provide a implies being able to provide b, and being able to provide b implies being able to provide c, then being able to provide a should imply being able to provide c.

Everything that has been said here applies equally to functionalities. In fact, what we view as resources and 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.

Resources and Functionalities

Resources and functionalities are modeled by preorders. An element a in the preorder expresses the guarantee *"I can provide a"*. An arrow $a \to b$ means *"being able to provide a implies being able to provide b"*.

Remark 2.1. There are some fine points here that may be skipped on a first reading. Observe that our guarantee does not specify who we are providing a to. The recipient might be another agent or our (future) selves. More generally, there are other interpretations we could give the elements and arrows in our preorders. Two of them are discussed in the following boxes. Ultimately, these just provide a different flavor to the theory. Our preferred interpretation fits best with thinking about choice.

Functionality Interpretation

Following the logic of Co-Design, a resource is determined by the functionality it can provide. From this perspective, we could alternatively set $a \to b$ whenever a provides at least the same level of functionality as b does. In this case, in terms of the functionality on offer, being able to provide a does implies being able to provide b. Conversely, if being able to provide a implies being able to provide b, then a must offer at least the same level of functionality as b. This shows that the two interpretations align.

Transformation Interpretation

We could also interpret $a \to b$ as meaning that we can freely and instantly transform a into b. If a freely transforms into b, then being able to provide a implies being able to provide b. Conversely, if being able to provide a implies being able to provide b, then given a, we should be able to provide b. This can be seen as a free transformation of a into b.

2.2 Feasibility Relations

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

Example 2.2. Consider buying groceries. Our resources $\mathcal{R}$ will be money as in Example 2.1. The possible groceries we can buy are the functionalities $\mathcal{F}$. The feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ will describe what groceries we can purchase given a specific amount of cash. For example, a `carrot` might cost

1\$. Hence, $\Phi(1\$, \texttt{carrot}) = \texttt{true}$. On the other hand, $\Phi(0\$, \texttt{carrot}) = \texttt{false}$, since we can't get a `carrot` without paying anything. Suppose now that a `bag of carrots` costs 3\$. This means $\Phi(3\$, \texttt{bag of carrots}) = \texttt{true}$. Observe that also $\Phi(3\$, \texttt{carrot}) = \texttt{true}$ should hold. Either we can pick up a single carrot in the store and pay for it with 3\$, or we could buy the whole bag for 3\$ and this would also give us a single `carrot`.

This shows that feasibility interacts with the implication arrows we have for the resources and functionalities. In our example,

$$3\$ \to 1\$ \text{ and } \Phi(1\$, \texttt{carrot}) = \texttt{true} \text{ implied } \Phi(3\$, \texttt{carrot}) = \texttt{true}.$$

Similarly,

$$\texttt{bag of carrots} \to \texttt{carrot} \text{ and } \Phi(3\$, \texttt{bag of carrots}) = \texttt{true}$$

implied that $\Phi(3\$, \texttt{carrot}) = \texttt{true}$.

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

Definition 2.2 (Feasibility Relation)**.** Given preorders $\mathcal{R}$ and $\mathcal{F}$, a *feasibility relation* $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ is a relation $\Phi \subseteq \mathcal{R} \times \mathcal{F}$ such that

(i) if $r \to s$ in $\mathcal{R}$ and $(s, f) \in \Phi$, then $(r, f) \in \Phi$,

(ii) if $f \to g$ in $\mathcal{F}$ and $(r, f) \in \Phi$, then $(r, g) \in \Phi$.

If $(r, f) \in \Phi$ we write $\Phi(r, f) = \texttt{true}$ and 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. We put an arrow between an $r \in \mathcal{R}$ and $f \in \mathcal{F}$ iff $\Phi(r, f) = \texttt{true}$. The arrows coming from Φ can be thought of the transformations that Φ enables. Observe that the closure conditions in Definition 2.2 mean that the arrows coming from Φ are transitively closed 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.

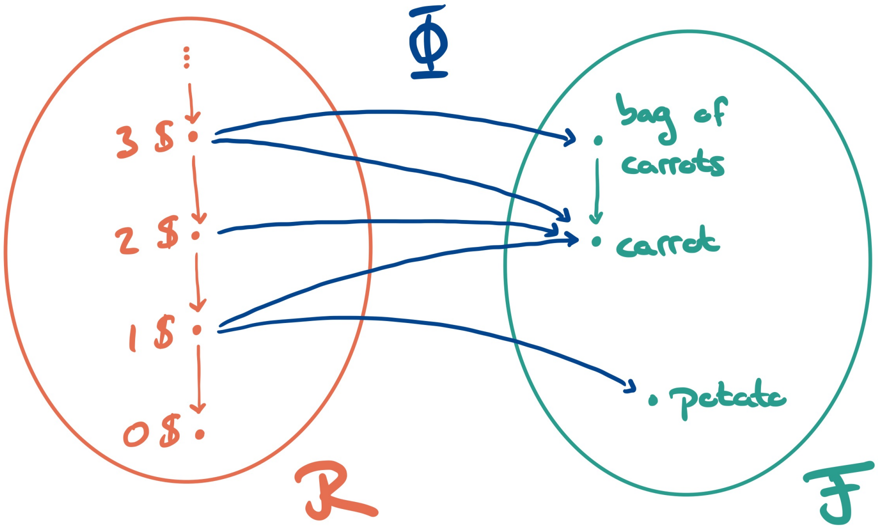

Figure 1: Internal diagram of a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$.

Example 2.2 (continued)*.* We can now formally implement our example of grocery shopping. Recall that our resource preorder $\mathcal{R}$ consists of money, while our functionality preorder $\mathcal{F}$ describes groceries we can purchase. Figure 1 shows part of an internal diagram of a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$. The blue arrows indicate the feasible pairs. For example,

$$\Phi(3\$, \texttt{bag of carrots}) = \texttt{true}.$$

It is worth contemplating why Φ satisfies the monotonicity conditions presented in Definition 2.2.

We could have also defined feasibility relations as special types of monotone maps. For this we need to introduce several concepts.

Definition 2.3. Let Bool denote the preorder generated by $\{\texttt{false} \to \texttt{true}\}$. Explicitly, Bool contains the arrows $\texttt{false} \to \texttt{false}$, $\texttt{false} \to \texttt{true}$, and $\texttt{true} \to \texttt{true}$.

Definition 2.4 (Opposite Preorder)**.** Given a preorder $(\mathcal{P}, \to)$, we define its *opposite preorder* $\mathcal{P}^{\mathrm{op}}$ to be the preorder that is obtained by reversing all the arrows in $\mathcal{P}$. That is, we set $a \to b$ in $\mathcal{P}^{\mathrm{op}}$ if and only if $b \to a$ in $\mathcal{P}$.

Definition 2.5 (Product of Preorders)**.** Let $\mathcal{P}$ and $\mathcal{Q}$ be preorders. Their *product* $\mathcal{P} \times \mathcal{Q}$ consists of all pairs (p, q) where $p \in \mathcal{P}$ and $q \in \mathcal{Q}$ where we set $(p, q) \to (p', q')$ if and only if $p \to p'$ in $\mathcal{P}$ and $q \to q'$ in $\mathcal{Q}$.

Definition 2.6 (Monotone Map)**.** Let $\mathcal{P}$ and $\mathcal{Q}$ be preorders. A function $f : \mathcal{P} \to \mathcal{Q}$ is called *monotone* if it preserves the order in $\mathcal{P}$. That is, whenever $a \to b$ in $\mathcal{P}$, we have $f(a) \to f(b)$ in $\mathcal{Q}$.

Lemma 2.1. *There is a one-to-one correspondence between feasibility relations $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ and monotone maps $\mathcal{R}^{op} \times \mathcal{F} \to \mathsf{Bool}$.*

Proof. Given a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ as defined in Definition 2.2, we obtain a function $F : \mathcal{R}^{\text{op}} \times \mathcal{F} \to \mathsf{Bool}$ by sending $(r, f) \mapsto \texttt{true}$ if and only if $(r, f) \in \Phi$. Conditions (i) and (ii) of Definition 2.2 say that this function is monotone. Conversely, given a monotone map $F : \mathcal{R}^{\text{op}} \times \mathcal{F} \to \mathsf{Bool}$, we can obtain a relation Φ by setting $(r, f) \in \Phi$ if and only if $F(r, f) = \texttt{true}$. The monotonicity of this function implies that Φ satisfies conditions (i) and (ii). It is apparent that these processes are inverse to one another. ∎

Feasibility Relations

A feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ expresses which functionalities in $\mathcal{F}$ can be obtained from which resources in $\mathcal{R}$. We write $\Phi(r, f) = \texttt{true}$ to indicate that $f \in \mathcal{F}$ can be obtained from $r \in \mathcal{R}$. The relation Φ needs to satisfy the two monotonicity conditions in Definition 2.2 which can be concisely expressed by saying the Φ is a monotone map $\mathcal{R}^{\text{op}} \times \mathcal{F} \to \mathsf{Bool}$.

3 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 3.1 (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 3.1 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.

3.1 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 $\bigsqcap_{a \in A} a$, while for a forced choice from a set B we write $\bigsqcup_{b \in B} b$. We will now see how such choices should interact with the free transformations we have in a resource preorder, as well as each other.

3.1.1 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

$$\bigsqcap_{a \in A} a \to a, \quad \forall a \in A. \tag{1}$$

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,

$$\forall a \in A : t \to a \quad \Rightarrow \quad t \to \bigsqcap_{a \in A} a. \tag{2}$$

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

$$b \to \bigsqcup_{b \in B} b, \quad \forall b \in B. \tag{3}$$

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,

$$\forall b \in B : b \to t \quad \Rightarrow \quad \bigsqcup_{b \in B} b \to t. \tag{4}$$

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 $b \in B$.

3.1.2 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

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

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

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

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

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

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

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

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

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 chose 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

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

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

3.1.3 Laws for the Choice Connectives

We have seen that the choice connectives $\sqcap$ and $\sqcup$ 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 $\mathsf{U}(\mathsf{L}(\mathcal{R}))$ satisfy the following universal property:

> There is an order embedding $i : \mathcal{R} \hookrightarrow \mathsf{U}(\mathsf{L}(\mathcal{R}))$, such that for any monotone function $f : \mathcal{R} \to M$ into a completely distributive lattice M, there is a unique complete homomorphisms $u : \mathsf{U}(\mathsf{L}(\mathcal{R})) \to M$ (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 $u : \mathsf{U}(\mathsf{L}(\mathcal{R})) \to M$, preserving choices.

3.2 The Upper-Lower Model

In this section we introduce and interpret the choice structure on a resource preorder $\mathcal{R}$ given by $\mathsf{U}(\mathsf{L}(\mathcal{R}))$. 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 3.1.1 and 3.1.2. 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.

3.2.1 Formally using the Upper-Lower Model

We begin by describing $\mathsf{U}(\mathsf{L}(\mathcal{R}))$ in greater detail. We again fix $\sqcap$ as free choice and $\sqcup$ as forced choice. First, we define the lattice of lower sets.

Definition 3.1 (Lower Sets)**.** Let $(P, \to)$ be a preorder. A subset $L \subseteq P$ is a lower set, iff for any $l \in L$, $x \to l$ implies $x \in L$. The set of lower sets of P, ordered by inclusion is denoted by $\mathsf{L}(P)$.

Lemma 3.1. *For any preorder P, $\mathsf{L}(P)$ is a completely distributive lattice with meets given by set intersection and joins by set union.*

Definition 3.2 (Lower Closure)**.** Given a subset $A \subseteq P$, its lower closure is the lower set given by

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

Lemma 3.2. *For any preorder P, we can embed $i : P \hookrightarrow \mathsf{L}(P)$ by taking lower closures $p \mapsto \downarrow \{p\}$. This map is injective and satisfies $p \to q \Leftrightarrow i(p) \subseteq i(q)$.*

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

Definition 3.3 (Upper Sets)**.** Let $(P, \to)$ be a preorder. A subset $U \subseteq P$ is an upper set, iff for any $u \in U$, $u \to x$ implies $x \in U$. The set of upper sets of P, ordered by containment is denoted by $\mathsf{U}(P)$.

Lemma 3.3. *For any preorder P, $\mathsf{U}(P)$ is a completely distributive lattice with meets given by set union and joins by set intersection.*

Definition 3.4 (Lower Closure)**.** Given a subset $A \subseteq P$, its lower closure is the lower set given by

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

Lemma 3.4. *For any preorder P, we can embed $i : P \hookrightarrow \mathsf{U}(P)$ by taking upper closures $p \mapsto \uparrow \{p\}$. This map is injective and satisfies $p \to q \Leftrightarrow i(p) \supseteq i(q)$.*

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

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

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

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

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

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

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

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

Example 3.2 (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

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

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

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

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

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

Example 3.3 (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 $\{\texttt{curry}, \texttt{beef}\}$ and those on the second are $\{\texttt{casserole}, \texttt{chicken}\}$. In terms of our connectives, this choice becomes

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

In the upper-lower model we represent

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

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

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

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

Calculation Rules

Lemma 3.5. *For arbitrary upper closures of $A, B \subseteq \mathsf{L}(\mathcal{R})$ 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) $\uparrow A \cap \uparrow B = \uparrow \{a \cup b : a \in A, b \in B\} = \{a \cup b : a \in \uparrow A, b \in \uparrow B\}$

(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 $a = \bigcup a \in \{\bigcup a_i : a_i \in \uparrow A_i \text{ for all } i\}$.
If $a \in \{\bigcup a_i : a_i \in \uparrow A_i \text{ for all } i\}$, 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 $a \in \uparrow \{\bigcup a_i : a_i \in A_i \text{ for all } i\}$. ∎

Lemma 3.6. *The following equalities hold for $a, b, \in \mathcal{R}$:*

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

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

Proof. Use Lemma 3.5 and the fact that $\downarrow \{a\} \cup \downarrow \{b\} = \downarrow \{a, b\}$. ∎

3.2.2 Interpretation-Based Reconstruction

In this section we will reconstruct $\mathsf{U}(\mathsf{L}(\mathcal{R}))$ based on interpretation. For this we fix $\sqcap$ as free choice and $\sqcup$ as forced choice. Because of distributivity, we can always represent any choice as a free choice over a set of forced choices.

We start by considering a forced choice of resources from a set $A \subseteq \mathcal{R}$. We will argue that $\bigsqcup_{a \in A} a \cong \bigsqcup_{a \in \downarrow A} a$, in the sense that we have the same information on the outcome of the choice in both cases. Observe that we have the following monotonicity law

$$a \to b \quad \Leftrightarrow \quad a \sqcup b \cong b.$$

If $a \to b$, then since $b \to b$, we have $a \sqcup b \to b$ by law (4). By law (3), $b \to a \sqcup b$, so $a \sqcup b \cong b$. Conversely, if $a \sqcup b \cong b$, then $a \to a \sqcup b \cong b$.

More generally,

$$\bigsqcup_{a \in A} a \cong b \quad \Leftrightarrow \quad a \to b, \, \forall a \in A.$$

In particular, for every $a \in A$, we have $\bigsqcup_{x \in \downarrow \{a\}} x \cong a$. Now $\downarrow A = \bigcup_{a \in A} \downarrow a$. Therefore,

$$\bigsqcup_{a \in \downarrow A} a = \bigsqcup_{x \in \bigcup_{a \in A} \downarrow \{a\}} x = \bigsqcup_{a \in A} \bigsqcup_{x \in \downarrow \{a\}} x \cong \bigsqcup_{a \in A} a.$$

This shows that for each forced choice, we can choose a representative of that forced choice, where the set that is being chosen from is a lower set. Hence, $\mathsf{L}(\mathcal{R})$ can be seen as the set of equivalence classes of forced choices between elements of $\mathcal{R}$.

By duality we have

$$\bigsqcap_{a \in \uparrow A} a \cong \bigsqcap_{a \in A} a.$$

This implies that we can represent each free choice (up to isomorphism) by an upper set. Hence, $\mathsf{U}(P)$ is the set of equivalence classes of free choices between elements of any preorder P. In particular, if we set $P := \mathsf{L}(\mathcal{R})$ we see that we can represent the set of free choices among forced choices among elements of $\mathcal{R}$ by $\mathsf{U}(\mathsf{L}(\mathcal{R}))$.

It remains to show that the natural meets and joins on $\mathsf{U}(\mathsf{L}(\mathcal{R}))$ align with the operations given by our interpretation. Let $A \in \mathsf{U}(\mathsf{L}(\mathcal{R}))$ represent a free choice over some forced choices a_i. That is, $A = \{a_i\}_{i \in I}$ represents

$\bigsqcap_{i\in I} a_i$. Similarly, suppose $B \in \mathsf{U}(\mathsf{L}(\mathcal{R}))$ is given by $B = \{b_j\}_{j\in J}$, representing $\bigsqcap_{j\in J} b_j$. Because

$$\bigsqcap_{i\in I} a_i \sqcap \bigsqcap_{j\in J} b_j = \bigsqcap_{a\in A} a \sqcap \bigsqcap_{b\in B} b \cong \bigsqcap_{x\in A\cup B} x,$$

we see that $A \cup B$ represents the free choice between A and B.

To see how forced choice is represented, we take $A, B \in \mathsf{U}(\mathsf{L}(\mathcal{R}))$. Then $A = \{a_i\}_{i\in I}$, where $a_i = \{r_{i,k}\}_{k\in K_i}$. Similarly, $B = \{b_l\}_{l\in L}$, where $b_l = \{r_{l,m}\}_{m\in M_l}$. Hence A represents $\bigsqcap_{i\in I} \bigsqcup_{k\in K_i} r_{i,k}$ and B represents $\bigsqcap_{l\in L} \bigsqcup_{m\in M_l} r_{l,m}$, where we assume that all index sets are disjoint.

Using distributivity

$$\begin{aligned}\bigsqcap_{i\in I} \bigsqcup_{k\in K_i} r_{i,k} \sqcup \bigsqcap_{l\in L} \bigsqcup_{m\in M_l} r_{l,m} &\cong \bigsqcap_{i\in I} \bigsqcap_{l\in L} \left(\bigsqcup_{k\in K_i} r_{i,k} \sqcup \bigsqcup_{m\in M_l} r_{l,m} \right) \\ &\cong \bigsqcap_{(i,l)\in I\times L} \bigsqcup_{j\in K_i \cup M_l} r_{h(j),j},\end{aligned}$$

where $h(j) = \begin{cases} i, & \text{if } j \in K_i \\ l, & \text{if } j \in M_l \end{cases}$.

Therefore, the set representing the forced choice between A and B is $\{a \cup b : a \in A, b \in B\}$, which be Lemma 3.5 (iii) is precisely equal to $A \cap B$.

In summary, we have shown that $\mathsf{U}(\mathsf{L}(\mathcal{R}))$ corresponds exactly to the set of equivalence classes of free choices among forced choices among elements of $\mathcal{R}$. Moreover, the meet $A \cup B$ in $\mathsf{U}(\mathsf{L}(\mathcal{R}))$ represents the free choice between the choices represented by A and B. The join $A \cap B$ represents the forced choice between the choices represented by A and B. Analogous arguments can be made for arbitrary meets and joins.

4 Feasibility between Choices

Having constructed a model for choices between the resources in a preorder $\mathcal{R}$, we turn to the task of extending feasibility relations to the model. The first task will be to lift a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ to the upper-lower models. We will see that such lifted feasibility relations display a certain kind of optimality in comparison to arbitrary feasibility relations between the upper-lower models.

4.1 Lifting Feasibility to Upper-Lower Models

We have seen in Section 3.2.2 that the choice of model is essentially determined by the laws we fixed in Sections 3.1.1 and 3.1.2. In a similar spirit, the way we are going to naturally lift feasibility relations is already determined by the choices we have made so far.

Let $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ be a feasibility relation between preorders $\mathcal{R}$ and $\mathcal{F}$. Lift this to a feasibility relation $\mathsf{L}(\Phi) : \mathsf{L}(\mathcal{R}) \nrightarrow \mathsf{L}(\mathcal{F})$ by setting

$$\mathsf{L}(\Phi)(A, B) = \texttt{true} \iff \forall a \in A\, \exists b \in B : \Phi(a, b) = \texttt{true}.$$

This is a feasibility relation since 'for all' is preserved by taking subsets and 'there exists' by taking supersets. Note that the choice of quantifiers is completely determined by the fact that we are ordering lower sets by inclusion. This choice was in turn fixed by the fact that forced choice is represented by lower sets, and forced choices over set can be weakened to forced choices over a superset. Hence, our lifting is fixed by the fact that forced choice behaves like a join.

Moving up a level, let $\Psi : \mathsf{L}(\mathcal{R}) \nrightarrow \mathsf{L}(\mathcal{F})$ be a feasibility relation. Lift this to $\mathsf{U}(\Psi) : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ by setting

$$\mathsf{U}(\Psi)(A, B) = \texttt{true} \iff \forall b \in B\, \exists a \in A : \Psi(a, b) = \texttt{true}.$$

This is again a feasibility relation because the choice of quantifiers is compatible with the fact that we order the upper sets by containment. Dual to the above, the way we lift is fixed by the fact that we want free choice to act like a meet.

Putting both lifts together, given $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$, we can lift it to $\mathsf{U}(\mathsf{L}(\Phi)) : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ by iterating the two lifts. Explicitly,

$$\mathsf{U}(\mathsf{L}(\Phi))(A, B) = \texttt{true} \iff \forall b \in B\, \exists a \in A : \forall x \in a\, \exists y \in b : \Phi(a, b).$$

4.2 Characterizing Lifted Feasibility Relations

4.2.1 Lower Set Level

Suppose we have a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ which we lift to $\mathsf{L}(\Phi) : \mathsf{L}(\mathcal{R}) \nrightarrow \mathsf{L}(\mathcal{F})$ by setting

$$\mathsf{L}(\Phi)(A, B) = \texttt{true} \iff \forall a \in A\, \exists b \in B : \Phi(a, b) = \texttt{true}.$$

Decomposition Property If $\mathsf{L}(\Phi)(A, B) = \texttt{true}$, then

$$\forall a \in A \, \exists b \in B : \Phi(a, b) = \texttt{true}.$$

In particular, for every $a \in A$ there is $b \in B$ such that

$$\forall x \in \downarrow \{a\} \, \exists y \in \downarrow \{b\} : \Phi(x, y) = \texttt{true}.$$

This is because for any $x \in \downarrow \{a\}$ we have $\Phi(x, b)$ by monotonicity, and $b \in \downarrow \{b\}$. Hence, for every $a \in A$ there is $b \in B$ such that $\mathsf{L}(\Phi)(\downarrow \{a\}, \downarrow \{b\})$.

Union Property Assume $\mathsf{L}(\Phi)(X_i, B) = \texttt{true}$ for each X_i in a collection $\{X_i\}_{i \in I}$. Then for each X_i we have that

$$\forall x \in X_i \, \exists b \in B : \Phi(x, b) = \texttt{true}.$$

This implies that

$$\forall x \in \bigcup_{i \in I} X_i \, \exists b \in B : \Phi(x, b) = \texttt{true},$$

whence $\mathsf{L}(\Phi)(\bigcup_{i \in I} X_i, B) = \texttt{true}$.

Now assume that we have an arbitrary feasibility relation $\Psi : \mathsf{L}(\mathcal{R}) \nrightarrow \mathsf{L}(\mathcal{F})$ satisfying decomposition property and the union property:

- If $\Psi(A, B) = \texttt{true}$, then for every $a \in A$ there is $b \in B$ such that $\Psi(\downarrow \{a\}, \downarrow \{b\}) = \texttt{true}$.
- For any collection $\{X_i\}_{i \in I} \subseteq \mathsf{L}(\mathcal{R})$,

$$\Psi(X_i, B) = \texttt{true} \;\; \forall i \in I \quad \Rightarrow \quad \Psi(\bigcup_{i \in I} X_i, B) = \texttt{true}.$$

We can obtain a feasibility relation $\tilde{\Psi} : \mathcal{R} \nrightarrow \mathcal{F}$ by setting

$$\tilde{\Psi}(r, f) := \Psi(\downarrow \{r\}, \downarrow \{f\}).$$

If $s \to r$, then $\downarrow \{s\} \subseteq \downarrow \{r\}$, so in this case

$$\tilde{\Psi}(r, f) \Leftrightarrow \Psi(\downarrow \{r\}, \downarrow \{f\}) \Rightarrow \Psi(\downarrow \{s\}, \downarrow \{f\}) \Leftrightarrow \tilde{\Psi}(r, f).$$

On the other hand if $f \to g$, then $\downarrow\{f\} \subseteq \downarrow\{g\}$, hence again

$$\tilde{\Psi}(r,f) \Leftrightarrow \Psi(\downarrow\{r\}, \downarrow\{f\}) \Rightarrow \Psi(\downarrow\{r\}, \downarrow\{g\}) \Leftrightarrow \tilde{\Psi}(r,g).$$

We show that $\mathsf{L}(\tilde{\Psi}) = \Psi$. This requires proving that $\mathsf{L}(\tilde{\Psi})(A,B) = \mathtt{true} \Leftrightarrow \Psi(A,B) = \mathtt{true}$. Suppose $\mathsf{L}(\tilde{\Psi})(A,B) = \mathtt{true}$. This means that

$$\forall a \in A\, \exists b \in B : \Psi(\downarrow\{a\}, \downarrow\{b\}) = \mathtt{true}.$$

In particular, by monotonicity,

$$\forall a \in A\; \Psi(\downarrow\{a\}, B) = \mathtt{true}.$$

By the union property in the source, $\Psi(\bigcup_{a\in A}\downarrow\{a\}, B) = \mathtt{true}$. It remains to observe that $\bigcup_{a\in A}\downarrow\{a\} = \downarrow\bigcup_{a\in A} a = \downarrow A = A$, since A is a lower set. Hence, $\Psi(A,B) = \mathtt{true}$.

Conversely, suppose $\Psi(A,B) = \mathtt{true}$. Then for each $a \in A$ there is $b \in B$ satisfying $\Psi(\downarrow\{a\}, \downarrow\{b\}) = \mathtt{true}$. In symbols,

$$\forall a \in A\, \exists b \in B : \Psi(\downarrow\{a\}, \downarrow\{b\}) = \mathtt{true}.$$

But this is precisely the condition showing that $\mathsf{L}(\tilde{\Psi})(A,B) = \mathtt{true}$. Therefore Ψ is induced by $\tilde{\Psi}$.

The above proves the following characterization of induced feasibility relations on the lower set level.

Proposition 4.1 (Induction Criterion for Lower Sets)**.** *Let $\Psi : \mathsf{L}(\mathcal{R}) \nrightarrow \mathsf{L}(\mathcal{F})$ be a feasibility relation. Then $\Phi = \mathsf{L}(\tilde{\Phi})$ for some $\tilde{\Phi} : \mathcal{R} \nrightarrow \mathcal{F}$ if and only if Φ satisfies the following:*

- *If $\Psi(A,B) = \mathtt{true}$, then for every $a \in A$ there is $b \in B$ such that $\Psi(\downarrow\{a\}, \downarrow\{b\}) = \mathtt{true}$.*

- *For any collection $\{X_i\}_{i\in I} \subseteq \mathsf{L}(\mathcal{R})$,*

$$\Psi(X_i, B) = \mathtt{true}\ \forall i \in I \quad \Rightarrow \quad \Psi(\bigcup_{i\in I} X_i, B) = \mathtt{true}.$$

4.2.2 Upper Set Level

We want to derive a similar criterion for upper sets. Let $\Psi : \mathsf{U}(\mathcal{R}) \nrightarrow \mathsf{U}(\mathcal{F})$ be a feasibility relation. Observe that $\mathsf{U}(\mathcal{R}) = \mathsf{L}(\mathcal{R}^{\mathrm{op}})^{\mathrm{op}}$. Hence we can view Ψ as being a feasibility relation $\mathsf{L}(\mathcal{R}^{\mathrm{op}})^{\mathrm{op}} \nrightarrow \mathsf{L}(\mathcal{F}^{\mathrm{op}})^{\mathrm{op}}$, which corresponds bijectively to a feasibility relation $\check{\Psi} : \mathsf{L}(\mathcal{F}^{\mathrm{op}}) \nrightarrow \mathsf{L}(\mathcal{R}^{\mathrm{op}})$. By the induction criterion (Proposition 4.1) we know that $\check{\Psi}$ is $\mathsf{L}(\tilde{\Psi})$ for $\tilde{\Psi} : \mathcal{F}^{\mathrm{op}} \nrightarrow \mathcal{R}^{\mathrm{op}}$ iff we satisfy the decomposition and union properties:

- If $\check{\Psi}(A, B) = \texttt{true}$, then for every $a \in A$ there is $b \in B$ such that $\check{\Psi}(\downarrow \{a\}, \downarrow \{b\}) = \texttt{true}$.
- For any collection $\{X_i\}_{i \in I}$, where $X_i \in \mathsf{L}(\mathcal{F}^{\mathrm{op}})$,
$$\check{\Psi}(X_i, B) = \texttt{true}\ \forall i \in I \quad \Rightarrow \quad \check{\Psi}(\bigcup_{i \in I} X_i, B) = \texttt{true}.$$

Since $\check{\Psi}(A, B) = \Psi(B, A)$, and lower closures become upper closures under taking opposites, this means that $\check{\Psi}$ is induced iff

- If $\Psi(B, A) = \texttt{true}$, then for every $a \in A$ there is $b \in B$ such that $\Psi(\uparrow \{b\}, \uparrow \{a\}) = \texttt{true}$.
- For any collection $\{X_i\}_{i \in I}$, where $X_i \in \mathsf{U}(\mathcal{F})$,
$$\Psi(B, X_i) = \texttt{true}\ \forall i \in I \quad \Rightarrow \quad \Psi(B, \bigcup_{i \in I} X_i) = \texttt{true}.$$

Because all the correspondences we used to get from $\Psi : \mathsf{U}(\mathcal{R}) \nrightarrow \mathsf{U}(\mathcal{F})$ to $\check{\Psi} : \mathsf{L}(\mathcal{F}^{\mathrm{op}}) \nrightarrow \mathsf{L}(\mathcal{R}^{\mathrm{op}})$ are invertible, this means that the original Ψ is induced by $\tilde{\Psi} : \mathcal{F}^{\mathrm{op}} \nrightarrow \mathcal{R}^{\mathrm{op}}$ iff the above conditions are fulfilled. We conclude by observing that $\tilde{\Psi} : \mathcal{F}^{\mathrm{op}} \nrightarrow \mathcal{R}^{\mathrm{op}}$ corresponds bijectively to a feasibility relation $\mathcal{R} \nrightarrow \mathcal{F}$. Therefore,

Proposition 4.2 (Induction Criterion for Upper Sets)**.** *Let $\Psi : \mathsf{U}(\mathcal{R}) \nrightarrow \mathsf{U}(\mathcal{F})$ be a feasibility relation. Then $\Phi = \mathsf{L}(\tilde{\Phi})$ for some $\tilde{\Phi} : \mathcal{R} \nrightarrow \mathcal{F}$ if and only if Φ satisfies the following:*

- *If $\Psi(A, B) = \texttt{true}$, then for every $b \in B$ there is $a \in A$ such that $\Psi(\uparrow \{a\}, \uparrow \{b\}) = \texttt{true}$.*
- *For any collection $\{X_i\}_{i \in I} \subseteq \mathsf{U}(\mathcal{F})$,*
$$\Psi(A, X_i) = \texttt{true}\ \forall i \in I \quad \Rightarrow \quad \Psi(A, \bigcup_{i \in I} X_i) = \texttt{true}.$$

4.2.3 Upper-Lower Induction

We prove the following characterization directly.

Theorem 4.1. *A feasibility relation* $\Psi : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ *has the form* $\mathsf{U}(\mathsf{L}(\tilde{\Psi}))$ *for* $\tilde{\Psi} : \mathcal{R} \nrightarrow \mathcal{F}$ *if and only if the following three conditions are fulfilled:*

(i) If $\Psi(A, B) = \texttt{true}$*, then*

$$\forall b \in B \; \exists a \in A : \forall r \in a \; \exists y \in b : \Psi(\uparrow\downarrow\{r\}, \uparrow\downarrow\{f\}) = \texttt{true}.$$

(ii) For any collection $\{X_i\}_{i \in I} \subseteq \mathsf{U}(\mathcal{F})$*,*

$$\Psi(A, X_i) = \texttt{true} \;\; \forall i \in I \quad \Rightarrow \quad \Psi(A, \bigcup_{i \in I} X_i) = \texttt{true}.$$

(iii) For any collection $\{X_i\}_{i \in I} \subseteq \mathsf{U}(\mathcal{R})$*,*

$$\Psi(X_i, B) = \texttt{true} \;\; \forall i \in I \quad \Rightarrow \quad \Psi(\bigcap_{i \in I} X_i, B) = \texttt{true}.$$

In that case, $\tilde{\Psi}$ *is obtained by setting* $\tilde{\Psi}(r, f) = \Psi(\uparrow\downarrow\{r\}, \uparrow\downarrow\{f\})$*.*

Proof. First, we show that given a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$, the induced feasibility relation $\mathsf{U}(\mathsf{L}(\Phi)) : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ satisfies (i)-(iii).

(i)

(ii)

(iii) Next, assume we have a feasibility relation $\Psi : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ satisfying (i)-(iii). We show that $\Psi = \mathsf{U}(\mathsf{L}(\tilde{\Psi}))$, where $\tilde{\Psi}(r, f) = \Psi(\uparrow\downarrow\{r\}, \uparrow\downarrow\{f\})$.

Suppose that $\Psi(A, B) = \texttt{true}$. Then by (i) we know that

$$\forall b \in B \; \exists a \in A : \forall r \in a \; \exists y \in b : \Psi(\uparrow\downarrow\{r\}, \uparrow\downarrow\{f\}) = \texttt{true}.$$

But this implies, by definition, that $\mathsf{U}(\mathsf{L}(\tilde{\Psi}))(A, B) = \texttt{true}$.

Conversely, suppose $\mathsf{U}(\mathsf{L}(\tilde{\Psi}))(A, B) = \texttt{true}$. Then we know that

$$\forall b \in B \; \exists a \in A : \forall r \in a \; \exists y \in b : \Psi(\uparrow\downarrow\{r\}, \uparrow\downarrow\{f\}) = \texttt{true}.$$

Proposition 4.3. *Suppose* $\Psi : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ *is induced as in Theorem 4.1. Then the following equivalences hold for principal upper sets*

(i) $\Psi(\bigcup_{i\in I} X_i, \uparrow\{z\}) \Leftrightarrow \Psi(X_i, \uparrow\{z\})$ *for some* $i \in I$.

(ii) $\Psi(\uparrow\{\downarrow\{s\}\}, \bigcap_{i\in I} X_i) \Leftrightarrow \Psi(\uparrow\{\downarrow\{s\}\}, X_i)$ *for some* $i \in I$.

Proof. Note that for both (i) and (ii) the right-hand sides imply the left by monotonicity.

For (i) suppose $\Psi(\bigcup_{i\in I} X_i, \uparrow\{z\})$ holds. Then we know that

$$\forall b \in \uparrow\{z\}\ \exists a \in \bigcup_{i\in I} X_i : \forall r \in a\ \exists f \in b : \tilde{\Psi}(r, f).$$

In particular, for z we have an $a \in \bigcup_{i\in I} X_i : \forall r \in a\ \exists f \in z : \tilde{\Psi}(r, f)$. This a will lie in some X_i, so we have an $a \in X_i : \forall r \in a\, \exists f \in z : \tilde{\Psi}(r, f)$. It remains to show that this choice of a also works for any $b \in \uparrow\{z\}$. Let $b \in \uparrow\{z\}$. Then $z \subseteq b$. Hence, for the $a \in X_i$ we have that for all $r \in a$ there is $f \in z \subseteq b$ such that $\tilde{\Psi}(r, f)$. This shows that for some X_i,

$$\forall b \in \uparrow\{z\}\ \exists a \in X_i : \forall r \in a\ \exists f \in b : \tilde{\Psi}(r, f),$$

meaning that $\Psi(X_i, \uparrow\{z\})$ for some $i \in I$.

For (ii) suppose $\Psi(\uparrow\{\downarrow\{s\}\}, \bigcap_{i\in I} X_i) = \texttt{true}$. Recall from Lemma 3.5 that $\bigcap_{i\in I} X_i = \{\cup x_i : x_i \in X_i\}$. Hence,

$$\forall b \in \{\cup x_i : x_i \in X_i\}\ \exists a \in \uparrow\{\downarrow\{s\}\} : \forall r \in a\ \exists f \in b : \tilde{\Psi}(r, f).$$

Fix some $\cup x_i$. Then there is $a \supseteq \downarrow\{s\}$ such that for all $r \in a$ there is $f \in \cup x_i$ satisfying $\tilde{\Psi}(r, f)$. In particular, the same holds if we restrict a to $\downarrow\{s\}$. Hence,

$$\forall b \in \{\cup x_i : x_i \in X_i\}\ \forall r \in \downarrow\{s\}\ \exists f \in b : \tilde{\Psi}(r, f).$$

Since

Suppose that $\Psi(\uparrow\{\downarrow\{s\}\}, X_i) = \texttt{false}$ for all i. That is, for all $i \in I$,

$$\exists b_i \in X_i\ \forall a \in \uparrow\{\downarrow\{s\}\} : \exists r \in a\ \forall f \in b_i : \tilde{\Psi}(r, f) = \texttt{false}.$$

In particular for any such b_i we have that

$$\exists r \in \downarrow\{s\}\ \forall f \in b_i : \tilde{\Psi}(r, f) = \texttt{false}.$$

By monotonicity, this means in particular that $\forall f \in b_i : \tilde{\Psi}(s,f) = \texttt{false}$, from which it follows that $\tilde{\Psi}(s,f) = \texttt{false}$ for all $f \in \cup b_i$. Therefore $\cup b_i \in \{\cup x_i : x_i \in X_i\}$ is an element of the intersection for which there exists $r \in \downarrow \{s\}$, (namely s) such that for all $f \in \cup b_i$ we have $\tilde{\Psi}(s,f) = \texttt{false}$. This contradicts our assumption that $\Psi(\uparrow \{\downarrow \{s\}\}, \bigcap_{i \in I} X_i) = \texttt{true}$. $\lightning$

4.3 Formulas for Queries on Induced Feasibility

Let $\Psi : \mathsf{U}(\mathsf{L}(\mathcal{R})) \nrightarrow \mathsf{U}(\mathsf{L}(\mathcal{F}))$ be an induced feasibility relation, that is $\Psi = \mathsf{U}(\mathsf{L}(\tilde{\Psi}))$ for $\tilde{\Psi}(r,f) = \Psi(\uparrow\downarrow \{r\}, \uparrow\downarrow \{f\})$. We define the following sets called queries.

Definition 4.1 (Querying Resources)**.** For a fixed $A \in \mathsf{U}(\mathsf{L}(\mathcal{R}))$ define

$$r(A) := \{f \in \mathcal{F} : \Psi(A, i(f))\}.$$

The result is an upper set in $\mathcal{F}$.

Definition 4.2 (Querying Functionalities)**.** For a fixed $B \in \mathsf{U}(\mathsf{L}(\mathcal{F}))$ define

$$f(B) := \{r \in \mathcal{R} : \Psi(i(r), B)\}.$$

The result is a lower set in $\mathcal{R}$.

Using the characterization in Theorem 4.1, we can immediately obtain formulas for queries between choices. Recall from Section 3.2 that for $A \subseteq \mathsf{U}(\mathsf{L}(\mathcal{R}))$ we define

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

Proposition 4.4. *For*

(i) $r(\bigsqcap_{a \in A} a) = r(\bigcup_{a \in A} a) = \bigcup_{a \in A} r(a)$

(ii) $r(\bigsqcup_{a \in A} a) = r(\bigcap_{a \in A} a) = \bigcap_{a \in A} r(a)$

(iii) $f(\bigsqcap_{b \in B} b) = f(\bigcup_{b \in B} b) = \bigcap_{b \in B} f(b)$

(iv) $f(\bigsqcup_{b \in B} b) = f(\bigcap_{b \in B} b) = \bigcup_{b \in B} f(b)$

Proof. We prove each in turn.

(i) Follows from 4.3 (i)

(ii) By definition $B \in r(\bigcap_{a \in A} a) \Leftrightarrow \Psi(\bigcap_{a \in A} a, B)$. By monotonicity and Theorem 4.1 (iii) we know $\Psi(\bigcap_{a \in A} a, B) \Leftrightarrow \Psi(a, B), \forall a \in A$. In summary, $B \in r(\bigcap_{a \in A} a) \Leftrightarrow B \in r(a), \forall a \in A \Leftrightarrow B \in \bigcap_{a \in A} r(A)$.

(iii) Obeserve that by monotonicity and Theorem 4.1 (ii) we have $\Psi(A, \bigcup_{b \in B}) \Leftrightarrow \Psi(A, b), \forall b \in B$. Hence $A \in f(\bigcup_{b \in B}) \Leftrightarrow \Psi(A, \bigcup_{b \in B}) \Leftrightarrow \Psi(A, b), \forall b \in B, \Leftrightarrow A \in f(b), \forall b \in B \Leftrightarrow A \in \bigcap_{b \in B} f(b)$.

(iv) Follows from 4.3 (ii) ⌝⌞

4.4 General Interpretation for Feasibility between Upper-Lower Models

4.4.1 Induced Feasibility

5 Choice between Transformations

6 Towards Modeling Dependencies

A Laws for Choice

Derive some general laws for distributivity ect... Especially those used is section on interpretation based reconstruction

Lemma A.1.

$$\bigsqcap_{a \in A} a \sqcap \bigsqcap_{b \in B} b \cong \bigsqcap_{x \in A \cup B} x$$

and its dual.

Lemma A.2. *The folowing holds for sets* $\{r_i\}_{i \in I}$ *and* $\{r_l\}_{l \in L}$*:*

$$\bigsqcap_{i \in I} r_i \sqcup \bigsqcap_{l \in L} r_l = \bigsqcap_{i \in I} \bigsqcap_{l \in L} (r_i \sqcup r_l)$$

Proof. Write

$$\bigsqcap_{i \in I} r_i \sqcup \bigsqcap_{l \in L} r_l = \bigsqcup_{X \in \{I,L\}} \bigsqcap_{j \in X} r_{X,j}.$$

Let F be the set of choice functions f from $\{I, L\}$ to $I \cup L$, such that $f(I) \in I$ and $f(L) \in L$. Observe that $F \cong I \times L$. Hence,

$$\begin{aligned}
\bigsqcup_{X \in \{I,L\}} \bigsqcap_{j \in X} r_{X,j} &= \bigsqcap_{f \in F} \bigsqcup_{X \in \{I,L\}} r_{X,f(X)} \\
&= \bigsqcap_{f \in F} r_{I,f(I)} \sqcup r_{L,f(L)} \\
&= \bigsqcap_{(i,l) \in I \times L} r_{I,i} \sqcup r_{L,l} \\
&= \bigsqcap_{i \in I} \bigsqcap_{l \in L} (r_i \sqcup r_l).
\end{aligned}$$

and its dual.

References

[Cen15] Andrea Censi. “A Mathematical Theory of Co-Design”. In: (2015). arXiv: 1512.08055. URL: http://arxiv.org/abs/1512.08055.

[MCR07] C. E. Martin, S. A. Curtis, and I. Rewitzky. “Modelling angelic and demonic nondeterminism with multirelations”. In: *Science of Computer Programming* 65.2 (2007), pp. 140–158. ISSN: 01676423. DOI: 10.1016/j.scico.2006.01.007.

[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. URL: https://link.springer.com/chapter/10.1007/978-3-540-27764-4_15.