Modeling Choice in Co-Design

Marius Furter

July 20, 2021

Abstract

This report describes a method for modeling free and forced choice within Co-Design. In a free choice among a set, one has control over which option is selected, while in a forced choice one does not. Given a preorder $\mathcal{P}$ describing resources or functionalities, a free choice among a subset of $\mathcal{P}$ acts like a meet. Dually, a forced choice acts like a join. Moreover, the two types of choice distribute over one another. Based on this, we construct a universal model for choice on a preorder using the free completely distributive lattice $\mathsf{UL}\mathcal{P}$. Feasibility relations are then extended to these models. Along the way, we illustrate how to work within $\mathsf{UL}\mathcal{P}$ and provide results that simplify calculations. The definitions presented here have been implemented in Haskell.

Acknowledgements and Code Resources

I'd like to sincerely thank Jonathan Lorand who supervised this project and dedicated many hours to listening to my ideas. I'm also grateful to Gioele Zardini who gave valuable inputs throughout. Finally, I want to thank Prof. Alberto Cattaneo for implementing the theory in Haskell. The code can be found on GitHub: MariusFurter/Choice-in-CoDesign or on the UZH gitlab. This report can be cited using its DOI: 10.3929/ethz-b-000532280 and is licensed under CC-BY.

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 can freely choose the 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 programmer's best interest, while forced choice describes 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.

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 motors as resources, and 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 outputting 2 kW then we would mark $(A, 2\text{ kW})$ as feasible.

Certain resources or functionalities may imply others. If motor A can provide 2 kW, it should also be able to provide 1 kW. We will indicate this by writing $2\text{ kW} \to 1\text{ kW}$. Hence, our sets of resources and functionalities possess 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 place 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 2.2. (Preorder) A *preorder* $\mathcal{P}$ is a set with a binary relation $R \subseteq \mathcal{P} \times \mathcal{P}$ satisfying reflexivity (r) and transitivity (t):

(r) $(a, a) \in R$ for all $a \in \mathcal{P}$,

(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. When defining preorders, we will usually refrain from mentioning the pairs stemming from reflexivity.

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 a. 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 2.3. 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. Moreover, 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 imply 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 if we have an a at our disposal, 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 $\Phi\colon \mathcal{R} \nrightarrow \mathcal{F}$ between them expresses which functionalities we can obtain from which resources. We write $\Phi(r, f) = \texttt{true}$ to indicate that f can be obtained from r. In the language of the previous section, being able to provide r implies being able to provide f.

Example 2.4 (Gorcery Shopping 1). 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 2.5 (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 $(r, g) \in \Phi$.

Whenever $(r, f) \in \Phi$, 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 as the transformations that Φ enables. Observe that the closure conditions in Definition 2.5 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.

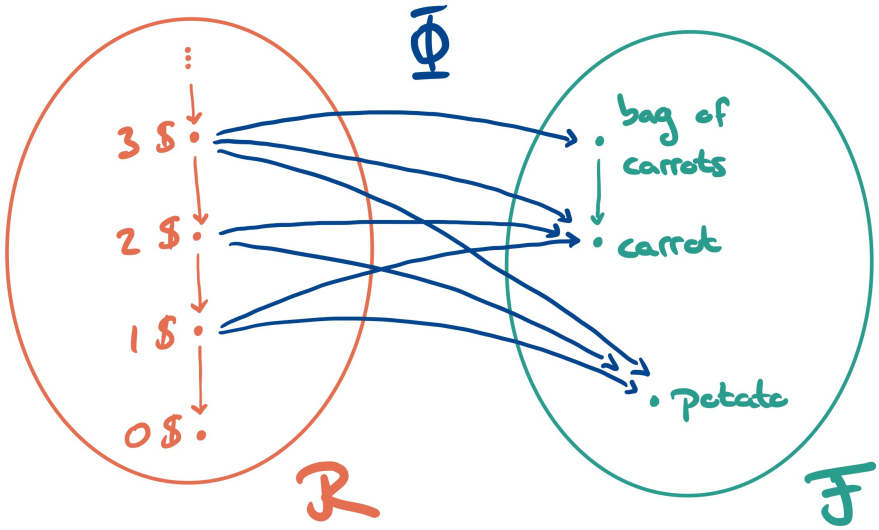

Figure 1: Internal diagram of a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$.

Example 2.4 (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 how Φ satisfies the monotonicity conditions presented in Definition 2.5. ◇

For any preorders $\mathcal{R}$ and $\mathcal{F}$, setting $\Psi := \mathcal{R} \times \mathcal{F}$ yields a feasibility relation. In particular, every relation $R \subseteq \mathcal{R} \times \mathcal{F}$ is contained within a feasibility relation. This allows us to compactly write a feasibility relation in terms of a generating relation.

Definition 2.6 (Feasibility Generated by a Relation)**.** Given a relation $R \subseteq \mathcal{R} \times \mathcal{F}$, the *feasibility relation generated by* R is the smallest feasibility relation containing R. It is given by

$$\bigcap_{\Phi \supseteq R} \Phi,$$

where Φ runs over all feasibility relations $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$.

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

Definition 2.7. 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 2.8 (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 2.9 (Product of Preorders)**.** Let $\mathcal{P}$ and $\mathcal{Q}$ be preorders. Their *product* $\mathcal{P} \times \mathcal{Q}$ consists of all pairs (p, q) for $p \in \mathcal{P}$ 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 2.10 (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 2.11. *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 2.5, we obtain a function $F : \mathcal{R}^{\mathrm{op}} \times \mathcal{F} \to \mathsf{Bool}$ by sending $(r, f) \mapsto \texttt{true}$ if and only if $(r, f) \in \Phi$. Conditions (i) and (ii) of Definition 2.5 say that this function is monotone. Conversely, given a monotone map $F : \mathcal{R}^{\mathrm{op}} \times \mathcal{F} \to \mathsf{Bool}$, 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 2.5 which can be concisely expressed by saying the Φ is a monotone map $\mathcal{R}^{\mathrm{op}} \times \mathcal{F} \to \mathsf{Bool}$.

3 Choice between Resources

We 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). A restaurant might offer the following menu options:

- Coffee or tea (customer choice)
- Asparagus or pumpkin 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 refer to the two modes of choice presented in Example 3.1 as *free choice* and *forced choice*, respectively. Alternatively, we could call them *choice* and *uncertainty*, or *internal* and *external* 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

Suppose we are given some preorder $\mathcal{P}$ of resources or functionalities. Our goal will be to augment $\mathcal{P}$ to include all possible free and forced choices among subsets $A \subseteq \mathcal{P}$. Let us call this choice-augmented preorder $\mathsf{C}(\mathcal{P})$. To this end, we introduce two connectives to express free and forced choice.

Definition 3.2 (Choice Connectives)**.** Given a set A, we denote

- the *free choice* among $a \in A$ by $\bigsqcap_{a \in A} a$,
- the *forced choice* among $a \in A$ by $\bigsqcup_{a \in A} a$.

When $A = \{a_1, \ldots, a_n\}$ is finite, we will also write

$$a_1 \sqcap \ldots \sqcap a_n := \bigsqcap_{a \in A} a, \qquad a_1 \sqcup \ldots \sqcup a_n := \bigsqcup_{a \in A} a.$$

We would like $\mathsf{C}(\mathcal{P})$ to contain all elements of $\mathcal{P}$ along with a single element for each possible choice. Moreover, we would like the preorder structure on $\mathsf{C}(\mathcal{P})$ to fit our established interpretation: An arrow $a \to b$ in $\mathsf{C}(\mathcal{P})$ should signify that being able to provide a implies being able to provide b. This poses the following questions:

(i) When are two choices as represented by the connectives distinct?

(ii) When must we add an arrow $a \to b$ if a or b represent choices?

We will answer these questions in what follows, beginning with (ii).

3.1.1 Arrows between Choices

In this section, we describe when we should add arrows in the choice-augmented preorder $\mathsf{C}(\mathcal{P})$. We start by considering free choice.

Suppose that we can provide a free choice among some set $A \subseteq \mathsf{C}(\mathcal{P})$. This means that regardless of which $x \in A$ is picked freely, we must be able to provide that x. In other words, being able to provide a free choice among A implies being able to provide any $x \in A$. Formally,

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

Let $A \subseteq \mathsf{C}(\mathcal{P})$ and suppose that there is a fixed $t \in \mathsf{C}(\mathcal{P})$ such that for every $a \in A$, being able to provide t implies being able to provide a. Being able to provide a free choice among A means that whatever $a \in A$ is picked freely, we can provide that a. By assumption, if we can provide t, we can provide any $a \in A$, and hence also the free choice among A. Formally,

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

We now turn to forced choice. Being able to provide a forced choice among $B \subseteq \mathsf{C}(\mathcal{P})$ means being able to provide some $y \in B$, without having to specify which. Therefore, being able to provide some $y \in B$ implies being able to provide a forced choice among B. Formally,

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

Finally, suppose that for every $b \in B$, being able to provide b implies being able to provide a fixed $t \in \mathsf{C}(\mathcal{P})$. Assume we can provide a forced choice among B. This means that we can provide some $b \in B$. However, regardless of which b we can provide, our hypothesis says that this implies we can provide t. Therefore, being able to provide a forced choice among B implies being able to provide t. Formally,

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

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

Definition 3.3 (Meets and Joins)**.** Let $(\mathcal{P}, \to)$ be a preorder and $A \subseteq \mathcal{P}$. A *meet* of the set A is an element $\bigwedge_{a \in A} a$ in $\mathcal{P}$ such that

(i) $\bigwedge_{a \in A} a$ is a lower bound of A, that is $\bigwedge_{a \in A} a \to a$ for all $a \in A$,

(ii) $\bigwedge_{a \in A} a$ is the greatest lower bound of A, that is if $t \to a$ for all $a \in A$, then $t \to \bigwedge_{a \in A} a$.

A *join* of the set A is an element $\bigvee_{a \in A} a$ in $\mathcal{P}$ such that

(i) $\bigvee_{a \in A} a$ is an upper bound of A, that is $a \to \bigvee_{a \in A} a$ for all $a \in A$,

(ii) $\bigvee_{a \in A} a$ is the least upper bound of A, that is if $a \to t$ for all $a \in A$, then $\bigvee_{a \in A} a \to t$.

A preorder in which every subset has a meet and a join is called a *complete lattice*.

Our discussion shows that our choice-augmented preorder $\mathsf{C}(\mathcal{P})$ will have meets and joins for any subset $A \subseteq \mathsf{C}(\mathcal{P})$. Therefore, $\mathsf{C}(\mathcal{P})$ will be a complete lattice.

3.1.2 Equality of Choices

The fact that free and forced choices are meets and joins in $\mathsf{C}(\mathcal{P})$ already implies many equalities between expressions consisting of the connectives $\sqcap$

and $\sqcup$. In this section, we will show that in addition to these, we also want the connectives to distribute over one another.

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

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

More generally, these laws 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 $\mathsf{C}(\mathcal{P})$ we have

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

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

This law is justified by the following argument. 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 an uncertainty 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

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

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

In summary, our discussion shows that in addition to being a complete lattice, $\mathsf{C}(\mathcal{P})$ should also satisfy the infinite distributive laws (5) and (6). A complete lattice satisfying infinite distributivity is called a *completely distributive lattice*.

Choice Connectives

We denote the *free choice* among $a \in A$ by $\bigsqcap_{a \in A} a$, and the *forced choice* among $a \in A$ by $\bigsqcup_{a \in A} a$. The connective $\sqcap$ acts like a meet, while $\sqcup$ acts like a join. Moreover, both connectives distribute over one another, even in the infinite case.

3.2 A Universal Model for Choice

Given a preorder $\mathcal{P}$, we want to construct a choice-augmented preorder $\mathsf{C}(\mathcal{P})$ that includes all possible choices between elements of $\mathcal{P}$. Our discussion in the previous section showed that $\mathsf{C}(\mathcal{P})$ should be a completely distributive lattice, where a free choice among a set A is given by the meet over A, and a forced choice over a set B is given by the join over B. Moreover, we want the new elements in $\mathsf{C}(\mathcal{P})$ to only fulfill those relations implied by our interpretation of these elements as choices. This leads us to consider the free completely distributive lattice generated by $\mathcal{P}$.

There are several notions of free completely 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 $\mathcal{P}$, denoted $\mathsf{UL}\mathcal{P}$, satisfy a universal property that suits our purposes. We will explain this in more detail after making the requisite definitions.

We start by defining special subsets of a preorder which are upwards and downward closed.

Definition 3.4 (Upper and Lower Sets)**.** Let $\mathcal{P}$ be a preorder. A subset $U \subseteq \mathcal{P}$ is called an *upper set*, if it is upward closed:

$$x \in U \text{ and } x \to y \text{ imply } y \in U.$$

A subset $L \subseteq \mathcal{P}$ is called a *lower set*, if it is downward closed:

$$x \in L \text{ and } y \to x \text{ imply } y \in L.$$

Definition 3.5 (Upper and Lower Closure)**.** Given a subset $A \subseteq \mathcal{P}$ of a preorder, we can form its

- *upper closure* $\uparrow A := \{p \in \mathcal{P} : a \to p \text{ for some } a \in A\}$,
- *lower closure* $\downarrow A := \{p \in \mathcal{P} : p \to a \text{ for some } a \in A\}$.

The upper closure is an upper set, while the lower closure is a lower set. Moreover, we can consider

- *upper sets* $\mathsf{U}\mathcal{P} := \{U \subseteq \mathcal{P} : U \text{ an upper set}\}$,
- *lower sets* $\mathsf{L}\mathcal{P} := \{L \subseteq \mathcal{P} : L \text{ an lower set}\}$.

We will be interested in monotone maps $f : \mathcal{P} \to \mathcal{Q}$ in which the implication "$p \to q$ implies $f(p) \to f(q)$" is actually an equivalence.

Definition 3.6 (Order Embedding)**.** Let $\mathcal{P}$ and $\mathcal{Q}$ be preorders. A function $f : \mathcal{P} \to \mathcal{Q}$ is called an *order embedding* provided

$$p \to q \Leftrightarrow f(p) \to f(q).$$

In this case we write $f : \mathcal{P} \hookrightarrow \mathcal{Q}$.

Lemma 3.7. *Given a preorder $\mathcal{P}$, its upper sets can be preordered by containment $(\mathsf{U}\mathcal{P}, \supseteq)$ and its lower sets can be ordered by inclusion $(\mathsf{L}\mathcal{P}, \subseteq)$. Both of these structures are completely distributive lattices. Moreover, we have order embeddings:*

$$\begin{array}{cc} i_{\mathsf{U}} : \mathcal{P} \hookrightarrow (\mathsf{U}\mathcal{P}, \supseteq) & i_{\mathsf{L}} : \mathcal{P} \hookrightarrow (\mathsf{L}\mathcal{P}, \subseteq) \\ p \mapsto \uparrow \{p\} & p \mapsto \downarrow \{p\} \end{array}$$

Proof. We will take for granted that the power set $(\mathsf{P}(\mathcal{P}), \subseteq)$, ordered by inclusion, is a completely distributive lattice with set intersection as meet and set union as join. This follows from a straightforward check of the definition for meets and joins (Def. 3.3) and verification of the infinite distributive laws (5) and (6). We now observe that unions and intersections of upper sets are again upper sets. The same holds for lower sets. Therefore, the completely distributive structure on $(\mathsf{P}(\mathcal{P}), \subseteq)$ restricts to completely distributive lattices $(\mathsf{U}\mathcal{P}, \subseteq)$ and $(\mathsf{L}\mathcal{P}, \subseteq)$. Finally, since $(\mathsf{U}\mathcal{P}, \subseteq)^{\mathrm{op}} = (\mathsf{U}\mathcal{P}, \supseteq)$, and the defining laws for completely distributive lattices are self-dual, we see that also $(\mathsf{U}\mathcal{P}, \supseteq)$ is completely distributive.

To prove that i_{U} is an order embedding, let $p \to q$ in $\mathcal{P}$. Suppose $x \in \uparrow\{q\}$. Then $q \to x$, whence $p \to x$ by transitivity. Hence, $x \in \uparrow\{p\}$, so $\uparrow\{p\} \supseteq \uparrow\{q\}$. Conversely, if $\uparrow\{p\} \supseteq \uparrow\{q\}$, then in particular $q \in \uparrow\{p\}$, since $q \in \uparrow\{q\}$. Hence $p \to q$, as desired. The proof that i_{L} is an order embedding is analogous. ⌐╙

We now have the tools to understand the free completely distributive lattice constructed in [Mor04]. Lemma 3.7 shows that given a preorder $\mathcal{P}$, we can obtain a completely distributive lattice $(\mathsf{L}\mathcal{P}, \subseteq)$. Since this is again a preorder, we can reapply the lemma and consider the completely distributive lattice $(\mathsf{UL}\mathcal{P}, \supseteq)$. In particular, [Mor04] shows that $\mathsf{UL}\mathcal{P}$ satisfies the following universal property:

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

If we augment a preorder $\mathcal{P}$ to include choice, we should have a monotone map $c : \mathcal{P} \to \mathsf{C}(\mathcal{P})$ which maps each element of $\mathcal{P}$ to the free (or forced) choice among the singleton $\{p\}$. Therefore, by the universal property above, there will be a unique map from $u : \mathsf{UL}\mathcal{P} \to \mathsf{C}(\mathcal{P})$, preserving the elements of $\mathcal{P}$ and the choices among them.

To see what this means, consider a free choice over $A \subseteq \mathcal{P}$. This will be represented by the meet $\bigwedge_{a \in A} c(a)$ in $\mathsf{C}(\mathcal{P})$, and by $\bigwedge_{a \in A} i(a)$ in $\mathsf{UL}\mathcal{P}$. Since the unique map u preserves meets, and $u \circ i = c$, we have

$$u\left(\bigwedge_{a \in A} i(a)\right) = \bigwedge_{a \in A} u(i(a)) = \bigwedge_{a \in A} c(a).$$

Hence each free choice represented in $\mathsf{C}(\mathcal{P})$ is the image of the corresponding choice represented in $\mathsf{UL}\mathcal{P}$. The same holds for forced choices. This shows that the part of $\mathsf{C}(\mathcal{P})$ we are interested lies in is the image of $\mathsf{UL}\mathcal{P}$ under the unique map u. In this sense, $\mathsf{UL}\mathcal{P}$ serves as a universal model for choice on $\mathcal{P}$, which we will call the *upper-lower model*.

Universal Model for Choice

Given a preorder $\mathcal{P}$ we can form the *upper-lower model* $\mathsf{UL}\mathcal{P}$, given by the upper sets (ordered by containment) of the lower sets (ordered by inclusion) of $\mathcal{P}$. For any other model $\mathsf{C}(\mathcal{P})$ of choice on $\mathcal{P}$, the free and forced choices in $\mathsf{C}(\mathcal{P})$ lie in the image of $\mathsf{UL}\mathcal{P}$ under a unique map.

3.3 The Upper-Lower Model

Apart from being universal, the upper-lower model lends itself to easy interpretation. This section will describe how. In addition, we present rules which simplify the evaluation of choices.

Given a preorer $\mathcal{P}$, we consider the upper-lower model $\mathsf{UL}\mathcal{P}$. By Lemma 3.7 we have an order embedding $i : \mathcal{P} \hookrightarrow \mathsf{UL}\mathcal{P}$ given by $p \mapsto \uparrow\downarrow\{p\}$. In addition, we know that meets and joins in $\mathsf{UL}\mathcal{P}$ are given by unions and intersections, respectively. Hence, we implement our connectives $\sqcap$ and $\sqcup$ by setting

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

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

If we are interested in choice between elements of $\mathcal{P}$, we first embed those elements into $\mathsf{UL}\mathcal{P}$ and then apply the corresponding operation. That is, if $B \subseteq \mathcal{P}$,

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

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

Remark 3.8. Observe that we are overloading the symbols $\sqcap$ and $\sqcup$ to express both choice between elements of $\mathsf{UL}\mathcal{P}$ and $\mathcal{P}$. The two uses only differ by the embedding $i : \mathcal{P} \hookrightarrow \mathsf{UL}\mathcal{P}$. It should be clear from context which meaning applies.

If $\mathcal{P}$ is finite, an element of $\mathsf{UL}\mathcal{P}$ has the form

$$\{\{p_{1,1}, \ldots, p_{1,n_1}\}, \ldots, \{p_{m,1}, \ldots, p_{m,n_m}\}\}$$

where the green brackets indicate an upper set, while the red brackets indicate lower sets. In the general case, an element of $\mathsf{UL}\mathcal{P}$ has the form

$$\{\{p_{k,j}\}_{k \in K_j}\}_{j \in J} \tag{7}$$

where again green indicates an upper set and red indicates lower sets.

We will interpret the element (7) as the choice

$$\bigsqcap_{j \in J} \left(\bigsqcup_{k \in K_j} p_{j,k} \right).$$

To justify this interpretation, we will need the following

Lemma 3.9. *Let $(\mathcal{P}, \to)$ be a preorder. Then for a collection $\{p_i\}_{i \in I} \subseteq \mathcal{P}$,*

$$\uparrow \{p_i\}_{i \in I} = \bigcup_{i \in I} \uparrow \{p_i\},$$

$$\downarrow \{p_i\}_{i \in I} = \bigcup_{i \in I} \downarrow \{p_i\}.$$

Proof. Suppose $x \in \uparrow \{p_i\}_{i \in I}$. Then $p_i \to x$ for some $i \in I$, whence $x \in \uparrow \{p_i\} \subseteq \bigcup_{i \in I} \uparrow \{p_i\}$. Conversely, if $x \in \bigcup_{i \in I} \uparrow \{p_i\}$, then $x \in \uparrow \{p_i\}$ for some $i \in I$. This means $p_i \to x$, whence $x \in \uparrow \{p_i\}_{i \in I}$. The second statement follows by applying the first to $\mathcal{P}^{\mathrm{op}}$. ⌐¬ ⌐

Lemma 3.10. *Let $\mathcal{P}$ be a preorder.*

(i) If $\{A_i\}_{i\in I} \subseteq \mathsf{UL}\mathcal{P}$ is a collection of elements of the upper-lower model, then

$$\bigcap_{i\in I} A_i = \{\bigcup_{i\in I} a_i : a_i \in A_i \text{ for each } i \in I\}.$$

(ii) If $\{a_j\}_{j\in J} \subseteq \mathsf{L}\mathcal{P}$ is a collection of lower sets, then

$$\bigcap_{j\in J} \uparrow \{a_j\} = \uparrow \{\bigcup_{j\in J} a_j\}.$$

(iii) If $\{p_k\}_{k\in K} \subseteq \mathcal{P}$ is a collection of elements of $\mathcal{P}$, then

$$\bigcap_{k\in K} \uparrow \{\downarrow \{p_k\}\} = \uparrow \{\downarrow \{p_k\}_{k\in K}\}.$$

Proof. To show (i), suppose $x \in \bigcap_{i\in I} A_i$. Then $x \in A_i$ for each $i \in I$. Writing $x = \bigcup_{i\in I} x$ shows that $x \in \{\bigcup_{i\in I} a_i : a_i \in A_i \text{ for each } i \in I\}$. Conversely, consider $x = \bigcup_{i\in I} a_i$ where $a_i \in A_i$ for all $i \in I$. Recall that the A_i are upper sets of $(\mathsf{L}\mathcal{P}, \subseteq)$. Hence, $a_i \in A_i$ and $a_i \subseteq \bigcup_{i\in I} a_i = x$ imply $x \in A_i$. This holds for every $i \in I$, so $x \in \bigcap_{i\in I} A_i$.

For (ii), we note that by (i),

$$\bigcap_{i\in I} \uparrow \{a_j\} = \{\bigcup_{j\in J} b_j : b_j \in \uparrow \{a_j\} \text{ for each } j \in J\}.$$

Since $b_j \in \uparrow \{a_j\}$ means $b_j \supseteq a_j$, we have

$$\bigcap_{i\in I} \uparrow \{a_j\} = \{\bigcup_{j\in J} b_j : b_j \supseteq a_j \text{ for each } j \in J\}.$$

Suppose $x = \bigcup_{j\in J} b_j$, where $b_j \supseteq a_j$ for all $j \in J$. Then $\bigcup_{j\in J} b_j \supseteq \bigcup_{j\in J} a_j$, whence $x \in \uparrow \{\bigcup_{j\in J} a_j\}$. Conversely, if $x \supseteq \bigcup_{j\in J} a_j$, then $x \supseteq a_j$ for each $j \in J$. Therefore, $x = \bigcup_{j\in J} x$ is an element of $\{\bigcup_{j\in J} b_j : b_j \supseteq a_j \text{ for each } j \in J\}$. In summary,

$$\bigcap_{i\in I} \uparrow \{a_j\} = \{\bigcup_{j\in J} b_j : b_j \supseteq a_j \text{ for each } j \in J\} = \uparrow \{\bigcup_{j\in J} a_j\}.$$

For (iii), we apply (ii) and Lemma 3.9 to obtain

$$\bigcap_{k\in K} \uparrow \{\downarrow \{p_k\}\} = \uparrow \{\bigcup_{k\in K} \downarrow \{p_k\}\} = \uparrow \{\downarrow \{p_k\}_{k\in K}\}.$$

Returning to the interpretation, consider the element

$$\{\{p_{j,k}\}_{k\in K_j}\}_{j\in J}.$$

Since the red brackets are a lower set and the green brackets upper sets, we can rewrite

$$\{\{p_{j,k}\}_{k\in K_j}\}_{j\in J} = \uparrow \{\downarrow \{p_{j,k}\}_{k\in K_j}\}_{j\in J}.$$

Using Lemmas 3.9 and 3.10 (ii),

$$\begin{aligned}
\{\{p_{j,k}\}_{k\in K_j}\}_{j\in J} &= \uparrow \{\downarrow \{p_{j,k}\}_{k\in K_j}\}_{j\in J} \\
&= \bigcup_{j\in J} \uparrow \{ \bigcup_{k\in K_j} \downarrow \{p_{j,k}\}\} \\
&= \bigcup_{j\in J} \bigcap_{k\in K_j} \uparrow \{\downarrow \{p_{j,k}\}\} \\
&= \bigsqcap_{j\in J} \left(\bigsqcup_{k\in K_j} p_{j,k} \right).
\end{aligned}$$

This equation confirms our interpretation.

Now that we know how to interpret elements of $\mathsf{UL}\mathcal{P}$, we show how free and forced choices are represented in the upper-lower model. Suppose $\{p_j\}_{j\in J}$ is a collection of elements of $\mathcal{P}$. Each p_j embeds into $\mathsf{UL}\mathcal{P}$ as $\uparrow \{\downarrow \{p_j\}\}$. The free choice among $\{p_j\}_{j\in J}$ is represented by

$$\bigsqcap_{j\in J} p_j = \bigcup_{j\in J} \uparrow \{\downarrow \{p_j\}\} = \uparrow \{\downarrow \{p_j\}\}_{j\in J}, \tag{8}$$

where we applied Lemma 3.9.

On the other hand, the forced choice among these is given by

$$\bigsqcup_{j\in J} p_j = \bigcap_{j\in J} \uparrow \{\downarrow \{p_j\}\} = \uparrow \{\downarrow \{p_j\}_{j\in J}\}, \tag{9}$$

where we have used Lemma 3.10 (iii).

Let us see how this works in practice.

Example 3.11 (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

$$\mathcal{P} := \{\texttt{curry}, \texttt{casserole}, \texttt{chicken}, \texttt{beef}\}.$$

Using rule (9) described above, we represent this choice in $\mathsf{UL}\mathcal{P}$ by

$$\begin{aligned}\texttt{curry} \sqcup \texttt{casserole} &= \uparrow \{\downarrow \{\texttt{curry}, \texttt{casserole}\}\} \\ &= \uparrow \{\{\texttt{curry}, \texttt{casserole}\}\}.\end{aligned}$$

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

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

Example 3.12 (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 in 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

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

Using rule (8), we represent

$$\begin{aligned}\texttt{curry} \sqcap \texttt{beef} &= \uparrow \{\downarrow \{\texttt{curry}\}, \downarrow \{\texttt{beef}\}\} \\ &= \uparrow \{\{\texttt{curry}\}, \{\texttt{beef}\}\}.\end{aligned}$$

Similarly, $\texttt{casserole} \sqcap \texttt{chicken} = \uparrow \{\{\texttt{casserole}\}, \{\texttt{chicken}\}\}$. Hence, using Lemma 3.10 (ii),

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

As seen in the examples, it is easiest to represent upper and lower sets by upper and lower closures. The following calculation rules describe how to write choices among elements represented in this fashion.

Calculation Rules for Upper and Lower Closures

Let $\mathcal{P}$ be a preorder and let $\{A_j\}_{j\in J}$ and $\{A_k\}_{k\in K}$ be indexed families of subsets of $\mathcal{P}$. Then the following rules hold for finite choices in $\mathsf{UL}\mathcal{P}$:

(i) $\uparrow\{\downarrow A_j\}_{j\in J} \sqcap \uparrow\{\downarrow A_k\}_{k\in K} = \uparrow\{\downarrow A_i\}_{i\in J\cup K}$

(ii) $\uparrow\{\downarrow A_j\}_{j\in J} \sqcup \uparrow\{\downarrow A_k\}_{k\in K} = \uparrow\{\downarrow (A_j \cup A_k)\}_{j,k\in J\times K}$

More generally, if $\{A_{j,k}\}_{j\in J, k\in K_j}$ is a double indexed family of subsets of $\mathcal{P}$, then the following rules apply for infinite choice in $\mathsf{UL}\mathcal{P}$:

(iii) $\bigsqcap_{j\in J} \uparrow\{\downarrow A_{j,k}\}_{k\in K_j} = \uparrow\{\downarrow A_i\}_{i\in \bigcup K_j}$

(iv) $\bigsqcup_{j\in J} \uparrow\{\downarrow A_{j,k}\}_{k\in K_j} = \uparrow\{\downarrow \bigcup_{j\in J} A_{j,f(j)}\}_{f\in F}$
where F denotes the set of choice functions $f : J \to \coprod_{j\in J} K_j$ with $f(j) \in K_j$ for all $j \in J$.

Proof. Note that (i) and (ii) are special cases of (iii) and (iv). Regarding (ii), this follows because choice functions $\{j,k\} \to J \cup K$ correspond to elements of the cartesian product $J \times K$.

For (iii) we observe that by Lemma 3.9,

$$\begin{aligned}
\bigsqcap_{j\in J} \uparrow\{\downarrow A_{j,k}\}_{k\in K_j} &= \bigcup_{j\in J} \uparrow\{\downarrow A_{j,k}\}_{k\in K_j} && \text{(Definition)}\\
&= \uparrow \bigcup_{j\in J} \{\downarrow A_{j,k}\}_{k\in K_j} && \text{(Lemma 3.9)}\\
&= \uparrow\{\downarrow A_i\}_{i\in \bigcup K_j}
\end{aligned}$$

For (iv) we apply Lemma 3.9, the infinite distributive law (6), and Lemma 3.10,

$$\begin{aligned}
\bigsqcup_{j\in J} \uparrow \{\downarrow A_{j,k}\}_{k\in K_j} &= \bigcap_{j\in J} \uparrow \{\downarrow A_{j,k}\}_{k\in K_j} && \text{(Definition)}\\
&= \bigcap_{j\in J} \bigcup_{k\in K_j} \uparrow \{\downarrow A_{j,k}\} && \text{(Lemma 3.9)}\\
&= \bigcup_{f\in F} \bigcap_{j\in J} \uparrow \{\downarrow A_{j,f(j)}\} && \text{(Distributivity)}\\
&= \bigcup_{f\in F} \uparrow \{\bigcup_{j\in J} \downarrow A_{j,f(j)}\} && \text{(Lemma 3.10)}\\
&= \uparrow \{\bigcup_{j\in J} \downarrow A_{j,f(j)}\}_{f\in F} && \text{(Lemma 3.9)}\\
&= \uparrow \{\downarrow \bigcup_{j\in J} A_{j,f(j)}\}_{f\in F} && \text{(Lemma 3.9)}
\end{aligned}$$

Working in the Upper-Lower Model

Given a preorder $\mathcal{P}$ we have an order embedding $i : \mathcal{P} \hookrightarrow \mathsf{UL}\mathcal{P}$ into its upper-lower model given by $p \mapsto \uparrow \{\downarrow \{p\}\}$. Free and forced choices are implemented in $\mathsf{UL}\mathcal{P}$ as

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

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

A general element of $\mathsf{UL}\mathcal{P}$ has the form $\{\{p_{k,j}\}_{k\in K_j}\}_{j\in J}$ and is interpreted as $\bigsqcap_{j\in J} \bigsqcup_{k\in K_j} p_{j,k}$. Choices in $\mathsf{UL}\mathcal{P}$ can be calculated using the rules presented in Lemmas 3.9, 3.10, and the box above.

4 Feasibility between Choices

Given preorders of resources $\mathcal{R}$ and functionalities $\mathcal{F}$, we can model free and forced choices by considering their upper-lower models $\mathsf{UL}\mathcal{R}$ and $\mathsf{UL}\mathcal{F}$. In this section, we will lift feasibility relations $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ to the upper-lower models. This yields a new feasibility relation $\mathsf{UL}\Phi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$, which extends Φ to choice. Moreover, we will see that such lifted feasibility relations $\mathsf{UL}\Phi$ enjoy special properties that distinguish them from arbitrary feasibility relations between the upper-lower models. Finally, we will consider what functionalities we can obtain from a given choice of resources, and conversely, what resources are required to enable a given choice of functionalities.

4.1 Lifting Feasibility to Upper-Lower Models

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

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

This is a feasibility relation since 'for all' is preserved by taking subsets and 'there exists' by taking supersets.

Moving up a level, let $\Psi : \mathsf{L}\mathcal{R} \nrightarrow \mathsf{L}\mathcal{F}$ be a feasibility relation. Lift this to $\mathsf{U}\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ 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 ordering the upper sets by containment.

Putting both lifts together yields the desired definition.

Definition 4.1 (Lifting Feasibility)**.** Given a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$, we can *lift* it to a feasibility relation $\mathsf{UL}\Phi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ by setting

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

Remark 4.2. The quantifiers in the above definitions are fixed by the ordering we place on upper sets and lower sets. This ordering is in turn determined by the fact that the upper sets represent free choices and the lower sets forced choices. For example, if $A \subseteq B$ we want an arrow $\bigsqcap_{b \in B} b \to \bigsqcap_{a \in A} a$, representing a weakening of free choice. This shows the upper sets have to be ordered by containment.

Example 4.3 (Grocery Shopping 2). Suppose we are grocery shopping. We model our money and groceries by

$$\mathcal{R} := \{2\$ \to 1\$\} \qquad \mathcal{F} := \{\texttt{carrot}, \texttt{eggplant}\}$$

Let $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ be the feasibility relation describing whether we can buy a given item with a given amount of money. Assume that Φ is generated by the following assignments:

$$\Phi(1\$, \texttt{carrot}) = \texttt{true},$$

$$\Phi(2\$, \texttt{eggplant}) = \texttt{true}.$$

By (8) and (9) we represent the following choices in $\mathsf{UL}\mathcal{F}$:

$$i(\texttt{carrot}) \sqcap i(\texttt{eggplant}) = \{\{\texttt{carrot}\}, \{\texttt{eggplant}\}, \{\texttt{carrot}, \texttt{eggplant}\}\}$$

$$i(\texttt{carrot}) \sqcup i(\texttt{eggplant}) = \{\{\texttt{carrot}, \texttt{eggplant}\}\}.$$

The elements 1\$ and 2\$ are represented in $\mathsf{UL}\mathcal{R}$ by

$$i(1\$) = \uparrow \{\downarrow \{1\$\}\} = \uparrow \{\{1\$, 2\$\}\} = \{A : A \supseteq \{1\$, 2\$\}\} = \{\{1\$, 2\$\}\},$$

$$i(2\$) = \uparrow \{\downarrow \{2\$\}\} = \uparrow \{\{2\$\}\} = \{A : A \supseteq \{2\$\}\} = \{\{1\$, 2\$\}, \{2\$\}\}.$$

We can now ask whether $\mathsf{UL}\Phi(i(1\$), i(\texttt{carrot}) \sqcup i(\texttt{eggplant})) \stackrel{?}{=} \texttt{true}$. By Definition 4.1 this is the case iff for every $b \in \{\{\texttt{carrot}, \texttt{eggplant}\}\}$ there is $a \in i(1\$) = \{\{1\$, 2\$\}\}$ such that for all $r \in a$ there is $f \in b$ with $\Phi(r, f)$ holds. Since the source and target sets are singletons, this condition is simply asking whether for all $r \in \{1\$, 2\$\}$ there is $f \in \{\texttt{carrot}, \texttt{eggplant}\}$ such that $\Phi(r, f)$. This is indeed the case since $\Phi(1\$, \texttt{carrot})$ and $\Phi(2\$, \texttt{carrot})$ both hold. Therefore, indeed $\mathsf{UL}\Phi(i(1\$), i(\texttt{carrot}) \sqcup i(\texttt{eggplant})) = \texttt{true}$. This is saying that for 1\$ the store will be willing to sell you either a `carrot` or an `eggplant`, provided they get to decide which you will receive.

On the other hand, $\mathsf{UL}\Phi(i(1\$), i(\texttt{carrot}) \sqcap i(\texttt{eggplant})) = \texttt{false}$, since for $b = \{\texttt{egglant}\}$, the only $a \in i(1\$) = \{\{1\$, 2\$\}\}$ is $\{1\$, 2\$\}$, and we know $\Phi(1\$, \texttt{eggplant}) = \texttt{false}$. This is saying that for 1\$ the store will not sell you the option of choosing between a `carrot` or an `eggplant`, since they would loose money if you chose the `eggplant`. $\diamond$

Lifting Feasibility to the Upper-Lower Model

Given a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ we can *lift* it to a feasibility relation between the upper-lower models $\mathsf{UL}\Phi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ by setting

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

This assignment extends Φ and captures feasibility between choices.

4.2 Feasibility between Upper-Lower Models

Consider an arbitrary feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$. Since both upper-lower models are again preorders, we can interpret Ψ as we would any other feasibility relation: $\Psi(A, B) = \texttt{true}$ means that being able to provide A implies being able to provide B. What is special in the case of upper-lower models is that their elements may be interpreted as choices among subsets of the underlying preorder.

The method described in Section 4.1 starts with a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ and lifts it to a feasibility relation $\mathsf{UL}\Phi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$. This lifting extends feasibility to choices in "the best way possible", given the feasibility among the underlying preorders. The sense in which lifted feasibility relations are optimal is explored further in Sections 4.3 and 5.3.

Example 4.4 (Double Order). Suppose a restaurant offers `soup` for 5\$ and `salad` for 5\$. We model our money by setting $\mathcal{R} := (\mathbb{N}, \geq)$ as in Example 2.1, and model the menu options $\mathcal{F}$ by the discrete preorder $\{\texttt{soup}, \texttt{salad}\}$. We let Φ be the feasibility relation generated by the assignments

$$\Phi(5\$, \texttt{soup}) = \texttt{true},$$

$$\Phi(5\$, \texttt{salad}) = \texttt{true}.$$

Applying the lifting definition 4.1 shows that

$$\mathsf{UL}\Phi(5\$, \texttt{soup} \sqcap \texttt{salad}) = \texttt{true}.$$

We interpret this as saying that being able to providing 5\$ implies being able to provide a free choice between `soup` or `salad`. The lifted relation is optimal in the following sense: If the restaurant can separately provide either `soup` or `salad` given 5\$, then it can provide the free choice between the options for the same price. ◇

On the other hand, one can directly describe a feasibility relation that already includes the desired relations among choices. This amounts to specifying a feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \twoheadrightarrow \mathsf{UL}\mathcal{F}$. However, that such a Ψ will not in general be the lift of some feasibility relation $\Phi : \mathcal{R} \twoheadrightarrow \mathcal{F}$. The conditions for when this is the case are described in Section 4.3.

Example 4.4 (continued). We will now directly specify a feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \twoheadrightarrow \mathsf{UL}\mathcal{F}$ for $\mathcal{R}$ and $\mathcal{F}$ defined above. We let Ψ be the feasibility relation generated by the assignments

$$\Psi(5\$, \texttt{soup}) = \texttt{true},$$

$$\Psi(5\$, \texttt{salad}) = \texttt{true},$$

$$\Psi(6\$, \texttt{soup} \sqcap \texttt{salad}) = \texttt{true}.$$

The resulting feasibility relation is similar to Φ from earlier, except that the restaurant is now charging a 1\$ fee for the free choice between `soup` and `salad`. Indeed, since neither $\texttt{soup} \to \texttt{soup} \sqcap \texttt{salad}$ nor $\texttt{salad} \to \texttt{soup} \sqcap \texttt{salad}$ in $\mathsf{UL}\mathcal{F}$, we must have

$$\Psi(5\$, \texttt{soup} \sqcap \texttt{salad}) = \texttt{false}.$$

This shows that Ψ is not a lift. At the same time it illustrates that our general feasibility relation Ψ is sub-optimal in the sense that the restaurant is charging extra for giving us a choice, although it need not do so based on the underlying feasible pairs. ◇

Feasibility between Upper-Lower Models

A feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \twoheadrightarrow \mathsf{UL}\mathcal{F}$, whether lifted as in Section 4.1 or defined directly, may be interpreted as any other feasibility relation: $\Psi(A, B) = \texttt{true}$ means that being able to provide A implies being able to provide B. Since we are working within upper-lower models, A and B are interpreted as choices between subsets of $\mathcal{R}$ or $\mathcal{F}$, respectively.

4.3 Characterizing Lifted Feasibility Relations

The lifted feasibility relations described in Section 4.1 enjoy special properties as compared to an arbitrary feasibility relation between upper-lower models. These properties can be used to characterize whether such a feasibility relation is lifted or not.

Theorem 4.5. *A feasibility relation* $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ *has the form* $\mathsf{UL}\tilde{\Psi}$ *for* $\tilde{\Psi} : \mathcal{R} \nrightarrow \mathcal{F}$ *if and only if the following three conditions are fulfilled:*

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

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

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

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

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

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

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

We will attempt to explain what these conditions mean before giving the lengthy proof of this theorem. Firstly, if a feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ is lifted, then its pre-lifted form $\tilde{\Psi}$ is given by simply restricting Ψ to the embeddings of $\mathcal{R}$ and $\mathcal{F}$ into their upper-lower models.

Condition (i) expresses the fact that feasibility between each pair (A, B) in the upper-lower models is obtained via the lifting definition from $\tilde{\Psi}$. Conditions (ii) and (iii) express a certain optimality of a lifted Ψ with respect to forced choice in the source and free choice in the target. Condition (ii) says that if being able to provide A is sufficient to guarantee X_i for all i, then providing A should also be sufficient to guarantee a free choice among the X_i. This seems tautological when thinking about feasibility between choices as stemming from feasibility between elements. However, a general feasibility relation $\mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ need not fulfil this condition and can be seen as expressing a type of sub-optimality that can occur in reality. More on this in Section 5.3. Similar remarks apply to condition (iii), which says that if X_i is sufficient to obtain B for all i, then the forced choice among the X_i should be sufficient to obtain B.

Proof of Theorem 4.5. First, we show that given a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$, the induced feasibility relation $\mathsf{UL}\Phi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ satisfies (i)-(iii).

(i) Suppose $\Psi(r, f) = \mathtt{true}$. Let $x \in \downarrow\{r\}$. Then by monotonicity of Ψ, we have $\Psi(x, f) = \mathtt{true}$. Therefore, for all $x = r \in \downarrow\{r\}$, there exists $y := f \in \downarrow\{f\}$ such that $\Psi(x, y)$. Hence, $\mathsf{L}\Psi(\downarrow\{r\}, \downarrow\{f\}) = \mathtt{true}$ (see Section 4.1).

Next, assume $\mathsf{L}\Psi(\downarrow\{r\}, \downarrow\{f\}) = \mathtt{true}$ and let $y \in \uparrow\{\downarrow\{f\}\}$. Then by monotonicity of $\mathsf{L}\Psi$, we have $\mathsf{L}\Psi(\downarrow\{r\}, y)$. Therefore, for all $y \in \uparrow\{\downarrow\{f\}\}$, there exists $x := \downarrow\{r\} \in \uparrow\{\downarrow\{r\}\}$ such that $\mathsf{L}\Psi(x, y)$. Hence by the definition in Section 4.1, $\mathsf{UL}\Psi(\uparrow\{\downarrow\{r\}\}, \uparrow\{\downarrow\{f\}\}) = \mathtt{true}$, as desired.

(ii) Let $\{X_i\}_{i\in I} \subseteq \mathsf{UL}\mathcal{F}$. Since $\bigcup_{i\in I} X_i \supseteq X_i$, we have that $\Psi(A, \bigcup_{i\in I} X_i) = \mathtt{true}$ implies $\Psi(A, X_i) = \mathtt{true}$ for all $i \in I$ by monotonicity. Conversely, suppose $\Psi(A, X_i) = \mathtt{true}$ for all $i \in I$. Then for each $i \in I$ we have by definition

$$\forall x_i \in X_i \, \exists a \in A : \forall r \in a \, \exists f \in x_i : \Psi(r, f).$$

This implies that

$$\forall x \in \bigcup_{i\in I} X_i \, \exists a \in A : \forall r \in a \, \exists f \in x : \Psi(r, f).$$

Therefore, $\Psi(A, \bigcup_{i\in I} X_i) = \mathtt{true}$.

(iii) Let $\{X_i\}_{i\in I} \subseteq \mathsf{UL}\mathcal{R}$. Since $X_i \supseteq \bigcap_{i\in I} X_i$, by monotonicity $\Psi(\bigcap_{i\in I} X_i, B) = \mathtt{true}$ implies $\Psi(X_i, B) = \mathtt{true}$ for all $i \in I$. Conversely, suppose $\Psi(X_i, B) = \mathtt{true}$ for all $i \in I$. Then, for each $i \in I$, we have by definition

$$\forall b \in B \, \exists x_{i,b} \in X_i : \forall r \in x_{i,b} \, \exists f \in b : \Psi(r, f).$$

By Lemma 3.10,

$$\bigcap_{i\in I} X_i = \{\bigcup_{i\in I} y_i : y_i \in X_i \text{ for each } i \in I\}.$$

In particular, $\bigcup_{i\in I} x_{i,b} \in \bigcap_{i\in I} X_i$. Now suppose $b \in B$ and consider $\bigcup_{i\in I} x_{i,b}$. If $r \in \bigcup_{i\in I} x_{i,b}$, then $r \in x_{i,b}$ for some i. Hence there is $f \in b$ such that $\Psi(r, f)$. This shows that

$$\forall b \in B \, \exists a = \bigcup_{i\in I} x_{i,b} \in \bigcap_{i\in I} X_i : \forall r \in a \, \exists f \in b : \Psi(r, f).$$

Therefore, $\Psi(\bigcap_{i\in I} X_i, B) = \texttt{true}$.

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

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

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

This implies, by definition, that $\mathsf{UL}\tilde{\Psi}(A, B) = \texttt{true}$.

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

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

Let $b \in B$. Because b is a lower set, we have $\downarrow\{f\} \subseteq b$ for any $f \in b$. This implies that $\uparrow\{\downarrow\{f\}\} \supseteq \uparrow\{b\}$. Since f was arbitrary, we have by monotonicity,

$$\forall b \in B\, \exists a \in A : \forall r \in a : \Psi(\uparrow\{\downarrow\{r\}\}, \uparrow\{b\}).$$

By property (iii), $\Psi(\uparrow\{\downarrow\{r\}\}, \uparrow\{b\})$ for all $r \in a$ implies $\Psi(\bigcap_{r\in a} \uparrow\{\downarrow\{r\}\}, \uparrow\{b\})$. Hence, we have

$$\forall b \in B\, \exists a \in A : \Psi(\bigcap_{r\in a} \uparrow\{\downarrow\{r\}\}, \uparrow\{b\}).$$

Using Lemma 3.10 (iii), and the fact that a is a lower set, we note that

$$\bigcap_{r\in a} \uparrow\{\downarrow\{r\}\} = \uparrow\{\bigcup_{r\in a} \downarrow\{r\}\} = \uparrow\{a\}.$$

Hence, $\forall b \in B\, \exists a \in A : \Psi(\uparrow\{a\}, \uparrow\{b\})$. Since A is an upper set, $A \supseteq \uparrow\{a\}$ for any $a \in A$, whence by monotonicity, $\forall b \in B : \Psi(A, \uparrow\{b\})$. Applying property (ii) finally yields $\Psi(A, \bigcup_{b\in B} \uparrow\{b\}) = \Psi(A, B) = \texttt{true}$. We have thus shown that $\Psi(A, B) = \texttt{true}$ if and only if $\mathsf{UL}\tilde{\Psi}(A, B) = \texttt{true}$, so the two feasibility relations coincide. ⊣⊢

The previous theorem begs the question whether lifted feasibility fulfills similar relations concerning unions in the source and intersections in the target. This is addressed in the following

Proposition 4.6. *Suppose* $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ *is induced as in Theorem 4.5. Then the following equivalences hold:*

(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.

(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 $z \in \uparrow \{z\}$ 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 $i \in I$.

(ii) Suppose $\Psi(\uparrow \{\downarrow \{s\}\}, \bigcap_{i\in I} X_i) = \texttt{true}$. Recall from Lemma 3.10 that $\bigcap_{i\in I} X_i = \{\bigcup_{i\in I} x_i : x_i \in X_i\}$. Hence,

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

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

$$\forall b \in \{\bigcup_{i\in I} x_i : x_i \in X_i\}\ \forall r \in \downarrow \{s\}\ \exists f \in b : \tilde{\Psi}(r, f). \tag{10}$$

Now suppose for the sake of contradiction 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, setting $a := \downarrow\{s\}$, we have that for each b_i,

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

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 $f \in \bigcup_{i\in I} b_i$. Therefore $\bigcup_{i\in I} b_i \in \{\bigcup_{i\in I} x_i : x_i \in X_i\}$ is an element of the intersection for which there exists $r \in \downarrow\{s\}$, (namely s) such that for all $f \in \bigcup_{i\in I} b_i$ we have $\tilde{\Psi}(s, f) = \texttt{false}$. This contradicts statement (10). ⌐¬

Characterizing Lifted Feasibility Relations

Feasibility Relations $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ that have been lifted from $\tilde{\Psi} : \mathcal{R} \nrightarrow \mathcal{F}$ can be characterized via the three conditions presented in Theorem 4.5. In particular, such Ψ satisfy

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

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

Proposition 4.6 proves that, in addition, lifted Ψ satisfy

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

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

4.4 Queries on Induced Feasibility

Often we are interested in what functionalities we can obtain given a fixed resource, or what resources enable us to obtain a fixed functionality. This process is called querying and is formalized in the following. Moreover, we derive formulas for the query of a choice.

Fix a feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$. We define the following sets called queries.

Definition 4.7 (Querying Resources)**.** For a fixed $A \in \mathsf{UL}\mathcal{R}$ define

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

The result is an upper set in $\mathcal{F}$ which describes all functionalities which can be obtained from the choice of resources A.

Definition 4.8 (Querying Functionalities)**.** For a fixed $B \in \mathsf{UL}\mathcal{F}$ define

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

The result is a lower set in $\mathcal{R}$ which describes all resources which enable the choice of functionalities B. We will omit the subscript when the feasibility relation is clear from context.

Example 4.9. Let $\mathcal{R} := (\mathbb{N}, \geq)$ and $\mathcal{F} := \{\texttt{carrot}, \texttt{eggplant}\}$, as in Example 4.3. Consider the feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ generated by

$$\Psi(1\$, \texttt{carrot}) = \texttt{true},$$

$$\Psi(2\$, \texttt{eggplant}) = \texttt{true}.$$

Then $\mathfrak{f}_\Psi(1\$) = \{\texttt{carrot}\}$ and $\mathfrak{f}_\Psi(2\$) = \{\texttt{carrot}, \texttt{eggplant}\}$. ◇

Proposition 4.10. *Consider a lifted feasibility relation* $\mathsf{UL}\Phi : \mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$. *For subsets* $A \subseteq \mathsf{UL}\mathcal{R}$ *and* $B \subseteq \mathsf{UL}\mathcal{F}$ *we have the following:*

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

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

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

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

Proof. We prove each in turn, denoting $\mathsf{UL}\Phi =: \Psi$.

(i) This follows from Proposition 4.6 (i): We have $\Psi(\bigcup_{a\in A} a, i(f))$ if and only if $\Psi(a, i(f))$ for some $a \in A$. Hence, $f \in \mathfrak{f}(A)$ if and only if $f \in \bigcup_{a\in A} \mathfrak{f}(a)$.

(ii) By monotonicity and Theorem 4.5 (iii) we know $\Psi(\bigcap_{a\in A} a, i(f))$ if and only if $\Psi(a, i(f))$ for all $a \in A$. Hence, $f \in \mathfrak{f}(\bigcap_{a\in A} a)$ if and only if $f \in \mathfrak{f}(a)$ for all $a \in A$. Therefore, $\mathfrak{f}(\bigcap_{a\in A} a) = \bigcap_{a\in A} \mathfrak{f}(a)$.

(iii) By monotonicity and Theorem 4.5 (ii) we know $\Psi(i(r), \bigcup_{b\in B} b)$ if and only if $\Psi(i(r), b)$ for all $b \in B$. Hence, $r \in \mathfrak{r}(\bigcup_{b\in B} b)$ if and only if $r \in \mathfrak{r}(b)$ for all $b \in B$. Therefore, $\mathfrak{r}(\bigcup_{b\in B} b) = \bigcap_{b\in B} \mathfrak{r}(b)$.

(iv) This follows from Proposition 4.6 (ii): We have $\Psi(i(r), \bigcap_{b\in B} b)$ if and only if $\Psi(i(r), b)$ for some $b \in B$. Hence, $r \in \mathfrak{r}(B)$ if and only if $r \in \bigcup_{b\in B} \mathfrak{r}(b)$.

Queries on Induced Feasibility

Given a feasibility relation $\Psi : \mathsf{UL}\mathcal{R} \twoheadrightarrow \mathsf{UL}\mathcal{F}$ and fixed $A \in \mathsf{UL}\mathcal{R}$, we can ask which functionalities in $\mathcal{F}$ are obtainable from A. Analogously, given $B \in \mathsf{UL}\mathcal{F}$, we can ask which resources in $\mathcal{R}$ enable us to get B. This process is called *querying*. When Ψ is lifted, the query of a choice among a set X decomposes into a union or intersection over the corresponding queries of elements $x \in X$, as shown in Proposition 4.10.

5 Future Directions

In this section, we will sketch some avenues for future exploration.

5.1 Choice between Feasibility

We have described how to model free and forced choices between resources and functionalities. However, Co-Design also deals with feasibility relations. Hence, we can ask how we would model free and forced choices between these. For example, there may be two rival grocery stores Φ and Ψ for which we may describe corresponding feasibility relations $\Phi, \Psi : \mathcal{R} \nrightarrow \mathcal{F}$. A free choice $\Phi \sqcap \Psi$ would mean that we have access to the store of our choice. Intuitively, we should have $\Phi \sqcap \Phi := \Phi \cup \Psi$.

Here are some thoughts on how to proceed. First, we observe that the set of feasibility relations $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$, denoted $\mathsf{Feas}(\mathcal{R}, \mathcal{F})$, is itself a preorder where $\Phi \to \Psi$ iff $\Phi \supseteq \Psi$. Therefore, we can consider $\mathsf{ULFeas}(\mathcal{R}, \mathcal{F})$ as we would for any other preorder. This already allows us to think about free and forced choice among feasibility relations.

A next step might be to decompose a given feasibility relation into a choice. We typically assume we can choose freely which feasible pair we use to realize some transformation: If both $\Phi(5\$, \texttt{soup})$ and $\Phi(5\$, \texttt{salad})$, we assume that if we provide 5\$ we can choose whether we use it to purchase `soup` or `salad`. Therefore, it may be reasonable to interpret a feasibility relation Φ as $\bigsqcap_{(r,f)\in\Phi}(r, f)$. This poses the problem that the individual feasible pairs are not in general feasibility relations. However, we note that since a feasibility relation correspond to upper sets of $\mathcal{R}^{\mathrm{op}} \times \mathcal{F}$, we can instead write $\Phi = \bigsqcap_{(r,f)\in\Phi} \uparrow \{(r, f)\}$.

By the decomposition strategy described above, it is sufficient to consider $\mathsf{ULPrinc}(\mathcal{R}^{\mathrm{op}} \times \mathcal{F})$, where $\mathsf{Princ}(\mathcal{R}^{\mathrm{op}} \times \mathcal{F})$ is the set of principal upper sets $\uparrow \{(r, f)\}$ of $\mathcal{R}^{\mathrm{op}} \times \mathcal{F}$. In this picture,

$$\Phi \sqcap \Psi = \bigsqcap_{(r,f)\in\Phi} \uparrow \{(r, f)\} \sqcap \bigsqcap_{(r',f')\in\Psi} \uparrow \{(r', f')\} = \bigsqcap_{(r,f)\in\Phi\cup\Psi} \uparrow \{(r, f)\},$$

which corresponds to the intuition above. On the other hand,

$$\Phi \sqcup \Psi = \bigsqcap_{(r,f)\in\Phi} \uparrow \{(r, f)\} \sqcup \bigsqcap_{(r',f')\in\Psi} \uparrow \{(r', f')\} = \bigsqcap_{(p,q)\in\Phi\times\Psi} \uparrow \{p\} \sqcup \uparrow \{q\},$$

where we have abbreviated $(r, f) =: p$ and $(r', f') =: q$ for legibility.

The above shows that a free choice between decomposed feasibility relations can again be interpreted as a decomposed feasibility relation. However, in order to represent the forced choice $\Phi \sqcup \Psi$ as a free choice over principal upper sets, one would have to define $\uparrow\{p\} \sqcup \uparrow\{q\}$ in a way that makes it principal. The obvious choice of taking the intersection does not fulfill this criterion for arbitrary preorders. One option is to just continue working within $\mathsf{ULPrinc}(\mathcal{R}^{\mathrm{op}} \times \mathcal{F})$, where such forced choices can be represented. Alternatively, one could attempt to "flatten" the forced choice $\Phi \sqcup \Psi$ into a new feasibility relation between upper-lower models $\mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$ by transferring the forced choice to elements of $\mathcal{R}$ and $\mathcal{F}$, respectively. For example, if $\Phi(r, f) = \texttt{true}$ and $\Psi(r, g) = \texttt{true}$, then a flattened forced choice should fulfill $\Phi \sqcup \Psi(r, f \sqcup g) = \texttt{true}$, but not in general $\Phi \sqcup \Psi(r, f) = \texttt{true}$ or $\Phi \sqcup \Psi(r, g) = \texttt{true}$. Intuitively, not being able to choose whether you purchase `soup` or `salad` using 5\$ is the same as being able to purchase the forced choice $\texttt{soup} \sqcup \texttt{salad}$ using 5\$.

In summary, it seems that thinking about choice between feasibility relations will amount to a special case of choice on a preorder. In particular, feasibility relations can be decomposed into principal upper sets of $\mathcal{R}^{\mathrm{op}} \times \mathcal{F}$ allowing us to use $\mathsf{ULPrinc}(\mathcal{R}^{\mathrm{op}} \times \mathcal{F})$ to model choices between them. Moreover, moving to upper-lower models on the source and target of feasibility relations could allow for certain flattening operations, where an element of $\mathsf{ULPrinc}(\mathcal{R}^{\mathrm{op}} \times \mathcal{F})$ could be converted to an element of $\mathsf{Feas}(\mathsf{UL}\mathcal{R}, \mathsf{UL}\mathcal{F})$.

5.2 Additional Connectives

Sometimes the preorder $\mathcal{P}$ may have additional structure. In these cases, we can ask whether this structure can be lifted to its upper-lower model $\mathsf{UL}\mathcal{P}$.

An example of an operation that would be of interest is the addition operator $+$, where $a + b$ means that we are able to independently provide a and b. This operation is commutative and has a unit 0, representing being able to provide 'nothing'. Moreover, being able to provide a implies being able to provide 'nothing' for any a, so $a \to 0$ for all $a \in \mathcal{P}$. Next, we would want $a+b \to a$ for all $a \in \mathcal{P}$, since being able to provide a and b independently in particular implies being able to provide a. Finally, $a \to a'$ and $b \to b'$ imply $a + a' \to b + b'$ by applying transformations in parallel.

We illustrate how one would lift an operation like $+$. First we determine how the operator should interact with free and forced choices. In our case, we would want

$$\bigsqcap_{a\in A} a + \bigsqcap_{b\in B} b = \bigsqcap_{(a,b)\in A\times B} a+b,$$

$$\bigsqcup_{a\in A} a + \bigsqcup_{b\in B} b = \bigsqcup_{(a,b)\in A\times B} a+b.$$

For $U, V \in \mathsf{ULP}$, this suggests setting

$$U{+}V := \uparrow \{u{+}v : u \in U, v \in V\} := \uparrow \{\downarrow \{x{+}y : x \in u, y \in v\} : u \in U, v \in V\}.$$

It then remains to verify which properties of $+$ transfer to the upper-lower level.

5.3 Exploring General Feasibility between Upper-Lower Models

We have seen that lifted feasibility relations are optimal in a sense explored in Section 4.3 and Example 4.4. One would expect that in the real world this optimality would often be violated. A free choice might be more expensive than the maximum cost of the individual options, or a forced choice cheaper than the minimum cost of the options. For example, booking a flight on standby is cheaper than a ticket for any given travel date. These considerations show that we should investigate general feasibility relations between upper-lower models $\mathsf{UL}\mathcal{R} \nrightarrow \mathsf{UL}\mathcal{F}$, as illustrated in Section 4.2.

One way to obtain such general feasibility relations is to explore alternative forms of lifting a feasibility relation $\Phi : \mathcal{R} \nrightarrow \mathcal{F}$ to the upper-lower models. For example, if $\mathcal{R}$ represents money, we could introduce a surcharge of 1\$ whenever a free choice occurs in between elements of $\mathcal{F}$. The magnitude of this surcharge will reflect how much overhead is involved in allowing for a choice.

Finally, one could think about introducing a measure for how far a general feasibility relation is from being a lift. This would allow for a quantification of the surplus cost charged for choice. This could be achieved formally by assigning weights to the arrows in the upper-lower models.

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.