A Response to Iterated Best Response: A Possible Solution to the Problem of Free Choice

by

Sander van Wincoop

Supervised by

Dr. Rick Nouwen

Utrecht University
Utrecht, The Netherlands

July 2021

Abstract

The Free Choice problem is a problem that arises when translating certain sentences pertaining to permission from natural language to logic. A solution to this problem was proposed by Franke called Iterated Best Response. Fox and Katzir recently pointed out a flaw in this solution. In this research, I approached this problem and managed to construct a solution that allowed the IBR algorithm to once again be a viable solution to the Free Choice problem. This solution is then also critically examined to see if it definitively solves the Free Choice problem.

Table of Contents

1 Introduction

Intelligent agents make a lot of choices. A significant portion of these choices is made as a response to input given in natural language by humans. To allow the agent to make the correct choices according to the given input, it is important the agent is able to correctly interpret the given input.

To interpret a sentence in natural language the sentence has to be converted to logic formulas, after which the agent can interpret the input and make its next choice accordingly. When translating natural language sentences to logic, several problems can arise that lead to an agent interpreting sentences in a different way than the person originally giving it intended. In this research, one of these problems will be discussed, namely the Free Choice problem. The Free Choice problem arises when translating a sentence with a modal modifier and a disjunction from natural language to logic. Finding a solution to this problem would allow agents to better interpret natural language the way we want them to, and make their choices accordingly.

The Free Choice Problem was first proposed in Kamp (1973). This is a problem regarding the interpretation of sentences pertaining to permissions. When reading sentence (1) below, the conclusion that sentence (2) holds, as well as sentence (3) can generally be made in natural language by humans. Should sentence (2) hold, but not sentence (3) for example, sentence (1) would not be uttered, but rather just sentence (2).

However, when we look at the logical equivalents of these sentences, this conclusion does not follow. When we know the statement $\Diamond(a \vee b)$ holds it does not follow that $\Diamond a$ holds or that $\Diamond b$ holds (or better yet, that $\Diamond a \wedge \Diamond b$ holds). This is because the statement $\Diamond(a \vee b)$ only tells us that there is a possibility that $a \vee b$ holds, which does not guarantee that a and b are both possible since it is feasible that only a is possible, for instance.

(1) You may eat an apple or a banana. $\Diamond(a \vee b)$

(2) You may eat an apple. $\Diamond a$

(3) You may eat a banana. $\Diamond b$

Because this conclusion does not follow through Kripke semantics we will need to find a different method to interpret sentences such as (1).

Franke (2010) proposes a solution to this problem. This solution involves an iterative process called Iterated Best Response (IBR) using a game-theoretical approach. This solution uses two players; the player sending the message and the player receiving the message and interpreting the meaning. These players first use the logical meanings of the possible messages according to Kripke semantics but evolve their response according to what facilitates the best response from the other player. This iterative process eventually converges and is able to interpret sentences (1), (2), and (3) the same as humans interpret these sentences.

Fox and Katzir (2020) criticize Franke's approach. They showed that, although Franke's IBR reaches the correct conclusion for sentences such as sentence (1), it fails to do so for sentences containing a disjunction made up of more than two disjuncts, such as sentence (4).

(4) You may eat an apple, a banana, or a cherry.

In sentence (4) the interpretation found by humans would be that you are allowed to eat an apple, a banana, as well as a cherry, but perhaps not all at once. When there are more alternative interpretations possible, IBR halts too soon and is not able to interpret every sentence correctly anymore. IBR is able to correctly interpret sentences (2) and (3) but converges before it can reach an interpretation for sentences (1) and (4).

In chapter two Franke's model will be explained in further detail, as well as the problem as described by Fox and Katzir.

The main goal of this research is to attempt to solve this problem described by Fox and Katzir. The research question I will aim to answer is the following: "Is the problem Fox and Katzir present a general problem for Iterated Best Response or just for Franke's version of Iterated Best Response?". I will answer this question by seeing if this problem can be circumvented by adjusting the assumptions made by Franke's IBR. I will show that, with some new assumptions, we can construct a solution that is not impaired by the problem from Fox and Katzir. Subsequently, the follow-up question to this will be "how sufficient is the solution I suggest?". I will first outline how the IBR algorithm operates and demonstrate the problem with IBR that Fox and Katzir described. Then I will investigate possible adjustments to the IBR algorithm and assess whether these adjustments help solve the problem. I will show that, with some adjustments to the IBR algorithm, the problem Fox and Katzir pose can be resolved. Lastly, I will analyse my suggested solution and discuss whether this can be seen as a feasible solution to the Free Choice problem as a whole.

2 Current solution and problem

2.1 Franke's Iterated Best Response

2.1.1 Model framework

Franke's proposal (2010) uses an interpretation game. In this game, a sender has to convey information to the receiver. The sender has the information which state t the world is in, which the receiver does not. The sender can send a message m to the receiver and from this message the receiver has to derive the correct world state t.

To put this into some context, we can construct an interpretation game for the sentences mentioned in section 1.2.

(5) You may eat an apple or a banana. $m_{\Diamond(a \vee b)}$

(6) You may eat an apple. $m_{\Diamond a}$

(7) You may eat a banana. $m_{\Diamond b}$

We can then select a subset of these messages for each world state possible such that the messages are true in that world state according to Kripke semantics:

$$t_a = \{m_{\Diamond a}, m_{\Diamond(a \vee b)}\}$$
$$t_b = \{m_{\Diamond b}, m_{\Diamond(a \vee b)}\}$$
$$t_{ab} = \{m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond(a \vee b)}\}$$

Here, t_a is the world state where you may take an apple, but not a banana, t_b is the world state where you may take a banana but not an apple and t_{ab} is the world state where you may take an apple and a banana. Note that t_a does not contain m_b as a valid message since that message directly contradicts the state. Similarly, the state t_b does not contain m_a as a valid message.

Now we can look at a possible strategy for the sender and receiver. A strategy is a function from states to messages for the sender and from messages to states for the receiver. Here, s is the sender's strategy and r is the receiver's strategy.

$$s = \left\{\begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_{ab} & \mapsto m_{\Diamond(a\vee b)} \end{array}\right\} \qquad r = \left\{\begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond(a\vee b)} & \mapsto t_{ab} \end{array}\right\}$$

This can be read as follows: if the sender is presented with the state t_a, they will give the receiver the message m_a. If the receiver is presented with message m_a, they will guess they are in world state t_a, which would be correct. These two strategies work well together; in every possible state the sender sends a message that allows the receiver to correctly deduce the world state.

2.1.2 The model

In Franke's model, the strategies from the receiver and sender are both based on the strategy of the other player and adapt over several iterations. Every new strategy iteration is generated in such a way to maximize the chance of a world state being deduced correctly, to optimize success. This process iterates until it reaches an equilibrium and does not change over iterations. This is reached when either a receiver strategy is the same as a previous receiver strategy or a sender strategy is the same as a previous sender strategy. Below is an example of the model for the world states and messages we looked at earlier in 2.1.1, accompanied by tables displaying the different probabilities of each message being sent in a certain state.

$$R_0 = \left\{\begin{array}{ll} m_{\Diamond a} & \mapsto t_a, t_{ab} \\ m_{\Diamond b} & \mapsto t_b, t_{ab} \\ m_{\Diamond(a\vee b)} & \mapsto t_a, t_b, t_{ab} \end{array}\right\} \qquad S_1 = \left\{\begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_{ab} & \mapsto m_{\Diamond a}, m_{\Diamond b} \end{array}\right\}$$

$$R_2 = \left\{\begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond(a\vee b)} & \mapsto t_a, t_b, t_{ab} \end{array}\right\} \qquad S_3 = \left\{\begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_{ab} & \mapsto m_{\Diamond(a\vee b)} \end{array}\right\}$$

$$R_4 = \left\{\begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond(a\vee b)} & \mapsto t_{ab} \end{array}\right\} \qquad S_5 = \left\{\begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_{ab} & \mapsto m_{\Diamond(a\vee b)} \end{array}\right\}$$

R_0	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond(a\vee b)}$
t_a	1/2	0	1/3
t_b	0	1/2	1/3
t_{ab}	1/2	1/2	1/3

S_1	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond(a\vee b)}$
t_a	1	0	
t_b	0	1	
t_{ab}	1/2	1/2	

R_2	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond(a\vee b)}$
t_a	1	0	1/3
t_b	0	1	1/3
t_{ab}	0	0	1/3

S_3	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond(a\vee b)}$
t_a	1	0	0
t_b	0	1	0
t_{ab}	0	0	1

R_4	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond(a\vee b)}$
t_a	1	0	0
t_b	0	1	0
t_{ab}	0	0	1

S_5	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond(a\vee b)}$
t_a	1	0	0
t_b	0	1	0
t_{ab}	0	0	1

Table 1. *The probability distributions at different strategy iterations for a situation with two disjuncts*

To read these tables it is important to distinguish between the tables for the receiver and the tables for the sender. The tables for the receiver, table R_0 for instance, can be read as follows: The receiver is presented a message, corresponding to a column. They have to evaluate which state the message is most likely to portray. In R_0, which is based solely on Kripke semantics, the message $m_{\Diamond a}$ is true in states t_a and t_{ab} so both these states would get a probability of one in two for the message $m_{\Diamond a}$. Whereas message $m_{\Diamond(a\vee b)}$ is true in all three states. Thus, this message gets a probability of one in three for all three states. The tables for the sender are the other way around. The sender is given a state and has to evaluate which message is best to send. In S_1, which is based on R_0, it is calculated that in state t_a it is best to send message $m_{\Diamond a}$ since this message has a one in two chance to be guessed as state t_a as opposed to message $m_{\Diamond(a \vee b)}$'s lower one in three chance. Here, in state t_{ab}, it is actually best to send message $m_{\Diamond a}$ or message $m_{\Diamond b}$, rather than message $m_{\Diamond(a \vee b)}$ as these first two messages will have a one in two chance to be guessed as state t_{ab}, whereas message $m_{\Diamond(a \vee b)}$ once again only has a one in three chance.

As I mentioned the process starts with the receiver strategy R_0. This strategy only takes the semantic meaning of the messages into account. We can see that it assumes every message could imply the world state t_{ab} and the message $m_{\Diamond(a \vee b)}$ could imply every possible world state. The sender strategy S_1 is then generated according to R_0. The next receiver strategy R_2 is in turn based on the sender strategy S_1. We can see that the message $m_{\Diamond(a \vee b)}$ is not included for any world state in strategy S_1. This means the sender will never send message $m_{\Diamond(a \vee b)}$. Should the receiver still receive this message anyway, it is then called a surprise message. If the receiver receives a surprise message, they will simply interpret it to be true according to Kripke semantics. This means that, for message $m_{\Diamond(a \vee b)}$, the receiver will still hold all world states as equally possible options. After this, strategy S_3 is based on R_2 and then strategy R_4 is based on S_3. At this point the model reaches an equilibrium, the next sender strategy, S_5, is the same as the previous sender strategy, S_3. Because of this, the next receiver strategy will also be the same as the previous receiver strategy R_4, since this strategy was already based on S_3.

When we assess these resulting strategies we can see they perform quite well. In every possible world state the message sent by the sender is deduced as the correct world state by the receiver. Furthermore, we can note that this works in a way that humans would find intuitive; the messages the sender uses are not seen as inappropriate for any given world state.

2.2 A problem for Iterated Best Response

As mentioned before, Fox and Katzir described a problem of Franke´s Iterated Best Response (2020). This problem arises when the sentence contains a disjunction made up of more than two disjuncts. An example of this, along with the corresponding messages, can be seen below.

(8) You may eat an apple, a banana, or a cherry. $m_{\Diamond(a \vee b \vee c)}$

(9) You may eat an apple or a cherry. $m_{\Diamond(a \vee c)}$

(10) You may eat a cherry. $m_{\Diamond c}$

If we attempt to run the model on this new situation we get the following iterations of strategies:

$$R_0 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_a, t_{ab}, t_{ac}, t_{abc} \\ m_{\Diamond b} & \mapsto t_b, t_{ab}, t_{bc}, t_{abc} \\ m_{\Diamond c} & \mapsto t_c, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(b \vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \end{array} \right\}$$

$$S_1 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_{ab} & \mapsto m_{\Diamond a}, m_{\Diamond b} \\ t_{ac} & \mapsto m_{\Diamond a}, m_{\Diamond c} \\ t_{bc} & \mapsto m_{\Diamond b}, m_{\Diamond c} \\ t_{abc} & \mapsto m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond c} \end{array} \right\}$$

$$R_2 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond c} & \mapsto t_c \\ m_{\Diamond(a \vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(b \vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \end{array} \right\}$$

$$S_3 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_{ab} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \\ t_{ac} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \\ t_{bc} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \\ t_{abc} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \end{array} \right\}$$

$$R_4 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond c} & \mapsto t_c \\ m_{\Diamond(a \vee b)} & \mapsto t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee c)} & \mapsto t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(b \vee c)} & \mapsto t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \end{array} \right\}$$

$$S_5 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_{ab} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \\ t_{ac} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \\ t_{bc} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \\ t_{abc} & \mapsto m_{\Diamond(a \vee b)}, m_{\Diamond(a \vee c)}, m_{\Diamond(b \vee c)} \end{array} \right\}$$

R_0	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond c}$	$m_{\Diamond(a\vee b)}$	$m_{\Diamond(a\vee c)}$	$m_{\Diamond(b\vee c)}$	$m_{\Diamond(a\vee b\vee c)}$
t_a	1/4	0	0	1/6	1/6	0	1/7
t_b	0	1/4	0	1/6	0	1/6	1/7
t_c	0	0	1/4	0	1/6	1/6	1/7
t_{ab}	1/4	1/4	0	1/6	1/6	1/6	1/7
t_{ac}	1/4	0	1/4	1/6	1/6	1/6	1/7
t_{bc}	0	1/4	1/4	1/6	1/6	1/6	1/7
t_{abc}	1/4	1/4	1/4	1/6	1/6	1/6	1/7

S_1	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond c}$	$m_{\Diamond(a\vee b)}$	$m_{\Diamond(a\vee c)}$	$m_{\Diamond(b\vee c)}$	$m_{\Diamond(a\vee b\vee c)}$
t_a	1	0	0	0	0	0	0
t_b	0	1	0	0	0	0	0
t_c	0	0	1	0	0	0	0
t_{ab}	1/2	1/2	0	0	0	0	0
t_{ac}	1/2	0	1/2	0	0	0	0
t_{bc}	0	1/2	1/2	0	0	0	0
t_{abc}	1/3	1/3	1/3	0	0	0	0

R_2	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond c}$	$m_{\Diamond(a\vee b)}$	$m_{\Diamond(a\vee c)}$	$m_{\Diamond(b\vee c)}$	$m_{\Diamond(a\vee b\vee c)}$
t_a	1	0	0	1/6	1/6	0	1/7
t_b	0	1	0	1/6	0	1/6	1/7
t_c	0	0	1	0	1/6	1/6	1/7
t_{ab}	0	0	0	1/6	1/6	1/6	1/7
t_{ac}	0	0	0	1/6	1/6	1/6	1/7
t_{bc}	0	0	0	1/6	1/6	1/6	1/7
t_{abc}	0	0	0	1/6	1/6	1/6	1/7

S_3	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond c}$	$m_{\Diamond(a\vee b)}$	$m_{\Diamond(a\vee c)}$	$m_{\Diamond(b\vee c)}$	$m_{\Diamond(a\vee b\vee c)}$
t_a	1	0	0	0	0	0	0
t_b	0	1	0	0	0	0	0
t_c	0	0	1	0	0	0	0
t_{ab}	0	0	0	1/3	1/3	1/3	0
t_{ac}	0	0	0	1/3	1/3	1/3	0
t_{bc}	0	0	0	1/3	1/3	1/3	0
t_{abc}	0	0	0	1/3	1/3	1/3	0

R_4	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond c}$	$m_{\Diamond(a\vee b)}$	$m_{\Diamond(a\vee c)}$	$m_{\Diamond(b\vee c)}$	$m_{\Diamond(a\vee b\vee c)}$
t_a	1	0	0	0	0	0	1/7
t_b	0	1	0	0	0	0	1/7
t_c	0	0	1	0	0	0	1/7
t_{ab}	0	0	0	1/4	1/4	1/4	1/7
t_{ac}	0	0	0	1/4	1/4	1/4	1/7
t_{bc}	0	0	0	1/4	1/4	1/4	1/7
t_{abc}	0	0	0	1/4	1/4	1/4	1/7

S_5	$m_{\Diamond a}$	$m_{\Diamond b}$	$m_{\Diamond c}$	$m_{\Diamond(a\vee b)}$	$m_{\Diamond(a\vee c)}$	$m_{\Diamond(b\vee c)}$	$m_{\Diamond(a\vee b\vee c)}$
t_a	1	0	0	0	0	0	0
t_b	0	1	0	0	0	0	0
t_c	0	0	1	0	0	0	0
t_{ab}	0	0	0	1/3	1/3	1/3	0
t_{ac}	0	0	0	1/3	1/3	1/3	0
t_{bc}	0	0	0	1/3	1/3	1/3	0
t_{abc}	0	0	0	1/3	1/3	1/3	0

Table 2. *The probability distributions at different strategy iterations for a situation with three disjuncts*

We can see that the model is able to link messages $m_{\Diamond a}$, $m_{\Diamond b}$, and $m_{\Diamond c}$ to states t_a, t_b, and t_c respectively, rather quickly. Yet after this, it fails to reach a distinction for the other messages and world states. Because the sender strategy S_5 is the same as the previous sender strategy S_3, the model halts after this iteration. If we evaluate the found strategies S_5 and R_4 we see that sadly, it does not perform well. States t_{ab}, t_{ac}, t_{bc}, and t_{abc} do not have a definitive solution. In each of these states there is only a one in four chance of the state being guessed correctly. This can be explained as the model not being able to make a distinction between the messages $m_{\Diamond(a\vee b)}$, $m_{\Diamond(a\vee c)}$, and $m_{\Diamond(b\vee c)}$ regarding the states t_{ab}, t_{ac}, t_{bc}, and t_{abc}.

Because of the addition of this newly introduced third disjunct, the model fails to reach a correct solution. This means we need to change the way this model works in order for it to be able to deal with more disjuncts.

3 Adjusting the IBR algorithm

3.1 Background

The algorithm initially fails to distinguish between the states t_{ab}, t_{ac}, t_{bc}, and t_{abc}; all states that would require disjunctions in natural languages. The outcome of the IBR algorithm is highly dependent on what is initially entered into it. For instance, what messages are taken into account and in what way they are interpreted. Because of this, it might be an option to add new messages that aid it in distinguishing between these aforementioned states. For instance, if I perhaps add a message that says ”You may eat an apple or a banana but not a cherry”, it could be found to be state t_{ab}, as we would interpret it.

3.2 Disjunct Negations

The first messages added were messages that stated that certain disjuncts are not true. These were chosen since they could possibly help distinguish between the states mentioned before.

(11) You may not eat an apple. $\neg\Diamond a$ $m_{\neg\Diamond a}$

(12) You may not eat a banana. $\neg\Diamond b$ $m_{\neg\Diamond b}$

(13) You may not eat a cherry. $\neg\Diamond c$ $m_{\neg\Diamond c}$

If we run the IBR algorithm together with these three new messages, we get the following iterations:

$$R_0 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_a, t_{ab}, t_{ac}, t_{abc} \\ m_{\Diamond b} & \mapsto t_b, t_{ab}, t_{bc}, t_{abc} \\ m_{\Diamond c} & \mapsto t_c, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a\vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a\vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(b\vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a\vee b\vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\neg\Diamond a} & \mapsto t_b, t_c, t_{bc} \\ m_{\neg\Diamond b} & \mapsto t_a, t_c, t_{ac} \\ m_{\neg\Diamond c} & \mapsto t_a, t_b, t_{ab} \end{array} \right\} \qquad S_1 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\neg\Diamond b}, m_{\neg\Diamond c} \\ t_b & \mapsto m_{\neg\Diamond a}, m_{\neg\Diamond c} \\ t_c & \mapsto m_{\neg\Diamond a}, m_{\neg\Diamond b} \\ t_{ab} & \mapsto m_{\neg\Diamond c} \\ t_{ac} & \mapsto m_{\neg\Diamond b} \\ t_{bc} & \mapsto m_{\neg\Diamond a} \\ t_{abc} & \mapsto m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond c} \end{array} \right\}$$

$$R_2 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_{abc} \\ m_{\Diamond b} & \mapsto t_{abc} \\ m_{\Diamond c} & \mapsto t_{abc} \\ m_{\Diamond(a\vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a\vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(b\vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond(a\vee b\vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\neg\Diamond a} & \mapsto t_{bc} \\ m_{\neg\Diamond b} & \mapsto t_{ac} \\ m_{\neg\Diamond c} & \mapsto t_{ab} \end{array} \right\} \qquad S_3 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond(a\vee b)}, m_{\Diamond(a\vee c)} \\ t_b & \mapsto m_{\Diamond(a\vee b)}, m_{\Diamond(b\vee c)} \\ t_c & \mapsto m_{\Diamond(a\vee c)}, m_{\Diamond(b\vee c)} \\ t_{ab} & \mapsto m_{\neg\Diamond c} \\ t_{ac} & \mapsto m_{\neg\Diamond b} \\ t_{bc} & \mapsto m_{\neg\Diamond a} \\ t_{abc} & \mapsto m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond c} \end{array} \right\}$$

$$R_4 = \left\{\begin{array}{ll} m_{\Diamond a} & \mapsto t_{abc} \\ m_{\Diamond b} & \mapsto t_{abc} \\ m_{\Diamond c} & \mapsto t_{abc} \\ m_{\Diamond (a \vee b)} & \mapsto t_a, t_b \\ m_{\Diamond (a \vee c)} & \mapsto t_a, t_c \\ m_{\Diamond (b \vee c)} & \mapsto t_b, t_c \\ m_{\Diamond (a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\neg \Diamond a} & \mapsto t_{bc} \\ m_{\neg \Diamond b} & \mapsto t_{ac} \\ m_{\neg \Diamond c} & \mapsto t_{ab} \end{array}\right\} \qquad S_5 = \left\{\begin{array}{ll} t_a & \mapsto m_{\Diamond (a \vee b)}, m_{\Diamond (a \vee c)} \\ t_b & \mapsto m_{\Diamond (a \vee b)}, m_{\Diamond (b \vee c)} \\ t_c & \mapsto m_{\Diamond (a \vee c)}, m_{\Diamond (b \vee c)} \\ t_{ab} & \mapsto m_{\neg \Diamond c} \\ t_{ac} & \mapsto m_{\neg \Diamond b} \\ t_{bc} & \mapsto m_{\neg \Diamond a} \\ t_{abc} & \mapsto m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond c} \end{array}\right\}$$

Strategy S_5 is the same as strategy S_3 so the model converges here. If we evaluate the found strategies R_4 and S_5 we find that this solution does not perform well. The world states t_a, t_b and t_c only have a one in two chance to be evaluated correctly. More importantly though, the interpretation of the messages does not correspond with the interpretation of humans when read in natural language. For instance, $m_{\Diamond a}$, $m_{\Diamond b}$, and $m_{\Diamond c}$ are all interpreted by the receiver as world state t_{abc}. Humans would not interpret these messages this way but rather as states t_a, t_b, and t_c, respectively.

Another method with negations embedded in the original messages was also tested (see appendix A). In this method the receiver was able to deduce all states correctly, but the model failed to reach an interpretation that was in line with the human interpretation.

3.3 Conjunctions

The next four messages added are similar to the messages used originally but use conjunctions instead of disjunctions. These messages were chosen since humans, in natural language, might use the word ‘and’ to specify when several options are allowed.

(14) You may eat an apple and a banana. $\Diamond(a \wedge b)$ $m_{\Diamond (a \wedge b)}$

(15) You may eat an apple and a cherry. $\Diamond(a \wedge c)$ $m_{\Diamond(a\wedge c)}$

(16) You may eat a banana and a cherry. $\Diamond(b \wedge c)$ $m_{\Diamond(b\wedge c)}$

(17) You may eat an apple and a banana and a cherry. $\Diamond(a \wedge b \wedge c)$ $m_{\Diamond(a\wedge b\wedge c)}$

If we run the IBR algorithm together with these four new messages, we get the following iterations:

$$R_0 = \left\{\begin{array}{ll}
m_{\Diamond a} & \mapsto t_a, t_{ab}, t_{ac}, t_{abc} \\
m_{\Diamond b} & \mapsto t_b, t_{ab}, t_{bc}, t_{abc} \\
m_{\Diamond c} & \mapsto t_c, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond(a\vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond(a\vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond(b\vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond(a\vee b\vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond(a\wedge b)} & \mapsto t_{ab}, t_{abc} \\
m_{\Diamond(a\wedge c)} & \mapsto t_{ac}, t_{abc} \\
m_{\Diamond(b\wedge c)} & \mapsto t_{bc}, t_{abc} \\
m_{\Diamond(a\wedge b\wedge c)} & \mapsto t_{abc}
\end{array}\right\}
\qquad
S_1 = \left\{\begin{array}{ll}
t_a & \mapsto m_{\Diamond a} \\
t_b & \mapsto m_{\Diamond b} \\
t_c & \mapsto m_{\Diamond c} \\
t_{ab} & \mapsto m_{\Diamond(a\wedge b)} \\
t_{ac} & \mapsto m_{\Diamond(a\wedge c)} \\
t_{bc} & \mapsto m_{\Diamond(b\wedge c)} \\
t_{abc} & \mapsto m_{\Diamond(a\wedge b\wedge c)}
\end{array}\right\}$$

$$R_2 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond c} & \mapsto t_c \\ m_{\Diamond (a \vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (a \vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (b \vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (a \wedge b)} & \mapsto t_{ab} \\ m_{\Diamond (a \wedge c)} & \mapsto t_{ac} \\ m_{\Diamond (b \wedge c)} & \mapsto t_{bc} \\ m_{\Diamond (a \wedge b \wedge c)} & \mapsto t_{abc} \end{array} \right\} \qquad S_3 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_{ab} & \mapsto m_{\Diamond (a \wedge b)} \\ t_{ac} & \mapsto m_{\Diamond (a \wedge c)} \\ t_{bc} & \mapsto m_{\Diamond (b \wedge c)} \\ t_{abc} & \mapsto m_{\Diamond (a \wedge b \wedge c)} \end{array} \right\}$$

This version of the model converges rather quickly and when we evaluate it we find that it performs well in the game-theoretical sense. Take note that this only means that, in every world state, the message sent by the sender is interpreted as the correct world state by the receiver. This does not mean that the messages are interpreted similar to the human interpretation. In this case, most of the messages are interpreted similar to how humans interpret their natural language equivalents. However, all the messages still containing disjunctions ($m_{\Diamond (a \vee b)}$, $m_{\Diamond (a \vee c)}$, $m_{\Diamond (b \vee c)}$ and $m_{\Diamond (a \vee b \vee c)}$) are all surprise messages since they are not used by the sender in any state and as such, have the truth-conditions of Kripke semantics. They still have not reached a definitive interpretation.

Although this version performs well, we cannot say that it can be seen as a correct solution to the Free Choice problem. The Free Choice problem arises because a sentence containing a disjunction such as "You may eat an apple or a banana" is not interpreted correctly and in these found strategies those sentences are still not being interpreted correctly, if we can even say they are being interpreted at all. If we choose to ignore these messages and instead only focus on the messages containing conjunctions instead it still does not offer a fitting solution since these sentences do not carry the implication that, although more than one disjuncts are allowed, you perhaps are not allowed to eat more than one at the same time. This point is in line with Chemla and Bott's outlook (2014). It is important the original sentences, particularly the sentences with disjunctions, are interpreted correctly, rather than new messages taking over their role.

3.4 Implied negation

The problem with adding new messages seems to be that, by adding new messages, IBR does not reach new interpretations for the old messages. Despite the fact that reaching interpretations for particularly these messages is what is needed to provide a solution to the Free Choice problem. Instead of adding new messages, it might be required to adapt the way the original messages are first interpreted. If the original semantic reading of the messages is adjusted it could possibly allow the algorithm to distinguish between the states mentioned in section 3.1.

A possible way of adjusting the semantic reading of the messages I constructed, I will label *implied negation*. In this reading it is assumed that, if a disjunct is not mentioned in a certain message, it is implied that that disjunct is not true. For example, the sentence "You may eat an apple or a banana" is interpreted here as "You may eat an apple or a banana but not a cherry". Or in other words $\Diamond(a \vee b)$ is interpreted as $\Diamond(a \vee b) \wedge \neg\Diamond c$. Because of this, message $m_{\Diamond(a\vee b)}$ is now originally interpreted to be true in only states t_a, t_b, and t_{ab} and not states t_{ac}, t_{bc}, and t_{abc} anymore. This exhaustive implicature can be compared to the exhaustive interpretation constructed by Schulz and Van Rooij (2006).

Running IBR with this new reading gives us the following iterations:

$$R_0 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond c} & \mapsto t_c \\ m_{\Diamond(a\vee b)} & \mapsto t_a, t_b, t_{ab} \\ m_{\Diamond(a\vee c)} & \mapsto t_a, t_c, t_{ac} \\ m_{\Diamond(b\vee c)} & \mapsto t_b, t_c, t_{bc} \\ m_{\Diamond(a\vee b\vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \end{array} \right\} \qquad S_1 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_{ab} & \mapsto m_{\Diamond(a\vee b)} \\ t_{ac} & \mapsto m_{\Diamond(a\vee c)} \\ t_{bc} & \mapsto m_{\Diamond(b\vee c)} \\ t_{abc} & \mapsto m_{\Diamond(a\vee b\vee c)} \end{array} \right\}$$

$$R_2 = \begin{Bmatrix} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond c} & \mapsto t_c \\ m_{\Diamond (a \vee b)} & \mapsto t_{ab} \\ m_{\Diamond (a \vee c)} & \mapsto t_{ac} \\ m_{\Diamond (b \vee c)} & \mapsto t_{bc} \\ m_{\Diamond (a \vee b \vee c)} & \mapsto t_{abc} \end{Bmatrix} \qquad S_3 = \begin{Bmatrix} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_{ab} & \mapsto m_{\Diamond (a \vee b)} \\ t_{ac} & \mapsto m_{\Diamond (a \vee c)} \\ t_{bc} & \mapsto m_{\Diamond (b \vee c)} \\ t_{abc} & \mapsto m_{\Diamond (a \vee b \vee c)} \end{Bmatrix}$$

Note that the IBR algorithm still operates the same. The new reading only changes the first strategy R_0. If we now evaluate this outcome it looks quite promising. The model performs well and, more importantly, in a way that directly mimics the human interpretation. The receiver interprets every message the same way humans would interpret the natural language equivalent.

If we are to use implied negation, it should also work for the original case where we had a sentence containing a disjunction made up of only two disjuncts. Let us test IBR with implied negation for the original sentences (1), (2), and (3) mentioned in chapter 1:

$$R_0 = \begin{Bmatrix} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond (a \vee b)} & \mapsto t_a, t_b, t_{ab} \end{Bmatrix} \qquad S_1 = \begin{Bmatrix} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_{ab} & \mapsto m_{\Diamond (a \vee b)} \end{Bmatrix}$$

$$R_2 = \begin{Bmatrix} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond (a \vee b)} & \mapsto t_{ab} \end{Bmatrix} \qquad S_3 = \begin{Bmatrix} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_{ab} & \mapsto m_{\Diamond (a \vee b)} \end{Bmatrix}$$

We find that implied negation still works for sentences containing disjunctions made up of only two disjuncts. Here, it finds the same solution as before with these messages, albeit slightly faster. We can conclude that IBR still finds the 'correct' solution (i.e. in line with human interpretation) using implied negation.

IBR with implied negation also operates well with sentences containing disjunctions made up of four disjuncts (see Appendix B). Here, the algorithm still finds the interpretation humans find.

3.5 Proof for any number of disjuncts

We can take the number of disjuncts a step further and prove that IBR with implied negation works for sentences containing a disjunction made up of any number of disjuncts. We will show that IBR with implied negation will end with every message being interpreted the way humans would. This will be done by proving that, when the first sender strategy S_1 is generated, it will find the correct, and only the correct message, at this iteration. Here, 'the correct message' means the message that humans would give. To show this, we will need to prove that, for every state, the correct message will have the highest probability for that state in R_0 out of all possible messages.

Let us take a situation with an arbitrary number of disjuncts. Take an arbitrary state t where the number of disjuncts allowed is n. The disjuncts allowed in state t can be gathered into a set A_t. When the first sender strategy S_1 is generated, it will find the correct, and only the correct message m_c, at this iteration. This will be the message that includes all disjuncts allowed in state t and no more (i.e. set A_t). We will take an arbitrary message m. The disjuncts mentioned in message m can be gathered into a set D_m. We will show that message m will only be linked to state t in S_1 if it mentions those disjuncts, and only those disjuncts, mentioned in state t (i.e. if $D_m = A_t$). We proceed with a proof by cases. Every message falls into one of three cases: the message does not contain all disjuncts allowed in state t, the message contains exactly those disjuncts allowed in state t, and the message contains all disjuncts allowed in state t, as well as others. Here, the second case is the correct message m_c. We want to show this case will have the highest probability for state t. We will compare the probability for state t in all cases and show that this second case will provide us with the highest probability for state t in R_0.

Case 1: The message does not contain all disjuncts allowed in state t ($A_t \nsubseteq D_m$).

If message m does not contain all disjuncts mentioned in state t, it follows that there is at least one allowed disjunct $d \in A_t$ such that $d \notin D_m$. This means $A_t \not\subseteq D_m$ and likewise $A_t \neq D_m$. Since this disjunct d is not mentioned in message m, it is assumed to be false through implied negation. This leads to state t not being a valid state from message m in R_0.

Case 2: The message contains exactly those disjuncts allowed in state t ($A_t = D_m$).
If message m contains exactly those disjuncts allowed in state t, we know that $A_t = D_m$. In other words, message m is the correct message m_c for state t. Since message m contains all disjuncts allowed in state t, it also follows that message m entails state t in R_0, among other states.

We can calculate the probability for state t from message m_c ($P(t|m_c)$) in R_0. To do this, we need the amount of states that could be true given message m_c. Message m_c contains only those disjuncts allowed in state t, so message m_c contains n disjuncts. The number of states that would be valid given message m_c would then be $2^n - 1$ (here, -1 is included since there is no state where no disjuncts are allowed). Since every state is assumed to have an equal probability the probability for state t, given message m_c, would then be $\frac{1}{2^n-1}$ in R_0.

Case 3: The message contains all disjuncts allowed in state t, as well as others ($A_t \subset D_m$).
If message m contains all disjuncts allowed in state t, as well as other disjuncts it follows that $A_t \neq D_m$: There is at least one mentioned disjunct $d \in D_m$ that is not allowed in state t, and thus, is not present in A_t. Since $d \in D_m$ and $d \notin A_t$ it follows that $A_t \neq D_m$. However, since every disjunct allowed in state t is also mentioned in message m, it follows that $A_t \subset D_m$. This means state t will still be a valid state for message m in R_0.

Similar to in case 2, we can also calculate the probability for state t from message m ($P(t|m_c)$) in R_0. We do not know the specific amount of disjuncts mentioned in message m, however, we do know it is more than the number of disjuncts allowed in state t. Let x be the number of disjuncts mentioned in message m. It then follows that $n < x$. The number of states that would be valid given message m would be $2^x - 1$. Since every state is assumed to have an equal probability the probability for state t, given message m ($P(t|m)$), would then be $\frac{1}{2^x-1}$ in R_0.

Now compare this probability with the probability for state t given message m_c:

$$\begin{gathered} n < x \\ 2^n < 2^x \\ 2^n - 1 < 2^x - 1 \\ \frac{1}{2^n - 1} > \frac{1}{2^x - 1} \\ P(t|m_c) > P(t|m) \end{gathered}$$

We see that $P(t|m_c) > P(t|m)$. This means that, although message m does entail state t, message m_c will entail state t with a higher probability.

We see that in R_0 message m_c (seen in case 2) is the message that gives the highest probability for state t. All other messages either have a lower probability (seen in case 3) or cannot entail state t at all (seen in case 1). When generating strategy S_1 it then follows that the only message the sender will send in state t will be the desired message m_c, since this message has the highest probability in R_0 to be guessed as state t of all messages. Since state t was arbitrary this will be the case for every possible state. This means that in S_1 the sender will send the desired message and only the desired message in every state. After this, the receiver strategy R_2 is generated according to strategy S_1. Every possible message will have exactly one state associated with it in S_1. This means that every message in R_2 will link to the state that originally linked to it in S_1. Since every state in S_1 is linked to the desired message, this means every message in R_2 will be linked to its corresponding desired state. The same happens when the next sender strategy S_3 is generated according to R_2. Because of this, strategy S_3 will be the same as strategy S_1 and the IBR algorithm will stop.

When evaluating the found strategies R_2 and S_3 we can conclude that the model reached the interpretation directly in line with the human interpretation. The receiver interprets every message the way humans would as well. Since the number of disjunctions in this situation was arbitrary, we can conclude that IBR with implied negation finds the correct interpretation for sentences containing a disjunction with any number of disjuncts.

4 Discussion

By using the implied negation reading I was able to adjust the IBR algorithm in such a way that it was able to reach the interpretation of the messages in line with human interpretation. Since the original messages that caused the Free Choice problem in the first place are now interpreted correctly, it could be argued that this solves the Free Choice problem for sentences containing disjunctions made up of any number of disjuncts.

Let us recall our research question: "Is the problem Fox and Katzir present a general problem for Iterated Best Response or just for Franke's version of Iterated Best Response?". I have shown that the problem argued by Fox and Katzir (2020) can be solved by changing the initial reading of the messages to be more in line with human implicatures. This shows that the problem argued by Fox and Katzir only presents a problem for Franke's version of IBR and not for IBR as a whole. If we go further and look at the follow-up question "how sufficient is the solution I suggest?", I can argue my solution is quite satisfactory, although some criticisms can be made.

Firstly, it can be argued that implied negation takes away of the original strength of IBR. IBR originally reaches the implied negation reading on its own; $\Diamond a \land \neg \Diamond b$ is assigned to $\Diamond a$ before, but now we explicitly include this implication $\Diamond a \rightarrow \Diamond a \land \neg \Diamond b$. This exhaustivity is often aimed to be emergent behaviour, rather than explicitly taught (Franke, 2011). Franke's IBR is initially able to reach the desired interpretations based on Kripke semantics alone, without any additional modification. Now that implied negation is added explicitly it does lessen this strength.

Secondly, it can be argued that adding the implied negation could be seen as similar to simply adding the Free Choice implication; $\Diamond(a \vee b) \rightarrow \Diamond a \vee \Diamond b$, which we would prefer to avoid. We wish to construct a method for finding interpretations of natural language sentences that finds the solution to these cases itself, without explicitly being told how to handle them. A method where finding these interpretations is emergent behaviour stemming from relatively 'simple' rules would be favored over adding these rules specifically. We can note, however, that the addition of implied negation does cause the conjunctive reading of disjunctions to emerge, without adding the rule itself. The Free Choice implication is not a direct consequence of implied negation; $\Diamond(a \vee b)$ does not entail that $\Diamond a \wedge \Diamond b$ under implied negation. It only entails that other disjuncts are false.

Reflecting on my work I can conclude that IBR can still be seen as a valid approach to the Free Choice problem. Although it seems to have some flaws, these can be addressed by adjusting what is put into the algorithm to begin with. In this research I approached a problem with IBR and managed to solve it. I also observed that adding new messages does not seem to be as suitable as they do not assist in finding a correct interpretation of problem-messages but rather take over those messages altogether. We cannot say for certain that IBR is now without flaw as it is still very much possible unforeseen issues can arise. However, IBR does seem to be a promising approach to the Free Choice problem.

Future research could assess if there are any problems with IBR and implied negation not seen here. For instance, I only demonstrated that this method works for sentences that require a Free Choice reading, but this is not the case for every sentence containing a disjunction. Because of this, future research could analyze how IBR with implied negation operates on other sentences containing disjunctions and see if problems emerge there. Future research could also apply this method for interpreting free choice sentences to an agent and assess if it operates well together with humans or if misunderstandings arise.

References

Lewis Bott and Emmanuel Chemla. *Processing inferences at the semantics/pragmatics frontier: disjunctions and free choice*. 2014.

Danny Fox and Roni Katzir. *Notes on iterated rationality models of scalar implicatures*. 2020.

Michael Franke. *Free Choice from Iterated Best Response*. 2010.

Michael Franke. *Quantity implicatures, exhaustive interpretation, and rational conversation*. 2011.

Hans Kamp. *Free Choice Permission*. In *Proceedings of the Aristotelian Society*, pages 57–74. Oxford University Press, 1973.

Katrin Schulz and Robert van Rooij. *Pragmatic Meaning and Non-monotonic Reasoning: The Case of Exhaustive Interpretation*. 2006.

5 Appendix

A. Adding negations in original messages

$$R_0 = \left\{ \begin{array}{ll}
m_{\Diamond a} & \mapsto t_a, t_{ab}, t_{ac}, t_{abc} \\
m_{\Diamond b} & \mapsto t_b, t_{ab}, t_{bc}, t_{abc} \\
m_{\Diamond c} & \mapsto t_c, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond (a \vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond (a \vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond (b \vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond (a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond (a \wedge \neg b)} & \mapsto t_a, t_{ac} \\
m_{\Diamond (a \wedge \neg c)} & \mapsto t_a, t_{ab} \\
m_{\Diamond (a \wedge \neg b \wedge \neg c)} & \mapsto t_a \\
m_{\Diamond (b \wedge \neg a)} & \mapsto t_b, t_{bc} \\
m_{\Diamond (b \wedge \neg c)} & \mapsto t_b, t_{ab} \\
m_{\Diamond (b \wedge \neg a \wedge \neg c)} & \mapsto t_b \\
m_{\Diamond (c \wedge \neg a)} & \mapsto t_c, t_{bc} \\
m_{\Diamond (c \wedge \neg b)} & \mapsto t_c, t_{ac} \\
m_{\Diamond (c \wedge \neg a \wedge \neg b)} & \mapsto t_c \\
m_{\Diamond ((a \vee b) \wedge \neg c)} & \mapsto t_a, t_b, t_{ab} \\
m_{\Diamond ((a \vee c) \wedge \neg b)} & \mapsto t_a, t_c, t_{ac} \\
m_{\Diamond ((b \vee c) \wedge \neg a)} & \mapsto t_b, t_c, t_{bc}
\end{array} \right\}
\qquad
S_1 = \left\{ \begin{array}{ll}
t_a & \mapsto m_{\Diamond (a \wedge \neg b \wedge \neg c)} \\
t_b & \mapsto m_{\Diamond (b \wedge \neg a \wedge \neg c)} \\
t_c & \mapsto m_{\Diamond (c \wedge \neg a \wedge \neg b)} \\
t_{ab} & \mapsto m_{\Diamond (a \wedge \neg c)}, m_{\Diamond (b \wedge \neg c)} \\
t_{ac} & \mapsto m_{\Diamond (a \wedge \neg b)}, m_{\Diamond (c \wedge \neg b)} \\
t_{bc} & \mapsto m_{\Diamond (b \wedge \neg a)}, m_{\Diamond (c \wedge \neg a)} \\
t_{abc} & \mapsto m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond c}
\end{array} \right\}$$

$$R_2 = \left\{ \begin{array}{ll} m_{\Diamond a} & \mapsto t_{abc} \\ m_{\Diamond b} & \mapsto t_{abc} \\ m_{\Diamond c} & \mapsto t_{abc} \\ m_{\Diamond (a \vee b)} & \mapsto t_a, t_b, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (a \vee c)} & \mapsto t_a, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (b \vee c)} & \mapsto t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\ m_{\Diamond (a \wedge \neg b)} & \mapsto t_{ac} \\ m_{\Diamond (a \wedge \neg c)} & \mapsto t_{ab} \\ m_{\Diamond (a \wedge \neg b \wedge \neg c)} & \mapsto t_a \\ m_{\Diamond (b \wedge \neg a)} & \mapsto t_{bc} \\ m_{\Diamond (b \wedge \neg c)} & \mapsto t_{ab} \\ m_{\Diamond (b \wedge \neg a \wedge \neg c)} & \mapsto t_b \\ m_{\Diamond (c \wedge \neg a)} & \mapsto t_{bc} \\ m_{\Diamond (c \wedge \neg b)} & \mapsto t_{ac} \\ m_{\Diamond (c \wedge \neg a \wedge \neg b)} & \mapsto t_c \\ m_{\Diamond ((a \vee b) \wedge \neg c)} & \mapsto t_a, t_b, t_{ab} \\ m_{\Diamond ((a \vee c) \wedge \neg b)} & \mapsto t_a, t_c, t_{ac} \\ m_{\Diamond ((b \vee c) \wedge \neg a)} & \mapsto t_b, t_c, t_{bc} \end{array} \right\} \qquad S_3 = \left\{ \begin{array}{ll} t_a & \mapsto m_{\Diamond (a \wedge \neg b \wedge \neg c)} \\ t_b & \mapsto m_{\Diamond (b \wedge \neg a \wedge \neg c)} \\ t_c & \mapsto m_{\Diamond (c \wedge \neg a \wedge \neg b)} \\ t_{ab} & \mapsto m_{\Diamond (a \wedge \neg c)}, m_{\Diamond (b \wedge \neg c)} \\ t_{ac} & \mapsto m_{\Diamond (a \wedge \neg b)}, m_{\Diamond (c \wedge \neg b)} \\ t_{bc} & \mapsto m_{\Diamond (b \wedge \neg a)}, m_{\Diamond (c \wedge \neg a)} \\ t_{abc} & \mapsto m_{\Diamond a}, m_{\Diamond b}, m_{\Diamond c} \end{array} \right\}$$

When we evaluate this we see that it does perform well, in every state the correct state is deduced, although not intuitively to humans. Here, t_{abc} can send the messages $m_{\Diamond a}$, $m_{\Diamond b}$, and $m_{\Diamond c}$. Additionally, the state t_{ab} sends the messages $m_{\Diamond (a \wedge \neg c)}$ and $m_{\Diamond (b \wedge \neg c)}$ although humans would sooner use message $m_{\Diamond ((a \vee b) \wedge \neg c)}$.

B. IBR with implied negation with four disjuncts

$$R_0 = \left\{ \begin{array}{ll}
m_{\Diamond a} & \mapsto t_a \\
m_{\Diamond b} & \mapsto t_b \\
m_{\Diamond c} & \mapsto t_c \\
m_{\Diamond d} & \mapsto t_d \\
m_{\Diamond (a \vee b)} & \mapsto t_a, t_b, t_{ab} \\
m_{\Diamond (a \vee c)} & \mapsto t_a, t_c, t_{ac} \\
m_{\Diamond (a \vee d)} & \mapsto t_a, t_d, t_{ad} \\
m_{\Diamond (b \vee c)} & \mapsto t_b, t_c, t_{bc} \\
m_{\Diamond (b \vee d)} & \mapsto t_b, t_d, t_{bd} \\
m_{\Diamond (c \vee d)} & \mapsto t_c, t_d, t_{cd} \\
m_{\Diamond (a \vee b \vee c)} & \mapsto t_a, t_b, t_c, t_{ab}, t_{ac}, t_{bc}, t_{abc} \\
m_{\Diamond (a \vee b \vee d)} & \mapsto t_a, t_b, t_d, t_{ab}, t_{ad}, t_{bd}, t_{abd} \\
m_{\Diamond (a \vee c \vee d)} & \mapsto t_a, t_c, t_d, t_{ac}, t_{ad}, t_{cd}, t_{acd} \\
m_{\Diamond (b \vee c \vee d)} & \mapsto t_b, t_c, t_d, t_{bc}, t_{bd}, t_{cd}, t_{bcd} \\
m_{\Diamond (a \vee b \vee c \vee d)} & \mapsto t_a, t_b, t_c, t_d, t_{ab}, t_{ac}, t_{ad}, t_{bc}, t_{bd}, \\
 & \quad t_{cd}, t_{abc}, t_{abd}, t_{acd}, t_{bcd}, t_{abcd}
\end{array} \right\}
\qquad
S_1 = \left\{ \begin{array}{ll}
t_a & \mapsto m_{\Diamond a} \\
t_b & \mapsto m_{\Diamond b} \\
t_c & \mapsto m_{\Diamond c} \\
t_d & \mapsto m_{\Diamond d} \\
t_{ab} & \mapsto m_{\Diamond (a \vee b)} \\
t_{ac} & \mapsto m_{\Diamond (a \vee c)} \\
t_{ad} & \mapsto m_{\Diamond (a \vee d)} \\
t_{bc} & \mapsto m_{\Diamond (b \vee c)} \\
t_{bd} & \mapsto m_{\Diamond (b \vee d)} \\
t_{cd} & \mapsto m_{\Diamond (c \vee d)} \\
t_{abc} & \mapsto m_{\Diamond (a \vee b \vee c)} \\
t_{abd} & \mapsto m_{\Diamond (a \vee b \vee d)} \\
t_{acd} & \mapsto m_{\Diamond (a \vee c \vee d)} \\
t_{bcd} & \mapsto m_{\Diamond (b \vee c \vee d)} \\
t_{abcd} & \mapsto m_{\Diamond (a \vee b \vee c \vee d)}
\end{array} \right\}$$

$$R_2 = \left\{\begin{array}{ll} m_{\Diamond a} & \mapsto t_a \\ m_{\Diamond b} & \mapsto t_b \\ m_{\Diamond c} & \mapsto t_c \\ m_{\Diamond d} & \mapsto t_d \\ m_{\Diamond(a\vee b)} & \mapsto t_{ab} \\ m_{\Diamond(a\vee c)} & \mapsto t_{ac} \\ m_{\Diamond(a\vee d)} & \mapsto t_{ad} \\ m_{\Diamond(b\vee c)} & \mapsto t_{bc} \\ m_{\Diamond(b\vee d)} & \mapsto t_{bd} \\ m_{\Diamond(c\vee d)} & \mapsto t_{cd} \\ m_{\Diamond(a\vee b\vee c)} & \mapsto t_{abc} \\ m_{\Diamond(a\vee b\vee d)} & \mapsto t_{abd} \\ m_{\Diamond(a\vee c\vee d)} & \mapsto t_{acd} \\ m_{\Diamond(b\vee c\vee d)} & \mapsto t_{bcd} \\ m_{\Diamond(a\vee b\vee c\vee d)} & \mapsto t_{abcd} \end{array}\right\} \qquad S_3 = \left\{\begin{array}{ll} t_a & \mapsto m_{\Diamond a} \\ t_b & \mapsto m_{\Diamond b} \\ t_c & \mapsto m_{\Diamond c} \\ t_d & \mapsto m_{\Diamond d} \\ t_{ab} & \mapsto m_{\Diamond(a\vee b)} \\ t_{ac} & \mapsto m_{\Diamond(a\vee c)} \\ t_{ad} & \mapsto m_{\Diamond(a\vee d)} \\ t_{bc} & \mapsto m_{\Diamond(b\vee c)} \\ t_{bd} & \mapsto m_{\Diamond(b\vee d)} \\ t_{cd} & \mapsto m_{\Diamond(c\vee d)} \\ t_{abc} & \mapsto m_{\Diamond(a\vee b\vee c)} \\ t_{abd} & \mapsto m_{\Diamond(a\vee b\vee d)} \\ t_{acd} & \mapsto m_{\Diamond(a\vee c\vee d)} \\ t_{bcd} & \mapsto m_{\Diamond(b\vee c\vee d)} \\ t_{abcd} & \mapsto m_{\Diamond(a\vee b\vee c\vee d)} \end{array}\right\}$$

When we evaluate R_2 and S_3 we find that the interpretations are perfectly in line with the human equivalents.