The Revelation Principle for Mechanism Design with Signaling Costs

by

Andrew Kephart

Department of Computer Science

Duke University

Date: _____

Approved:

_____

Vincent Conitzer, Supervisor

_____

Ronald Parr

_____

Saša Pekeč

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in the Department of Computer Science

in the Graduate School of Duke University

2017

Abstract

The *revelation principle* is a key tool in mechanism design. It allows the designer to restrict attention to the class of truthful mechanisms, greatly facilitating analysis. This is also borne out in an algorithmic sense, allowing certain computational problems in mechanism design to be solved in polynomial time. Unfortunately, when not every type can misreport every other type (the *partial verification* model), or—more generally—misreporting can be costly, the revelation principle can fail to hold. This also leads to NP-hardness results.

The primary contribution of this work consists of characterizations of conditions under which the revelation principle still holds when reporting can be costly. (These are generalizations of conditions given earlier for the partial verification case) In fact, our results extend to cases where, instead of being able to report types directly, agents may be limited to sending signals that do not directly correspond to types. In this case, we obtain conditions for when the mechanism designer can restrict attention to a given (but arbitrary) mapping from types to signals without loss of generality. We also study associated computational problems.

Contents

Abstract **iii**

List of Abbreviations and Symbols **vii**

Acknowledgements **viii**

1 Introduction **1**

1.1 Related Work 4

1.2 Introductory Example - Inspection Game 7

1.3 Model 9

1.4 Revelation Principle 11

1.5 Running Example: Stocking Food Banks 13

2 Results **17**

2.1 Intuition Behind Main Results 17

2.2 Summary of Main Results 20

3 Variable Transfers, Variable Utilities **21**

3.1 Relation to Partial Verification 26

3.2 Revisiting the Running Example 28

4 Variable Transfers, Fixed Utilities **29**

4.1 Revisiting the Running Example 32

5 Fixed Transfers, Variable Utilities **34**

5.1 Special Case: No Transfers To The Agent 37
5.2 Special Case: No Transfers At All, Type-Reporting Setting 38
5.3 Special Case: No Transfers At All, Signaling Setting 39
5.4 Revisiting the Running Example 39

6 Fixed Transfers, Fixed Utilities 41

6.1 Revisiting the Running Example 44

7 Revelation Principle for Known Valuations 46

8 Known Valuations, Variable Transfers, Variable Utilities 47

9 Known Valuations, Variable Transfers, Fixed Utilities 48

10 Known Valuations, Fixed Transfers, Variable Utilities, Type-Reporting Setting 49

11 Known Valuations, Fixed Transfers, Variable Utilities, General Signaling Setting 53

11.0.1 Example: Difference in Revelation Principles for Known and Unknown Valuations 53

11.1 KFTVU Condition 55

11.1.1 Example: Difference in Revelation Principles for Fixed Transfers and No Transfers to the Agent 57

11.2 Special Case: No Transfers To The Agent 58

11.2.1 Example: Difference in Revelation Principles for No Transfers to the Agent and No Transfers At All 61

11.3 Special Case: No Transfers At All, Type-Reporting Setting 62

11.4 Revisiting the Running Example 64

11.5 Special Case: No Transfers At All, Signaling Setting 64

12 Known Valuations, Fixed Transfers, Fixed Utilities 65

13 Revelation Principle for Fully Specified Instances 66

14 Conclusions 71

Bibliography **73**

List of Abbreviations and Symbols

Abbreviations

RP	Revelation Principle
VTVU	Variable Transfers, Variable Utilities
TRVTVU	Type Reporting, Variable Transfers, Variable Utilities
VTFU	Variable Transfers, Fixed Utilities
FTVU	Fixed Transfers, Variable Utilities
FTFU	Fixed Transfers, Fixed Utilities
KFTVU	Known Valuations, Fixed Transfers, Variable Utilities
KNTGVU	Known Valuations, No Transfers to the Agent, Variable Utilities
KNTAAVU	Known Valuations, No Transfers At All, Variable Utilities

Acknowledgements

We are thankful for support from ARO under grants W911NF-12-1-0550 and W911NF-11-1-0332, NSF under awards IIS-1527434, IIS-0953756, CCF-1101659, and CCF-1337215, and a Guggenheim Fellowship. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

1

Introduction

Mechanism design concerns making decisions based on the private information of one or more agents, who will not report this information truthfully if they do not see this to be in their interest. The goal is to define a *mechanism*—a game to be played by the agent(s)—that defines actions for the agents to take and maps these actions to outcomes, such that in equilibrium, the agents' true types map to outcomes as desired. This may, at first, appear to leave us in the unmanageable situation of having to search through all possible games we could define. Fortunately, there is the *revelation principle*, which states that anything that can be implemented by a mechanism can also be implemented by a *truthful* mechanism, where agents in equilibrium report their types truthfully. This vastly reduces the search space, and indeed under certain circumstances allows optimal mechanisms to be found in polynomial time (e.g., if the number of agents is constant and randomized outcomes are allowed [Conitzer and Sandholm, 2002, 2004]). And of course, the revelation principle is also extremely useful in the process of obtaining analytical results in mechanism design.

Unfortunately[1], there are situations where the revelation principle fails to hold. Notably, this is the case in settings with so-called *partial verification*.[2] This means that not every type is able to report every other type (without detection). For example, an agent in an online marketplace may be able to pretend to arrive later than she really does, but not earlier. Indeed, due to this absence of a revelation principle in this context, the problem of determining whether a particular choice function can be implemented becomes, in general, NP-hard [Auletta et al., 2011]. It is thus natural to seek to characterize those conditions on the misreporting graph—under which represents which types can misreport which other types—the revelation principle holds. Such a characterization can help us obtain analytical results in mechanism design and algorithms for efficiently finding mechanisms. Indeed, these conditions have been previously characterized [Green and Laffont, 1986; Yu, 2011]; we review them later as they come up as special cases of our characterizations.

In practice, it is not always black and white whether one type can misreport another. Often, one type can misreport another *at some cost*. We call this *mechanism design with reporting costs*. Reporting costs may correspond to financial expense or to expended effort. For example, a minor may acquire a fake driver's license at some price. A consumer may engage in tricks to improve his credit score. In college admissions, a student may improve on his/her "natural" SAT score by extra prepping for the test.[3] Generally, for every type $\theta \in \Theta$ and every report $\hat{\theta} \in \Theta$ that the agent might report, there is a non-negative cost $c(\theta, \hat{\theta})$ for doing so (with $c(\theta, \theta) = 0$). Traditional mechanism design is the special case where $c(\theta, \hat{\theta}) = 0$ everywhere; partial

[1] This is only unfortunate from analytical and algorithmic viewpoints. Indeed, often a non-truthful mechanism in these settings will perform better than any truthful one. We show several examples of this.

[2] This setting is also known as that of *hard evidence*.

[3] We have in mind here not prepping that has value beyond the test—e.g., studying algebra in order to be able to solve more problems—but rather acquiring tricks—e.g., about how best to guess when unsure—that improve the score on the test but are otherwise of no societal value.

verification is the special case where $c(\theta, \hat{\theta}) \in \{0, \infty\}$ and $c(\theta, \theta) = 0$. Because partial verification is a special case, it immediately follows that the revelation principle does not hold *in general* with costly reporting. However, in this work, we identify necessary and sufficient conditions for the revelation principle to still hold, generalizing the earlier conditions for the partial revelation case [Green and Laffont, 1986; Yu, 2011].

In fact, we present our results for a more general setting,[4] namely the one where agents may be restricted to send (costly) signals that do not necessarily correspond directly to types. We call this *mechanism design with signaling costs*. Here we say an agent *emits* a signal, rather than *reports* a type. This puts us in the domain of signaling games. Consider, for example Spence's famous model of education [Spence, 1973]. In this model, agents signal their type (capabilities) by attaining a certain level of education; the idea is that higher-ability agents can more easily attain a higher level of education, so that in equilibrium the different types separate and employers take the level of education into account in hiring decisions. In this case, there is no *ex-ante* correspondence between types and signals.

Now consider an employer who can commit to a mapping from signals (levels of education) to hiring decisions. This employer then is in a position to design a mechanism, but cannot restrict attention to "truthful" mechanisms in a straightforward sense, because it is not clear what reporting truthfully would mean. It would be very helpful to the employer to know a mapping from types to signals (that the agent would be incentivized to follow) with the following property: if there is *any* such mapping that suffices for the employer's objective, then so does this one. In a standard mechanism design context, that special mapping is the truthful one. In this case, it is not clear which mapping, if any, fits the bill. But, for any given mapping, we provide necessary and sufficient conditions that this mapping needs to

[4] The EC'16 version of this work did not consider this more general setting.

satisfy to have the desired property. This, of course, generalizes the case discussed before where signals do correspond to types and reporting truthfully is costless; in this case, the mapping of interest is the truthful one.

In Sections 7 and 13, we also consider cases where some aspects beyond the signaling costs, such as the valuation function and choice function, are known.

1.1 Related Work

In earlier work, we studied the computational complexity of deciding whether a given choice function can be implemented when misreporting is costly [Kephart and Conitzer, 2015]; these results will be relevant in Section 13 of this paper.

Several other papers study the revelation principle in the setting of partial verification, including the more general variant of that where the signal space is not identical to the type space (but we still have $c(\theta, s) \in \{0, \infty\}$ for all θ, s). Bull and Watson [2007] and Deneckere and Severinov [2008] consider conditions which Koessler and Perez-Richet [2014] characterize as *normality*. When these conditions hold, each type has some "maximal evidence" it can emit, which does as much to distinguish it from other types as any other signal it can emit. Thus, we can restrict attention to mechanisms where each type emits its maximal evidence. In Bull and Watson [2007] this corresponds to a specific signal for each type, while in Deneckere and Severinov [2008] it corresponds to each type being able to emit all its signals simultaneously.

When normality does not hold, both Bull and Watson [2007] and Deneckere and Severinov [2008] show that it is sufficient to consider dynamic mechanisms of the following form. First, the agent emits a costless cheap talk signal announcing her type. Then, the center requests she follow up with a 'verifying' signal or signals which

are the same as those she would emit in the non-truthful implementation. If the agent does not emit the verifying signal, she is allocated a punishment outcome. Building on these ideas, Strausz [2016] proposes an alternate interpretation of the revelation principle. If we consider the emission of the verifying signal to be part of outcome imposed by the center, then the only 'reporting' that the agent is really doing is emitting her cheap talk type signal. Hence, in a sense, the revelation principle in standard form still holds.

In our setting with costs, the normality condition is not as natural. One might assume that for any two "basic" signals s_1 and s_2, there exists another signal $s_1 \cup s_2$ such that for any θ, we have $c(\theta, s_1 \cup s_2) = c(\theta, s_1) + c(\theta, s_2)$. This is not always reasonable, but even if it does hold, we generally cannot restrict attention to mechanisms where each type emits its maximal evidence signal. The combined cost of all these signals may be so large that the agent would rather stay home or pretend to be an incompetent type that cannot emit any signals. (Of course this can also happen even if the maximal evidence signal is not the union of a number of basic signals.)

The second approach, in which a signal is requested from an agent who has sent a cheap-talk message about her type, works in our context. Specifically, given any mechanism at all that involves signaling, we can turn it into a mechanism where first the agent is requested to report a type in a cheap-talk way, and then, based on this report, she is requested to signal as she would have signaled in the original mechanism. This shows that, in a particular weak sense, the revelation principle does hold here. Unfortunately, this is of little use in our context to the designer in the search through the space of mechanisms, because it does not help in determining which signal each type should eventually (after cheap talk) emit.[5] And that search

[5] It is somewhat more helpful when a non-truthful mechanism might involve multiple rounds of signaling as is the case in Bull and Watson [2007] and Deneckere and Severinov [2008].

corresponds to an NP-hard problem [Auletta et al., 2011]. In contrast, if we know which signal each type is supposed to emit, the problem becomes easy. This is what the revelation principle does for us in the traditional mechanism design setting, and what we would like any revelation principle to reproduce here.[6]

A variety of papers consider revelation type principles in more limited partial verification or costly signaling settings. These include: Lacker and Weinberg [1989] for contracts in an exchange economy; Bull and Watson [2004] for enforcement of contract disputes; Kartik et al. [2014] for when agents have preferences for honesty; Koessler and Perez-Richet [2014] for when each type has a signal which can be used to distinguish it from all other types. Deneckere and Severinov [2014] shows that when costs are monotonically increasing with a signal's distance from 'truth', increasing the number of signals available for agents to use expands the set of implementable choice functions. Additionally, they characterize the optimal mechanism for a setting with a one-dimensional type space.

Other papers that explore implementability of choice functions in costly signaling-like settings, but do not derive revelation principles. Kartik and Tercieux [2012] gives a necessary condition, *evidence-monotonicity*, which is required for a choice function to be implementable with no signaling costs incurred in equilibrium. Caragiannis et al. [2012] characterizes truthfully implementable choice functions in the setting of *probabilistic verification*, in which there is a certain *probability* that a lying agent will be caught.

We also consider our work related to machine learning in contexts where what is being classified is an agent that may put in some effort to resist accurate classification. The most obvious version of this is *adversarial classification*: detecting spam, intrusions,

[6] One caveat is that, in the general signaling setting, we do not provide an algorithm for finding the without-loss-of-generality mapping from types to signals.

fraud, etc. when the perpetrators wish to evade detection [Dalvi et al., 2004; Barreno et al., 2010]. However, as the examples in the introduction indicate, there are many situations which are not zero-sum. This is also brought out in more recent work on "strategic classification," [Hardt et al., 2016] which has a heavier focus on the machine learning aspect.

1.2 Introductory Example - Inspection Game

The "Inspection Game" example illustrates our model as well as the failure, in general, of the revelation principle in the mechanism design with reporting costs setting.[7] We will introduce another, more complex example in Section 1.5.

Suppose Beth is considering buying a crate of produce from Pam. The produce can be fresh, decent, or rotten (and Pam will know which it is). Beth can either accept or reject the crate. If the produce is fresh or decent, Beth would like to accept it, otherwise she would like to reject it. This gives:

$\Theta = \{fresh,\ decent,\ rotten\}$ — the set of types.
$O = \{accept,\ reject\}$ — the set of outcomes.
$F = \{fresh \to accept,\ decent \to accept,\ rotten \to reject\}$ — the choice function Beth seeks to implement.

Before making her decision, Beth can inspect the appearance of the produce. However, at some cost Pam can add dyes or scents to the produce, which will alter how it appears to Beth. Thus we also have:
$S = \{\hat{fresh},\ \hat{decent},\ \hat{rotten}\}$ – the set of signals.

[7] We present a similar example in Kephart and Conitzer [2015]. This is the smallest example where a choice function is implemented by some non-truthful mechanism, but not by any truthful one.

Since we are in the reporting costs setting, this set is the same as the set of types. We add the " ˆ " symbol to each signal to distinguish it from its corresponding type. The following matrix gives the cost of making a crate of type θ give off the signal s. For example, at a cost of 30, Pam can make a crate of rotten produce appear fresh. Since we are in the reporting costs setting, we must have $c(\theta, \hat{\theta}) = 0$

$c(\theta, s) =$

	$\hat{fresh}$	$\hat{decent}$	$\hat{rotten}$
fresh	0	0	0
decent	10	0	0
rotten	30	10	0

The value that Pam receives if the crate is accepted is 20:[8]

$v_\theta(accept) = 20$

$v_\theta(reject) = 0$

Beth needs to commit to a mechanism for choosing an outcome based on how the produce appears. The naïve mechanism of accepting the produce whenever it does not appear rotten, $H = \{\hat{fresh} \to accept, \hat{decent} \to accept, \hat{rotten} \to reject\}$ would fail. It would be worth it to Pam to pay the cost of 10 to make rotten produce appear to be decent, netting 10 value. Hence Beth would end up inadvertently accepting rotten produce.

What Beth should do instead is use $N = \{\hat{fresh} \to accept, \hat{decent} \to reject, \hat{rotten} \to reject\}$. Under N, if the produce really is rotten it will not be worth it for Pam to alter it, and it will be rejected. On the other hand, if the produce is decent, Pam will make it appear to be fresh and it will be accepted, resulting in the implementation of Beth's desired choice function.

[8] In general, the value of each outcome to the agent can depend on the type.

This example shows that the revelation principle does not hold in this case. There exists a set of outcomes, choice function, and valuation function (namely those given here) such that there exists a mechanism (namely N) that implements the choice function. But, there exists no mechanism (H would be our best candidate) that both incentivizes the agent to report truthfully and implements the choice function.

This example had the property that signals are type reports, and reporting one's type is costless. In the more general setting where types emit signals that do not directly correspond to the types, it is not immediately clear what truthful reporting means; in this context, we will be interested in whether we can, without loss of generality, restrict our attention to some mapping G from types to signals. In the example above, the mapping of interest was $G = \{\textit{fresh} \to \hat{\textit{fresh}},\ \textit{decent} \to \hat{\textit{decent}},\ \textit{rotten} \to \hat{\textit{rotten}}\}$, and we showed that we cannot restrict attention to it without loss of generality.

1.3 Model

As is common in this type of setting, we focus on the case of a single *signal-emitting* agent; this corresponds to holding the other agents' signals fixed.

In a fully specified instance, we have a set of *types* Θ; we will generally use $\theta, \theta_1, \theta_2, \ldots$ to denote variable types and $a, b, c, \ldots$ to denote specific types. We have a set of *signals* S; with $s, s_1, s_2 \ldots$ denoting variable signals and x, y, z denoting specific signals. (When agents report types directly, we have $S = \Theta$.)

We then have the *revelation principle mapping* $G : \Theta \to S$ which designates a signal for each type. The question is whether the designer can restrict attention, without loss of generality, to mechanisms in which each type θ emits signal $G(\theta)$.

(When agents report types directly, the mapping G of interest is always the identity function, corresponding to truthful reporting.)

We assume throughout that G maps each type to a unique signal which has finite cost for that type. This automatically holds in the type-reporting case. In the general signaling model, it is not entirely without loss of generality,[9] but if G maps two types to the same signal then they can never receive distinct outcomes.

This assumption allows us to overload our notation by using $\theta, \theta_1, \ldots, a, b, \ldots$ also to indicate the designated signal of the type with the same name. That is, we use a shorthand where θ refers both to the type θ and the signal $G(\theta)$. This is natural in the type-reporting setting and remains convenient in the general signaling setting. Because of this, we will rarely mention G explicitly (it is generally held fixed), but G is implicit in how the signals are named.

We also have a set of *outcomes* (alternatives) O. There is a *valuation function* $v : \Theta \times O \to \mathbb{R}$, where $v_\theta(o)$ is the valuation that type θ has for outcome o.

Finally, there is a *cost function* $c : \Theta \times S \to \mathbb{R}_{\geqslant 0}$, where $c(\theta, s)$ denotes the cost type θ incurs when emitting s. We often use the shorthand ax for $c(a, x)$. Combining this shorthand with that for G, we will often use ab to mean $c(a, G(b))$.

A *mechanism* is defined by, first, an *allocation function* $A : S \to O$, where $A(s) = o$ denotes that the mechanism chooses outcome o when signal s is emitted. When we allow for transfers, then another part of the mechanism is the *transfer function* $T : S \to \mathbb{R}$, where $T(s)$ denotes the transfer (payment) *received* by the agent when emitting s. (Hence, $T(s) < 0$ implies the agent is making a transfer.) The agent's

[9] For example, consider a setting with two types and only one possible signal. Here, there is only one possible mapping G and so, technically, the revelation principle holds for this mapping, even though it maps two types to the same signal.

utility for having type θ, emitting signal s, and receiving outcome o and transfer t is $u(\theta, s, o, t) = v_\theta(o) - c(\theta, s) + t$.

Let $R : \Theta \to S$ denote a *response* for the agent to the mechanism, where $R(\theta) = s$ denotes that the agent emits s when her true type is θ. We say R is *optimal* for mechanism $M = (A, T)$ if for all θ and s, $u(\theta, R(\theta), A(R(\theta)), T(R(\theta))) \geqslant u(\theta, s, A(s), T(s))$.

We are generally interested in *implementing* a *choice function* $F : \Theta \to O$. We sometimes use the shorthand o_a for $F(a)$. A mechanism $M = (A, T)$ together with response R *implements* F if R is optimal for M and for all θ, $A(R(\theta)) = F(\theta)$. (Moreover, it implements it with transfer $T(R(\theta))$ and utility $u(\theta, R(\theta), A(R(\theta)), T(R(\theta)))$ for type θ.)

We say a mechanism is *truthful* if all types emit in accordance with G, and *non-truthful* otherwise. Similarly, we say a type θ *misemits* if it emits a signal other than $G(\theta)$.

We will sometimes use N to denote a not-necessarily truthful mechanism and H to denote a truthful one.[10] Additionally, R_N and R_H will refer to optimal responses for the respective mechanisms.

1.4 Revelation Principle

We say that the revelation principle (RP) holds on a mapping G whenever we can, without loss of generality, restrict our attention to truthful mechanisms when trying to implement the choice function.

[10] Here, H stands for "honest" to prevent confusion with the transfer function T.

Thus, given Θ, S, and c (we will refer to a combination of Θ, S, and c as an *instance*), the *revelation principle holds on* G if for any O, v, $N = (A_N, T_N)$, and R_N which is optimal for N, there exists another mechanism $H = (A_H, T_H)$ such that the truthful-emitting response R_H (with $R_H(\theta) = \theta$) is optimal for H, and for all θ, $A_H(\theta) = A_N(R_N(\theta))$. (Note that here we are using the shorthand described earlier, using θ to refer to $G(\theta)$, so that $A(\theta) = A(G(\theta)) = A(R_H(\theta))$.) Hence, any choice function F that can be implemented can be implemented truthfully.

Sometimes, we will also wish to implement the choice function with specific transfers and/or utilities. The revelation principle holds with *fixed transfers* (to the agent) if the mechanism H can always be chosen such that $T_H(\theta) = T_N(R_N(\theta))$ for all θ, and with *fixed utilities* if the mechanism H can always be chosen such that $u(\theta, \theta, A_H(\theta), T_H(\theta)) = u(\theta, R_N(\theta), A_N(R_N(\theta)), T_N(R_N(\theta)))$ for all θ. If transfers or utilities are not fixed, we say they are *variable*.

Note that if the revelation principle holds with fixed transfers, then it also holds when we wish to have *no transfers to the agent* (but we do allow negative transfers to signals the agent does not emit). This is simply the special case where transfers to the agent are fixed to 0. This does not automatically imply that the revelation principle holds when we have *no transfers at all*, i.e., $T(\cdot) = 0$. The intuition for why there is a difference is that there may be some signals that we would prefer the agent never to use, but we cannot prevent the agent from doing so because we do not have sufficiently unappealing outcomes available.

But, as we will show, in the type-reporting setting, with unknown valuations the revelation principle is the same, and with known valuations it is essentially the same. Finding useful conditions that ensure the revelation principle holds with no transfers at all when we are in the signaling setting is currently an open problem.

We will use acronyms to refer to our various revelation principles. The possible words used to describe a revelation principle and the letter(s) used to represent the words are as follows: **F**ixed; **V**ariable; **T**ransfers; **U**tilities; **N**o **T**ransfers (to the) a**G**ent; **N**o **T**ransfers **A**t **A**ll; **K**nown (valuations). For instance, *FTFU* means fixed transfers and fixed utilities; *KNTGVU* means known valuations, no transfers to the agent, and variable utilities.

1.5 Running Example: Stocking Food Banks

We will use a running example of stocking food banks to help illustrate the various instantiations of our framework. This example is type-reporting, in the sense that the signal space and the type space coincide. (This example is purely for illustration purposes.) Imagine that a city has four districts: *North*, *East*, *South*, *West*, each of which has a food bank and a population of people living in it. A person's type thus consists of the location in which she lives.

Based on demographics and health conditions, the city government has determined how much it values the population of each district receiving certain types of food. For example, it may wish to distribute milk in a district with many families with young children, and vegetables in a district with many single, middle-aged people. Note that the city's objective here is different from maximizing the sum of the utilities of the people who will make use of the food bank. For example, the middle-aged people may want to consume tasty food, whereas the city may prefer for them to eat healthy food in order to reduce the burden on the health care system. In this example, we assume there are three food types, (those high in) *fiber*, *protein*, and *vitamins*.

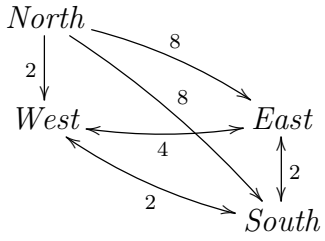
Determining which food to stock in each bank would be straightforward except that there is a possibility that the population from one district might travel to another district if it prefers the latter district's food. This would correspond to them "lying" about their district. We assume here that the food banks cannot check the home address of people arriving at them.

That being said, such "misreporting" is not without cost, because it requires traveling to another district. The transportation costs to residents for traveling between districts are summarized as follows:

$c(\theta, \hat{\theta}) =$

	North	*West*	*East*	*South*
North	0	2	8	8
West	∞	0	4	2
East	∞	4	0	2
South	∞	2	2	0

It can also be useful to visualize this reporting cost structure as a graph. The directed edge between two districts represents the cost of traveling from one to the other. We leave off the zero-cost self reporting edges.[11] All other edges not shown are assumed to have infinite cost.[12]



Suppose that the people of each district value the food as follows.

[11] If we were in the signaling setting these edges might not be zero-cost.

[12] We know this is slightly unrealistic for travel costs, but it helps us better illustrate the revelation principle.

$v_\theta(o) =$

	fiber	*protein*	*vitamins*
North	1	9	1
West	2	3	3
East	1	3	6
South	1	1	1

Moreover, suppose that the city's objective function is as follows.

$J(\theta, o) =$

	fiber	*protein*	*vitamins*
North	10	0	0
West	0	10	5
East	0	10	5
South	0	5	10

The city's problem is to specify a mechanism—that is, which food each district's food bank distributes—such that it is happy with the equilibrium result of this mechanism. Possibly, the city also has the option to distribute a (welfare) transfer to each person who comes to the food bank.

We assume that there are no capacity constraints on any food bank—that is, the food bank does not run out of food if too many people come to it. Hence, this is effectively a mechanism design problem for a single agent—the person turning up at the food bank—because other agents' decisions do not affect this agent.

There are multiple variants of this problem, depending on whether the city can make transfers, whether it wants these transfers to be a certain amount, whether it cares about the transportation costs incurred by people traveling to a different district, etc.

If the revelation principle holds for the variant in question, the city's problem is significantly easier; it can focus on truthful implementations, i.e., mechanisms such that nobody would travel to another district. On the other hand, if it does not hold,

the city needs to consider mechanisms that do incentivize some districts' populations to travel to the food bank in another district, because this may result in better outcomes than any truthful mechanism can achieve. We will see examples of both cases.

2

Results

We now move on to our contributions.

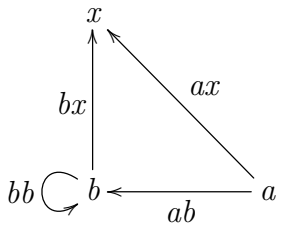
2.1 Intuition Behind Main Results

We first provide a high level overview of the intuition behind our main results.

The revelation principle holds iff for any choice function F implemented by a non-truthful mechanism N, there also exists a truthful mechanism H which implements F. For the purpose of providing intuition, we limit ourselves here to the case where only one type signals non-truthfully in N. Under N let b obtain $F(b)$ by emitting some signal $x \neq b$.

To create H where $A(b) = F(b)$, b must now obtain $F(b)$ by emitting b. This risks that some type a now prefers emitting b to truthfully emitting a.

We can visualize these types and the relevant signaling costs between them as follows[1]:



Note that this way of displaying the costs makes heavy use of the fact that we conflate types and the signals those types emit to, which is natural in the type-reporting case but a bit more subtle in the general signaling case. An edge in this graph corresponds to the type at the beginning of the edge emitting the signal at the end of the edge, and the weight on the edge is the cost for doing so. In general, some signals in the graph may not have a type associated with them; such signals cannot have outgoing edges in the graph. In the type-reporting setting, every vertex has a self-edge with cost 0, but this is not true in the general signaling setting.

There are two ways to use transfers to keep a truthful in H.

First: Pay an agent extra for emitting a. This keeps a honest but may cause other types to misemit to a under H. So we pay off all the types which can emit a too, and so on. This works as long as the process terminates without reaching b. If we had to pay b, the extra transfers would cancel out and all would be for naught. Thus, with variable transfers and variable utilities, if there exists no path of finite emission cost edges along signals in $range(G)$ from b to a, the revelation principle holds. (Note that signals not in $range(G)$ cannot have outgoing edges.)

Second: Pay less to an agent for emitting b. We can safely subtract $bx - bb$ from the transfer that b received for emitting x under N (making b equally happy under

[1] Other types not shown in the graph may exist.

H and N).[2] Note that a did not emit x under N. Consider how much greater a's incentive is to emit b under H than to emit x under N. It is $(ax - ab) - (bx - bb)$, which will be nonpositive iff $ax \leqslant ab + bx - bb$. In this case a still will not misemit. Thus, with variable transfers, if $ax \leqslant ab + bx - bb$ holds, then the revelation principle holds with fixed utilities (and thus with variable utilities as well). Note that in the case where $bb = 0$ this condition is equivalent to the triangle inequality holding on the signaling cost structure.

If we have fixed transfers (but variable utilities), neither of the previous two techniques are allowed. So, a must not want to emit b outright, given it did not want to emit x under N. To ensure this, we must have $ab \geqslant ax$. Additionally, this condition is only necessary when b can actually emit x.[3] So we have $bx < \infty \implies ab \geqslant ax$.

Here, our limited example leaves out an aspect that will affect the general condition. The additional aspect is that a might emit some signal $y \neq a$ in the non-truthful mechanism.[4] Hence an honest emission of a by a would leave it $aa - ay$ worse off in the truthful mechanism, which affects a's incentives to misemit to b. To keep this from happening, (when $bx < \infty$) we now need $ab - (aa - ay) \geqslant ax$, i.e. $ab - aa \geqslant ax - ay$.

Finally, with fixed transfers and fixed utilities, b must receive the same utility in both H and N. This requires that there exists some δ_b such that when $bx \neq \infty$ then $bx = bb = \delta_b$. This then reduces to the partial verification setting and we can use the known revelation principle for that case. In our notation this becomes: $(ab = \delta_a \wedge bx = \delta_b) \implies ax = \delta_a$.

[2] Subtracting any more might cause b to misemit under H.

[3] We don't need to explicitly check this for any of the other cases as bx is already present in the conditions.

[4] This aspect does not play a part in the variable transfers revelation principles as we can compensate a for any change in signaling costs.

As it turns out, our reasoning above captures all the key aspects, so our general results involve the exact same conditions as those presented here (though the proofs are more intricate).

2.2 Summary of Main Results

The revelation principle holds with {variable, fixed} transfers and {variable, fixed} utilities *iff* for all ordered doubles $a \neq b$ of types and all ordered doubles x, y of signals:

	Variable Utilities	Fixed Utilities
Variable Transfers	$ab + bx - bb \geqslant ax$, *or* no b to a path[6]	$ab + bx - bb \geqslant ax$
Fixed Transfers	$bx < \infty \implies ab - aa \geqslant ax - ay$	$bx \in \{\delta_b, \infty\}$, *and* $(ab = \delta_a \wedge bx = \delta_b) \implies ax = \delta_a$

To recover the revelation principles for the type-reporting case set bb, aa, ay[7], δ_a, and δ_b to 0.

[6] Along signaling cost edges between signals in $range(G)$.

[7] $ay = 0$ in the type-reporting case as this is always the minimum possible value for it, which is when $y = a$.

3

Variable Transfers, Variable Utilities

Consider the case where we allow the transfers and utilities agents achieve to vary between non-truthful and truthful implementations. That is, all that is needed for the revelation principle to hold is that for any non-truthful implementation of a choice function, there exists a truthful implementation as well—but the transfers and utilities received by types may be different in the latter.

Definition 1 (VTVU Condition)**.** *An instance satisfies the* VTVU condition *if for all ordered doubles $a \neq b$ of types and every signal x, either:*

(i) There exists no path from b to a of finite signaling cost edges between signals in $range(G)$, or

(ii) $ax \leqslant ab + bx - bb$.

Theorem 2. *The RP holds with variable transfers and variable utilities iff the VTVU condition holds.*

Proof. VTVU condition $\implies$ RP holds. Consider the signaling graph consisting of the vertices in $range(G)$ and the edges with finite signaling costs between signals in $range(G)$. Consider the strongly connected components of this graph; the graph decomposes into a DAG over these components.

Suppose there is a mechanism N that, together with a response R_N, non-truthfully implements a choice function F. Let us restrict our attention to types inside a single component. For these types, for *any* signal (even one outside the component), part (ii) of the VTVU condition must hold since there clearly is a path between every pair of types in the component. Hence, within this component the RP holds even with variable transfers and fixed utilities (Theorem 6), and hence there exists a truthful implementation within this component. That is, every type θ in the component can receive $F(\theta)$ and some payment at θ, while also having no incentive to misemit to any other signal inside the component.

Choose such an internally truthful implementation (including transfers) with fixed utilities within each component. For signals not in any component (i.e., signals not in $range(G)$), let the implementation be the same as N. There will be no incentive to misemit inside each component by internal truthfulness. Nor will there be incentive to misemit to a signal that is outside every component because the utility each agent receives is the same as in N . But, this does not necessarily result in a mechanism that is truthful overall, because there may be incentives to misemit across components. We can fix this as follows.

We order the components in a way consistent with the DAG, such that types in a later component can emit types in an earlier component, but not vice versa. Additionally, for each component we specify an additional transfer that all types in that component will receive. By making this transfer sufficiently larger for later components in this

order, no type will wish to misemit to an earlier component (and misemitting to a later component comes at infinite cost).

(Specifically, if component C_1 comes before C_2, then the additional transfers $\pi^{\text{add}}_{C_1}$ and $\pi^{\text{add}}_{C_2}$ should be such that for all $\theta_1 \in C_1$ and $\theta_2 \in C_2$,

$$v_{\theta_2}(F(\theta_2)) + \pi^{\text{orig}}_{\theta_2} + \pi^{\text{add}}_{C_2} - \theta_2\theta_2 \geqslant v_{\theta_2}(F(\theta_1)) + \pi^{\text{orig}}_{\theta_1} + \pi^{\text{add}}_{C_1} - \theta_2\theta_1$$

$$\Leftrightarrow$$

$$\pi^{\text{add}}_{C_2} - \pi^{\text{add}}_{C_1} \geqslant v_{\theta_2}(F(\theta_1)) + \pi^{\text{orig}}_{\theta_1} - \theta_2\theta_1 - v_{\theta_2}(F(\theta_2)) - \pi^{\text{orig}}_{\theta_2} + \theta_2\theta_2$$

where the π^{orig} are the transfers obtained from applying the VTFU revelation principle within the components.)

Since the additional transfer to a component is the same for all types in it, internal truthfulness is maintained. Since the additional transfers are positive no type will want to emit a signal that is outside every component. So, since no type will misemit within a component, to a signal outside every component, or to another component, the implementation is truthful.

VTVU not holding $\implies$ *RP does not hold.* Given that the VTVU condition does not hold, we must have some $a \neq b$ and x with:

$$ax > ab + bx - bb$$

for which there is a path of types connected by finite-cost edges: $b = \theta_1, \theta_2, \ldots, \theta_{l-1}, \theta_l = a$ (where we consider $l + 1 = 1$).

We first consider the cases where x is equal to a or b:

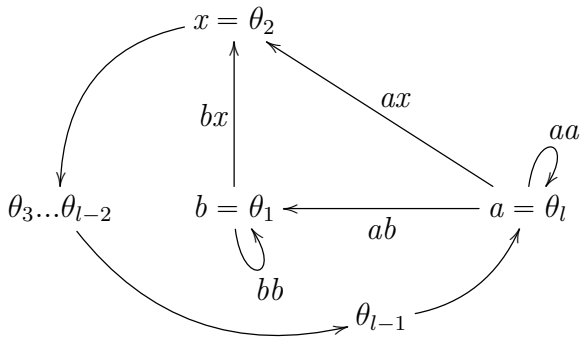
– If $x = b$, then $ab > ab + bb - bb$, which clearly cannot be the case.

– If $x = a$, then $aa > ab + ba - bb$, i.e. $aa - ab > ba - bb$. Consider the choice function $F(\cdot) = o$ for some arbitrary outcome o. F is clearly non-truthfully implementable.

  But F is not truthfully implementable by any mechanism H. To keep a from misemitting to b, we need $T_H(a) - T_H(b) \geqslant aa - ab$. And, to keep b from misemitting to a, we need $ba - bb \geqslant T_H(a) - T_H(b)$. But this leads to a contradiction as $aa - ab > ba - bb$. Hence the revelation principle fails.

So, from now on we assume $x \neq a, b$.

If the path goes through x, we can assume without loss of generality that $x = \theta_2$. We can visualize this as follows.



Define λ s.t. $ax > \lambda > ab + bx - bb$.

Consider an outcome set with a separate outcome o_{θ_i} for every type θ_i (except $o_b = o_x$), and a valuation function with:

$v_{\theta_i}(o_{\theta_i}) = \theta_i\theta_i$

$v_{\theta_i}(o_{\theta_{i+1}}) = \theta_i\theta_{i+1}$

except,

$v_a(o_b) = \lambda$

$v_b(o_x = o_b) = bx$

and $v = 0$ elsewhere.

Consider the mechanism N where $A_N(\theta_i) = o_{\theta_i}$, except $A_N(b) = o_a$. Moreover, let T_N be a large constant value on all the θ_i and x, and 0 everywhere else, so that none of the θ_i would misemit somewhere outside that set.

Then an optimal response has $R_N(\theta_i) = \theta_i$, except $R_N(b) = x$, since:

- For any $\theta_i \notin \{a, b\}$, the only viable alternative is to misemit θ_{i+1}, but the cost of doing so exactly cancels out the benefit.
- For b, emitting x results in utility $T_N + bx - bx = T_N$, which is no worse than emitting truthfully and getting $T_N + v_b(o_a) - bb \leqslant T_N$.
- For a, emitting x would give utility $T_N + \lambda - ax$, which is less than the $T_N + aa - aa$ it gets for emitting truthfully.

Hence, this is a non-truthful implementation of some choice function with $F(\theta_i) = o_{\theta_i}$.

On the other hand, such a choice function cannot be implemented truthfully. This is because we have a Rochet-style negative cycle Rochet [1987]. For truthful emissions we must have $T_H(\theta_i) \geqslant T_H(\theta_{i+1})$ for each θ_i, except:

$$T_H(b) + v_b(o_b) - bb \geqslant T_H(x) + v_b(o_x) - bx, \text{ i.e.}$$

$$T_H(b) + bx - bb \geqslant T_H(x)$$

and

$$\begin{aligned} T_H(a) + v_a(o_a) - aa &\geqslant T_H(b) + v_a(o_b) - ab, \text{ i.e.} \\ T_H(a) &\geqslant T_H(b) + v_a(o_b) - ab, \text{ i.e.} \\ T_H(a) &\geqslant T_H(b) + \lambda - ab \\ &> T_H(b) + (ab + bx - bb) - ab \\ &= T_H(b) + bx - bb \\ &\geqslant T_H(x) \end{aligned}$$

Hence $T_H(a) > T_H(x)$. But this leads to a contradiction as we follow the inequalities around the cycle.

For the case where the path does not go through x, we can make a similar argument, treating x as before but using a separate outcome o_{θ_2} for θ_2 and letting $v_b(o_{\theta_2}) = b\theta_2$. By doing so, b is indifferent between x and θ_2 in the non-truthful implementation which therefore still works[1], and for the cycle we have:

$$\begin{aligned} T_H(b) + v_b(o_b) - bb &\geqslant T_H(\theta_2) + v_b(o_{\theta_2}) - b\theta_2, \text{ i.e.} \\ T_H(b) + bx - bb &\geqslant T_H(\theta_2), \end{aligned}$$

allowing us to have $T_H(a) > T_H(\theta_2)$ and thus we get the same contradiction around the cycle. □

3.1 Relation to Partial Verification

In the partial verification case (type reporting costs are zero or infinity), if we allow transfers and utilities to vary, it is known that the revelation principle is characterized by the following "Strong Decomposability" condition Yu [2011] in the reporting graph (where there is an edge from a to b if and only if a can report b, at zero cost):

[1] So b still emits x.

(1) Every strongly connected component is fully connected, i.e., every type in the component can report every other type.

(2) All types within the same strongly connected component have the same image set, i.e., the set of types they can report is the same.

Definition 3 (Type Reporting VTVU Condition)**.** *An instance satisfies the* type reporting VTVU condition (TRVTVU) *if it is a type reporting instance (i.e., for all* θ*,* $\theta\theta = 0$ *and* $S = \Theta$*), and it satisfies the VTVU condition, which in type-reporting instances comes down to: for all ordered doubles of types* $a \neq b$*, and type* x*, either:*

(i) There exists no path from b *to* a *of finite reporting cost edges, or*

(ii) $ax \leqslant ab + bx$*.*

Proposition 4. *In the partial verification case (reporting costs are zero or infinity), strong decomposability is equivalent to the TRVTVU condition.*

Of course, if both results are correct, this must in fact be the case. Nevertheless, it is instructive to verify it directly.

Proof. The TRVTVU condition $\implies$ *Strong Decomposability.* First, to show (1), assume $a \neq b$, and x are in the same strongly connected component (so in particular, a is reachable from b with edges of zero reporting cost), and that $ab = 0$ and $bx = 0$. Then, by TRVTVU, $ax \leqslant ab + bx = 0$, so $ax = 0$. This implies transitivity, and hence full connectivity, within every strongly connected component.

Second, to show (2), suppose a and b are in the same strongly connected component—so by (1), $ab = 0$—and $bx = 0$. Because a is reachable from b, by TRVTVU for any x, $ax \leqslant ab + bx = 0$, so $ax = 0$. Hence, nodes in a strongly connected component have the same image set.

Strong Decomposability $\implies$ *the TRVTVU condition.* Suppose $ab = bx = 0$ and a is reachable from b with edges of zero reporting cost; it suffices to show that $ax = 0$. But in fact, because a and b are in the same strongly connected component, by (2), $bx = 0 \implies ax = 0$.[2] □

3.2 Revisiting the Running Example

Suppose the city wants to maximize its objective without regard to the transfers it makes or the resulting utilities of the different types of agent. When we consider the conditions for the revelation principle we just obtained, we see that part (ii) of the VTVU condition is violated by the following triples: (*North, West, East*) and (*North, West, South*). However, in both cases there is no path back to *North*.

Hence, in fact the revelation principle holds. Thus, any choice function that is implementable is also implementable truthfully. So the city only needs to search through the space of truthful mechanisms to maximize its objective. In fact, the following truthful mechanism achieves the best conceivable objective value of 40.

$H =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*protein*	*protein*	*vitamins*
T	6	0	1	0

[2] The careful reader may wonder why we did not need to use condition (1) in this part of the proof. This is because condition (1) in Strong Decomposability is in fact redundant—condition (2) implies condition (1).

4

Variable Transfers, Fixed Utilities

We now consider the case where, given a non-truthful mechanism, we wish to obtain a truthful mechanism that implements the same choice function and maintains the utility that each type obtains, but we allow the two mechanisms to be different in the transfers they make to the agents.

Definition 5 (VTFU Condition)**.** *An instance satisfies the* VTFU condition *if for all ordered doubles $a \neq b$ of types, for every signal x,*

$$ax \leqslant ab + bx - bb$$

Note that if $bb > ab + bx$, the VTFU condition cannot hold as ax must be non-negative.

Theorem 6. *The RP holds with variable transfers and fixed utilities iff the VTFU condition holds.*

Proof. VTFU condition $\implies$ RP holds. We will show that for any F, for any non-truthful implementation, we can construct a truthful implementation. Let N together with R_N implement F. Let $H = N$ except:

$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$
$T_H(\theta) = T_N(R_N(\theta)) - \theta R_N(\theta) + \theta\theta$ for $\theta \in \Theta$

If H is truthful, it clearly implements F with fixed utilities. So now we show this is the case.

By the VTFU condition, the optimality of R_N, and the definition of H, the following series of inequalities holds for any $a \neq b$, and $x = R_N(b)$:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa \\
= \; & v_a(o_a) + T_N(R_N(a)) - aR_N(a) \\
\geqslant \; & v_a(o_b) + T_N(x) - ax \\
\geqslant \; & v_a(o_b) + T_N(x) - bx - ab + bb \\
= \; & v_a(o_b) + (T_H(b) + bx - bb) - bx - ab + bb \\
= \; & v_a(o_b) + T_H(b) - ab
\end{aligned}$$

Similarly, for any type a and signal $z \notin range(G)$ we have:

$$\begin{aligned}
& v_a(o_a) + T_H(a) - aa \\
= \; & v_a(o_a) + T_N(R_N(a)) - aR_N(a) \\
\geqslant \; & v_a(A_N(z)) + T_N(z) - az \\
= \; & v_a(A_H(z)) + T_H(z) - az
\end{aligned}$$

This shows that under H, a is willing to emit a over any other signal. Thus, H is truthful.

VTFU condition not holding $\implies$ *RP does not hold.* Choose an instance where the VTFU condition is violated, i.e., there exist $a \neq b \in \Theta$ and $x \in S$ such that:

$$ax > ab + bx - bb$$

By the same reasoning as in this part of the proof of Theorem 2 (variable rather than fixed utilities) we consider the cases where x equals a or b. So, from now on we assume $x \neq a, b$.

Define λ s.t. $ax > \lambda > ab + bx - bb$.

Choose an arbitrary outcome o and let $F(\cdot) = o$. N defined by:

$A_N(\cdot) = o$
$T_N(x) = \lambda$
$T_N(a) = aa$
$T_N(\theta) = -bb$ for $\theta \neq x, a$

implements F and is non-truthful as b will emit x or a to obtain a utility of at least $v_b(o) + \lambda - bx > v_b(o) + ab - bb \geqslant v_b(o) - bb$. And since $ax > \lambda$, a will emit truthfully over emitting x and thus obtain a utility of $v_a(o)$.

Consider any mechanism H where for all θ, $A_H(\theta) = o$, and truthful emission gives it the same utility it achieved under N. Thus:

$T_H(b) \geqslant \lambda - bx + bb$, and
$T_H(a) = aa$

If the revelation principle is to hold, this mechanism must be truthful. But in fact, by emitting b, a can obtain a utility of:

$$\begin{aligned} & v_a(o) + T_H(b) - ab \\ \geqslant\ & v_a(o) + \lambda - bx + bb - ab \\ >\ & v_a(o) \end{aligned}$$

which is what it would get for emitting a. So the revelation principle does not hold. □

4.1 Revisiting the Running Example

The mechanism in 3.2 maximized the city's objective but resulted in uneven utilities for the different types of agent: *North* obtained 7, *West* obtained 3, *East* obtained 4, and *South* only 1.

Suppose now the city would like equal utilities for all types. (Note that, because the city can make arbitrary transfers, any mechanism where all types receive utility u_0 can easily be transformed into another mechanism where all types receive utility u_1 and that implements the same choice function, simply by adding $u_1 - u_0$ to each transfer.) The best truthful mechanism is the following, resulting in an objective of 30 while ensuring each type achieves a utility of exactly 3 (again, this utility can easily be transformed into any other number).

$H =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*vitamins*	*protein*	*protein*
T	2	0	0	2

But might there be a non-truthful mechanism that achieves a higher objective? When we check the VTFU condition, we see that the following triples violate it:

$a = North,\ b = West,\ x = East$

$a = North,\ b = West,\ x = South$

Thus, the revelation principle does not hold. Indeed, in this case there is in fact a better non-truthful mechanism. Consider the following non-truthful mechanism under which *West* travels to the *South* food bank. It results in an objective of 35 while still giving all types utility 3.

$N =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*fiber*	*protein*	*protein*
T	2	0	0	2

5

Fixed Transfers, Variable Utilities

Definition 7 (FTVU Condition)**.** *An instance satisfies the* FTVU condition *if for all ordered doubles $a \neq b$ of types, for all ordered doubles x, y of signals,*

$$bx < \infty \implies ab - aa \geqslant ax - ay$$

Theorem 8. *The RP holds with fixed transfers and variable utilities iff the FTVU condition holds.*

Proof. FTVU condition $\implies$ RP holds. We will show that for any F, for any non-truthful implementation we can construct a truthful implementation. Let N together with R_N implement F. Define $H = N$ except:

$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$

$T_H(\theta) = T_N(R_N(\theta))$ for $\theta \in \Theta$

$T_H(s) = -L$ for some L large enough that no type would ever emit s, for $s \notin range(G)$ [1]

[1] This does not contradict our definition of fixed transfers as the definition allows transfers to signals that no type emits.

If H is truthful, it clearly implements F with fixed transfers. So now we show this is the case.

By the FTVU condition, the definition of H, and the optimality of R_N, the following series of inequalities holds for any pair of types $a \neq b$, and signals $x = R_N(b)$[2], $y = R_N(a)$:

$$\begin{aligned} & v_a(o_a) + T_H(a) - aa \\ = \; & v_a(o_a) + T_N(y) - aa \\ \geqslant \; & v_a(o_a) + T_N(y) - ay + ax - ab \\ \geqslant \; & v_a(o_b) + T_N(x) - ax + ax - ab \\ = \; & v_a(o_b) + T_N(x) - ab \\ = \; & v_a(o_b) + T_H(b) - ab \end{aligned}$$

By the definition of H, for any type a, and signal $z \notin range(G)$ we have:

$$\begin{aligned} & v_a(o_a) + T_H(a) - aa \\ \geqslant \; & v_a(A_H(z)) + T_H(z) - az \end{aligned}$$

This shows that under H, a is willing to emit a over any other signal. Thus, H is truthful.

FTVU condition not holding $\implies$ *RP does not hold.* Let $a \neq b \in \Theta$ and $x, y \in S$ violate the FTVU condition. Thus:

$$bx < \infty, \text{ and } ab - aa < ax - ay$$

Note that in the following proof, unlike those for variable transfers, we do not need to specially address when any of the following equivalencies (or pair of them) hold:

[2] This will imply $bx \neq \infty$.

$b = x$, $x = y$, or $y = a$. (Though if all three hold simultaneously, we have $a = b$, which is ruled out by definition).

We will show that we can define outcomes and valuation functions, as well as a non-truthful mechanism N (with an associated optimal response R_N) without transfers at all[3] that implement a choice function F that no truthful mechanism with fixed transfers would implement. That is, H with $A_H = A_N \circ R_N$ and $T_H(\theta) = 0$ for all θ is not truthful.

Create outcomes o_a, o_b, $\varnothing$ with:

$v_a(o_a) = ay$

$v_a(o_b) = ax$

$v_a(\varnothing) = 0$

$v_b(o_a) = 0$

$v_b(o_b) = bx$

$v_b(\varnothing) = 0$

Consider the mechanism N defined by $T_N = 0$, $A_N(y) = o_a$, $A_N(x) = o_b$, and otherwise $A_N(\cdot) = \varnothing$. Associated with it is some optimal response R_N for which $R_N(a) = y$ and $R_N(b) = x$.

[3] To prove the FTVU it is not necessary that N has no transfers at all. But, we will reuse this part of the proof in to prove Theorems 9 (no transfers to the agent) and 10 (no transfers at all), in which case having this condition is necessary.

However, a mechanism H with $T_H(a) = T_H(b) = 0$, $A_H(a) = o_a$, and $A_H(b) = o_b$ is not truthful. We have:

$$\begin{aligned} & v_a(o_b) - ab \\ = \; & ax \quad - ab \\ > \; & ay \quad - aa \\ = \; & v_a(o_a) - aa \end{aligned}$$

Hence a would prefer to misemit b. □

5.1 Special Case: No Transfers To The Agent

We now consider the special case where we have no transfers to the agent. That is, the agent never receives any transfers (although we may allow transfers to signals no type would emits). Of course, if the FTVU condition holds in general, then the revelation principle will also hold for this special case.

However, we may wonder whether the full FTVU condition is still necessary, or if a more relaxed condition will do. It turns out that the full FTVU condition is still necessary.

Theorem 9. *The RP holds with no transfers to the agent and variable utilities iff the* FTVU condition *holds.*

Proof. FTVU condition $\implies$ *RP holds.* This follows immediately from Theorem 8 as no transfers to the agent is a special case of fixed transfers.

FTVU condition not holding $\implies$ *RP does not hold.* In this part of the proof of Theorem 8 we used a transfer function with no transfers at all, and thus no transfers to the agent. Hence it carries over unchanged. □

5.2 Special Case: No Transfers At All, Type-Reporting Setting

We now consider the special case where we are in a type-reporting setting and have no transfers at all; that is, transfers are fixed at zero.

There is not necessarily any relation between this special case and the case of fixed transfers to the agent,[4] but as it turns out here, the revelation principle will be the same.

Theorem 10. *The RP holds with no transfers at all and variable utilities in the type-reporting setting iff the* FTVU condition *holds.*

Proof. FTVU condition $\implies$ *RP holds.* In this part of the proof of Theorem 8 we showed that if the FTVU condition held: for any choice function F, for any non-truthful mechanism N implementing it, we could create a truthful mechanism H which also implemented F. In particular, we had $H = N$ except:

$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$

$T_H(\theta) = T_N(R_N(\theta))$ for $\theta \in \Theta$

$T_H(s) = -L$ for some L large enough that no type would ever emit s, for $s \notin range(G)$

Consider a N with no transfers at all in a type-reporting setting. The H as defined above is is truthful and implements the same F as N. Since N has no transfers at all, i.e. $T_N(\cdot) = 0$, then we have $T_H(\theta) = 0$ for $\theta \in \Theta$. Since we are in a type reporting setting, $\Theta = S$. Hence H has no transfers at all.

So since H is truthful, implements F, and has no transfers at all, the revelation principle holds.

[4] Indeed, in the setting of known valuations the conditions will turn out to differ.

FTVU condition not holding $\implies$ *RP does not hold.* In this part of the proof of Theorem 8 we used a transfer function with no transfers at all. Hence it carries over unchanged. □

5.3 Special Case: No Transfers At All, Signaling Setting

Finding useful conditions that ensure the revelation principle holds when we are in the signaling setting and have no transfers at all is currently an open problem.

5.4 Revisiting the Running Example

In this setting, the city no longer cares that the types all receive the same utility, but now the city is unable to make welfare transfers. That is, transfers are fixed at zero. The *FTVU condition* does not hold for this cost function, the following assignments to a, y, b, and x all violate the condition:

$a = North,\ y = North,\ b = West,\ x = East$

$a = North,\ y = West,\ b = West,\ x = East$

$a = North,\ y = North,\ b = West,\ x = South$

$a = North,\ y = West,\ b = West,\ x = South$

$a = West,\ y = West,\ b = South,\ x = East$

$a = East,\ y = East,\ b = South,\ x = West$

Thus, we may wonder whether a non-truthful mechanism exists that is better than any other mechanism. However, it turns out that the revelation principle still holds *for the agent's specific valuation function* that we are considering here, and so in fact we can restrict attention to truthful mechanisms. We will return to this example

in 11.4, at which point we will have given conditions for the revelation principle to hold for specific valuation functions in the FTVU case.

6

Fixed Transfers, Fixed Utilities

Definition 11 (FTFU Condition)**.** *An instance satisfies the* FTFU condition *if there exists some* $0 \leqslant \delta_\theta < \infty$ *for each type* θ *such that for all pairs of type* b *and signal* x*,*

$$bx \in \{\delta_b, \infty\}$$

and for all ordered doubles $a \neq b$ *of types, and signal* x*,*

$$(ab = \delta_a \wedge bx = \delta_b) \implies ax = \delta_a$$

Indeed, in the partial verification case (which, in our notation, is the type-reporting case where for all $a \neq b \in \Theta : ab \in \{0, \infty\}$), the condition $(ab = 0 \wedge bx = 0) \implies ax = 0$ is known as the *nested range condition* Green and Laffont [1986], which is known to characterize when the revelation principle holds in that case, with no transfers at all. Note that utilities are also necessarily fixed because no agent will ever incur nonzero reporting costs.

We first provide intuition as to why the FTFU condition is essentially the same as the nested range condition even though we have: a general signaling rather than

type-reporting setting; fixed transfers rather than no transfers at all; and signaling costs that may be δ_θ rather than 0.

- In the proof for the nested range condition, the fact that x may have a type corresponding to it does not play a role. Thus, we can extend it to a signaling setting with a, b as types and x as a signal.

- The key difference between implementations with fixed transfers and with no transfers at all is the ability to give a large negative transfer to a signal that no type emits. This is equivalent to banning its emission.

  This ban is useful for a 'nuisance' signal for which putting any outcome at it (without a negative transfer) would cause at least one type to misemit to it. Since we have fixed utilities, the presence of a nuisance signal affects both truthful and non-truthful implementations. Thus the ability to ban it will not change when the revelation principle holds.

- Finally, for each type b, replacing 0 by δ_b shifts the utility function by a constant and does not affect behavior.

Theorem 12. *The RP holds with fixed transfers and fixed utilities iff the FTFU condition holds.*

Proof. FTFU condition $\implies$ RP holds. We will show that for any F, for any non-truthful implementation we can construct a truthful implementation. Let N together with R_N implement F. Define $H = N$ except:
$A_H(\theta) = F(\theta)$ for $\theta \in \Theta$
$T_H(\theta) = T_N(R_N(\theta))$ for $\theta \in \Theta$

If H is truthful, it clearly implements F with fixed transfers and fixed utilities. (Note that because of the FTFU condition, the reporting cost for a type is always the same.) So now we show this is the case.

By the FTFU condition, the definition of H, and the optimality of R_N, the following series of inequalities holds for any pair of types $a \neq b$, and signals $x = R_N(b)$, $y = R_N(a)$:

$$\begin{aligned} & v_a(o_a) + T_H(a) - aa \\ = \; & v_a(o_a) + T_N(y) - ay \\ \geqslant \; & v_a(o_b) + T_N(x) - ax \\ \geqslant \; & v_a(o_b) + T_H(b) - ab \end{aligned}$$

By the optimality of R_N, and the definition of H, for any type a, signal $y = R_N(a)$, and signal $z \notin \mathit{range}(G)$ we have:

$$\begin{aligned} & v_a(o_a) + T_H(a) - aa \\ = \; & v_a(o_a) + T_N(y) - ay \\ \geqslant \; & v_a(A_N(z)) + T_N(z) - az \\ = \; & v_a(A_H(z)) + T_H(z) - az \end{aligned}$$

This shows that under H, a is willing to emit a over any other signal. Thus, H is truthful.

FTFU condition not holding $\implies$ RP does not hold. Consider first the case where for all $s \in S$, $s\theta \in \{\delta_\theta, \infty\}$, but there exist some $a \neq b$ and x such that $ab = \delta_a$, $bx = \delta_b$, and $ax = \infty$.

Let N be a mechanism with $A_N(\cdot) = o$, with o being any fixed outcome, and $T_N(\cdot) = 0$, except $T_N(x) = 1$. We have $R_N(b) = x$, so $T_N(R_N(b)) = 1$. Also, we know that $T_N(R_N(a)) = 0$ since a cannot emit x.

Now consider any mechanism H with $A_H = A_N \circ R_N$ and $T_H = T_N \circ R_N$. We then have $T_H(b) = 1$ and $T_H(a) = 0$. Since we also have $A_H(a) = A_H(b) = o$ and $aa = ab$, a would emit b over a and thus H is not truthful.

Hence there exists no truthful mechanism with the same transfers as N that implements the choice function. So the revelation principle fails.

Now, all that is left to do is to show that if for some b there is no δ_b (i.e., $bx \neq bb$, and $bx \neq \infty$ for some $b \in \Theta$ and $x \in S$) then the revelation principle fails.

Consider a mechanism N where A_N maps all signals to some fixed outcome o and T_N pays $bx + 1$ to any type that emits x, and 0 otherwise. For an optimal response R_N we have $R_N(b) = x$, so that b's utility is $1 + v_b(o)$.

Now consider any truthful mechanism H with $A_H(b) = o$ and where b receives the same utility as under N, which is $1 + v_b(o)$. $T_H(b)$ must be $bb + 1$ for this to be the case. But, since $bb \neq bx$ we will not have fixed transfers and thus the revelation principle fails. □

6.1 Revisiting the Running Example

Suppose the city is unable to make transfers and wants each type of agent to receive the same utility. To make the example more interesting in this context, we add another outcome $\varnothing$, which consists of giving the agent nothing, for which all types

have utility 0 and for which the city has objective 0 for all types. Clearly, the FTFU condition does not hold because there are edge costs that are neither 0 nor ∞. The optimal truthful mechanism in this context is simply to give all agents $\varnothing$; this is simply because there is no value other than 0 that the types would all be able to get as their utility in this context. On the other hand, the following is an optimal non-truthful implementation (combined with the response where *West* and *East* misreport *South*, but *North* and *South* report truthfully), resulting in an objective value of 35 and each type obtaining utility 1.

$N =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	$\varnothing$	$\varnothing$	*protein*
T	0	0	0	0

Now let us again modify the example, and replace the $\varnothing$ outcome with *cod liver oil*, which all types dislike (utility 0.1) but the city would love for the agent to take (objective 11 for all types). Obviously, in this case the optimal mechanism is to give all types *cod liver oil*, which is in fact truthful. In this case, unlike in 5.4, what is happening is not that the revelation principle holds for the specific valuation function at hand, but rather that it holds for the specific combination of both the valuation function and the choice function at hand—that is, for the specific instance at hand. We will return to this in Section 13.

7

Revelation Principle for Known Valuations

So far, we have always considered whether the revelation principle holds for a given combination of Θ and $c : \Theta \times S \to \mathbb{R}_{\geqslant 0}$. For it to hold meant that *no matter what* the valuation function v and the choice function F (and, possibly, the transfer and/or utilities to be achieved) are, it is either truthfully implementable or not implementable at all. But if we have a particular valuation function in mind, we may not care whether the revelation principle holds for other valuation functions; we just want to know whether *for this valuation function* we can restrict our attention to truthful mechanisms. As we already alluded to in 5.4, the revelation principle may hold for a specific valuation function even when it does not hold for all valuation functions. Accordingly, in this section, we consider the case where the valuation function is known (or fixed). Hence, in this section, an *instance* consists not only of Θ, S and c, but also O and v. Later, in Section 13, we will consider the case where *everything* is fixed, including the choice function (and, possibly, the transfer and/or utilities to be achieved).

8

Known Valuations, Variable Transfers, Variable Utilities

Finding useful conditions that ensure that the RP holds in this case is currently an open problem. We conjecture that verifying whether the revelation principle holds in this case is NP-complete.

9

Known Valuations, Variable Transfers, Fixed Utilities

It turns out that in the VTFU case, it makes no difference whether we know the valuation function.

Theorem 13. *The RP holds with known valuations, variable transfers, and fixed utilities iff the VTFU condition holds.*

Proof. VTFU condition holds $\implies$ *RP holds.* By Theorem 6, the VTFU condition implies that the RP holds for all possible valuation functions, so it will continue to hold for a specific valuation function.

VTFU condition not holding $\implies$ *RP does not hold.* In the corresponding part of the proof of Theorem 6 (without known valuations), we used a constant choice function whose choice of outcome did not matter. Hence, this part of the proof carries over unmodified to this case. □

10

Known Valuations, Fixed Transfers, Variable Utilities, Type-Reporting Setting

In the FTVU case, if we are in a type-reporting setting, it makes no difference whether we know the valuation function.

Theorem 14. *In the type-reporting setting the RP holds with known valuations, fixed transfers, and variable utilities iff the FTVU condition holds.*

Proof. FTVU condition holds $\implies$ *RP holds.* By Theorem 8, the FTVU condition implies that the RP holds for all possible valuation functions, so it will continue to hold for a specific valuation function.

FTVU condition not holding $\implies$ *RP does not hold.* Let $a \neq b \in \Theta$ and $x, y \in \Theta$ violate the FTVU condition. Thus:

$$bx < \infty, \text{ and } ab - aa < ax - ay$$

Since we are in the type-reporting setting, we note that aa is zero. Additionally, since ay is always non-negative, we also have:

$$bx < \infty, \text{ and } ab < ax$$

We will show that for any set of outcomes and valuation function, we can define a non-truthful mechanism N (with an associated optimal response R_N) that implements a choice function F that no truthful mechanism with fixed transfers would implement. That is, H with $A_H = A_N \circ R_N$ and $T_H = T_N \circ R_N$ is not truthful.

We consider the following two cases separately:
(i) $bx \leqslant ba$ or $bx \leqslant ax$
(ii) $bx > ba$ and $bx > ax$

First consider (i):

Choose an arbitrary outcome o, and let $F(\cdot) = o$. Define N as follows:

$A_N(\cdot) = o$
$T_N(x) = bx + ax$
$T_N(a) = bx$
$T_N(\theta) = 0$ for $\theta \neq x, a$

Since $T_N(x) - T_N(a) = ax$, we can have a emit truthfully and receive bx. b will emit x over b since $T_N(x) > bx$, since $ax > ab \geqslant 0$. If $bx \leqslant ba$, b will emit x over a since

$T_N(x) > T_N(a)$. If $bx \leqslant ax$, we can have b still emit x over a since:

$$\begin{aligned} & v_b(o) + T_N(x) - bx \\ = \; & v_b(o) + ax \\ \geqslant \; & v_b(o) + bx \\ \geqslant \; & v_b(o) + bx - ba \\ = \; & v_b(o) + T_N(a) - ba \end{aligned}$$

Hence N is non-truthful[1] and gives a transfer of $bx + ax$ to b and a transfer of bx to a.

For H to have fixed transfers we must have $T_H(b) = bx + ax$ and $T_H(a) = bx$. If the revelation principle is to hold, this mechanism must be truthful. But, by emitting b, a can obtain a utility of:

$$\begin{aligned} & v_a(o) + bx + ax - ab \\ > \; & v_a(o) + bx \end{aligned}$$

which is what it would get for emitting a. So the revelation principle does not hold when $bx \leqslant ba$ or $bx \leqslant ax$.

Now consider (ii):

Choose an arbitrary outcome o, and let $F(\cdot) = o$. Define N as follows:

$A_N(\cdot) = o$

$T_N(x) = bx$

$T_N(\theta) = 0$ for $\theta \neq x$

[1] The FTVU condition not holding implies $x \neq b$.

N is non-truthful as a will emit x over a[2] since $T_N(x) = bx > ax$. And since $T_N(x) = bx$, we can have b emit truthfully and receive 0 transfer.

For H to have fixed transfers we must have $T_H(b) = 0$ and $T_H(a) = bx$. If the revelation principle is to hold, this mechanism must be truthful. But in fact, by emitting a, b can obtain a utility of:

$$\begin{aligned} & v_b(o) + bx - ba \\ > \; & v_b(o) \end{aligned}$$

which is what it would get for emitting b. So the revelation principle does not hold when $bx > ba$ and $bx > ax$.

Since our two cases cover all the possibilities, the revelation principle does not hold.

□

[2] (ii) implies $x \neq a$.

11

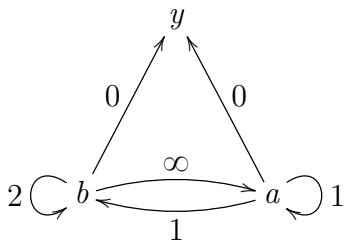
Known Valuations, Fixed Transfers, Variable Utilities, General Signaling Setting

Unlike in the case of variable transfers and fixed utilities, the condition here turns out not to be the same as in the unknown valuations case when we move beyond the type-reporting setting. Thus, we provide a separate condition for this case. It is not elegant, but, as we show, it can still be checked in polynomial time.

11.0.1 Example: Difference in Revelation Principles for Known and Unknown Valuations

We first give an example showing that the FTVU condition is not necessary for the revelation principle to hold when we are in the signaling setting, and have known valuations, fixed transfers, and variable utilities. (It is, of course, sufficient.) Consider an instance with two types $a \neq b$ and an additional signal y. Let $aa = 1,\ ab = 1,\ ay = 0,\ bb = 2,\ ba = \infty$, and $by = 0$. We can visualize these types and signaling costs as

follows:



The FTVU condition does not hold on this instance. If we let $x = b$ we have:

$$ab - aa = 1 - 1 < 1 - 0 = ax - ay$$

Yet, we show with known valuations the revelation principle can still hold on this instance.

Suppose that we have just a single outcome that a and b both value at 0. When considering the design of truthful mechanisms, we need not worry about y, because we can put a sufficiently negative payment there. Given this, it is easy to see that a mechanism H is truthful if and only if $T_H(a) \geqslant T_H(b)$. Hence, for the revelation principle to be violated, there must exist some non-truthful mechanism N where $T_N(R_N(b)) > T_N(R_N(a))$. But this leads to a contradiction no matter the choice of R_N:

- If both b and a emit y, then we cannot give them different transfers.
- If just a emits y (so b emits b), then for a to prefer emitting y to emitting b we must have $T_N(y) \geqslant T_N(b) - 1$. But then b would also prefer to emit y.
- If just b emits y, if $T_N(y) > T_N(R_N(a))$, then a would also prefer to emit y.

Hence the revelation principle holds for these valuations.

We now move on to the revelation principle condition for this case.

11.1 KFTVU Condition

Definition 15 (KFTVU Condition)**.** *An instance satisfies the* KFTVU condition *if for all ordered doubles $a \neq b$ of types and ordered doubles x, y of signals, there* ***do not exist*** *outcomes o_a and o_b, allocation function $A : S \to O$, and transfer function $T : S \to \mathbb{R}$ such that the following hold:*

$$v_a(o_b) + T(x) - ab > v_a(o_a) + T(y) - aa \tag{11.1}$$

$$A(y) = o_a \tag{11.2}$$

$$A(x) = o_b \tag{11.3}$$

$$(\forall s \in S)\ v_b(o_b) + T(x) - bx \geqslant v_b(A(s)) + T(s) - bs \tag{11.4}$$

$$(\forall s \in S)\ v_a(o_a) + T(y) - ay \geqslant v_a(A(s)) + T(s) - as \tag{11.5}$$

Effectively, this condition asks whether we can directly construct a counterexample to the revelation principle. In particular, (1) ensures that a will always misemit to b if we try to implement our choice function truthfully.

Note that if the FTVU condition holds on $a \neq b$, and x, y, then the KFTVU condition also holds on a, b, and x, y. This is because if (1) holds, then (5) does not when $s = x$.

Naïvely, checking this condition requires searching through all allocation functions A and transfer functions T, which would take exponential time. However, the condition can in fact be checked efficiently.

Proposition 16. *The KFTVU condition can be checked in polynomial time.*

Proof. For each $a \neq b$ and x, y, we can efficiently check whether the KFTVU condition holds for every o_a, o_b, as follows.

Set $A(y) = o_a$, $A(x) = o_b$, and the rest of A arbitrarily. This satisfies (2) and (3).

Any transfer function that satisfies (1), (4), and (5) also satisfies the following constraints:

$$v_a(o_b) + T(x) - ab > v_a(o_a) + T(y) - aa$$

$$v_b(o_b) + T(x) - bx \geqslant v_b(o_a) + T(y) - by$$

$$v_a(o_a) + T(y) - ay \geqslant v_a(o_b) + T(x) - ax$$

Furthermore, any transfer function T that satisfies these constraints satisfies (1), and can be transformed into a T' which also satisfies (4) and (5). Let $T' = T$ except for x, y where $T' = T + L$ for some very large L.

Thus a transfer function that satisfies (1), (4), and (5) exists iff one exists that satisfies the constraints. And we can check this using a simple linear feasibility program. □

Theorem 17. *The RP holds with known valuations, fixed transfers, and variable utilities iff the* KFTVU condition *holds.*

Proof. KFTVU condition not holding $\implies$ the RP does not hold. Let $a \neq b, x, y, o_a, o_b$, $A : S \to O$, and $T : S \to \mathbb{R}$ be a witness for the violation of the KFTVU condition.

Consider the mechanism N defined by $A_N = A$ and $T_N = T$. By the conditions, there exists an optimal response R_N where $R_N(a) = y$ and $R_N(b) = x$, so that $A_N(R_N(a)) = o_a$ and $A_N(R_N(b)) = o_b$.

But the function $A_H \triangleq A_N \circ R_N$ is not truthfully implementable with $T_H \triangleq T_N \circ R_N$ (these are both taken to be restricted to signals in $range(G)$ here), because a would prefer to misemit to b by (11.1).

RP not holding $\implies$ *KFTVU condition does not hold.* Let mechanism N with transfers T_N, allocation A_N, and optimal response profile R_N be a witness to the violation of the revelation principle, i.e., the mechanism H with $T_H \triangleq T_N \circ R_N$ and $A_H \triangleq A_N \circ R_N$ is not truthful.

For that to be the case, there must be some types $a \neq b$ and signals x, y such that in H, a strictly prefers emitting b to emitting a, and in N, $R_N(b) = x$ and $R_N(a) = y$.

(a must prefer emitting to some b corresponding to a type as WLOG we can assume large negative transfers on signals corresponding to no type in H.)

Now, let $o_a = A_H(a)$, $o_b = A_H(b)$, $A = A_N$, and $T = T_N$. Then, in the KFTVU conditions,
(11.1) holds because a prefers misemitting b in H;
(11.2) holds because $o_a = A_H(a) = A_N(R_N(a)) = A(y)$;
(11.3) holds because $o_b = A_H(b) = A_N(R_N(b)) = A(x)$;
(11.4) holds because $R_N(b) = x$; and
(11.5) holds because $R_N(a) = y$.

Hence the KFTVU condition is violated. □

11.1.1 Example: Difference in Revelation Principles for Fixed Transfers and No Transfers to the Agent

The following example shows that in the type-reporting, known-valuations setting, the 'fixed transfers' and 'no transfers to the agent' revelation principles must differ.

Consider an instance with types a, b, c, outcome o, and costs:

$ab = 0$

$bc = 0$

$\theta\theta = 0$ (since we are in the type-reporting setting)

and all other costs are ∞.

The revelation principle does not hold with fixed transfers. We can non-truthfully implement $F(\cdot) = o$ with a mechanism where b gets a transfer of 10 for reporting c, and a gets 0 at a. But, we cannot truthfully implement it with the same transfers, because a would misemit to b.

On the other hand, with no transfers to the agent the revelation principle holds. The only possible choice function is $F(\cdot) = o$, and we must have that for all θ, $T(R(\theta)) = 0$. This is truthfully implementable, so the revelation principle holds.

11.2 Special Case: No Transfers To The Agent

We again consider the special case of no transfers to the agent. Unlike in the case of unknown valuations, the condition will be different here.

Definition 18 (KNTGVU Condition)**.** *An instance satisfies the* KNTGVU condition *if it satisfies the modified version of the KFTVU conditions where we only consider transfer functions T satisfying the following conditions:*

(i) $\forall s$, $T(s) = 0$ *or* $-L$*, where* $-L$ *is a very negative number such that any type would prefer any outcome and a transfer of zero at any (finite-cost) signal, to receiving a transfer of* $-L$ *(alongside any outcome at any signal).*

(ii) $\forall \theta$, $\exists s$ *s.t.* $\theta s < \infty$ *and* $T(s) = 0$.

Restriction (i) ensures that no type receives a transfer of more than 0. *Restriction (ii) ensures that each type can receive a transfer of at least* 0. *Note that these restrictions imply* $T(x) = T(y) = 0$.

The KNTGVU condition differs from the KFTVU condition. Hence we must separately prove that the KNTGVU condition can be checked in polynomial time.

Naïvely, checking this would require searching through all allocation functions A and transfer functions with $T(s) \in \{0, -L\}$, which would require exponential time. However, again, the condition can in fact be checked efficiently.

Proposition 19. *The KNTGVU condition can be checked in polynomial time.*

Proof. For each $a \neq b$ and x, y, we can efficiently check whether the KNTGVU condition holds for every two outcomes o_a and o_b, as follows.

What we need to check is that there exists a combination of outcomes and transfers for every $s \neq x, y$ that satisfies (4), (5), (i), and (ii).

First we check whether there exists a combination of outcomes for every $s \neq x, y$ that satisfies (11.4) and (11.5) when $T(s) = 0$. Whether an outcome satisfies these conditions for a single s is independent of which outcome we choose for any other s'. Hence, all that needs to be checked is, for each s individually, whether there exists an outcome satisfying (4) and (5). If this is the case, we can set $T(\cdot) = 0$. Hence, (ii) holds by default, so we are done.

Otherwise, for each s which has a satisfying outcome, give it that outcome with a transfer of 0. For each s which does not, to satisfy (4) and (5) we have no other option than to give it a transfer of $-L$ (and an arbitrary outcome). So now we simply check whether (ii) still holds. □

Theorem 20. *The RP holds with known valuations, no transfers to the agent and variable utilities iff the* KNTGVU condition *holds.*

Proof. KNTGVU condition not holding $\implies$ *the RP does not hold.* Let $a \neq b, x, y, o_a, o_b$, $A : S \to O$, and $T : S \to \mathbb{R}$ be a witness for the violation of the KNTGVU condition.

Consider the mechanism N defined by $A_N = A$ and $T_N = T$. By the conditions, there exists an optimal response R_N where $R_N(a) = y$ and $R_N(b) = x$, so that $A_N(R_N(a)) = o_a$ and $A_N(R_N(b)) = o_b$. Additionally we have $T_N(R_N(\theta)) = 0$ for all θ by (i) and (ii).

But then the function $A_H \triangleq A_N \circ R_N$ is not truthfully implementable with $T_H \triangleq T_N \circ R_N$ (these are both taken to be restricted to signals in $range(G)$ here), because a would prefer to misemit to b by (11.1).

RP not holding $\implies$ *KNTGVU condition does not hold.* Let mechanism N with transfers T_N, allocation A_N, no transfers to the agent, and optimal response profile R_N be a witness to the violation of the revelation principle. Thus any mechanism H with $A_H \triangleq A_N \circ R_N$ (these are both taken to be restricted to signals in $range(G)$ here), and no transfers to the agent is not truthful.

For that to be the case, there must be some types $a \neq b$ and signals x, y such that in H, a strictly prefers emitting b to emitting a, and in N, $R_N(b) = x$ and $R_N(a) = y$.

(a must prefer emitting to some b corresponding to a type as WLOG we can assume large negative transfers on signals corresponding to no type in H.)

Create $N' = N$ except that $T_{N'} = -L$ for any signal not emitted by any type under N. Since no type would ever want to receive a transfer of $-L$, $R_{N'} = R_N$. Hence N' has no transfers to the agent. And since $A_{N'} = A_N$, we have $A_H = A_{N'} \circ R_{N'}$.

Now, let $o_a = A_H(a)$, $o_b = A_H(b)$, $A = A_{N'}$, and $T = T_{N'}$. Then, in the KNTGVU conditions,

(11.1) holds because a prefers misemitting b in H;

(11.2) holds because $o_a = A_H(a) = A_{N'}(R_{N'}(a)) = A(y)$;

(11.3) holds because $o_b = A_H(b) = A_{N'}(R_{N'}(b)) = A(x)$;

(11.4) holds because $R_{N'}(b) = x$;

(11.5) holds because $R_{N'}(a) = y$;

(i) holds by construction of N';

(ii) holds because under N' each type θ can receive a transfer of 0 at $R_N(\theta)$.

Hence the KFTVU condition is violated. □

11.2.1 Example: Difference in Revelation Principles for No Transfers to the Agent and No Transfers At All

The following example shows that in the type-reporting, known-valuations setting, the 'no transfers to the agent' and 'no transfers at all' revelation principles must differ.

Consider an instance with types a, b, c, outcomes: o_1, o_2, valuations:

$v_a(o_1) = 2, v_a(o_2) = 0$

$v_b(o_1) = v_b(o_2) = 1$

$v_c(o_1) = 0, v_c(o_2) = 2,$

and costs:

$ab = cb = 0$

$ba = bc = 1$

$ac = ca = 10$

$\theta\theta = 0$ (since we are in the type-reporting setting).

If we have no transfers *at all*, the revelation principle holds. In any implementation b will always report b, and a and c will never report each other. WLOG assume a non-truthful implementation involves a (but not c) reporting b. We can create a truthful implementation of the same choice function by putting the outcome from b at a as well, and c will still not report a or b.

If we have no transfers *to the agent* we are allowed to put negative transfers on types no one will report. Then we can non-truthfully implement $F(a) = o_2$, $F(b) = o_2$, and $F(c) = o_1$ by having a and b report a, c report c, and putting a large negative transfer at b. But we cannot do this truthfully, because if we put o_2 at b then c will report b. Hence the revelation principle does not hold.

11.3 Special Case: No Transfers At All, Type-Reporting Setting

We again consider the special case where we allow no transfers at all. When we are in a type-reporting setting the condition is essentially the same as that for fixed transfers.

Definition 21 (KNTAAVU Condition)**.** *An instance satisfies the* KNTAAVU condition *if it satisfies the modified version of the KFTVU condition where we only consider transfer functions T with $T(\cdot) = 0$.*

The KNTAAVU condition differs from the KFTVU condition. Hence we must separately prove that the KNTAAVU condition can be checked in polynomial time.

Naïvely, checking this would require searching through all allocation functions A, which would require exponential time. However, again, the condition can in fact be checked efficiently.

Proposition 22. *The KNTAAVU condition can be checked in polynomial time.*

Proof. For each $a \neq b$ and x, y, we can efficiently check whether the KNTAAVU condition holds for every two outcomes o_a and o_b, as follows.

What we need to check is that there exists a combination of outcomes for every $s \neq x, y$ that satisfies (11.4) and (11.5). Whether an outcome satisfies these conditions for a single s is independent of which outcome we choose for any other s'. Hence, all that needs to be checked is, for each s individually, whether there exists an outcome satisfying the conditions. □

Theorem 23. *The RP holds with known valuations,no transfers at all and variable utilities in the type-reporting setting iff the* KNTAAVU condition *holds.*

Proof. KNTAAVU condition not holding $\implies$ the RP does not hold. In this part of the proof of Theorem 17 (with fixed rather than no transfers at all) at no point did we require the transfer function not be fixed at zero. Thus we can carry it over unchanged except that we require $T(\cdot) = 0$ for any transfer function T.

RP not holding $\implies$ KNTAAVU condition does not hold. In this part of the proof of Theorem 17 (with fixed rather than no transfers at all) the only point we required the transfer function not to be fixed at zero was to prevent type a from emitting a signal not corresponding to any type in the truthful mechanism. But here we are in

the type-reporting setting so this is not an issue. Thus we can carry over the proof unchanged except that we require $T(\cdot) = 0$ for any transfer function T. □

11.4 Revisiting the Running Example

We now return to the running example for the FTVU case. Recall from 5.4 that for the case where the city does not care that the types of agent all receive the same utility, but is unable to make welfare transfers, the revelation principle does not hold for *all* valuation functions for the cost function under consideration here.

However, it does hold for the specific valuation function under consideration here. This is because for it to be violated, there would need to be a pair of outcomes such that the difference in valuations between them could incentivize *West* to travel to *East* or *South*, or *South* to *East* or *West*. But there exists no such pair of outcomes. Thus we only need to search through the space of truthful mechanisms to find an optimal mechanism. The following truthful mechanism obtains the best objective value of 30.

$H =$

	$\hat{North}$	$\hat{West}$	$\hat{East}$	$\hat{South}$
A	*fiber*	*vitamins*	*vitamins*	*vitamins*
T	0	0	0	0

11.5 Special Case: No Transfers At All, Signaling Setting

Finding useful conditions that ensure that the RP holds in this case is currently an open problem.

12

Known Valuations, Fixed Transfers, Fixed Utilities

Theorem 24. *The RP holds with known valuations, fixed transfers, and fixed utilities iff the FTFU condition holds.*

Proof. FTFU condition holds $\implies$ *RP holds.* By Theorem 12, the FTFU condition implies the RP holds for all possible valuation functions, so it will continue to hold for a specific valuation function.

FTFU condition not holding $\implies$ *RP does not hold.* In the corresponding part of the proof of Theorem 12 (FTFU without known valuations), we used constant choice functions whose choice of outcome did not matter. Hence, this part of the proof carries over unmodified to this case. □

13

Revelation Principle for Fully Specified Instances

In the previous section, we considered the case where we already know the valuation function, and wish to know if the revelation principle holds for that valuation function. Still, all that is needed to violate of the revelation principle is that there is *some* choice function (possibly together with transfers and/or utilities) that can be non-truthfully, but not truthfully, implemented. But this may be of little interest if we already know the precise choice function (etc.) we wish to implement. Indeed, even if the revelation principle does not hold for all choice functions (etc.), it may yet hold for the one we care about. This is what we study in this section. Hence, an instance is now a *fully specified instance*, consisting of Θ, S, c, O, and v as before, but also F, possibly a specific transfer function $T^* : \Theta \to \mathbb{R}$, and possibly a specific utility function $U^* : \Theta \to \mathbb{R}$, which we wish to implement.

Definition 25 (Revelation Principle on an Fully Specified Instance)**.** *We say the RP is true for a fully specified instance if either (1) a truthful mechanism T exists that implements the choice function (with the required utilities and/or transfers), or (2) no (possibly non-truthful) mechanism N does.*

As it turns out, deciding whether the RP holds on individual fully specified instances even in the type-reporting setting comes down to the computational problem of deciding whether a (possibly non-truthful) implementation exists. The following lemma makes this clear.

Lemma 26. *Determining whether the revelation principle fails to hold on a given fully specified instance is computationally exactly as hard as determining whether there is a (not necessarily truthful) implementation for that instance.*

Proof. In each case, we can efficiently verify whether there is a truthful implementation for that instance:

- If there are no transfers, then there is only one possibility for what the mechanism does for the signals in $range(G)$. But, we must also assign outcomes to the signals outside $range(G)$ such that no type misemits to them. Whether there exists such an outcome for a given s outside $range(G)$ is independent of which outcome we choose for any other such s'. Hence, all that needs to be checked is, for each s individually, whether there exists an outcome such that no type will misemit to it.

- If there are transfers but they are fixed (or implicitly fixed because utilities are), again there is only one possibility for what the mechanism does for signals in $range(G)$. In this case, we can always ensure that no type will misemit outside $range(G)$, by putting a sufficiently negative transfer on those signals.

- Finally, if neither transfers nor utilities are fixed, then it is a simple linear feasibility problem to determine whether transfers exist that implement the choice function on the signals in $range(G)$. And again, we can ensure that

no type misemits to signals outside $range(G)$ by putting sufficiently negative transfers there.

Hence, we can reduce the problem of determining whether a (not necessarily truthful) implementation exists for a fully specified instance to the problem of checking whether the RP holds on a fully specified instance, as follows. First, check whether a truthful implementation exists; if so the answer is "yes." Otherwise, there is an implementation if and only if the revelation principle fails to hold on this instance.

Conversely, we can reduce the problem of checking whether the RP holds on a fully specified instance to the problem of determining whether a (not necessarily truthful) implementation exists for a fully specified instance, as follows. Again, first, check whether a truthful implementation exists; if so the answer is "yes." Otherwise, the revelation principle holds if and only if there is no implementation. □

Theorem 27. *Computing whether the revelation principle holds on a given fully specified instance is coNP-complete (whether or not transfers and/or utilities are fixed). This is true even in the type-reporting setting.*

Proof. Variable Transfers

When we have variable transfers, for both variable and fixed utilities, the problem of determining whether a (not necessarily truthful) implementation for an instance exists is NP-complete Auletta et al. [2011]; Kephart and Conitzer [2015].

Fixed Transfers, Variable Utilities

Auletta et al. [2011] showed that implementation is NP-complete in the type-reporting setting with no transfers at all and variable utilities. In particular, for an arbitrary 3-SAT instance they show how to construct a mechanism design with partial verifi-

cation instance with choice function F, such that the 3-SAT instance is satisfiable if and only if F is implementable with no transfers at all.

This does not necessarily imply that the case with no transfers *to the agent* is NP-complete. These cases may differ, as it is possible that non-truthful implementation requires giving negative transfers to types not reported.

But, as it turns out, the proof that they give can also show that implementation with no transfers to the agent is NP-complete. On their instance, F is implementable with no transfers at all *if and only if* it is implementable with no transfers to the agent. The 'only if' is automatic as no transfers at all is a special case of of no transfers to the agent. For the 'if' to hold, we must show that allowing negative transfers to types not reported cannot cause F to become implementable. This is equivalent to asking if forbidding certain types from being reported is helpful. For the instances they construct, it is not. The fundamental difficulty in implementation in their instances stems from reconciling the following two needs:

- Every clause type needs to receive an outcome of 'true' at some literal type in the clause.
- Every variable type needs to receive an outcome of 'false' at one of its two literal types.

Forbidding certain type reports will not help satisfy either of these conditions. Hence, implementability with no transfers to the agent and variable utilities is NP-complete, and thus fixed transfers to the agent and variable utilities is as well.

Fixed Transfers, Fixed Utilities

Finally, Kephart and Conitzer [2015] showed that in the partial verification case, when we have no transfers to the agent, the NP-completeness of the variable utilities

case implies the NP-completeness of the fixed utilities case. Hence implementability with no transfers to the agent with fixed utilities is NP-complete, and thus implementability with fixed transfers to the agent is as well.

So, the problem of determining whether a (not necessarily truthful) implementation for an instance exists is NP-complete in all the cases. Hence, Lemma 26 implies that determining whether the revelation principle **fails** to hold is NP-complete, proving the theorem. □

14

Conclusions

In this work, we studied mechanism design with signaling costs. Because the revelation principle is the foundation for so much of existing mechanism design theory, we focused on determining necessary and sufficient conditions for it to hold, under various circumstances (allowing transfers/utilities to vary or requiring them to remain fixed from the non-truthful to the truthful implementation; knowing the valuation function/choice function, or not).

We believe that the framework of mechanism design with signaling costs is one that will be of increasing importance. This is because the parties that run mechanisms increasingly have data on the other agents, as opposed to knowing nothing about them *ex ante* and only being able to ask them about their preferences. Now, an agent can often change the data that the mechanism has about it. We discussed several examples in the introduction; another possibility is for the agent to actively (and possibly selectively) avoid having data collected on it, for example by logging out of systems, erasing cookies, avoiding using credit cards that identify them, filing

"right to be forgotten" requests, etc. All of these, though, come at some effort (or other) cost. Hence, the standard mechanism design framework where an agent can report any type at no cost—the "anonymous bidder walking into a Sotheby's auction" model[1]—does not exactly fit such applications. But the mechanism design with costly signaling framework does.

[1] Of course, the standard mechanism design framework where misreporting is costless can perfectly well address situations where the party running the mechanism has prior information over the agent. The point is that the standard framework does *not* address the agent being able to *change* this prior information at some cost.

Bibliography

Auletta, V., Penna, P., Persiano, G., and Ventre, C. (2011), "Alternatives to truthfulness are hard to recognize," *Autonomous Agents and Multi-Agent Systems*, 22, 200–216.

Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. (2010), "The security of machine learning," *Machine Learning*, 81, 121–148.

Bull, J. and Watson, J. (2004), "Evidence disclosure and verifiability," *Journal of Economic Theory*, 118, 1–31.

Bull, J. and Watson, J. (2007), "Hard evidence and mechanism design," *Games and Economic Behavior*, 58, 75–93.

Caragiannis, I., Elkind, E., Szegedy, M., and Yu, L. (2012), "Mechanism design: from partial to probabilistic verification," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 266–283, Valencia, Spain.

Conitzer, V. and Sandholm, T. (2002), "Complexity of Mechanism Design," in *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 103–110, Edmonton, Canada.

Conitzer, V. and Sandholm, T. (2004), "Self-interested Automated Mechanism Design and Implications for Optimal Combinatorial Auctions," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 132–141, New York, NY, USA.

Dalvi, N. N., Domingos, P., Mausam, Sanghai, S. K., and Verma, D. (2004), "Adversarial Classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 99–108, Seattle, WA, USA.

Deneckere, R. and Severinov, S. (2008), "Mechanism design with partial state verifiability," *Games and Economic Behavior*, 64, 487–513.

Deneckere, R. and Severinov, S. (2014), "Optimal Screening with Costly Misrepresentation," Working Paper.

Green, J. and Laffont, J.-J. (1986), "Partially verifiable information and mechanism design," *Review of Economic Studies*, 53, 447–456.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. (2016), "Strategic Classification," in *Innovations in Theoretical Computer Science (ITCS)*, Cambridge, MA, USA.

Kartik, N. and Tercieux, O. (2012), "Implementation with evidence," *Theoretical Economics*, 7, 323–355.

Kartik, N., Tercieux, O., and Holden, R. (2014), "Simple mechanisms and preferences for honesty," *Games and Economic Behavior*, 83, 284–290.

Kephart, A. and Conitzer, V. (2015), "Complexity of Mechanism Design with Signaling Costs," in *Proceedings of the Fourteenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 357–365, Istanbul, Turkey.

Koessler, F. and Perez-Richet, E. (2014), "Evidence based mechanisms," Tech. rep., working paper.

Lacker, J. and Weinberg, J. (1989), "Optimal Contracts under Costly State Falsification," *Journal of Political Economy*, 97, 1345–63.

Rochet, J.-C. (1987), "A necessary and sufficient condition for rationalizability in a quasi-linear context," *Journal of Mathematical Economics*, 16, 191–200.

Spence, M. (1973), "Job Market Signaling," *Quarterly Journal of Economics*, 87, 355–374.

Strausz, R. (2016), "Mechanism Design with Partially Verifiable Information," .

Yu, L. (2011), "Mechanism design with partial verification and revelation principle," *Autonomous Agents and Multi-Agent Systems*, 22, 217–223.