

A million answers to twenty questions: choosing by checklist*

Michael Mandler

Paola Manzini

Royal Holloway College, University of London

University of St Andrews and IZA

Marco Mariotti[†]

University of St Andrews

This version: February 2011

Abstract

Several decision models in marketing science and psychology assume that a consumer chooses by proceeding sequentially through a checklist of desirable properties. These models are contrasted to the utility maximization model of rationality in economics. We show on the contrary that the two approaches are nearly equivalent. Since the number of preference discriminations that an agent can make increases exponentially in the number of properties used, checklists provide a rapid procedural basis for utility maximization.

JEL codes: D01

Keywords: Bounded rationality, utility maximization, choice function, lexicographic utility.

*We are grateful to Claire Blackman, Vincent Crawford, Peter Hammond, Jacob Leshno, Marco Li Calzi, Mark Machina, Ben Polak, Ludovic Renou, Ariel Rubinstein, Chris Tyson, the Associate Editor and two referees for comments and discussions.

[†]Mandler: Dept. of Economics, Royal Holloway College, University of London, Egham, Surrey TW20 0EX, UK. Manzini and Mariotti: School of Economics & Finance, Castlecliffe, The Scores, University of St Andrews KY16 9AL, UK.

1 Introduction

In shopping for a car online, suppose you first decide on a maximum price, then narrow down your search to cars with a manual transmission, then look to see if any sport cars are available, then any Italian sport cars ... and you end up buying a red Alfa Romeo.

In this example you make your decision when facing a set of alternatives using only *properties* of the alternatives. A property is simply a subset of alternatives, e.g., all sports cars. You go through your checklist of properties until you are able to narrow down the set of alternatives sufficiently. At each step you eliminate the alternatives that do not have the specified property, or, if no alternative has the property, you do not eliminate any options and move on to the next property. No maximization of utility or of preferences is invoked: all that is required is an ordered list of desirable attributes. That the list is ordered means that earlier properties always trump later properties; if the car buyer checks car color only with his final property, then color can never take precedence over the properties checked earlier on. This lexicographic feature of ordered properties makes choosing by checklist appear distant from the classical economic agent's pursuit of utility. Moreover, a checklist is easy to execute, while maximizing utility may seem to be a daunting task. In the words of Herbert Simon [26]:

The assumption of a utility function postulates a consistency of human choice that is not always evidenced in reality. The assumption of maximization may also place a heavy (often unbearable) computational burden on the decision maker.

(p. 16)

Checklists present a challenge to Simon's view. Although easy to use, checklists implicitly impose a utility ordering on alternatives; the checklist and utility models are in fact nearly equivalent. Checklists in addition can make fine preference discriminations using only a handful of properties; from the checklist point of view, utility maximization is computationally undemanding.

The sequential elimination of alternatives by whether or not they possess properties

underlies several decision making models in psychology¹ and marketing science.² Since any decision procedure that follows a flowchart of ‘yes or no’ questions can be written as a checklist, checklists can serve as normative guides in fields such as clinical medicine.³ The specific checklist model we present is a simplified (deterministic) version of Tversky’s [27] ‘elimination by aspects’, which

is relatively easy to apply... involves no numerical computations and... is easy to explain and justify (p. 489 in [28]).

Decision-making with a checklist is considered basic precisely because it eschews any use of preference relations over alternatives, the hallmark of economic analysis. Its attraction is its simplicity: in the language of Gigerenzer and Todd [12], it generates ‘fast and frugal’ heuristics, appropriate when time, knowledge and computational power are scarce. Gigerenzer and Todd indeed emphasize the contrast between such heuristics and ‘demonic rationality’, by which they mean preference or utility maximization.

As the views of Simon and the psychologists illustrate, it is not clear at first sight that there is a connection between checklists and the economic model of maximization. And the fact that discriminations among alternatives made by one property can never be overturned by later properties suggests that the only maximizing agents that the model can capture are agents with lexicographic preferences on \mathbb{R}_+^n who do not make trade-offs among different types of goods or attributes (where, e.g., agents prefer more of good 1 and good 2 quantities are decisive only when good 1 quantities are tied).

We will see that the reverse is the case: the agents who use our benchmark model of checklists – where all the alternatives that will be rejected are eliminated in finitely many steps – always make choices that maximize some utility function. Since lexicographic preferences on \mathbb{R}_+^n cannot be represented by a utility function, it follows that checklist users cannot in this sense be lexicographic. So, whatever goes on in the minds of checklist users, they act like classical maximizers. While we can extend the benchmark model of checklists

¹See Tversky [27], Bereby-Meyer, Assor and Katz [2], Brandstätter, Gigerenzer and Hertwig [3] and Katsikopoulos and Martignon [16].

²See e.g. Yee et al. [29]. The term ‘non-compensatory choice models’ is used in these fields to underscore the lack of ‘tradeoffs’ between earlier and later properties.

³ See for example Fischer et al. [8].

to cover agents without utility functions, such agents remain handicapped: their checklists will go on indefinitely eliminating options without end.

The lexicography example illustrates the broader principle, contra Simon, that having a utility function contributes to rather than detracts from decision-making efficiency. We will see that checklist users can sift through alternatives rapidly: the number of properties they must go through relative to the number of preference discriminations n that they make shrinks to 0 as n increases. Checklist users can in effect perform a binary search, which makes the number of preference discriminations they make an exponential function of the number of properties that they use. As a result, an agent who makes a 1,000,000 preference discriminations needs a checklist that is just 20 properties long.

Comparable conclusions hold for the agents of consumer theory who choose commodity bundles. It might seem that checklist users cannot exhibit the uncountably many indifference classes that textbook consumers have. But in fact the choice behavior of any utility-maximizer can be generated by some checklist. Moreover the checklist can be one of the benchmark checklists that execute quickly: for any finite set of alternatives, the agent will need to go through only finitely many properties on his or her checklist before coming to a decision.

So not only will any agent who uses the benchmark model of quickly executing checklists have a utility function but the converse holds as well: any utility-maximizer can make decisions with a quickly-executing checklist (though here some domain qualifications will come into play). The tractability that has attracted psychologists to checklists thus obtains if and only if checklist users display the trade-offs of utility maximizers.

We end up near the Gigerenzer and Todd [12] point of view but with a caveat. Checklists are indeed ‘fast and frugal’: they are a fast and frugal way to maximize utility.

2 Checklists

Fix a nonempty set of alternatives X . An agent faces a domain \mathcal{A} of choice sets, where each A in \mathcal{A} is a nonempty subset of X . For each choice set A in \mathcal{A} , the agent selects a nonempty $c(A) \subset A$. Following tradition, we call c a ‘choice function’ but each $c(A)$ is a set.

A decision maker who chooses by checklist decides on a $c(A)$ by going through a sequence of properties; for each property, if there is an alternative in A that has that property then the agent eliminates all those alternatives that do not. While an agent may use a large pool of properties to discriminate among alternatives, in our benchmark model we require that for every A a final selection is reached in a finite number of steps.

Formally, a *property* $P(i)$ is simply a set of alternatives, $P(i) \subset X$, and we say ‘alternative x has property $P(i)$ ’ when $x \in P(i)$. A *checklist* is a sequence of properties $P = (P(1), P(2), \dots) = P(i)_{i \in I}$ where the set of indices I is either $\{1, \dots, n\}$ or the entire set of natural numbers $\{1, \dots, n, \dots\}$.

Given a choice set $A \subset X$ and a checklist P , define inductively the following ‘survivor sets’ $S_i(A)$:

$$S_0(A) = A$$

$$S_i(A) = \begin{cases} S_{i-1}(A) \cap P(i) & \text{if } S_{i-1}(A) \cap P(i) \neq \emptyset \\ S_{i-1}(A) & \text{otherwise} \end{cases}$$

This sequence makes precise the elimination procedure we described. At each step i the agent checks whether the current set of surviving alternatives have the i th property. If some alternatives do, the alternatives that do not are thrown away. Otherwise, all alternatives survive to the next round. In both cases the agent moves to step $i + 1$.

Definition 1 *A choice function c defined on a domain \mathcal{A} has a checklist if and only if there exists a checklist P such that, for all $A \in \mathcal{A}$, there is a property $P(j)$ such that*

$$\begin{aligned} S_i(A) &= S_j(A) \text{ for all } i \geq j \\ c(A) &= S_j(A), \end{aligned} \tag{1}$$

and we then say that P is a checklist for c .

A choice function that has a checklist thus satisfies two features. First, the procedure ‘finitely terminates’: for any choice set A there exists a property in the checklist such that,

from that stage onwards, the set of survivors does not shrink any further.⁴ Second, this set of permanent survivors coincides with what the choice function selects from A .

Some properties in a checklist might have no impact on choice, e.g., when a property discriminates only between alternatives that properties earlier in the checklist have already ranked. The impact of a property on choice can also vary with A . In our opening car example, suppose $P(1)$ is the set of all cars cheaper than \$15,000. Then, if A contains no car cheaper than \$15,000, a checklist that omits $P(1)$ but is otherwise identical would make the same selection from A . The irrelevance of any property i with $P(i) \cap A = \emptyset$ ensures that checklists will always make a nonempty selection from A .

3 Checklist users are utility-maximizers

Consider again the car buyer that opened the paper and the lexicographic flavor of his decisions. This agent makes a series of categorical judgments, where any discrimination between cars made by a property trumps all discriminations made by properties later in the checklist. It might therefore seem that the only preference relation that such a checklist user could maximize is the lexicographic preference on \mathbb{R}_+^n . Recall that lexicographic preferences, on \mathbb{R}_+^2 for example, are defined by $x \succsim y$ if and only if $x_1 > y_1$ or ($x_1 = y_1$ and $x_2 \geq y_2$): more of good 1 always trumps any increase of good 2.

The following example shows how to express categorical judgments as properties. We use the following terminology: we call a complete and transitive binary relation on X a *preference relation* and say that a choice function c on the domain \mathcal{A} *maximizes a preference relation* \succsim if $c(A) = \{x \in A : x \succsim y \text{ for all } y \in A\}$ for all $A \in \mathcal{A}$. A choice function c on the domain \mathcal{A} *maximizes a utility function* if there exists a function $u : X \rightarrow \mathbb{R}$ such that $c(A) = \{x \in A : u(x) \geq u(y) \text{ for all } y \in A\}$ for all $A \in \mathcal{A}$. Any c that maximizes a utility function automatically maximizes a preference relation but not vice versa, as the classic example of lexicographic preferences on \mathbb{R}_+^2 shows.

⁴After reaching $P(j)$ in Definition 1, to execute a decision the agent must conclude that it would be pointless to consider any further properties. The agent can make this inference in two prominent cases: if S_j is a singleton or if S_j is a subset of a single indifference class (taking preferences as primitive in the latter case). The remaining cases are more problematic and ‘finite termination’ must be understood as an approximate description, as we will explain in section 7.

Example 1 (characteristics) In the spirit of Lancaster [18], we can recast the car example that opened the paper by viewing each car as a bundle of characteristics (horsepower, color, price, and so on). For any continuous characteristic, such as horsepower or price, there is a class of properties that we call ‘coordinate cutoffs.’ Suppose that there are two continuous characteristics and so $X = \mathbb{R}_+^2$. A coordinate cutoff is a property of the form $\{(x_1, x_2) : x_j \geq r\}$ or $\{(x_1, x_2) : x_j \leq r\}$ where j is the coordinate 1 or 2 and r is a real number. Coordinate cutoffs express categorical judgments about a single characteristic, e.g., if coordinate 1 is price and $P(1) = \{(x_1, x_2) : x_1 \leq 30,000\}$ then any car costing less than \$30,000 is ranked above any other car. For instance, suppose an agent has four coordinate cutoffs all of the form $\{(x_1, x_2) : x_j \geq r\}$ with cutoff levels as given in Figure 1 (the cutoff of property $P(i)$ is labeled $r(i)$). The choice function with this checklist maximizes a utility function: in Figure 1 the regions from worst to best are labeled 1 through 9.

Properties in \mathbb{R}_+^n of course do not have to be coordinate cutoffs. As a quick example, let coordinates 1 and 2 be two foods that make up a meal – say meat and potatoes. Then a property that placed a 800 calorie limit on the meal would be the ‘calorie cutoff’ $P(i) = \{(x_1, x_2) : k_1x_1 + k_2x_2 \leq 800\}$, where k_i is the number of calories per unit of food i . ■

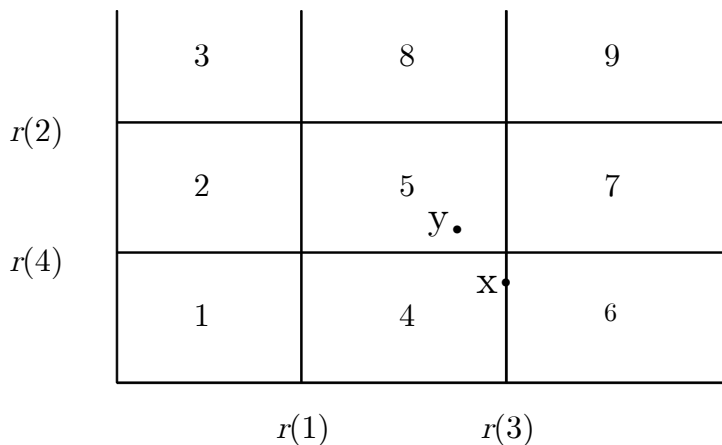


Figure 1: Coordinate cutoff preferences

The fact that the coordinate cutoff agent in Example 1 maximizes a utility function is no

fluke and is not due to the simplification that the agent deploys only finitely many properties.

Theorem 1 *If a choice function has a checklist then it maximizes a utility function.*

Thus an agent whose choices come from a checklist acts ‘as if’ he is maximizing a utility function. Of course the agent does not have to think about preferences or utility at all; the agent can just churn through his list of properties.

All proofs are in the appendix, but the argument behind Theorem 1 is simple. When a choice function has a checklist we can identify each $x \in X$ with a sequence of ‘ins’ and ‘outs’ that indicate in any coordinate i whether x is in or is not in property $P(i)$. Suppose we write down the ‘ins’ and ‘outs’ as a sequence of 1’s and 0’s respectively. When P has finitely many properties, use 0’s following the last property. For example, if P has four properties and $x \in P(1)$, $x \notin P(2)$, $x \notin P(3)$, $x \in P(4)$, the sequence for x is 1, 0, 0, 1, 0, 0, 0, Now we can read this sequence as the 0’s and 1’s of binary expansion of a number between 0 and 1; for the x above, this number is $\frac{1}{2} + \frac{1}{16} = .5625$. So, given a checklist P , each x has a sequence that defines a number $u(x)$ in the interval $[0, 1]$. Outside of a small class of exceptions, $u(x)$ can serve as the utility of x for an agent who uses P ! The reason is simply that $u(x) \geq u(y)$ if and only if, at the first digit where the sequences for x and y differ, x has a 1 and y has a 0 and therefore in the first property i that contains one of x and y but not both it is x that must be in $P(i)$.⁵

Since lexicographic preferences on \mathbb{R}_+^n cannot be represented by a utility function, we conclude from Theorem 1 that an agent who chooses with a checklist cannot have such preferences (even when the checklist consists of infinitely many coordinate cutoffs). Checklist users, who at first glance seem not to make trade-offs, turn out to fit the most traditional model of economic rationality.

In the next section, we will put a finer point on the problematic feature of choice behavior that maximizes lexicographic preferences on \mathbb{R}_+^n . As we will see, it is easy to extend the checklist model to cover such behavior. Rather the problem is that choices that maximize lexicographic preferences on \mathbb{R}_+^n can be produced only by checklists that fail to terminate

⁵The exceptions are numbers in $[0, 1]$ with two binary representations, but this difficulty can be bypassed by reading the sequence of 0’s and 1’s as a number written in base 3 (see the proof of Theorem 1).

in finitely many steps and are therefore unwieldy. As we argued in the introduction, the absence of a utility function detracts from decision-making efficiency.

Theorem 1 leaves some important questions unanswered. While lexicographic preferences on \mathbb{R}_+^n behavior cannot arise from the standard checklists of this section, what types of choice behavior can? That of all utility maximizers or just certain types? In particular, could the utility-maximizing choices of textbook economic consumers arise from standard checklists? Since such agents have uncountably many indifference classes, it might seem that they are in the same boat as agents with lexicographic preferences on \mathbb{R}_+^n and that their choices could not be the outcome of the practical checklists that finitely terminate. This turns out not to be the case: as we will see in section 6, the choice behavior of any utility-maximizer is the outcome of some standard checklist (putting domain qualifications aside for the moment). But the following example lays out the problem that the obvious way to build a checklist that generates a preference relation employs far too many properties to be applicable to textbook consumer choice.

Example 2 Suppose an agent uses the checklist $P(1), P(2), \dots, P(n)$ where the properties form a partition of X (each $x \in X$ is in exactly one property). It is easy to see that this agent also maximizes a preference relation \succsim with n indifference classes, $P(1), \dots, P(n)$ going from best to worst: given a choice set A , no eliminations occur until the property $P(i)$ that contains the elements of A that have the lowest property index and at that stage all other elements of A are eliminated. Notice that we could omit the last property $P(n)$ without changing the choice function that results. ■

Example 2 makes clear that the choices of any utility maximizer with finitely many indifference classes could be the outcome of a checklist that finitely terminates. But the construction in Example 2 uses the same number of properties as the number of indifference classes in \succsim (or one fewer). The decisions made by the textbook agents of consumer theory, who have uncountably many indifference classes, therefore could not arise from a standard checklist that follows the Example 2 model: textbook consumers have too many indifference classes and if unsatiated have no top indifference class to begin the checklist. We will see, however, that there are different constructions that do provide standard checklists in these

cases.

The same ‘slowness’ feature of Example 2 also raises the possibility that even agents who make finitely many discriminations can be inefficient decision-makers who proceed through a large number of properties. We address these questions in sections 5 and 6.

4 Extended checklists and preference maximization

We now present a more abstract version of the model, allowing sequences of properties that go beyond the ordinary counting numbers; the new version opens the door to checklists for arbitrary preference-maximizing behavior including the lexicographic cases that fail to have the standard checklists of section 3. This section is more technical; since we will not refer back to these ideas until section 8, it can be skipped.

In our earlier elimination procedure, each set of survivors $S_i(A)$ is a subset of its immediate predecessor $S_{i-1}(A)$. Since therefore $S_{i-1}(A) = \bigcap_{k < i} S_k(A)$, we could equivalently define the elimination by

$$S_0(A) = A$$

$$S_i(A) = \begin{cases} \bigcap_{k < i} S_k(A) \cap P(i) & \text{if } \bigcap_{k < i} S_k(A) \cap P(i) \neq \emptyset \\ \bigcap_{k < i} S_k(A) & \text{otherwise} \end{cases}$$

for each $i > 0$. The new definition has the advantage that it can be applied to ‘longer’ sets of properties: we can weaken the assumption that the indices I in a checklist are a set of natural numbers and suppose instead that I is well-ordered by some \leq , letting 0 be the least element of I .⁶ The assumption that I is well-ordered implies that each $i \in I$ has an immediate successor; thus the procession through the checklist of properties remains orderly. For an arbitrary well-ordered I , the above definition employs a variant of standard induction (transfinite induction) to specify each $S_i(A)$ as a function of its entire set of predecessors and $P(i)$.

⁶A set B is *well-ordered* by \leq if \leq is a linear order (a complete, transitive, and antisymmetric relation) on B such that every nonempty subset of B has a least element b : $b \leq x$ for all $x \in B$. See Halmos [13] for the set theory concepts we use in this section.

We say that a choice function c has an **extended checklist** if c satisfies Definition 1 except that the $S_i(A)$ are defined as above and I is permitted to be any well-ordered set whose least element is 0.⁷ The terminal step j continues to be defined as in Definition 1 but now need not be finite. Any of our earlier checklists, which we call ‘standard,’ qualifies as an extended checklist, and conversely, if c has an extended checklist that ‘finitely terminates’ – for each $A \in \mathcal{A}$, the index j identified in Definition 1 is finite – then c has a standard checklist since then we can excise all but the properties with finite indices.

The main advantage of extended checklists is that they give an exact characterization of preference maximization.

Theorem 2 *A choice function has an extended checklist if and only if it maximizes a preference relation.*

Since ‘having an extended checklist,’ as an assumption on choice functions, is equivalent to preference maximization, it is equivalent to any characterization of preference maximization for choice functions. For example, it is equivalent to the Richter [22] version of the strong axiom of revealed preference. Notice also that Theorem 2 imposes no restriction on the domain of the choice function; for example, it applies equally to budget sets in consumer theory and to finite sets.

Regarding the ‘only if’ half of Theorem 2, it is easy to detail the preference relation that an agent with an extended checklist implicitly maximizes. Recall from section 3, that when a choice function has a checklist, we can identify each x with the sequence of ‘ins’ and ‘outs’ that indicate in any coordinate i whether x is in or is not in property $P(i)$. We then declare $x \succsim y$ if the x and y sequences are identical or x scores an ‘in’ at the first coordinate where the sequences differ. Now if a eliminates b in the checklist’s sequential eliminations – that is, if both a and b have survived to some stage $i - 1$ but only a survives to stage i – then $a \succ b$ since the first property that has only one of a and b must have a and not b . So if x is \succsim -maximizing on some A then x could never be eliminated, and conversely if x is chosen

⁷In terms of ordinal numbers, the distinction between standard and extended checklists is that in the former case I must equal (technically, be order isomorphic to) an ordinal number equal to ω or less while the latter case imposes no restriction. Also, notice that if we apply an arbitrary well-ordered set of properties to a choice set A , it could happen that $S_i(A)$ is empty for some i (when $\bigcap_{k < i} S_k(A) = \emptyset$). But if c has an extended checklist then this possibility does not arise since we require $c(A) \neq \emptyset$ for $A \in \Sigma$.

from some A then $x \succ y$ must obtain for every y in A that is not chosen. So the choice function indeed maximizes the \succsim we have defined.⁸

For the ‘if’ half of Theorem 2, suppose we are given a choice function c that maximizes a preference relation \succsim . We can then build an extended checklist from a familiar item, the weakly-better-than sets of the preference relation \succsim : for each $x \in X$, set a property P_x equal to $\{y \in X : y \succsim x\}$, ignoring the duplicates that arise when $x \sim x'$. We then list in arbitrary order – technically, we well-order – these properties to form an extended checklist. When this checklist is applied to some A , the agent will eventually hit a property P_x where $x \succsim y$ for all $y \in A$, whereupon no further eliminations occur. The upper-contour properties used here are less natural than the properties we have seen so far in examples. But the upper contours do not have to be taken literally as properties that an agent would actually employ. As we will see in Proposition 1, any choice function that has a checklist (perhaps a checklist with ‘natural’ properties) also has a checklist that consists of upper contours; and for characterization purposes a checklist of upper contours is the easier object to work with.

Given Theorem 1, the checklists that produce choices that maximize lexicographic preferences on \mathbb{R}_+^n must be extended rather than standard. The following example shows what they look like.

Example 3 (characteristics revisited) To eliminate the puzzle of checklists and lexicographic preferences on \mathbb{R}_+^n , suppose all of the properties in Example 1 are coordinate cutoffs of the form $\{(x_1, x_2) : x_j \geq r\}$, as in Figure 1. So if, e.g., $P(1) = \{(x_1, x_2) : x_1 \geq r\}$ then any bundle with $x_1 \geq r$ is ranked above any bundle with $x_1 < r$ according to the preferences maximized by any c that has $P(1)$ as its first property. If we increase the number of properties and let the cutoff levels ‘fill in’ each axis (become dense in \mathbb{R}_+), we approach preferences that have a strictly increasing utility function. As Theorem 1 showed, no matter how many coordinate cutoff properties an agent uses in a standard checklist, the preferences that result cannot lexicographically rank bundles first according the level of coordinate 1

⁸A less general argument works via the weak axiom of revealed preference (WARP). A choice function with an extended checklist must satisfy WARP since if x is chosen when y is available it must be that if there is a first property $P(i)$ that contains either x or y but not both then $P(i)$ contains x , hence if y is chosen from any S that contains x then x must be chosen too. So on any domain where WARP implies that a choice function maximizes some preference relation, for example the finite subsets of X , a choice function with a checklist must also maximize a preference relation.

and second according the level of coordinate 2. But there is an extended checklist for such preferences that uses coordinate cutoffs: begin with a countably infinite set of properties of type $\{(x_1, x_2) : x_1 \geq r\}$ (where the r 's for these properties are dense in \mathbb{R}_+) and then proceed to a countably infinite set of properties of type $\{(x_1, x_2) : x_2 \geq r\}$ (again with the r 's dense in \mathbb{R}_+).

Having established that standard checklists with dense coordinate cutoff properties cannot lead to lexicographic preferences on \mathbb{R}_+^n , it would be interesting to know what type preferences they do in fact generate. As long as the cutoffs are of the $\{x : x_j \geq r\}$ form, it is easy to confirm that the preferences that result will be strictly monotone but we do not have an exact characterization. Observe also that even before becoming dense, coordinate cutoff properties rule out choice functions that maximize continuous preferences. Consider for example a sequence of points in region 4 in Figure 1 that converges to x . Any point in this sequence is strictly dispreferred to y in region 5 and yet the limit point x is strictly preferred to y . The failure of continuity persists no matter how many additional cutoffs we add.

Example 3 indicates that the problem with lexicographic preferences on \mathbb{R}_+^n is not that they cannot arise from checklists but only from checklists that are problematic from a procedural standpoint. Utility maximization in consumer theory, even though it typically involves uncountably many utility levels, does not suffer the same difficulty, but so far the only checklists we have seen that can generate such behavior (in the 'if' half of Theorem 2) are extended and hence need not finitely terminate.⁹ To show that tractable checklists can lead to textbook consumer behavior, in section 6 we find replacement checklists that are standard.

Mathematically, the trouble with lexicographic preferences on \mathbb{R}_+^n is not lexicography per se. Theorem 2 reported that any preference-maximizing choice function has a checklist, and a checklist embeds an ordering on X into the lexicographic ordering of $\{0, 1\}^I$ where I is a well-ordered set of indices.¹⁰ Lexicographic preferences on \mathbb{R}_+^n on the other hand coincide

⁹The problem shows up in the proof of Theorem 2 when we take the nonconstructive step of well-ordering the upper contour sets to create the extended checklist.

¹⁰Here 1 and 0 play the same role that 'in' and 'out' did in our earlier discussions: a I -sequence a is ranked strictly higher than I -sequence b if, at the first coordinate where a and b the sequences differ, a registers a 1.

with the lexicographic ordering of the n -fold copy of \mathbb{R}_+ . While the existence of an extended checklist for lexicographic preferences on \mathbb{R}_+^n shows that we can embed these preferences into a lexicographic ordering that uses $\{0, 1\}$ rather than \mathbb{R}_+ , the cost is that the set of indices must expand beyond the natural numbers, which is the mathematical counterpart of being procedurally unwieldy. We comment further on this trade-off in section 8.

5 Quick checklists 1: agents who finitely discriminate

Since checklists make for practical decision-making procedures only when the elimination of options concludes after finitely many steps, we need to pin down which preferences can arise from our benchmark model of ‘standard’ checklists. (Until further notice, all checklists will now be standard rather than extended.) Although finite termination might seem incompatible with preferences with uncountably many indifference classes, we will see that any case of utility-maximizing decision-making can be the outcome of a checklist that finitely terminates.

As Example 2 made clear, choice behavior that marks out a finite number of discriminations can always be the consequence of a checklist that finitely terminates. But finite termination is then too weak a test of practicality. Recall the worst-case scenario of the checklists in Example 2 where the number of properties equals the number of preference discriminations. An agent that uses such a checklist would spend a long time eliminating alternatives before coming to a decision, ending up with a procedure that is plodding and profligate instead of fast and frugal.

This section and the next address these points. Can a checklist make a given finite number of discriminations reasonably quickly? And can the agents of consumer theory use a checklist at all?

Our measure of decision-making speed is the number of properties per preference discrimination. While a checklist user could of course use a single property and thus make decisions rapidly, such an agent would be dividing the universe of alternatives X coarsely, into just two indifference classes. So instead we ask how the potential number of preference discriminations n varies as a function of the number of properties that an agent uses or,

equivalently, how many properties are needed to make n discriminations.

We define a checklist P to *make n discriminations* if there is a choice function c that maximizes a preference relation with n indifference classes and P is a checklist for c . To ensure that there are not multiple values for n that meet this definition, we assume in this section that the domain of choice sets \mathcal{A} includes every two-element subset of X . Notice that a checklist that makes n discriminations can make a final selection from some choice sets before coming to its final property; our measure of speed – the number of properties that will make n discriminations on the entire domain \mathcal{A} – therefore might not be appropriate if an agent has enough information about which choice sets he is likely to face. We briefly address expected decision-making speed at the end of this section.

To see how many properties are needed for a checklist to make n discriminations, let's label the indifference classes $1, \dots, n$, going from worst to best. Since the number of properties required depends only on the number of indifference classes, we may as well take each indifference class to be a singleton. The agent does not 'begin' with these indifference classes or even have to recognize their existence: $1, \dots, n$ merely denote the indifference classes of the preference that the checklist we now construct implicitly maximizes. With this notation, our question becomes, for $X = \{1, \dots, n\}$, how many properties are needed so that the choice function that results divides X into n indifference classes? Luckily the $n - 1$ answer given in Example 2 fails to be the minimum when $n > 3$.

Consider, as an example,

$$X = \{1, 2, 3, 4\}.$$

Given our labeling convention, the choice function c must maximize the usual order \geq on integers. It is easy to see that $P(1) = \{4, 3\}$, $P(2) = \{4, 2\}$ is a checklist for c . Following Theorem 3, we show how a checklist like this might arise.

Next, consider

$$X = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

with c again maximizing \geq . Define the checklist $P(1) = \{8, 7, 6, 5\}$, $P(2) = \{8, 7, 4, 3\}$, $P(3) = \{8, 6, 4, 2\}$. Again, it is easy to verify that this is a checklist for c . (It suffices to consider just the two-element subsets of X .)

Notice how the first example is nested in the second: the last two properties $P(2)$ and $P(3)$ of the second example treat $\{5, 6, 7, 8\}$ and $\{1, 2, 3, 4\}$ just as P in the first example treats $\{1, 2, 3, 4\}$, with the additional first property $P(1)$ serving only to separate the two chains. So, we have provided a checklist with two properties makes four discriminations, and a checklist with just one additional property that makes twice as many discriminations. This conclusion extends inductively:

Theorem 3 *If X contains at least n alternatives then there is a checklist that makes n discriminations with k properties, where k is the smallest integer such that $2^k \geq n$. Furthermore, any checklist that makes n discriminations must have at least k properties.*

The properties above that illustrate the proof of Theorem 3 may seem convoluted but they arise in natural ways. Suppose an agent makes decisions by proceeding through a series of attributes where each attribute forms a ‘linear’ scale that distinguishes between any pair of alternatives and is viewed by the agent as either a good or a bad. For cars, two examples of such linear attributes would be price and horsepower; roughly speaking, no two cars have the same price or horsepower. If each property in an agent’s checklist is based on a linear attribute and the agent seeks to make decisions quickly, then the properties must look like the ones we have provided. The first property, based on the first attribute, merely has to divide X into two parts in such a way that each alternative in part $P(1)$ has more of the first attribute than each alternative in the other part $X \setminus P(1)$. The second property must do more since the agent will want $P(2)$ to discriminate both among alternatives that are in $P(1)$ and among alternatives that are in not in $P(1)$; otherwise $P(2)$ would not be doing enough discriminatory work. So $P(2)$ must divide both $P(1)$ and $X \setminus P(1)$ into two parts, where again each alternative in one part has more of the second attribute than each alternative in the other part. And so on with any further properties.¹¹ For an agent who chooses in this way, the relationship between the number of indifference classes n that the agent implicitly displays and the number of properties k in the agent’s checklist will be given by Theorem 3.

¹¹This construction is laid out in more detail in Mandler [19].

Theorem 3 shows how checklists become more and more efficient as the number of preference discriminations increases: the maximum number of preference discriminations n is an exponential function of the checklist length k . Or, if we take n as primitive, then the minimum number of properties required is a less-than-polynomial function of n and hence the ratio of the minimum number of properties to n falls to zero as n increases. Since $2^{20} \geq 1,000,000$, Theorem 3 explains why a million preference discriminations require only twenty checklist properties.

We can compare the efficiency of a checklist to other choice procedures that make the same number of discriminations. Suppose an agent with n indifference classes wants to find the highest indifference class in a choice set; in the notation of the above examples, the agent seeks out, given $A \subset \{1, \dots, n\}$, the largest integer in A . The solution of this problem via ‘yes or no’ questions is a classic illustration of binary search: first ask ‘does A contain an integer between $\frac{n}{2}$ and n ?’, and then, if yes, ask ‘does A contain an integer between $\frac{3n}{4}$ and n ?’ and, if no, ask ‘does A contain an integer between $\frac{n}{4}$ and $\frac{n}{2}$?’, and so on. That a recursive computer program, where the choice of the i th question depends on earlier answers, can execute this algorithm in $\lceil \log_2 n \rceil$ steps is hardly news ($\lceil x \rceil$ denotes the least integer $\geq x$).¹² What is notable about a checklist is that it executes the algorithm nonrecursively. A property $P(i)$ does not change as a function of the eliminations that occur prior to i , and every property is used for every A . To do without input from earlier steps, each property in effect encodes a set of questions. Consider again $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and let m denote $\max A$. Then $P(1)$ ‘asks’ one question, ‘is $m \in \{8, 7, 6, 5\}$?’, $P(2)$ ‘asks’ two conditional questions, ‘if $m \in \{8, 7, 6, 5\}$ then is $m \in \{8, 7\}$?’ and ‘if $m \notin \{8, 7, 6, 5\}$ then is $m \in \{4, 3\}$?’, and $P(3)$ ‘asks’ four conditional questions. For $i > 1$, the eliminations prior to i ensure that only one of the antecedents of the $P(i)$ questions is satisfied. Property $P(i)$ therefore asks the right question, and without recursive instructions or an exhaustive tree of $n - 1$ ‘if then’ commands (where each answer to a command leads to a distinct subsequent command).

An optimal tree of ‘yes or no’ questions can in principle outperform a checklist. Suppose we can ask questions of the form ‘does A intersect $Y \subset \{1, \dots, n\}$?’. Then, depending on the

¹²See, e.g., Knuth [17], chapter 6, Theorem B.

probabilities that particular integers lie in A , the minimum expected number of questions can be less than $\lceil \log_2 n \rceil$. For example if it is highly likely that $m = \max A = 4$, then one can first ask ‘does A intersect $\{5, 6, 7, 8\}$?’ and if not ‘does A intersect $\{4\}$?’ But if each $x \in X$ is equally likely to be m then $\lceil \log_2 n \rceil$ is the minimum expected number of questions: the optimal tree does no better than the optimal checklist.¹³

6 Quick checklists 2: classical utility maximizers

Checklists with a finite number of properties are appealingly concrete: there is a *uniform* upper bound on the number of properties the decision maker has to examine before the choice procedure terminates. When checklists are not restricted to be finite (but are still standard rather than extended), it remains true that each choice set needs to be checked against only finitely many properties but there might not be any bound on the number of properties that serves simultaneously for *all* choice sets. This small difference gives checklists much greater reach when they are not required to be finite. Indeed, we will now see that, subject to a domain restriction, for any case of utility-maximizing choice behavior there is always a checklist that generates that behavior. For utility functions with uncountably many utility levels, such as those found in textbook consumer theory, these checklists are ‘quick’ in that the number of properties the agent must check is a negligible fraction of the infinite number of preference discriminations that the agent implicitly makes.

Example 3 illustrates how a quick checklist for a classical utility function might arise. Let X equal \mathbb{R}_+^2 and suppose, as in Example 3, that properties are the coordinate cutoffs $\{(x_1, x_2) : x_1 \geq r_1\}$ and $\{(x_1, x_2) : x_2 \geq r_2\}$, where r_1 and r_2 each vary over a countable set dense in \mathbb{R}_+ : when these properties are placed in some order, they define a checklist P . On the domain \mathcal{A} of all finite sets in X the checklist P defines a choice function c since for any pair of distinct bundles y and z there must be a coordinate j such that y_j and z_j differ; so there will be a property that contains one of y and z but not both. By Theorem 1, c must maximize a utility function and thus we have a checklist for a utility function with

¹³If questions of the form ‘is $m \in Y$?’ are permitted, which is exactly the game ‘Twenty questions,’ Huffman coding [15] generates the optimal tree. See also Zimmerman [30], and Gilbert [11] for the connection to our problem.

uncountably many utility levels. Of course an agent who chooses in this way never has to think about the preferences or utility he is implicitly maximizing.

This illustration only goes so far: while it shows that a checklist that generates uncountably many utility levels can arise from coordinate cutoffs, for an arbitrary utility on \mathbb{R}_+^n there might not be a coordinate-cutoff checklist that maximizes that utility. Subject to a domain qualification, however, we can at least conclude that any choice function that maximizes a utility does have some checklist. When the domain consists of finite sets, the reason is simple: given a choice function that maximizes a utility function u , we may define a checklist P by setting, for each rational number r , the property $P_r = \{x \in X : u(x) \geq r\}$ and then listing these properties in any order. The u -maximal alternatives in a A will never be eliminated: if at any stage i the set of survivors from the previous rounds contains some alternatives that are in $P(i)$, then the u -maximal alternatives must be among them. Conversely, any z in A that is not u -maximal will eventually be eliminated by a P_r such that r lies strictly between $u(z)$ and the maximum utility achieved by the alternatives in A . This checklist moreover has a procedural interpretation if we view each r as a numerical satisfaction threshold. At each stage, the consumer keeps only the alternatives that are ‘satisficing’, if any, and otherwise he keeps all of them. Then in the next stage he modifies the satisfaction threshold.

To give an exact characterization of the choice functions with checklists, let \succsim be a preference relation and let $A \subset X$ be called \succsim -top dense if and only if A contains a \succsim -maximal element x and for all y with $x \succ y$ there exists $z \in A$ such that $x \succ z \succ y$. If \mathcal{A} is a domain of choice sets, we will say that (\mathcal{A}, \succsim) satisfies the *domain restriction* if and only if $\{x \in X : x \text{ is a } \succsim\text{-maximal element of some } \succsim\text{-top dense } A \in \mathcal{A}\}$ intersects at most countably many \succsim -indifference classes. The pair (\mathcal{A}, \succsim) will necessarily satisfy the domain restriction if \mathcal{A} is a family of finite sets or if \succsim has countably many indifference classes.

Theorem 4 *A choice function c on a domain \mathcal{A} has a checklist if and only if there exists a utility u that c maximizes such that (\mathcal{A}, \succsim) satisfies the domain restriction, where u represents \succsim .*

Theorem 4 shows the reach of checklists. They can generate utility-maximizing behavior

that divides the universe of alternatives into a continuum of indifference classes and yet still eliminate the inferior alternatives from a choice set in a finite number of steps. Theorem 4 also slightly refines Theorem 1: if a choice function c has a checklist, we now know not only that c maximizes a utility but also that the domain restriction holds.

The following example illustrates why the domain restriction is needed: in the absence of a restriction, there are utility-maximizing choice functions that do not have a checklist. Given a preference relation \succsim , let an *upper contour* of \succsim be a set of the form $\{x \in X : x \succsim z\}$ for some $z \in X$.

Example 4 Let X be the interval $[0, 1]$, let the domain \mathcal{A} of c be the closed sets in X , and let c maximize the preference \geq and hence the utility defined by $u(x) = x$. Since each closed interval in X is \geq -top dense, (\mathcal{A}, \geq) fails to satisfy the domain restriction. But suppose nevertheless there is a checklist P for c . Since \geq is the only preference relation that c maximizes, the Proposition below lets us assume that the checklist consists only of properties $P(i)$ that are \geq -upper contours. That is, if there is a checklist \hat{P} for c then there is also a checklist P for c that consists solely of \geq -upper contours.

Assume then that there is a P that is a checklist for c that consists of \geq -upper contours. If $\min P(i)$ denotes the \geq -minimal element of $P(i)$, then there will be at most countably many $\min P(i)$ for the properties in P . Pick some $y \in X$ that is not one of these $\min P(i)$, and set $A = \{x \in X : x \leq y\}$. Then, for any i , $S_i(A)$ will equal the closed interval whose lower boundary equals $\max\{\min P(k) : \min P(k) < y \text{ and } k \leq i\}$ and whose upper boundary equals y . Since $S_i(A) \neq \{y\} = c(A)$ for all i , P could not in fact be a checklist for c . ■

That we may take a checklist in Example 4 to consist solely of upper contours reflects a wider principle that is essential for the proof of Theorem 4.

Proposition 1 (canonical checklists) *If c has a checklist then there is a preference relation \succsim such that c maximizes \succsim and has a checklist that consists of upper contours of \succsim .*

The preference \succsim whose existence is assured by the Proposition is in fact the ordering, mentioned in section 3 and used in several of our proofs, that identifies each $x \in X$ with

the sequence that assigns a 1 to coordinate i if x is in property $P(i)$ and otherwise assigns a 0, and then declares x better than y if, in the first coordinate where the x and y sequences differ, x scores a 1. Choice functions can have a vast array of checklists; the availability of canonical checklists allows us to narrow our focus to a particular highly tractable checklist, which makes a proof of Theorem 4 feasible.¹⁴

7 Utility maximizers always have quick approximate checklists

As we have seen, the choice behavior of utility maximizers does not coincide exactly with that of agents who use a checklist (a domain restriction is necessary), nor of agents who use an extended checklist (since then we go beyond utility maximization to preference maximization). The following picture summarizes our findings so far.

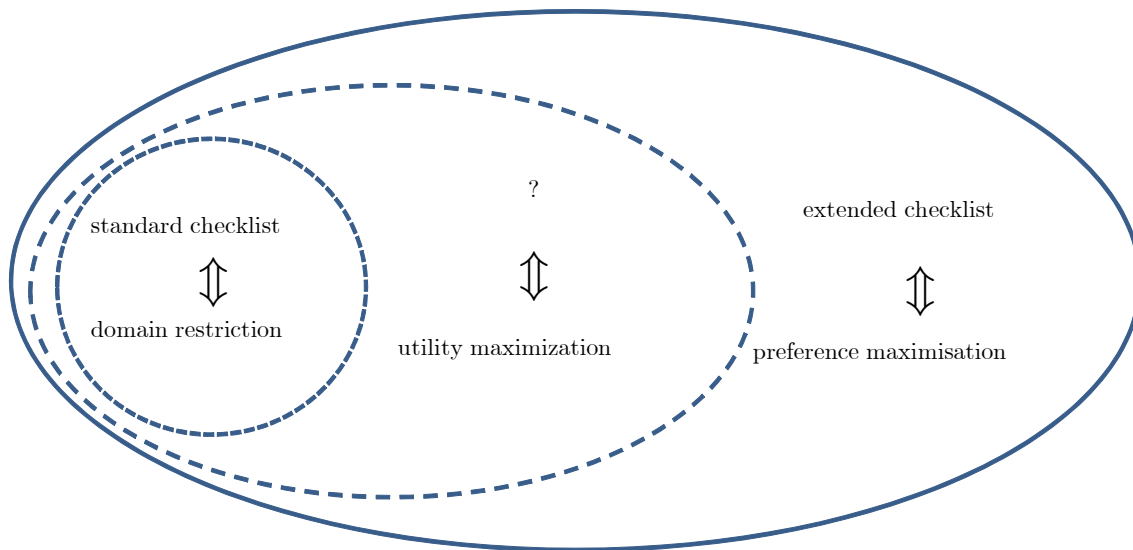


Figure 2: Relationships between various checklist notions

¹⁴Consistent with the Lemma, there can be a c and a \succsim such that c has a checklist and maximizes \succsim and yet not have a checklist of \succsim -upper contours; \succsim can even have a utility representation. For an example, define for any $r \in \mathbb{R}$ the set $A(r) = \{r\} \cup \{x \in \mathbb{Q} : x < r\}$, and set $\mathcal{A} = \{A(r) : r \text{ is irrational}\}$. A c on this domain that always selects an irrational number has the checklist $P(1) = \{\text{all irrational numbers}\}$. Although c also maximizes the utility given by $u(x) = x$ there can be no standard checklist for c consisting of u upper contours since for each irrational r we would have to include $\{x \in \mathbb{R} : x \geq r\}$ in the checklist (as in Example 4).

In this section we resolve the question mark in the picture by showing that checklists can closely *approximate* utility maximization regardless of the domain.

To capture the idea that a checklist can approximate the decision $c(A)$ we consider the limit of the set of survivors selected by a checklist: although the procedure never yields exactly the decision $c(A)$ at any finite step, it approximates $c(A)$ more and more accurately as the number of steps increases. In the limit, we get exact equivalence between the choices of checklist users and utility maximizers.

As no notion of distance is present in our set-up, we use a set-theoretic definition of the convergence of the $S_i(A)$. A choice function c on the domain \mathcal{A} has an **approximate checklist** if and only if there is a checklist P such that, for all $A \in \mathcal{A}$,

$$c(A) = \bigcap_{i \in I} S_i(A),$$

where the $S_i(A)$ are defined from P as in section 2. Although after any finite number of steps the set of survivors may still contain other alternatives beside the chosen ones, it is only the chosen alternatives that survive all steps of elimination: for any alternative rejected by the choice function, there exists a property that it does not have.

Theorem 5 *A choice function maximizes a utility function if and only if it has an approximate checklist.*

Approximate checklists help explain how a checklist that has the entire set of natural numbers as its set of indices would work practically. Such checklists can raise a termination problem: even when no further eliminations occur after some property $P(j)$, the agent may not know this fact. The agent will know it for choice functions that always select singletons or subsets of a single indifference class (see footnote 4). But in all other cases, the practical distinction between ordinary and approximate checklists is not sharp. For both models, the agent would have to declare at some point that the set of alternatives has been winnowed down adequately.

8 Multivalued properties and the representation of preferences

While so far we have focussed on checklists as decision-making procedures, they can also be seen as a representation device for preferences. This section explores this possibility and the connection to Chipman [4]’s theory of lexicographic utility.

We can rephrase our initial model of standard checklists by replacing each property $P(i)$ with the indicator function of $P(i)$ (that is, the function $u_i : X \rightarrow \{0, 1\}$ with $u_i(x) = 1$ if and only if $x \in P(i)$) and redefining $S_i(A)$ to equal $\arg \max u_i(x)$ s.t. $x \in S_{i-1}(A)$ for all $i > 0$. Each of these newly defined $S_i(A)$ will coincide with our originally defined $S_i(A)$. For the more general case of extended checklists, we can instead use $S_i(A) = \arg \max u_i(x)$ s.t. $x \in \bigcap_{k < i} S_k(A)$ for $i > 0$.

This reformulation suggests replacing the u_i above with functions that have a larger range (‘multivalued properties’). Among the prominent possibilities, we could admit any u_i that maps to a finite set with at least two elements, or any u_i that maps into \mathbb{R} . Indeed we could go one step further and instead of functions, use a complete and transitive relation R_i on X , and set

$$S_i(A) = \left\{ x \in \bigcap_{k < i} S_k(A) : x R_i y \text{ for all } y \in \bigcap_{k < i} S_k(A) \right\} \quad (2)$$

for $i > 0$. This last proposal is evidently the most general. Given any well-ordered set of indices I with least element 0 and a complete and transitive R_i for each $i \in I$, we call $\{R_i\}_{i \in I}$ a **multivalued checklist**. Without loss of generality we now represent each I by the ordinal number it is equivalent to. This definition of a multivalued checklist does *not* require $I \leq \omega$ (where ω denotes the ordinal number for \mathbb{N}). So multivalued checklists generalize extended rather than standard checklists. If each R_i has at most two indifference classes (our original model) we say $\{R_i\}_{i \in I}$ is *two-valued*, if the number of indifference classes of each R_i is finite we say $\{R_i\}_{i \in I}$ is *finite-valued*, and if each R_i has a real-valued utility representation we say $\{R_i\}_{i \in I}$ is *real-valued*.

With the $S_i(A)$ given by (2), we can define $\{R_i\}_{i \in I}$ to be a multivalued checklist for a choice function c by applying Definition 1.

Theorem 2 extends to any multivalued checklist: a choice function c has a multivalued checklist if and only if it maximizes a preference relation. The ‘if’ direction follows from the original argument we gave for Theorem 2. For the ‘only if’ direction, some minor adjustments to the proof of Theorem 2 show that if $\{R_i\}_{i \in I}$ is a multivalued checklist for c then c maximizes the binary relation \succsim_L on X that is induced by the lexicographic ordering of the R_i , that is,¹⁵

$$x \succsim_L y \Leftrightarrow [(yR_ix) \Rightarrow (\exists k \leq i \text{ with } xR_ky)]. \quad (3)$$

Theorem 2’s applicability to multivalued checklists suggests their use as a representation device. One way to proceed is to say that a multivalued checklist $\{R_i\}_{i \in I}$ represents the preference relation \succsim if $\{R_i\}_{i \in I}$ is a checklist for the choice function c , defined on say the finite subsets of X , that maximizes \succsim . But it is equivalent and simpler to omit any mention of choice functions and just say that a multivalued checklist $\{R_i\}_{i \in I}$ **represents the preference relation** \succsim if $\succsim = \succsim_L$ (as defined by (3)). Requiring that a checklist is n -valued (for $n = \text{two, finite, real}$) provides a correspondingly more restrictive definition of representation. Evidently there is a trade-off, already mentioned at the end of section 4, between the upper bound on the number of indifference classes (n) and the number of multi-valued properties (I): to represent a given \succsim , a smaller n can necessitate a larger I .

A real-valued checklist is the definition of representation that Chipman [4] proposed in his classical work on utility theory.¹⁶ To see that Chipman picked a n that deftly judged the trade-off, observe that with no restrictions on the admissible R_i , multivalued checklists can be trivial and have no value for representation purposes: any preference relation \succsim can be represented by the multivalued checklist that consists of the single relation \succsim . Moreover, there are preference relations that can be ‘concisely’ represented by a real-valued checklist – that is, I can be small – but that have neither a classical utility representation nor a ‘concise’ finite-valued checklist. The pertinent example is of course the lexicographic ordering on \mathbb{R}_+^2 , which can be represented by a real-valued checklist with $I = 2$ but where any finite-valued checklist representation must have a $I > \omega$ (this conclusion follows from Theorem 1). Thus

¹⁵This extension of Theorem 2 would not hold if the R_i were not required to be complete and transitive. See Apesteguia and Ballester [1], Houy and Tadenuma [14], and Manzini and Mariotti ([20], [21]).

¹⁶We thank Chris Tyson for stressing the connection between our work and Chipman’s. For a survey of Chipman’s theory and related developments, see Fishburn [10].

real-valued checklists are restrictive enough to be useful but not so restrictive that they are always unwieldy. In fact, Chipman’s construction would lose most of its value if we added even the smallest additional restriction on the admissible R_i , that each must have only countably many indifference classes: one may show that any such ‘countably-valued’ checklist that has an index set $I \leq \omega$ must represent a preference relation that could also be represented by a classical utility function. Therefore, to get a concise ($I \leq \omega$) representation when a classical utility is unavailable, a real-valued checklist is required.

In our terminology, the main theorem in Chipman [4] states that any preference relation \succsim can be represented by a real-valued checklist. Theorem 2 implies this result. Indeed, Chipman’s proof uses utility functions with ranges that take on two values; thus, he implicitly showed that any \succsim can be represented by a two-valued checklist, which is the content of Theorem 2.¹⁷ Outside of Theorem 2, our results do not intersect with lexicographic utility theory, for the very reason that we restrict the range of the admissible R_i . The range restriction indeed exposes a rich structure hiding inside Chipman’s theory; for example, the capacity of a two-valued checklist to make a finite number of preference discriminations that is an exponential function of the number of properties (see Theorem 3) has no parallel in the theory of real-valued checklists, since in the latter case one R_i can by itself make infinitely many discriminations.

Finally, we note that our original model of two-valued checklists performs *reasonably* well as a representation tool. Chipman [5] showed that there are preferences relations that can be represented by only those real-valued checklists $\{R_i\}_{i \in I}$ that use a I that is uncountable. Since Theorem 2 applies to such preference relations, they can be represented by two-valued checklists – as Chipman himself makes clear – though of course I must again be uncountable. Conversely, if a preference relation \succsim can be represented by a real-valued checklist $\{R_i\}_{i \in I}$ with a set of indices $I \leq \omega$ then one may show that \succsim can also be represented by a two-valued checklist where I is countable (though I may have to be larger than ω). Real-valued checklists still have an edge: as lexicographic preferences on \mathbb{R}_+^2 attest, there are preference relations \succsim that can be represented by a real-valued checklist such that I is finite but

¹⁷This result precedes Chipman in the mathematical literature on ordinal numbers, see Cuesta Dutari [6], [7] and Sierpinski [24].

where the only two-valued checklists that represent the same \succsim have to use a I greater than ω – though I can at least remain countable.¹⁸ Of course it is this ‘drawback’ of two-valued checklists that guarantees the tight connection between their tractability as a decision procedure – that they terminate after finitely many steps – and utility maximization. Two-valued checklists have to use a set of properties that goes beyond ω to represent a \succsim in just the cases where \succsim has no utility representation.

9 Concluding remarks

Since Simon’s [25] contribution, we have been used to thinking of ‘procedural rationality’ as entirely separate from, and even opposed to, ‘substantive rationality.’ This paper leads to a different view. We have considered a tractable, realistic procedure that can underpin utility maximization, blurring Simon’s distinction. While this procedure is by no means the only possible one for a procedural agent, it is tractable and realistic in many contexts, as demonstrated by its popularity with psychologists.

There are other ways to aggregate properties and for each method one may investigate whether the procedure generates rational behavior. For at least some procedures, the moral of this paper – that realistic procedures can mask utility-maximization – will continue to apply. For example, suppose an agent always chooses those alternatives that satisfy the most properties. Then, for any given standard checklist on some finite domain, the choice function that results will maximize the rational preference \succsim defined by $x \succsim y$ if and only if the number of properties that x has is at least as great as the number of properties than y has. For a converse, if we are given a preference \succsim then we can construct a checklist whose choices maximize \succsim by including, for each y , a property $P_y = \{x \in X : x \succsim y\}$. One can proceed similarly for utility maximization and nonfinite domains (with the necessary technical adjustments). The same conclusions hold if we were to aggregate by a weighted sum of the number of properties satisfied.

There are cases where a checklist will simplify choice and other cases where it will not.

¹⁸Following Example 3, lexicographic preferences on \mathbb{R}_+^2 can be represented by a two-valued checklist such that $I = \omega 2$ (the ordinal that orders 2 copies of ω by ranking each number in copy 1 below each number in copy 2).

If you have to choose from {cherries, peaches, apricots, figs, dates, oranges, apples, pears}, it may be natural to consider the properties ‘summer fruits’ or ‘winter fruits’. But it is not so easy to find properties that will make a choice from sets of disparate alternatives, for example, a decision about how to spend a lump-sum retirement distribution from the set {special vacation, new car, house remodeling, golf club membership, live-in housekeeper, down payment on kid’s house, increase in pension contributions, donation to charity}. Since a (standard) checklist will always lead to utility maximization, the very categories that an agent uses to describe decision options can determine whether substantively rational behavior materializes.

The fact that checklists can generate preference-maximizing behavior means that the model cannot be used to compare the decision making speed of preference-maximizers with other types of agents. Mandler [19] takes up this subject.

Although we believe that the checklist model is new to economics, we should mention Rubinstein [23], who underlines the potential importance of unary relations (what we call ‘properties’) in decision making. Although distantly related, that work was the initial stimulus for this project.

10 Appendix: Proofs

It is convenient to present the proofs of some results in reverse order.

Proof of Theorem 2: Let the choice function c have the extended checklist P . We identify each $x \in X$ with the vector $p_x \in \{0, 1\}^I$ given by $p_x(i) = 1$ if $x \in P(i)$ and $p_x(i) = 0$ if $x \notin P(i)$ (of course each p_x can be associated with many alternatives). We order $\{0, 1\}^I$ lexicographically: for $p, q \in \{0, 1\}^I$, define \geq_L by $p \geq_L q \Leftrightarrow (q(i) > p(i) \Rightarrow \exists k < i$ with $p(k) > q(k))$. The asymmetric and symmetric parts of \geq_L are labeled $>_L$ and $=_L$ respectively. To conclude that \geq_L is a linear order, we could appeal to the fact that the lexicographic order of any family of linear orders with well-ordered indices must itself be a linear order. But to argue directly, completeness follows from the fact that (1) if $p = q$ then $(q(i) > p(i) \Rightarrow \exists k < i$ with $p(k) > q(k))$ obtains vacuously, while (2) if $p \neq q$ then the well-ordering of I implies that $j = \min\{i : p(i) \neq q(i)\}$ is well-defined and hence $p >_L q$ if

$p(j) > q(j)$ and $q >_L p$ if $q(j) > p(j)$. Case (2) also yields antisymmetry. For transitivity, if $p =_L q =_L r$ then $p = q = r$ and hence $p =_L r$. If on the other hand $p \geq_L q >_L r$ or $p >_L q \geq_L r$ set $j = \min\{i : p(i) \neq q(i) \text{ or } q(i) \neq r(i)\}$. Then $p(j) \geq q(j) \geq r(j)$ with at least one strict inequality. Hence $p(j) > r(j)$ and $p(i) = r(i)$ for $i < j$, i.e., $p >_L r$.

Let \succsim now denote the relation on X given by $x \succsim y \Leftrightarrow p_x \geq_L p_y$: since \geq_L on $\{0, 1\}^I$ is a linear order, \succsim on X is a preference relation. To see that for any $A \in \mathcal{A}$, $c(A) = \{x \in A : x \succsim y \text{ for all } y \in A\}$, suppose first that $x \in c(A)$. If $y \succ x$ for some $y \in A$ and we set $j = \min\{i : p_x(i) \neq p_y(i)\}$ then the fact that $x \in S_i(A)$ for all $i < j$ implies that $y \in S_i(A)$ for all $i < j$. But since $y \in P(j)$ and $x \notin P(j)$, $x \notin S_j(A)$, contradicting $x \in c(A)$. Conversely suppose $x \in A$ and $x \succsim y$ for all $y \in A$. Then, since $c(A)$ is nonempty, $x \succsim z$ for some $z \in c(A)$. Since $z \in S_i(A)$ for all i , $x \succsim z$ implies $\{i : p_x(i) \neq p_z(i)\} = \emptyset$ (otherwise z would be eliminated at $\min\{i : p_x(i) \neq p_z(i)\}$). So $x \in S_i(A)$ for all i , i.e., $x \in c(A)$.

Now suppose that c maximizes some preference relation \succsim . To construct a checklist, let $I = X \cup \{0\}$ and let \leq be a well-ordering of I with $0 < x$ for any $x \in X$. (This is a nonconstructive step: the principle that any set can be well-ordered relies on the axiom of choice.) For each $x \in X$ define $P(x) = \{y \in X : y \succsim x\}$. Fix $A \in \mathcal{A}$ and some $x \in c(A)$. Then, for any $z \in X$ with $x \notin P(z)$, the fact that $x \succsim y$ for $y \in A$ and the transitivity of \succsim imply $y \notin P(z)$ for any $y \in A$. So, for any $z \in X$, if $x \in \bigcap_{w < z} S_w(A)$ then $x \in S_z(A)$. Since $x \in S_0(A)$, transfinite induction implies that $x \in S_z(A)$ for all $z \in X$. Moreover, for all $y \notin c(A)$, $y \notin P(x)$ and so $y \notin S_x(A)$. Finally observe that $S_z(A) = S_x(A)$ for all z such that $x \leq z$, so that the terminal step j in Definition 1 is well defined. ■

Proof of Theorem 1: Let c have a checklist $P : I \rightarrow 2^X$. As in Theorem 2, given P , each $x \in X$ can be associated with a unique $p_x \in \{0, 1\}^I$, where the i th component is defined by $p_x(i) = 1$ if $x \in P(i)$ and $p_x(i) = 0$ if $x \notin P(i)$. Define $u : X \rightarrow \mathbb{R}$ by

$$u(x) = \sum_{i \in I} \frac{p_x(i)}{3^i}.$$

Since $\sum_{j > i} \frac{1}{3^j} < \frac{1}{3^i}$ for any $i \in I$, this u is a utility representation for \succsim , where, as in the proof of Theorem 2, \succsim is the preference relation \succsim on X induced by the lexicographic order \geq_L on $\{0, 1\}^I$. (A utility representation for \succsim is a u such that $x \succsim y \Leftrightarrow u(x) \geq u(y)$.) The proof

of Theorem 2 also shows that $c(A) = \{x \in X : x \succsim y \text{ for all } y \in X\}$ for all $A \in \mathcal{A}$. Hence $c(A) = \{x \in X : u(x) \geq u(y) \text{ for all } y \in X\}$. ■

Proof of Theorem 3: Given a P that makes n discriminations and the choice function \hat{c} for which P is a checklist, we may without loss of generality let $1, \dots, n$ denote the indifference classes of the preference relation \succsim that \hat{c} maximizes and let the linear order over $\{1, \dots, n\}$ that c induces be \geq (the standard order on the integers). That is, $g \geq h$ for $g, h \in \{1, \dots, n\}$ if and only if, for all $x \in g$ and $y \in h$, $x \succsim y$. It is sufficient to consider only choice functions c defined on subsets of $\{1, \dots, n\}$ that always select the \geq -maximal element. Specifically, if \hat{c} is the choice function that maximizes \succsim , then let A be in the domain of c if and only if there is a \hat{A} in the domain of \hat{c} such that $g \in A \Leftrightarrow (\exists x \in \hat{A} \text{ such that } x \in g)$.

Both conclusions of the theorem hold for $n = 1$ since the empty set of properties can then serve as the desired checklist. So assume henceforth that $n > 1$.

For the second half of the theorem, suppose c has a checklist P with s properties. As in the proof of Theorem 2, identify each $x \in \{1, \dots, n\}$ with the $p_x \in \{0, 1\}^s$ given by $p_x(i) = 1$ if $x \in P(i)$ and $p_x(i) = 0$ if $x \notin P(i)$, which we may read as an integer's s -digit binary expansion. Since there are 2^s integers with s binary digits and given that $n > 1$, $2^s < n$ would imply that there is a distinct pair $x, y \in \{1, \dots, n\}$ identified with the same integer and so then $p_x = p_y$. Since the domain \mathcal{A} contains the two-element sets we have $c(\{x, y\}) = \{x, y\}$, contradicting the assumption that c maximizes \geq . So for this domain we cannot have $2^s < n$.

Regarding ‘there is a checklist that makes n discriminations with k properties, where k is the smallest integer such that $2^k \geq n$,’ suppose this claim holds for $1, \dots, n - 1$. Partition $\{1, \dots, n\}$ into $Z_l = \{1, \dots, m\}$ and $Z_u = \{m + 1, \dots, n\}$, where $m = n/2$ if n is even and $m = (n + 1)/2$ if n is odd. Then, since $n > 1$, we have $2^{k-1} \geq |Z_r|$ for both $r = l$ and $r = u$. The induction hypothesis implies that $c|_{Z_u}$ (the choice function defined by restricting c to subsets of Z_u) has a checklist $P = (P(1), \dots, P(k - 1))$ and that $c|_{Z_l}$ has a checklist $P' = (P'(1), \dots, P'(k - 1))$. Define the checklist Q by $Q(1) = Z_u$ and $Q(i + 1) = P(i) \cup P'(i)$ for $i = 1, \dots, k - 1$.

For any checklist R , let $S_i^R(A)$ denote the i th set of survivors when R is applied to the choice set A .

To see that Q is a checklist for c , notice first that if $A \in Z_u$ then $S_k^Q(A) = S_{k-1}^Q(A) = c|_{Z_u}(A \cap Z_u) = c(A)$, and similarly if $A \in Z_l$ then $S_k^Q(A) = c(A)$. For all A that contain both elements of Z_l and elements of Z_u , application of $Q(1)$ yields $S_1^Q(A) = A \cap Q(1) = A \cap Z_u$. Since $Q(i+1) \cap Z_u = P(i)$, for $i = 1, \dots, k-1$, application of properties $Q(2)$ through $Q(k)$ yields $S_k^Q(A) = S_{k-1}^P(A \cap Z_u) = c|_{Z_u}(A \cap Z_u) = c(A)$. ■

Proof of Proposition 1: Call $U \subset X$ an *upper set of a preference relation* \succsim on X if $(x \in U \text{ and } y \succsim x) \Rightarrow y \in U$.

Let c with domain \mathcal{A} have the (standard) checklist P and let I be the indices of P . We know from the proof of Theorem 2 that c maximizes the preference relation \succsim on X defined by $w \succsim z \Leftrightarrow p_w \geq_L p_z$, where $p_w \in \{0, 1\}^I$ is given by $p_w(k) = 1 \Leftrightarrow w \in P(k)$ and \geq_L is the lexicographic order on $\{0, 1\}^I$. Define the countable family \mathcal{P} of upper sets of \succsim by $P_q = \{w \in X : p_w \geq_L q\} \in \mathcal{P}$ if and only if $q \in \{0, 1\}^I$ has finitely many coordinates k such that $q(k) = 1$. Enumerate \mathcal{P} by a bijection $\kappa : \mathcal{P} \rightarrow \widehat{I}$, where \widehat{I} is \mathbb{N} or $\{1, \dots, n\}$, which defines a checklist \widehat{P} and thus, for any $A \in \mathcal{A}$, a sequence of survivor sets $\widehat{S}_i(A)$. Since P is a checklist for c , for any $A \in \mathcal{A}$ and any $y \in A \setminus c(A)$ there is a $i \in I$ with $y \notin S_i(A)$; let $i(y)$ denote the smallest such i . In addition, since for any given A there are only finitely many properties in P that eliminate some $y \in A \setminus c(A)$, the index $j(A) \equiv \max\{i(y) : y \in A \setminus c(A)\}$ is well-defined. Fix some A and $x \in c(A)$. Since each $\widehat{P}(k)$ is an upper set of \succsim and c maximizes \succsim , $x \in \widehat{S}_k(A)$ for all $k \in \widehat{I}$ (see the proof of Theorem 2). For any $y \in A \setminus c(A)$, we have $x \in P(i(y))$ and $y \notin P(i(y))$ while $x \in P(k) \Leftrightarrow y \in P(k)$ for $k < i(y)$. Thus

$$p_x \geq_L q^{i(y)} >_L p_y$$

where, for any index i , $q^i \in \{0, 1\}^I$ is defined by $q^i(k) = p_x(k)$ for $k \leq i$ and $q^i(k) = 0$ for $k > i$. Thus, for the index $l = \kappa(P_{q^{i(y)}})$, $x \in \widehat{P}(l)$ and $y \notin \widehat{P}(l)$ and so $y \notin \widehat{S}_l(A)$. Since for any $y \in A \setminus c(A)$ the index $\kappa(P_{q^{i(y)}})$ must be drawn from the finite set $J = \{\kappa(P_{q^i}) : i \leq j(A)\}$, $\widehat{S}_{\max J}(A) = c(A)$. Thus c has a checklist that consists of upper sets of \succsim .

To replace upper sets with upper contours, note by the proof of Theorem 1 that \succsim has a utility representation and hence there is a countable $D \subset X$ that is dense in \succsim , i.e., if $x \succ y$ then there is a $d \in D$ such that $x \succ d \succ y$. Define a checklist whose properties consist of

one upper contour $U(d) = \{x \in X : x \succsim d\}$ for each $d \in D$ and any $P_q \in \mathcal{P}$ such that P_q contains a \succsim -minimal element. To see that these properties will form a checklist \tilde{P} for c , we need observe only that if $x \in c(A)$ and $y \in A \setminus c(A)$ and the upper set in the previous paragraph $\hat{P}(l)$ that eliminated y is not a property in \tilde{P} , then $\hat{P}(l)$ has no \succsim -minimal element and hence, since $x \in \hat{P}(l)$, there will be a $d \in D \cap \hat{P}(l)$ such that $x \succsim d$. Since $x \in U(d)$ and $y \notin U(d)$, $U(d)$ eliminates y . ■

Proof of Theorem 4: Let c be a choice function on a domain \mathcal{A} that maximizes a utility function u that represents \succsim and such that (\mathcal{A}, \succsim) satisfies the domain restriction. Let $\Sigma \subset \mathcal{A}$ be the set of \succsim -top-dense sets in \mathcal{A} , let $\sigma : \Sigma \rightarrow X$ be a function that selects an arbitrary \succsim -maximal element from each set in Σ (σ is well-defined by the definition of a \succsim -top-dense set), and let $M = \{m : \sigma(A) = m \text{ for some } A \in \Sigma\}$. Define a property $P_r = \{x \in X : u(x) \geq r\}$ for each rational number r and a property $P^m = \{x \in X : x \succsim m\} = \{x \in X : u(x) \geq u(m)\}$ for each $m \in M$. Due to the domain restriction, $\mathcal{P} = \{P_r : r \in \mathbb{Q}\} \cup \{P^m : m \in M\}$ is countable. Hence we can build a standard checklist P by defining a bijection f from \mathcal{P} to \mathbb{N} and setting $P(f(P_r)) = P_r$ or $P(f(P^m)) = P^m$ for all $P_r, P^m \in \mathcal{P}$. To confirm that P is a checklist for c , given $A \in \mathcal{A}$ suppose y is u -maximal in A and that $y \in S_{i-1}(A)$. Then for any $z \in S_{i-1}(A) \cap P(i)$ we have $u(y) \geq u(z)$ and so, since $P(i)$ is an upper set, $y \in S_i(A)$. Hence y is never eliminated. On the other hand, suppose $z \in A$ is not u -maximal in A . If A is top-dense then there is a u -maximal $y \in A$ and a $m \in M$ such that $u(z) < u(m) = u(y)$. Then $P(f(P^m))$ satisfies $z \notin P(f(P^m))$ and $y \in P(f(P^m))$, and hence $z \notin S_{f(P^m)}(A)$. Alternatively if A is not top-dense then there is some $r \in \mathbb{Q}$ such that $u(z) < r \leq u(y)$. Then $P(f(P_r))$ satisfies $z \notin P(f(P_r))$ and $y \in P(f(P_r))$, and so $z \notin S_{f(P_r)}(A)$.

For the other direction, suppose that c on a domain \mathcal{A} has a standard checklist. By Proposition 1, there is a preference relation \succsim on X such that c maximizes \succsim and has a standard checklist P consisting of upper contours of \succsim . Now suppose by contradiction that $\{x \in X : x \text{ is a } \succsim\text{-maximal element of some } \succsim\text{-top dense } A \in \mathcal{A}\}$ intersects uncountably many \succsim -indifference classes. Since P , being standard, consists of only a countable set of properties there must then be a \succsim -top dense set $A \in \mathcal{A}$ with a \succsim -maximal element m such that $\{x : x \succsim m\}$ is not a property in P . Then for each $i \in \mathbb{N}$ there exists a $y \in A$ (depending on i) for which $m \succ y$ and $y \in S_i(A)$, i.e., y is not eliminated at stage i . For

suppose j were the least element of \mathbb{N} such that $S_j(A) = c(A)$ (i.e., j is the stage at which the final elimination of elements \succsim -worse than m occurs) and let $z \in S_{j-1}(A) \setminus c(A)$ be one of the alternatives eliminated by $P(j)$. Since $P(j)$ is an upper contour with $m \in P(j)$, and $P(j) \neq \{x : x \succsim m\}$, there exists $w \in P(j)$ with $m \succ w$ and, since $z \notin S_j(A)$, $w \succ z$. Since A is \succsim -top dense there must exist some $y \in A$ such that $m \succ y \succ w$. We must then have $y \in S_{j-1}(A)$ since if there were a $P(k)$ with $k < j$ that eliminated y then $P(k)$ would also have eliminated z (since $y \succ w \succ z$ and $P(k)$ is an upper contour). Since $P(j)$ is an upper contour and $y \succ w \in P(j)$, we have $y \in S_j(A)$, a contradiction. ■

Proof of Theorem 5: The part of the proof of Theorem 2 that shows that a c with a checklist $P : I \rightarrow 2^X$ maximizes the \succsim induced by the lexicographic order on $\{0, 1\}^I$ never uses the fact that P finitely terminates. The proof of Theorem 1 therefore also does not use finite termination, and so that proof establishes the ‘if’ part of the present Theorem. For the ‘only if’ part, where we are given a utility u that represents some \succsim and a c that maximizes u , we use the same checklist constructed in the proof of Theorem 4. Once again for any $y \in c(A)$ and $i \in I$, we have $y \in S_i(A) \Rightarrow y \in S_{i+1}(A)$ and therefore $y \in S_i(A)$ for all $i \in I$. And for all $z \in A \setminus \{c(A)\}$, where therefore $y \succ z$ for any $y \in c(A)$, there must exist $P(i) = P_r$ such that $u(z) < r < u(y)$. So it must be that $z \notin \bigcap_{i \in I} S_i(A)$, and thus $c(A) = \bigcap_{i \in I} S_i(A)$. ■

References

- [1] Apestequia, J. and M. A. Ballester (2009) “Rationalizability of Choice by Sequential Procedures”, mimeo, Universitat Autònoma de Barcelona and Universitat Pompeu Fabra.
- [2] Bereby-Meyer, Y., A. Assor and I. Katz (2004) “Children’s choice strategies: the effects of age and task demands,” *Cognitive Development* 19: 127-146.
- [3] Brandstätter, B., G. Gigerenzer, and R. Hertwig (2006) “The priority heuristic: making choices without trade-offs,” *Psychological Review* 413: 409-432.

- [4] Chipman, J. (1960) “The foundations of utility,” *Econometrica* 28:193-224.
- [5] Chipman, J. (1971) “On the lexicographic representation of preference orderings,” in J. Chipman, L. Hurwicz, M. Richter, and H. Sonnenschein, eds., *Preferences, Utility, and Demand*. Harcourt: New York.
- [6] Cuesta Dutari, N. (1943) “Teoria decimal de los tipos de orden,” *Revista Matematica Hispano-Americana* 3: 186-205, 242-268.
- [7] Cuesta Dutari, N. (1947) “Notas sobre unos trabajos de Sierpinski,” *Revista Matematica Hispano-Americana* 7: 128-131.
- [8] Fischer, J. E., F. Steiner, F. Zucol, C. Berger, L. Martignon, W. Bossart, M. Altwegg, and D. Nadal (2002) “Using simple heuristics to target macrolide prescription in children with community-acquired pneumonia,” *Archives of Pediatrics* 156, 1005–1008.
- [9] Fishburn, P. (1972) *Mathematics of Decision Theory*. Mouton: The Hague, Paris.
- [10] Fishburn, P. (1974) “Lexicographic orders, utilities and decision rules: a survey,” *Management Science* 20: 1442-1471.
- [11] Gilbert, E. (1962) “Games of identification or convergence,” *SIAM Review* 4: 16-24.
- [12] Gigerenzer, G., P. M. Todd and the ABC Research Group (1999) *Simple Heuristics That Make Us Smart*. Oxford University Press, New York.
- [13] Halmos, P.R. (1970) *Naive Set Theory*. Springer Verlag, New York.
- [14] Houy, N. and K. Tadenuma (2009) “Lexicographic compositions of multiple criteria for decision making”, *Journal of Economic Theory* 144: 1770-1782
- [15] Huffman, D. (1952) “A method for the construction of minimum-redundancy codes,” *Proceedings of the Institute of Radio Engineers* 40: 1098-1101.
- [16] Katsikopoulos, K. V. and L. Martignon (2006) “Naive heuristics for paired comparisons: some results on their relative accuracy,” *Journal of Mathematical Psychology* 50: 488-494.

- [17] Knuth, D. (1973) *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley: Reading MA.
- [18] Lancaster, K.L. (1966) “A new approach to consumer theory”, *The Journal of Political Economy*, 74: 132-157.
- [19] Mandler, M. (2010), “Rational agents are the quickest,” mimeo, Royal Holloway College, University of London.
- [20] Manzini, P. and M. Mariotti (2007) “Sequentially rationalizable choice,” *American Economic Review* 97: 1824-1839.
- [21] Manzini, P. and M. Mariotti (2009) “Choice by Lexicographic Semiorders”, *Theoretical Economics*, forthcoming.
- [22] Richter, M. (1966) “Revealed preference theory,” *Econometrica* 34: 635-645.
- [23] Rubinstein, A. (1996) “Why are certain properties of binary relations relatively more common in natural language?” *Econometrica* 64: 343-355.
- [24] Sierpinski, W. (1949) “Sur une propriete des ensembles ordonnes,” *Fundamenta Mathematicae* 36: 56-67.
- [25] Simon, H.A. (1976) “From Substantive to Procedural Rationality,” in S.J. Latsis, ed., *Methods and Appraisal in Economics*. Cambridge University Press, Cambridge.
- [26] Simon, H.A. (1990) “Bounded Rationality,” in J. Eatwell, M. Milgate and P. Newman, eds., *The New Palgrave: Utility and Probability*. Macmillan, London.
- [27] Tversky, A. (1972) “Elimination by Aspects: A Theory of Choice” *Psychological Review* 79: 281–299.
- [28] Tversky, A. and E. Shafir (2004) *Preference, belief, and similarity: selected writings*, MIT Press.
- [29] Yee, M., E. Dahan, J. R. Hauser and J. B. Orlin (2007) “Greedoid-based non-compensatory inference,” *Marketing Science*, 26: 532-549.

- [30] Zimmerman, S. (1959) “An optimal search procedure,” *American Mathematical Monthly* 66: 690-693.