

A Sequent Calculus for Urn Logic

Rohan French¹

Published online: 9 April 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract Approximately speaking, an urn model for first-order logic is a model where the domain of quantification changes depending on the values of variables which have been bound by quantifiers previously. In this paper we introduce a model-changing semantics for urn-models, and then give a sequent calculus for urn logic by introducing formulas which can be read as saying that “after the individuals a_1, \dots, a_n have been drawn, A is the case”.

Keywords Urn logic · Sequent calculus · Non-standard quantification · Proof theory

1 Introduction

Urn logic is a generalisation of first-order logic introduced in [Rantala \(1975\)](#). The idea behind urn logic is to think of the domain of quantification in a way reminiscent of the eponymous urns of balls used in probability theory. The metaphor here is to think of quantifying over objects in a domain as being analogous to drawing balls from an urn. Typically in first-order logic what we are doing is always quantifying over the same objects—our domain stays fixed every time we ‘draw’ from the domain. But, just as we can alter the contents of the urn based upon what we’ve drawn from it (for example, by simply not replacing it, or by putting more objects back into the urn of a kind based on what we’ve drawn out), we can also think about models for the quantifiers where the domain of quantification changes in the course of evaluating a formula.

✉ Rohan French
rohan.french@gmail.com

¹ Faculty of Philosophy, University of Groningen, Oude Boteringestraat 52, 9712 GL Groningen, The Netherlands

[Hintikka \(1975\)](#) argues that urn models can be used to give a satisfactory model theoretic account of impossible worlds (and thus deal with issues of logical omniscience). Hintikka's idea is, when giving our first-order epistemic logic, to have two kinds of worlds—invariant, and almost-invariant worlds—the invariant ones being those which are used for determining logical truths, and all worlds (invariant and almost-invariant alike) being used in the interpretation of the epistemic operators. These almost-invariant worlds are envisaged as being urn models where the domain from which we can 'draw' is constant up until we have made n draws, the n being determined by the agent's cognitive capacities. We will dwell no further on this potential application, outlined by Hintikka, of the notion of an urn model. Instead our focus will be largely technical (although we will mention another potential application at the end of the next section).

Relatively little work has been done on urn logic—notable exceptions being [Cresswell \(1982\)](#) and [Olin \(1978\)](#). In [Cresswell \(1982\)](#) a 'classical analogue' of Rantala's urn logic is investigated, using a modification of the standard Tarskian semantics for first-order logic as opposed to the game-theoretic semantics used by Rantala. Using these semantics Cresswell gives a Henkin-style completeness proof for a range of urn logics with various constraints placed upon how we can draw individuals from the domain. [Olin \(1978\)](#), also using a classical version of urn Logic, has investigated the model-theory of urn models, with particular attention to issues concerning satisfiability and categoricity.

Our plan here is as follows. In Sect. 2 we will describe the (standard/Tarskian) semantics for urn logic which we will be using, and comparing it to the semantic treatments given by Rantala, Cresswell and Olin. We will then proceed in Sect. 3 to give a sequent calculus for urn logic obtained by 'decorating' formulas so as to give us a way to keep track of changes in the domain. We then prove this sequent calculus to be sound and complete w.r.t. the semantics described in Sect. 2 before going on to show how to extend our sequent calculus in Sect. 5 so that it is sound and complete w.r.t. the class of urn models which Rantala is concerned with.

2 Semantics for Urn Logic

The language of urn logic is just the standard language of first-order logic without identity or individual constants. The restriction to first-order languages without individual constants is (aside from some minor complications discussed in Sect. 5.2) entirely for expository ease—unless noted all results here will hold for languages which contain individual constants. We will think of our models for urn logic in a similar vein to Cresswell, taking our models to be triples $\langle U, D, V \rangle$ where U is a non-empty set (the initial domain), V is a standard first-order interpretation, and D is a function from sequences of objects from U to subsets of U . We do this mostly so that we can talk about domains in the standard way: the value of D on a sequence $\langle a_1, \dots, a_n \rangle$ being the effective domain for the next quantification after $\langle a_1, \dots, a_n \rangle$ have been 'drawn'. In order for a structure $\langle U, D, V \rangle$ to be an urn model we will also require that it satisfies the following condition.

$$(Draw): D(\langle a_1, \dots, a_n \rangle) \neq \emptyset \Rightarrow a_n \in D(\langle a_1, \dots, a_{n-1} \rangle).$$

So our official definition of an urn model here will be that a structure $\langle U, D, V \rangle$ is an urn model iff U is a non-empty set, V is a function which assigns subsets of U^n to n -place primitive predicates, and D is a function from sequences of objects for U to subset of U which satisfies the condition *(Draw)* above. In his discussion, Rantala places two further constraints upon the class of models $\langle U, D, V \rangle$ he is concerned with, which we will discuss further in Sect. 5.

The semantic discussion of urn logic in Rantala (1975) is a modification of the game-theoretic semantics for first-order logic from Hintikka (1979). Rather than working within the framework of game-theoretic semantics, though, we will instead work within a more familiar looking semantic framework, somewhat reminiscent of that used in Cresswell (1982) and Olin (1978). Rantala’s reasons (Rantala 1975, p. 351) for using the game-theoretic semantics is so that we can have a ‘model-changing’ semantics in the sense discussed in Sect. 4 of Humberstone (2008), the semantics for urn logic not requiring any of the independent-choice style phenomena which game-theoretic semantics is particularly amenable to.

In the semantics for urn logic which we will be using, we will keep our model fixed, but evaluate formulas for truth relative, not just to a variable assignment as with standard Tarskian treatments of first-order logic, but also relative to a *draw*.

Definition 1 Given an urn model $\mathcal{M} = \langle U, D, V \rangle$ a sequence $\langle s_1, \dots, s_n \rangle$ of objects from U is a *draw from \mathcal{M}* iff (1) $s_1 \in D(\emptyset)$ and (2) $s_{i+1} \in D(\langle s_1, \dots, s_i \rangle)$ for all $1 \leq i \leq n - 1$.

One of the effects of the condition *(Draw)* is that whenever the domain determined by a sequence \underline{s} is non-empty, then \underline{s} is automatically a draw.

Definition 2 An assignment ν is an *x-variant of μ* iff $\nu(y) = \mu(y)$ for all variables y except (possibly) x .

Our fundamental notion, then, will be truth of a formula φ in an urn model $\mathcal{M} = \langle U, D, V \rangle$ relative to an assignment μ and a draw \underline{s} , which we will write as $\mathcal{M}, \underline{s} \models \varphi[\mu]$. Throughout we will use $\underline{s} \cdot t$ to denote the concatenation of \underline{s} with t .

$$\begin{aligned} \mathcal{M}, \underline{s} \models Fx_1 \dots x_n [\mu] &\iff \langle \mu(x_1), \dots, \mu(x_n) \rangle \in V(F). \\ \mathcal{M}, \underline{s} \models \neg\varphi [\mu] &\iff \mathcal{M}, \underline{s} \not\models \varphi [\mu]. \\ \mathcal{M}, \underline{s} \models \varphi \wedge \psi [\mu] &\iff \mathcal{M}, \underline{s} \models \varphi [\mu] \text{ and } \mathcal{M}, \underline{s} \models \psi [\mu]. \\ \mathcal{M}, \underline{s} \models \forall x\varphi [\mu] &\iff \mathcal{M}, \underline{s} \cdot \nu(x) \models \varphi[\nu] \\ &\text{for all } x\text{-variants } \nu \text{ of } \mu \text{ s.t.} \\ &\nu(x) \in D(\underline{s}). \end{aligned}$$

To give the reader an example of the way in which the urn semantics for first-order logic differs from the standard semantics for the quantifiers, consider the following classically valid formula:

$$(\forall y)((\forall x)Fx \rightarrow Fy).$$

This formula is not valid in the semantics outlined above—where we take a sentence to be valid whenever it is true in a model w.r.t. the empty sequence of draws. To see this consider the following simple urn model $\langle U, D, V \rangle$.

- $U = \{a, b\}$.
- $D(\emptyset) = \{a, b\}$, $D(a) = \{b\}$, $D(b) = \{a\}$, and $D(s) = U$ for all sequences of length 2 or more.
- $V(F) = \{a\}$.

Now to see that the above formula is false in this urn model note that we can assign b to y , forcing us to consider whether $\mathcal{M}, \langle b \rangle \models \forall x Fx \rightarrow Fy[\mu]$, but we know that $\mathcal{M}, \langle b \rangle \not\models Fy[\mu]$, as $\mu(y) = b$, so in order for the above formula to be true we must have $\mathcal{M}, \langle b \rangle \not\models \forall x Fx[\mu]$. But as $D(b) = \{a\}$, and $a \in V(F)$ it follows that $\mathcal{M}, \langle b \rangle \models \forall x Fx[\mu]$, and thus the above formula is invalid.

The above example also shows how our semantics differs from that of Cresswell and Olin. According to our semantics the truth of a quantified formula like $\exists x\varphi$ in a model is evaluated relative to a sequence of ‘draws’ from the domain which, when we are considering sentences, will be determined by the values which have been assigned to variables which do not occur in the formula under consideration. By contrast, (again considering sentences) Cresswell and Olin evaluate formulas relative to the effect which such assigning has had upon the formula being evaluated, having the relevant domain of quantification be determined by the values assigned to the free variables in the formula in question. In the above case, for example, Cresswell and Olin would evaluate $\forall x Fx$ relative to $D(\emptyset)$ (and consequently judge the above formula as valid).

2.1 Globally and Locally Determined Domains

This difference between the approach we are taking here, and that taken by Cresswell and Olin corresponds quite neatly to the distinction between the strongly and weakly exclusive interpretation of the quantifiers discussed in (Hintikka 1956, p. 230) and given a proof-theoretic treatment in Wehmeier (2004, 2009). The following terminology is more suggestive, though: call any semantics for the quantifiers where the domain quantified over is a non-constant function of the formula under investigation one with *locally determined domains*. Similarly, we will call any semantics for the quantifiers in which the domain quantified over is a non-constant function of the broader sentential-context in which the quantified expression occurs one with *globally determined domains*.¹ So the strongly-exclusive interpretation of the quantifiers (discussed in Wehmeier (2009)) is one where we have globally determined domains,

¹ Rather than limiting ourselves to sentential-context determining the domain (so that the broader formula in which a quantified expression is embedded determines the relevant domain) we could instead let the domain simply be determined by some context of assessment c , which is only in part determined by the sentential-context. On this approach we can give a pleasant account of (semantically) restricted quantification. For example suppose c was a context in which the speaker has just opened the fridge—a context which makes the fridge’s contents especially salient—and consider the utterance of a sentence such as “There’s no beer” relative to this context. Then we could think of this as being of the form $\neg\exists x\text{Beer}(x)$ with the quantifier ranging over the domain determined by c —namely the contents of the fridge. We will consider another similar example below.

and the weakly-exclusive interpretation (discussed in [Wehmeier \(2004\)](#)) one where we have locally determined domains. In the case of the semantics for urn logic discussed above, the semantics used by Cresswell and Olin is one with local domain determination, and that used by Rantala and myself one with global domain determination.

One issue which globally determined domains allows us to get clearer on is the status of what is called the ‘Binding Assumption’ in [Stanley \(2000\)](#). Approximately speaking, this is the assumption that if a part α of a clause semantically binds another part β of that clause, then this binding must show up in the logical form of the two clauses with a variable being bound in α which is a constituent of β . For example, consider the following sentences (both of which are variants of sentences from [Stanley \(2000\)](#)).

- (★) In all of John’s classes he fails three students.
- (★′) Whatever office you go to, the supervisor is always unavailable.

The most straightforward reading of (★) is one where in each of John’s classes x he fails three students in x , but the logical form which most closely mirrors its surface form is something like the following (where ‘ \exists^{\neq} ’ is an abbreviation for ‘there exist distinct...’).

$$\forall x(\text{John-Class}(x) \rightarrow \exists^{\neq} tuv(\text{Student}(t) \wedge \text{Student}(u) \wedge \text{Student}(v) \\ \wedge \text{Fail}(t) \wedge \text{Fail}(u) \wedge \text{Fail}(v))).$$

Similarly, the most forward reading of (★′) is that where for every office x which you go to, the supervisor of x is always unavailable, but the straightforward logical form of this sentence similarly does not exhibit the required kind of syntactic binding. Concerning cases like these Stanley says the following.

In each of the cases we have discussed, the domain of the second quantified expression varies with the values introduced by the initial quantifier expression. Therefore, given what we have been assuming about the relation between semantic binding and syntactic binding outlined above [i.e. the Binding Assumption], it follows that there are bindable variables in the logical form of sentences containing quantifier expressions whose values are quantifier domains. ([Stanley 2000](#), p. 421)

What this argument tells us, then, is that due to the fact that in order for sentences like (★) and (★′) to be given the straightforward truth conditions mentioned above the outer quantified expression must semantically bind the inner quantified expression. According to Stanley’s Binding Assumption this can only be so if the outer quantified expression binds variables which are otherwise free in the inner quantified expression, and thus the true logical form of sentences like (★) and (★′) must contain such bound variables, unlike the surface form given above.

Quantifiers with globally determined domains, like the urn quantifiers we will be investigating here, give us the technical machinery to undermine part of the plausibility of the Binding Assumption. If our quantifiers are to work just like standard Tarskian quantifiers then the only way for there to be semantic binding of quantifier expressions is, as the Binding Assumption tells us, by having free-variables in the semantically

bound expression being syntactically bound by the expression which is doing the binding. If the only treatment of quantifiers we are countenancing is the Tarskian one then the Binding Assumption seems warranted, but in the presence of alternative ways of registering semantic binding without having syntactic binding this warrant is at least somewhat weakened.

To see that we can do without the Binding Assumption, note that if we were to use quantifiers with globally determined domains then we could get the correct interpretation for (\star) using its surface logical form. For example, we will end up with the semantically bound reading of (\star) whenever, for all $a \in D(\emptyset) \cap \|\text{John-Class}\|$ we have $D(a) \cap \|\text{Student}\| = \|\text{InClass}(a)\|$, where $\|F\|$ is the extension of the predicate F . Here the predicate InClass , which is implicit in the standard interpretation of (\star) , is used implicitly to semantically restrict our domain of quantification. We might think of the process of domain determination on this picture as being a top-down pragmatic process, based on an assessment of speaker intent and features of the conversational context.

This kind of semantic (as opposed to syntactic) binding also has the additional advantage of allowing us to accommodate deviant interpretations of (\star) in a relatively straightforward manner. For example, suppose John finds teaching so stressful that for each class he teaches he finds that the only way he can unwind is by failing three students, and that he is thoroughly unconcerned which classes the students are in. In such a situation we can imagine an (understandably) confused interlocutor engaging with someone who has just uttered (\star) querying whether John fails three students from each class he teaches and being told “No, John just gets very stressed and can only get relief by failing students. He’s not overly concerned which class they’re in!”. Inappropriate behaviour for a teacher, perhaps, but a perfectly legitimate reading of (\star) all the same.

Our aim is not to necessarily endorse such a view here, but mostly to illustrate how being clearer about the various options we have for interpreting the quantifiers provides us with greater flexibility in giving the semantics of natural language without having to postulate additional bindable variables or hidden constituents.

3 A Sequent Calculus for Urn Logic

In this section we will provide a sequent calculus which is sound and complete w.r.t. the class of all urn models. To do this we will find it useful to modify the standard language of first-order logic as follows. Suppose that L is the set of wffs for a first-order language without identity. Then the wffs of L^U are all the formulas of L , as well as any formula of the form

$$\langle t_1, \dots, t_n \rangle : A$$

where $\langle t_1, \dots, t_n \rangle$ is a sequence of terms from L , and A is a wff from L . The idea behind this change to our object language is to read formulas like that inset above as saying that “after the objects denoted by t_1, \dots, t_n are drawn, A ”, to modify Rantala’s turn of phrase.

3.1 The System UC

Let us define the following sequent calculus, which we will call **UC** (mnemonic for ‘Urn Calculus’). Formally speaking our sequents are pairs of multisets of L^U -formulas $\langle \Gamma, \Delta \rangle$, which we will write as $\Gamma \succ \Delta$. Throughout we will use \underline{s} and \underline{t} as variables over sequences of terms, A and B as L -formulas. We will also occasionally find it useful to use α and β as schematic letters for L^U -formulas. Throughout t and u be schematic for terms, and a be schematic for free-variables (variously, in both cases, with and without subscripts). Where \underline{s} is the sequence $\langle s_1, \dots, s_n \rangle$ we will write $\underline{s} \cdot t$ to denote the sequence $\langle s_1, \dots, s_n, t \rangle$.

AXIOMS: All of the axiomatic sequents for **UC** are of the following form:

$$\Gamma, \underline{s} : Ft_1 \dots t_n \succ \underline{s}' : Ft_1 \dots t_n, \Delta$$

where \underline{s} and \underline{s}' are sequences of terms from L , and t_1, \dots, t_n are terms, and F is an n -ary atomic predicate of L .

OPERATIONAL RULES: **UC** has the following operational rules for the connectives \neg , \wedge and \forall .

$$\begin{array}{c} \frac{\Gamma \succ \underline{s} : A, \Delta}{\Gamma, \underline{s} : \neg A \succ \Delta} \quad (\neg L) \qquad \frac{\Gamma, \underline{s} : A \succ \Delta}{\Gamma \succ \underline{s} : \neg A, \Delta} \quad (\neg R) \\ \frac{\Gamma, \underline{s} : A, \underline{s} : B \succ \Delta}{\Gamma, \underline{s} : A \wedge B \succ \Delta} \quad (\wedge L) \qquad \frac{\Gamma \succ \underline{s} : A, \Delta \quad \Gamma \succ \underline{s} : B, \Delta}{\Gamma \succ \underline{s} : A \wedge B, \Delta} \quad (\wedge R) \\ \frac{\Gamma, \underline{s} \cdot t : A(t) \succ \Delta}{\Gamma, \underline{s} : \forall x A(x) \succ \Delta} \quad (\forall L) \qquad \frac{\Gamma \succ \Delta, \underline{s} \cdot a : A(a)}{\Gamma \succ \Delta, \underline{s} : \forall x A(x)} \quad (\forall R) \end{array}$$

where for some B and some (possibly empty) term sequence \underline{s}' we have $\underline{s} \cdot t \cdot \underline{s}' : B \in \Gamma \cup \Delta$ in $(\forall L)$, and a does not appear in $\Gamma \cup \Delta \cup \{\underline{s} : \forall x A(x)\}$ in $(\forall R)$.

STRUCTURAL RULES: The system **UC** has the following standard structural rules.

$$\begin{array}{c} \frac{\Gamma \succ \Delta}{\Gamma \succ \underline{s} : A, \Delta} \quad (\text{Weakening-R}) \qquad \frac{\Gamma \succ \Delta}{\Gamma, \underline{s} : A \succ \Delta} \quad (\text{Weakening-L}) \\ \frac{\Gamma \succ \underline{s} : A, \underline{s} : A, \Delta}{\Gamma \succ \underline{s} : A, \Delta} \quad (\text{Contraction-R}) \qquad \frac{\Gamma, \underline{s} : A, \underline{s} : A \succ \Delta}{\Gamma, \underline{s} : A \succ \Delta} \quad (\text{Contraction-L}) \end{array}$$

It also have the following variation on the standard (*Cut*) rule:

$$\frac{\Gamma \succ \underline{s} : A, \Delta \quad \Gamma, \underline{s} : A \succ \Delta}{\Gamma \succ \Delta} \quad (dCut)$$

provided that for some formula $B \in L$ and some term sequence \underline{s}' we have $\underline{s} \cdot \underline{s}' : B \in \Gamma \cup \Delta$.

URN- RULES: Let φ be a quantifier-free formula.

$$\frac{\Gamma, \underline{s} : \varphi \succ \Delta}{\Gamma, \underline{s} \cdot t : \varphi \succ \Delta} \quad (Draw-L) \qquad \frac{\Gamma \succ \underline{s} : \varphi, \Delta}{\Gamma \succ \underline{s} \cdot t : \varphi, \Delta} \quad (Draw-R)$$

Consider the following example of a sequent derivation in this system.

$$\begin{array}{c}
 a : Fa, \langle a, b \rangle : Fb \succ \langle a, b \rangle : Fa \quad a : Fa, \langle a, b \rangle : Fb \succ \langle a, b \rangle : Fb \\
 \hline
 \frac{}{a : Fa, \langle a, b \rangle : Fb \succ \langle a, b \rangle : Fa \wedge Fb} \text{(\wedge L)} \\
 \frac{}{a : Fa, a : \forall y(Fy) \succ \langle a, b \rangle : Fa \wedge Fb} \text{(\wedge L)} \\
 \frac{}{a : Fa \wedge \forall y(Fy) \succ \langle a, b \rangle : Fa \wedge Fb} \text{(\forall R)} \\
 \frac{}{a : Fa \wedge \forall y(Fx) \succ a : \forall y(Fa \wedge Fy)} \text{(\forall L)} \\
 \frac{}{\forall x(Fx \wedge \forall y(Fx)) \succ a : \forall y(Fa \wedge Fy)} \text{(\forall R)} \\
 \frac{}{\forall x(Fx \wedge \forall y(Fx)) \succ \forall x \forall y(Fx \wedge Fy)} \text{(\forall R)}
 \end{array}$$

3.2 Soundness

Let us abbreviate the claim that \underline{s} is a draw from \mathcal{M} by writing $draw^{\mathcal{M}}(\underline{s})$. Furthermore we will occasionally write $\mu(\underline{s})$ where μ is a variable assignment and \underline{s} is a sequence of terms from L , understanding that as being the result of applying the variable assignment to each member of the sequence in turn (e.g. $\mu(\langle a, b \rangle) = \langle \mu(a), \mu(b) \rangle$).

Definition 3 Let $\langle s_1, \dots, s_n \rangle : A$ be a formula from L^U . Then we will say that $\mathcal{M} \Vdash^+ \langle s_1, \dots, s_n \rangle : A[\mu]$ whenever:

$$draw^{\mathcal{M}}(\langle \mu(s_1), \dots, \mu(s_n) \rangle) \ \& \ \mathcal{M}, \langle \mu(s_1), \dots, \mu(s_n) \rangle \models A[\mu].$$

Let us say that $\mathcal{M} \Vdash^- \langle s_1, \dots, s_n \rangle : A[\mu]$ whenever:

$$draw^{\mathcal{M}}(\langle \mu(s_1), \dots, \mu(s_n) \rangle) \ \& \ \mathcal{M}, \langle \mu(s_1), \dots, \mu(s_n) \rangle \not\models A[\mu].$$

Definition 4 Say that a sequent $\Gamma \succ \Delta$ fails in an Urn-Model \mathcal{M} on an assignment μ whenever for all $\alpha \in \Gamma$ we have $\mathcal{M} \Vdash^+ \alpha[\mu]$ and for all $\beta \in \Delta$ we have $\mathcal{M} \Vdash^- \beta[\mu]$. Otherwise we will say that it holds in \mathcal{M} on μ and write $\mathcal{M} \models \Gamma \succ \Delta[\mu]$.

We will often find it convenient to write $\mathcal{M} \Vdash^+ \Gamma[\mu]$ (or $\mathcal{M} \Vdash^- \Gamma[\mu]$) for some set of formulas Γ , which should be understood as meaning that $\mathcal{M} \Vdash^+ \alpha[\mu]$ (resp. $\mathcal{M} \Vdash^- \alpha[\mu]$) for all formulas $\alpha \in \Gamma$, in which case the above definition tells us that a sequent fails in a model \mathcal{M} on an assignment μ whenever we have $\mathcal{M} \Vdash^+ \Gamma[\mu]$ and $\mathcal{M} \Vdash^- \Delta[\mu]$.

We will occasionally find the following lemma useful to appeal to in what follows.

Lemma 1 For all quantifier-free formulas φ , all models $\mathcal{M} = \langle U, D, V \rangle$, variable assignments μ , and all draws \underline{s} , and \underline{u} from \mathcal{M} we have the following:

$$\mathcal{M}, \underline{s} \models \varphi[\mu] \quad \text{if and only if} \quad \mathcal{M}, \underline{u} \models \varphi[\mu]$$

Proposition 1 (Soundness) If $\Gamma \succ \Delta$ is derivable in UC then, for all models \mathcal{M} and assignments μ , we have $\mathcal{M} \models \Gamma \succ \Delta[\mu]$.

Proof What we want to show is that if a sequent $\Gamma \succ \Delta$ is derivable in **UC** then, for all models \mathcal{M} and all assignments mu we have either it is not the case that $\mathcal{M} \Vdash^+ \Gamma[\mu]$ or it is not the case that $\mathcal{M} \Vdash^- \Delta[\mu]$ —i.e. that a sequent is derivable if it holds in all models. Put more explicitly, the consequent of this conditional means that, for all models \mathcal{M} and all assignments μ that either we have $\mathcal{M} \Vdash^+ \alpha$ for some $\alpha \in \Gamma$ or $\mathcal{M} \Vdash^- \beta$ for some $\beta \in \Delta$.

To prove this we will show that the axioms hold in all models, and show that each of the rules preserve the property of holding in a model (in particular by showing that if the conclusion of a rule *fails* in a model on an assignment, then (one of its) premise sequents also *fails* in a model on an assignment. Thus, given that the axioms hold in all models, and the rules preserve the property of holding in a model it follows by a simple induction on the length of derivations that if a sequent is derivable then it holds in all models, as desired.

Axiom Case: Suppose that $\Gamma \succ \Delta$ is an axiom. Then we have some formula $F t_1 \dots t_n$ and some term sequences \underline{s} and \underline{s}' s.t. $\underline{s} : F t_1 \dots t_n \in \Gamma$ and $\underline{s}' : F t_1 \dots t_n \in \Delta$. If we have either \underline{s} or \underline{s}' fail to be a draw then we're done, as this will mean that either $\mathcal{M} \Vdash^+ \underline{s} : F t_1 \dots t_n[\mu]$ or $\mathcal{M} \Vdash^- \underline{s}' : F t_1 \dots t_n[\mu]$. So suppose that we have both $draw^{\mathcal{M}}(\underline{s})$ and $draw^{\mathcal{M}}(\underline{s}')$. It follows now that either (1) $\mathcal{M}, \underline{s} \models F t_1 \dots t_n[\mu]$ or (2) $\mathcal{M}, \underline{s} \not\models F t_1 \dots t_n[\mu]$. If (1) then we have $\mathcal{M}, \underline{s}' \models F t_1 \dots t_n[\mu]$ by Lemma 1, and so $\mathcal{M} \Vdash^- \underline{s}' : F t_1 \dots t_n[\mu]$. The case for (2) follows similarly.

We now cover some representative cases of the rules.

\neg -cases: For $(\neg L)$ suppose that we have $\Gamma, \underline{s} : \neg A \succ \Delta$ failing in some model \mathcal{M} on an assignment μ . That is we have $\mathcal{M} \Vdash^+ \Gamma[\mu]$, $\mathcal{M} \Vdash^+ \underline{s} : \neg A[\mu]$ and $\mathcal{M} \Vdash^- \Delta[\mu]$. As we have $\mathcal{M} \Vdash^+ \underline{s} : \neg A[\mu]$ it follows that $draw^{\mathcal{M}}(\mu(\underline{s}))$ and $\mathcal{M}, \mu(\underline{s}) \models \neg A$, and hence that $\mathcal{M}, \mu(\underline{s}) \not\models A$, from which it follows that $\mathcal{M} \Vdash^- \underline{s} : A[\mu]$, resulting in $\Gamma \succ \underline{s} : A, \Delta$ failing in \mathcal{M} on μ as desired. For $(\neg R)$ suppose that we have $\Gamma \succ \Delta, \underline{s} : \neg A$ failing in some model \mathcal{M} on an assignment μ . That is, we have $\mathcal{M} \Vdash^+ \Gamma[\mu]$, $\mathcal{M} \Vdash^- \underline{s} : \neg A[\mu]$ and $\mathcal{M} \Vdash^- \Delta[\mu]$. From $\mathcal{M} \Vdash^- \underline{s} : \neg A[\mu]$ it follows that $draw^{\mathcal{M}}(\mu(\underline{s}))$ and $\mathcal{M}, \mu(\underline{s}) \not\models \neg A[\mu]$, and hence that $\mathcal{M}, \mu(\underline{s}) \models A[\mu]$ from which it follows that $\mathcal{M} \Vdash^+ \underline{s} A[\mu]$, resulting in $\Gamma, \underline{s} : A \succ \Delta$ failing in \mathcal{M} on μ as desired.

\forall -cases: For $(\forall L)$ suppose that we have $\Gamma, \underline{s} : \forall x A(x) \succ \Delta$ failing in a model \mathcal{M} on an assignment μ . That is to say, we have $\mathcal{M} \Vdash^+ \Gamma[\mu]$, $\mathcal{M} \Vdash^+ \underline{s} : \forall x A(x)[\mu]$ and $\mathcal{M} \Vdash^- \Delta[\mu]$. As we have $\mathcal{M} \Vdash^+ \underline{s} : \forall x A(x)[\mu]$ it follows that $draw^{\mathcal{M}}(\mu(\underline{s}))$ and $\mathcal{M}, \mu(\underline{s}) \models \forall x A(x)[\mu]$. So for all $e \in D(\underline{s})$ we have $\mathcal{M}, \mu(\underline{s}) \cdot e \models A(x)[\mu']$ where μ' is an x -variant of μ for which $\mu(x) = e$. Now by the side condition for the rule we know that for some formula B and (possibly empty) term sequence \underline{s}' we have $\underline{s} \cdot t \cdot \underline{s}' : B \in \Gamma \cup \Delta$ from which it follows that $draw^{\mathcal{M}}(\mu(\underline{s} \cdot t \cdot \underline{s}'))$, and thus that $draw^{\mathcal{M}}(\mu(\underline{s} \cdot t))$ by the (Draw) condition. Thus, letting $\mu'(x) = \mu(t)$ it follows that $\mathcal{M}, \mu(\underline{s} \cdot x) \models A(x)[\mu']$. As μ' is an x -variant of μ it follows that $\mathcal{M} \Vdash^+ \Gamma[\mu']$ and $\mathcal{M} \Vdash^- \Delta[\mu']$. Further, as $\mu'(x) = \mu'(t)$ we have $\mathcal{M} \Vdash^+ \underline{s} \cdot t : A(t)[\mu']$, from which we have $\Gamma, \underline{s} \cdot t : A(t) \succ \Delta$ failing in \mathcal{M} on μ' as desired.

For $(\forall R)$ suppose that we have $\Gamma \succ \Delta, \underline{s} : \forall x A(x)$ failing in some model \mathcal{M} on an assignment μ . That is to say, we have $\mathcal{M} \Vdash^+ \Gamma[\mu]$, $\mathcal{M} \Vdash^- \underline{s} : \forall x A(x)[\mu]$ and $\mathcal{M} \Vdash^- \Delta[\mu]$. It follows, then, that $draw^{\mathcal{M}}(\mu(\underline{s}))$ and that there is an object

$e \in D(\mu(\underline{s}))$ s.t. $\mathcal{M}, \mu(\underline{s}) \cdot e \not\models A(x)[\mu']$ where μ' is an x -variant of μ for which $\mu(x) = e$. Given that, by the side condition on $(\forall R)$, the term a does not occur in Γ, Δ or $\underline{s} : \forall x A(x)$ we can simply let our variable assignment assign e to a in which case we have $\mathcal{M}, \mu'(\underline{s}) \cdot a \not\models A(a)[\mu']$ —which given $draw^{\mathcal{M}}(\mu(\underline{s}) \cdot a)$ gives us $\mathcal{M} \Vdash^- \underline{s} \cdot a : A(a)[\mu']$. Further, as μ' is an x -variant of μ it follows that $\mathcal{M} \Vdash^+ \Gamma[\mu']$ and $\mathcal{M} \Vdash^- \Delta[\mu']$ from which it follows that $\Gamma \succ \Delta, \underline{s} \cdot a : A(a)$ fails in \mathcal{M} on μ' as desired.

(Draw)-cases: For the case of *(Draw-L)* suppose that $\Gamma, \underline{s} \cdot t : \varphi \succ \Delta$ fails in some model \mathcal{M} on an assignment μ , that is to say that we have $\mathcal{M} \Vdash^+ \Gamma[\mu], \mathcal{M} \Vdash^+ \underline{s} \cdot t : \varphi[\mu]$ and $\mathcal{M} \Vdash^- \Delta[\mu]$. It follows directly that $draw^{\mathcal{M}}(\underline{s} \cdot t)$ and hence that $draw^{\mathcal{M}}(\underline{s})$ by the *(Draw)* condition, and that $\mathcal{M}, \underline{s} \cdot t \models \varphi[\mu]$. Thus applying Lemma 1 it follows that $\mathcal{M}, \underline{s} \models \varphi[\mu]$, and so $\mathcal{M} \Vdash^+ \underline{s} : \varphi[\mu]$ as desired. The case for *(Draw-R)* follows similarly.

(dCut)-case: For the case of the rule of *(dCut)* suppose that the sequent $\Gamma \succ \Delta$ fails in the model \mathcal{M} on an assignment μ —i.e. that $\mathcal{M} \Vdash^+ \Gamma[\mu], \mathcal{M} \Vdash^- \Delta[\mu]$, and further that for some formula B and term sequent \underline{s}' we have $\underline{s} \cdot \underline{s}' : B \in \Gamma \cup \Delta$. From the above it follows by the *(Draw)* condition that $draw^{\mathcal{M}}(\mu(\underline{s}))$ and so either $\mathcal{M}, \mu(\underline{s}) \models A[\mu]$ or $\mathcal{M}, \mu(\underline{s}) \not\models A[\mu]$ —i.e. that either $\mathcal{M} \Vdash^+ \underline{s} : A[\mu]$ or $\mathcal{M} \Vdash^- \underline{s} : A[\mu]$, meaning that one of the sequents $\Gamma \succ \Delta, \underline{s} : A$ or $\Gamma, \underline{s} : A \succ \Delta$ must fail in \mathcal{M} on the variable assignment μ , as desired. □

4 Completeness

What we are going to do here is to construct a reduction tree. We will use reduction trees a few times in what is to follow so we will define the procedure in general terms here so that we can use it in the completeness proofs below. Our presentation below is largely a generalisation of that given in Takeuti (1987).

4.1 Reduction Rules

The first concept we will need is that of a *reduction rule*, which is a rule which tells us how we can extend our reduction tree. For example, the reduction rule for $(\neg R)$ has the following form.

$$(\neg R\text{-Red}) : \frac{\Gamma, \underline{s} : A \succ \Delta, \underline{s} : \neg A}{\Gamma \succ \Delta, \underline{s} : \neg A}$$

We apply this rule to a formula $\underline{s} : \neg A$ appearing on the RHS of the sequent turnstile by extending our reduction tree by the addition of the node (or in the case of multi-premise rules, nodes) above.

Unlike the reduction rules which one might commonly see in completeness proofs for the classical sequent calculus (like that in Takeuti (1987)), some of our reduction rules can only be applied under certain circumstances. For example, the reduction rule for $(\forall L)$ is of the following form.

$$(\forall L\text{-Red}): \frac{\Gamma, \underline{s} \cdot t : A(t), \underline{s} : \forall x A(x) \succ \Delta}{\Gamma, \underline{s} : \forall x A(x) \succ \Delta}$$

where for some B and some \underline{s}' we have $\underline{s} \cdot t \cdot \underline{s}' : B \in \Gamma \cup \Delta$

What this means is that in order for us to apply this rule to a formula $\underline{s} : \forall x A(x)$ appearing on the LHS of the sequent turnstile we require that there be some formula of the appropriate form appearing somewhere in the sequent. If the side conditions to a reduction rule are not met, then we do not extend the branch.

We will now explain how we can use reduction rules to construct a *reduction tree*. This is a tree, each node of which is a sequent. If the tip of a branch in the reduction tree is an axiom then we will say that that branch is *closed*, otherwise it is *open*.

Definition 5 Given a sequent $\mathcal{S} = \Gamma \succ \Delta$, and a set \mathbb{R} of reduction rules the \mathbb{R} -*reduction tree* of \mathcal{S} is the tree formed by the following iterative procedure.

- **Stage 0** At stage 0 we begin with $\Gamma \succ \Delta$ as the root node of the tree.
- **Stage $n + 1$** At stage $n + 1$ we apply all of the reduction procedures in \mathbb{R} in the order given as many times as possible to every formula introduced at an earlier stage which occurs in the tip of an open branch.

The tree which results of applying this procedure ω -many times is the \mathbb{R} -*reduction tree* of \mathcal{S} .

Theorem 1 (Completeness) *Let $\mathcal{S} = \Gamma \succ \Delta$ be a sequent. Then either there is a (dCut)-free proof of \mathcal{S} , or there is an urn model in which \mathcal{S} fails.*

Proof Given a sequent \mathcal{S} let $R_{\mathcal{S}}$ be the \mathbb{UC} -reduction tree of \mathcal{S} (Table 1). By the construction of the reduction tree either the tip of every branch in the tree is an axiom (in which case it is trivial work to convert the reduction tree into a proof of \mathcal{S}) or the tree has a (potentially infinite) open branch. Letting $\Gamma_0 \succ \Delta_0$ be \mathcal{S} we can think of this branch as being of the form

$$\Gamma_0 \succ \Delta_0, \Gamma_1 \succ \Delta_1, \dots, \Gamma_n \succ \Delta_n, \dots$$

and let $\mathbf{\Gamma} = \cup \Gamma_i$ and $\mathbf{\Delta} = \cup \Delta_i$. We will now use $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ to construct an urn model in which our sequent \mathcal{S} fails. Let us define the model $\mathcal{M}^{\mathcal{S}} = \langle U^{\mathcal{S}}, D^{\mathcal{S}}, V^{\mathcal{S}} \rangle$ as follows.

- $U^{\mathcal{S}}$ is the set of all terms from L^U .
- $D^{\mathcal{S}}(\emptyset) = \{t\}$ for some L -formula A and term sequence $\underline{s}, t \cdot \underline{s} : A \in \mathbf{\Gamma} \cup \mathbf{\Delta}$
- $D^{\mathcal{S}}(\underline{s}) = \{t\}$ for some L -formula A and term sequence $\underline{s}', \underline{s} \cdot t \cdot \underline{s}' : A \in \mathbf{\Gamma} \cup \mathbf{\Delta}$
- $V^{\mathcal{S}}(F) = \{\langle t_1, \dots, t_n \rangle \mid \text{for some sequence of terms } s_1, \dots, s_n, \langle s_1, \dots, s_n \rangle : Ft_1 \dots t_n \in \mathbf{\Gamma}\}$

All that remains to be done is to show that $\mathcal{M}^{\mathcal{S}}$ is an urn model, and that \mathcal{S} fails in $\mathcal{M}^{\mathcal{S}}$.

$\mathcal{M}^{\mathcal{S}}$ is an urn model: It is easy to verify that $\mathcal{M}^{\mathcal{S}}$ satisfies the condition (*Draw*) and thus is an urn model. Suppose that $D^{\mathcal{S}}(\underline{s} \cdot s_n) \neq \emptyset$. Then there is some b and some (possibly empty) term sequence \underline{s}' s.t. for some formula A we have $\underline{s} \cdot s_n \cdot b \cdot \underline{s}' : A \in$

Table 1 The UC-reduction rules

($\neg L$ -Red):	$\frac{\Gamma, \underline{s} : \neg A \triangleright \underline{s} : A, \Delta}{\Gamma, \underline{s} : \neg A \triangleright \Delta}$
($\neg R$ -Red):	$\frac{\Gamma, \underline{s} : A \triangleright \underline{s} : \neg A, \Delta}{\Gamma \triangleright \underline{s} : \neg A, \Delta}$
($\wedge L$ -Red):	$\frac{\Gamma, \underline{s} : A \wedge B, \underline{s} : A, \underline{s} : B \triangleright \Delta}{\Gamma, \underline{s} : A \wedge B \triangleright \Delta}$
($\wedge R$ -Red):	$\frac{\Gamma \triangleright \underline{s} : A, \underline{s} : A \wedge B, \Delta \quad \Gamma \triangleright \underline{s} : B, \underline{s} : A \wedge B, \Delta}{\Gamma \triangleright \underline{s} : A \wedge B, \Delta}$
($\forall L$ -Red):	$\frac{\Gamma, \underline{s} : \forall x A(x), \underline{s} \cdot t : A(t) \triangleright \Delta}{\Gamma, \underline{s} : \forall x A(x) \triangleright \Delta}$
where for some B and some \underline{s}' we have $\underline{s} \cdot t \cdot \underline{s}' : B \in \Gamma \cup \Delta$	
($\forall R$ -Red):	$\frac{\Gamma \triangleright \Delta, \underline{s} : \forall x A(x), \underline{s} \cdot a : A(a)}{\Gamma \triangleright \Delta, \underline{s} : \forall x A(x)}$
where a is the first free variable which does not occur in $\Gamma \cup \Delta \cup \{ \underline{s} : \forall x A(x) \}$.	
(<i>Draw L</i> -Red):	$\frac{\Gamma, \underline{s} \cdot t : \varphi, \underline{s} : \varphi \triangleright \Delta}{\Gamma, \underline{s} \cdot t : \varphi \triangleright \Delta}$
where φ is quantifier free.	
(<i>Draw R</i> -Red):	$\frac{\Gamma \triangleright \underline{s} \cdot t : \varphi, \underline{s} : \varphi, \Delta}{\Gamma \triangleright \underline{s} \cdot t : \varphi, \Delta}$
where φ is quantifier free.	

The reduction rules for UC consist of the following reduction rules applied in the order given (in the case of ($\forall R$ -Red) applying the rule for all applicable terms t)

$\Gamma \cup \Delta$. But this also means that there is some term sequence \underline{u} s.t. $\underline{s} \cdot s_n \cdot \underline{u} : A \in \Gamma \cup \Delta$, and so $s_n \in D^S(\underline{s})$, as desired.

\mathcal{M}^S is a counter-model to \mathcal{S} : What we need to show here is that:

$$\mathcal{M}^S \Vdash^+ \Gamma[i] \quad \& \quad \mathcal{M}^S \Vdash^- \Delta[i],$$

where i is the variable assignment s.t. $i(x) = x$ for all variables x .

We proceed by induction on the complexity of formulas, the basis case following from the definition of V^S , the cases for the boolean connectives are standard, the only case of interest being that of the quantifiers. We begin by noting that, by the construction of D^S that for every L -formula B and term sequence \underline{s} , if $\underline{s} : B \in \Gamma \cup \Delta$ then $draw^{\mathcal{M}^S}(i(\underline{s}))$.

Suppose, then, that $\underline{s} : \forall x A(x) \in \Gamma$. Then by the ($\forall L$ -Red) reduction step it follows that we have

$$\underline{s} \cdot t : A(t) \in \Gamma$$

for every term t s.t. for some formula B and term sequence \underline{s}' we have $\underline{s} \cdot t \cdot \underline{s}' : B \in \Gamma$. By the construction of \mathcal{M}^S these are precisely the terms which are in $D^S(i(\underline{s}))$. So by the induction hypothesis it follows that $\mathcal{M}^S \Vdash^+ \underline{s} \cdot t : A(t)[i]$ for all $t \in D^S(i(\underline{s}))$,

and thus $\mathcal{M}^S, i(\underline{s} \cdot t) \models A(t)[i]$. From which it follows that $\mathcal{M}^S \Vdash^+ \underline{s} : \forall x A(x)[i]$ as desired.

Suppose, now, that $\underline{s} : \forall x A(x) \in \Delta$. Then by the ($\forall R$ -Red) reduction step it follows that we have

$$\underline{s} \cdot a : A(a) \in \Delta.$$

So by the induction hypothesis it follows that $\mathcal{M}^S \Vdash^- i(\underline{s}) \cdot a : A(a)[i]$, and thus that $\mathcal{M}^S, i(\underline{s} \cdot a) \not\models A(a)[i]$, from which it follows that $\mathcal{M}^S, i(\underline{s}) \not\models \forall x A(x)[i]$, giving us $\mathcal{M}^S \Vdash^- \underline{s} : \forall x A(x)[i]$, as desired. \square

This gives us the following corollary.

Corollary 1 *The rule of ($dCut$) is eliminable from UC.*

5 Rantala’s Conditions

The system described in the previous section is sound and complete relative to the class of all urn models, as we have defined them above. As noted above, though, this is not quite the class of urn models which Rantala is concerned with, his models also satisfying the conditions (*Ran1*) and (*Ran2*) below.

$$(Ran1) : D(\emptyset) = U$$

$$(Ran2) : D(\langle a_1, \dots, a_n \rangle) \neq \emptyset \text{ if and only if } a_n \in D(\langle a_1, \dots, a_{n-1} \rangle).$$

What we will do in this section is to look at extensions of the sequent calculus UC which are sound and complete w.r.t. models which satisfy the above conditions (or in the case of (*Ran1*) a near neighbour thereof). We begin by looking at the sequent calculus for models which satisfy (*Ran2*).

5.1 The Condition (*Ran2*)

In order to capture the logic of Urn models which satisfy (*Ran2*) we need to extend our system by the following three rules—two structural, and one additional ($\forall L$)-rule.

$$\frac{\Gamma, \underline{s} \cdot a : A(a) \succ \Delta}{\Gamma, \underline{s} : \forall x A(x) \succ \Delta} (\forall L)\text{-Ran2} \qquad \frac{\Gamma, \underline{s} \cdot a : \varphi \succ \Delta}{\Gamma, \underline{s} : \varphi \succ \Delta} (\text{Ran2L}) \qquad \frac{\Gamma \succ \underline{s} \cdot a : \varphi, \Delta}{\Gamma \succ \underline{s} : \varphi, \Delta} (\text{Ran2R})$$

where in all three rules a is an eigenvariable (i.e. a does not occur in the lower sequent in any of these three reduction rules), and φ is a quantifier free formula. Call the extension of UC by the above rules UCR2.

These three rules force every draw to always be extendable. To see this first note that all three rules are sound in urn models which satisfy (*Ran2*).

Proposition 2 *UCR2 is sound for the class of all urn models which satisfy (*Ran2*).*

Proof For $(\forall L)$ -*Ran2* suppose that we have $\mathcal{M} \Vdash^+ \underline{s} : \forall x A(x)[\mu]$ — i.e. that $draw^{\mathcal{M}}(\mu(\underline{s}))$ and $\mathcal{M}, \mu(\underline{s}) \models \forall x A(x)[\mu]$. From $draw^{\mathcal{M}}(\mu(\underline{s}))$ we get that there must be some object $e \in D(\mu(\underline{s}))$ by $(Ran2)$. So we can simply let $\mu(a) = e$ and thus we have $\mathcal{M}, \mu(\underline{s} \cdot a) \models A(a)[\mu]$ as desired.

For the $(Ran2L)$ case suppose that we have $\mathcal{M} \Vdash^+ \underline{s} : \varphi[\mu]$. Then we have that $\mu(\underline{s})$ is a draw from \mathcal{M} and thus, by $(Ran2)$ that there is some object $\mu(\underline{s}) \cdot b$ is also a draw. Given that we also have $\mathcal{M}, \mu(\underline{s}) \models \varphi[\mu]$ it follows by Lemma 1 that $\mathcal{M}, \mu(\underline{s} \cdot a) \models \varphi[\mu]$ where $\mu(a) = e$, yielding $\mathcal{M} \Vdash^+ \underline{s} \cdot a : \varphi[\mu]$ as desired. The case of $(Ran2R)$ follows similarly. \square

Lemma 2 *Suppose that \mathcal{S} is a UCR2-unprovable sequent and that R is a reduction tree for \mathcal{S} for which the UCR2 (and hence UC) reduction rules have been applied, and that*

$$\Gamma_0 \succ \Delta_0, \dots, \Gamma_n \succ \Delta_n, \dots$$

is an open branch in R (there is guaranteed to be such a branch as the sequent is unprovable). Then for all i , if $\underline{s} : A \in \Gamma_i \cup \Delta_i$ then for some j , there is a quantifier-free formula φ_A and a sequence of terms \underline{s}' s.t. $\underline{s}s' : \varphi_A \in \Gamma_j \cup \Delta_j$.

Proof We prove something stronger, namely that for every formula A there is some such atomic formula (or formulas) φ_A s.t. for some sequence of terms \underline{s}' we have $\underline{s}s' : \varphi_A \in \Gamma_j \cup \Delta_j$.

To prove this we show that for every formula $\underline{s} : A \in \Gamma_k \cup \Delta_k$, there is a formula B of lower complexity such that, for some sequence of terms $\underline{s}', \underline{s}s' : B \in \Gamma_k \cup \Delta_k$. To see this note that the reduction steps for UC guarantee that at each stage we can produce a formula of the required kind whenever the main connective of A is not a quantifier. In the case where the main connective of A is \forall either we can produce the appropriate formula using $(\forall R\text{-Red})$, or using $(\forall L\text{-Red})$ (if the conditions are met), or (and here is where we require UCR2) we can produce it using $(\forall L\text{-Ran2-Red})$. \square

Proposition 3 *UCR2 is complete for the class of all urn models which satisfy $(Ran2)$.*

Proof The proof is just like that of Theorem 1, except that we now create a UCR2 reduction tree for our sequent \mathcal{S} (Table 2). To verify that this results in a countermodel to \mathcal{S} which satisfies condition $(Ran2)$ suppose that $s_{n+1} \in D^{\mathcal{S}}(\langle s_1, \dots, s_n \rangle)$. Then by the construction of $\mathcal{M}^{\mathcal{S}}$ there must be some formula of the form $\langle s_1, \dots, s_{n+1} \rangle \cdot \underline{s}' : B \in \Gamma \cup \Delta$. If any such formula has a non-empty sequence \underline{s}' then we are done, so suppose that all such formulas are of the form $\langle s_1, \dots, s_{n+1} \rangle : B$. Such a formula must have been in the tree at some stage k , and so we know by Lemma 2 that there is some later stage at which we have some sequence of terms \underline{s}'' and some quantifier-free formula φ s.t. $\langle s_1, \dots, s_{n+1} \rangle \cdot \underline{s}'' : \varphi \in \Gamma \cup \Delta$. Again, if \underline{s}'' is non-empty then we are done, so suppose it is empty. Then by (depending on whether it is in Γ or Δ) either the $(Ran2-L\text{-Red})$ or $(Ran2-R\text{-Red})$ reductions we know that, for some term a that $\langle s_1, \dots, s_{n+1} \rangle \cdot a : \varphi \in \Gamma \cup \Delta$, and so by construction of $\mathcal{M}^{\mathcal{S}}$ we have that for some term a , $a \in D^{\mathcal{S}}(\langle s_1, \dots, s_{n+1} \rangle)$, as desired. \square

Table 2 The UCR2-reduction rules

	$\frac{\Gamma, \underline{s} : \forall x A(x), \underline{s} \cdot a : A(a) \succ \Delta}{\Gamma, \underline{s} : \forall x A(x) \succ \Delta}$
($\forall L$ - <i>Ran2-Red</i>):	
where a is the first free variable which does not occur in $\Gamma \cup \Delta \cup \{\underline{s} : \forall x A(x)\}$.	
	$\frac{\Gamma, \underline{s} : \varphi, \underline{s} \cdot a : \varphi \succ \Delta}{\Gamma, \underline{s} : \varphi \succ \Delta}$
(<i>Ran2L-Red</i>):	
where a is the first free variable which does not occur in $\Gamma \cup \Delta \cup \{\underline{s} : \varphi\}$, and φ is quantifier free.	
	$\frac{\Gamma \succ \underline{s} : \varphi, \underline{s} \cdot a : \varphi, \Delta}{\Gamma \succ \underline{s} : \varphi, \Delta}$
(<i>Ran2R-Red</i>):	
where a is the first free variable which does not occur in $\Gamma \cup \Delta \cup \{\underline{s} : \varphi\}$, and φ is quantifier free.	

The reduction rules for UCR2 consist of all the reduction rules for UC followed by the reduction rules given below, with the rules given below being applied after all the UC reduction rules, in the order given below

5.2 (*Ran1*) and Almost-Rantala Models

What we will do in this section is to describe a system, which we will call **UCRant-** which agrees with the system described by Rantala when every formula which occurs in a sequent is a sentence.² In fact, **UCRant-** is a slightly stronger system, being sound and complete w.r.t. the class of all urn models which satisfy both (*Ran2*) as well as the following condition:

$$(\text{Ran1}') : \quad t \in D(\underline{s}) \Rightarrow t \in D(\emptyset).$$

This condition makes each $D(\underline{s})$ be a subset of $D(\emptyset)$. Let us call urn models which satisfy (*Ran1'*) and (*Ran2*) *almost-Rantala models*, and those which satisfy (*Ran1*) and (*Ran2*) *Rantala models*. It is relatively simple to see that any almost-Rantala model $\mathcal{M} = \langle U, D, V \rangle$ can be transformed into a Rantala model $\mathcal{M}' = \langle U', D, V' \rangle$ where U' is the union of all $D(\underline{s})$, where \underline{s} is a sequence of objects from $D(\emptyset)$, and where V' is just V restricted to U' in the obvious way. These two models are equivalent in the sense that for all L^U -formulas α (recalling that these formulas do not contain individual constants) and all assignments μ which are restricted to U' we have $\mathcal{M} \models \alpha[\mu]$ iff $\mathcal{M}' \models \alpha[\mu]$. In particular this means that for all L^U -formulas of the form $\emptyset : A$ where A is a sentence we have $\mathcal{M} \models \emptyset : A[\mu]$ iff $\mathcal{M}' \models \emptyset : A[\mu]$ for all assignments μ , and so almost-Rantala and Rantala models agree on all sentences when our language does not contain individual constants.

Let **UCRant-** be the result of adding the following rules to UCR2.

$$\frac{\Gamma, t : \varphi \succ \Delta}{\Gamma, \underline{s} \cdot t \cdot \underline{u} : \varphi \succ \Delta} \quad (\text{Ran1}'\text{-L}) \qquad \frac{\Gamma \succ t : \varphi, \Delta}{\Gamma \succ \underline{s} \cdot t \cdot \underline{u} : \varphi, \Delta} \quad (\text{Ran1}'\text{-R})$$

² It is important to recall at this point that the languages under consideration here do not contain individual constants. Up until now this has simply been a matter of convenience of exposition, here it matters.

Table 3 The UCRANT[−]-reduction rules

<p>(Ran1'-L)-Red: where φ is quantifier free.</p>	$\frac{\Gamma, \underline{s} \cdot t \cdot \underline{u} : \varphi, t : \varphi \succ \Delta}{\Gamma, \underline{s} \cdot t \cdot \underline{u} : \varphi \succ \Delta}$
<p>(Ran1'-R)-Red: where φ is quantifier free.</p>	$\frac{\Gamma \succ \underline{s} \cdot t \cdot \underline{u} : \varphi, t : \varphi, \Delta}{\Gamma \succ \underline{s} \cdot t \cdot \underline{u} : \varphi, \Delta}$

The reduction rules for UCRANT[−] consist of all the reduction rules for UCR2 followed by the reduction rules given below, with the rules given below being applied after all the UCR2 reduction rules, in the order given below

What we will now show is that the system UCRant[−] is sound and complete w.r.t. the class of all almost-Rantala models. Note that our proofs below do not show that the result of adding the above rules to UC forces it to satisfy (Ran1'), as our proofs make heavy use of Lemma 2.

Proposition 4 UCRant[−] is sound for the class of all urn models which satisfy both (Ran1') and (Ran2).

Proof What needs to be shown is that (Ran1'-L) and (Ran1'-R) are sound in models which satisfy (Ran1'). We treat just the cases of the left rule here (the right rule following similarly).

Suppose that a conclusion of an application of (Ran1'-L) were to fail in a model \mathcal{M} on an assignment μ . Then in particular we would have that $\mathcal{M} \Vdash^+ \underline{s} \cdot t \cdot \underline{u} : \varphi[\mu]$, and so $draw^{\mathcal{M}}(\mu(\underline{s} \cdot t \cdot \underline{u}))$. and $\mathcal{M}, \mu(\underline{s} \cdot t \cdot \underline{u}) \models \varphi[\mu]$. By (Draw) it follows that $\mu(t) \in D(\underline{s})$ and so by (Ran1') that $\mu(t) \in D(\emptyset)$, and thus that $\mu(t)$ is a draw from \mathcal{M} . Then by Lemma 1 it follows that $\mathcal{M}, \mu(t) \models \varphi[\mu]$ and thus $\mathcal{M} \Vdash^+ t : \varphi[\mu]$ as desired. □

Proposition 5 UCRant[−] is complete for the class of all urn models which satisfy (Ran1') and (Ran2).

Proof The proof is just like that of Theorem 1, except that we now create a UCRANT[−]-reduction tree for our sequent \mathcal{S} (Table 3). That this results in a countermodel to \mathcal{S} which satisfies condition (Ran2) follows as in Proposition 3. All that remains to be shown is that the resulting countermodel satisfies (Ran1').

Suppose, then, that $t \in D^{\mathcal{S}}(\underline{s})$ for some sequence \underline{s} . Then by the construction of $\mathcal{M}^{\mathcal{S}}$ it follows that for some formula B and sequence of terms \underline{s}' we have $\underline{s} \cdot t \cdot \underline{s}' : B \in \Gamma \cup \Delta$. Then by Lemma 2 there is some sequence of terms \underline{u} and variable free formula φ s.t. $\underline{s} \cdot t \cdot \underline{s}' \cdot \underline{u} : \varphi \in \Gamma \cup \Delta$. So by either (Ran1'-L)-Red or (Ran1'-R)-Red it follows that $t : \varphi \in \Gamma \cup \Delta$, and so by the construction of $D^{\mathcal{S}}$ that $t \in D^{\mathcal{S}}(\emptyset)$ as desired. □

6 Conclusion

One of the things which we have endeavoured to do here in our proof theoretic investigation of urn logic is to provide the first steps towards a more general framework

for thinking about the quantifiers more generally. In particular the alterations to the language we made in Sect. 3 were made specifically in order to induce extra structure into our formulas (and thus into our sequents) which we could adjust without altering the operational rules for our quantifiers in order to validate further inferences. Our approach here has been directed specifically at the goal of giving a proof theoretic characterisation of urn logic, rather than a general investigation of this approach to the proof theory of the quantifiers, the more general study having to wait until another occasion.

Acknowledgments I would like to thank Dave Ripley, Greg Restall and Lloyd Humberstone for their helpful comments and suggestions. I'm also grateful to an anonymous referee from this journal who provided a number of invaluable comments, and noticed a rather serious (but thankfully repairable) problem in an earlier version of this paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Cresswell, M. (1982). Urn models: A classical exposition. *Studia Logica*, 41, 109–130.
- Hintikka, J. (1956). Identity, variables, and impredicative definitions. *Journal of Symbolic Logic*, 21, 225–245.
- Hintikka, J. (1975). Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4, 475–484.
- Hintikka, J. (1979). Quantifiers in logic and quantifiers in natural language. In E. Saarinen (Ed.), *Game-theoretical semantics* (pp. 27–47). Dordrecht: D. Reidel Publishing Co.
- Humberstone, L. (2008). Can every modifier be treated as a sentence modifier? *Philosophical Perspectives*, 22, 241–275.
- Olin, P. (1978). Urn models and categoricity. *Journal of Philosophical Logic*, 7, 331–345.
- Rantala, V. (1975). Urn models: A new kind of non-standard model for first-order logic. *Journal of Philosophical Logic*, 4, 445–474.
- Stanley, J. (2000). Context and logical form. *Linguistics and Philosophy*, 23, 391–434.
- Takeuti, G. (1987). *Proof theory*. Amsterdam: North-Holland.
- Wehmeier, K. (2004). Wittgensteinian predicate logic. *Notre Dame Journal of Formal Logic*, 45(1), 1–11.
- Wehmeier, K. (2009). On ramsey's 'silly delusion' regarding tractatus 5.53. In G. Primiero & S. Rahman (Eds.), *Acts of knowledge-history, philosophy and logic* (pp. 353–368). London: College Publications.