CIRCULARITY AND CONVENTION (T)

MICHAŁ WALICKI

Abstract. We show that extending any FOL language with quantification over all sentences – of the extended language – does not increase expressive power, in particular, does not lead to any paradoxes. Further extension with sentential operators, by a form of definitional extension, does not lead to any paradoxes, either, but arbitrary such predicates can lead to paradoxes. We give some conditions ensuring their absence.

§ 1. Introduction. In terms truth, the first point of departure from the main tradition is that we say “All following sentences are true” rather than “All following sentence-names name true sentences.” To avoid diagonal lemma, we do not arithmetize syntax but apply truth directly to sentences so that, technically, it is an operator and not a predicate. Hopefully, the reader will excuse our referring to it as a sentential predicate. It functions like one and, combined with sentential quantifiers, resolves the worries about generalization and finite axiomatizability related to operators, allowing to write (T) as a single sentence.

This is achieved by the main novelty in treatment of circularity. Presentation of syntax as digraphs, and of semantics as their kernels, introduced and applied in [2, 8, 10], allows us to extend standard semantics to quantification over all sentences. Formalizations of sentential quantification, that are not modal and do not arithmetize syntax, e.g., [1, 4, 5], utilize some language $\mathcal{L}$ quantifying only over sentences of a sublanguage $\mathcal{L}^-$, due to unclear means for defining semantics of self-reference ensuing from unrestricted sentential quantification. “All sentences are true” is intuitively false and is formally so, if taken as an $\mathcal{L}$-sentence quantifying over $\mathcal{L}^-$-sentences. Formalization of the fact that, being false, it witnesses to its own falsity, has proven elusive. The graph technique allows us to define semantics of an extension of $\mathcal{L}^-$ to $\mathcal{L}$ with quantification over all $\mathcal{L}$-sentences, and to show that the resulting self-reference does not lead to paradoxes.

The approach should be applicable to logics possessing graph normal form, [2, 10], but our example throughout is FOL. Section 2 gives the background of graph syntax and kernel semantics for FOL. Section 3 extends this to quantification over all sentences and Section 4 shows that such an extension does not lead to any paradoxes, allowing to assign consistently boolean values to all sentences. In fact, sentential quantification over FOL is close to trivial, reducing to quantification over boolean values and staying within FOL, Section 4.1. Section 5 includes axioms into graph presentation, adds sentential predicates and establishes some conditions on their definitions ensuring absence of paradoxes.

§ 2. Background – FOL digraphs. The initial language $\mathcal{L}^-$ can be always instantiated by FOL with the following grammar (and missing connectives defined in the classical way):

\[
\begin{align*}
T_X &::= X | Const | Func(T_X|...|T_X) \\
A^>_X &::= P1(T_X|...|T_X) \\
F^>_X &::= A^>_X | \neg F^>_X | F^>_X \land F^>_X | \forall X.F^>_X
\end{align*}
\]  

$P1$ is the category of predicate names. We do not pay much attention to actual variables $X$, $Const$ and $Func$ symbols, assuming always a domain $M$ with their standard interpretation. In general, the free algebra of terms over $M$, denoted by $T_M$, plays important role, but we present results for FOL without equality, assuming also $T_M = M$. Although “domain $M$” means thus here set $M$, it is used because results hold also for terms $T_M \neq X$ and equality, by extension following [10]. Elements of $T_M$ appear in $\mathcal{L}^>_M$-sentences, i.e., $\mathcal{L}^-$-sentences over terms $T_M$, denoted by $S^>_M$.
For every domain \( M \), \( \mathcal{L}^- \)-sentences form a subset \( S^- \subseteq S^-_M \subseteq F^-_M \). An \( \mathcal{L}^- \)-structure is a pair \((M, \rho)\) with a domain \( M \) and valuation \( \rho \in 2^{A_M} \) of atoms.

For a language \( \mathcal{L}^- \) and a domain \( M \), we form the language graph \( S_M(\mathcal{L}^-) \), over vertices \( S^-_M \) with some auxiliaries. \( L^- \)-sentences play thus semantic role and it may be simpler, while equally adequate, to take as \( S^-_M \) only a unique representative from each class of sentences identical up to correct renaming of bound variables. By “graph” we mean a directed graph, so “tree” means arborescence, with edges directed from the root and all internal vertices to their children. By \( ar(F) \) we denote the arity of \( F \), i.e., the number of non-instantiated arguments, or free variables, in \( F \). We use this notation also for partial applications; if \( ar(B) = 3 \), then \( ar(B(a)) = 2 \) after a partial application \( B(a) \). Graph \( S_M(\mathcal{L}^-) \) contains, for each sentence \( A \in S^-_M \), a tree \( T_M(A) \) with \( A \) as the root. In \( \mathcal{L}^- \), this tree is the same as the subgraph rooted at \( A \), denoted by \( S_M(A) \), but the two will become distinct in the next section, so we introduce both notations at once.

**Definition 2.2.** The language graph \( S_M(\mathcal{L}^-) \), for a domain \( M \), is given by:

1. vertices, \( V = S^-_M \cup \text{AUX} \), with some auxiliary vertices \( \text{AUX} \), shown by examples below.
2. Each atomic sentence \( A \in A^-_M \) has a 2-cycle to its negation: \( A \vdash \neg A \).
3. Each non-atomic sentence \( S \in S^-_M \) is the root of the subgraph \( S_M(S) \):

\[
\begin{align*}
(a) \quad & \neg F & \rightarrow & \quad S_M(F), \\
(b) \quad & F_1 \land F_2 & \rightarrow & \quad S_M(\neg F_i), \text{ for } i \in \{1, 2\}, \\
(c) \quad & \forall xFx & \rightarrow & \quad S_M(\neg F(d)), \text{ for each } d \in T_M^{ar(F)}.
\end{align*}
\]

Aux vertices represent also sentences \( S^-_M \), with which they could be identified, but keeping them separated makes the structure clearer, without affecting the results. When \( M \) is inessential, we may drop it, writing \( S(\mathcal{L}^-) \) or \( S(A) \). \( T_M(A) \), which now equals \( S_M(A) \), reminds of \( A \)'s parse tree but, primarily, it reflects the semantics of the formula constructors \((\neg, \land, \lor)\) in terms of kernels, presented in Section 2.1. Out-branching represents conjunction (or universal quantification), and each edge negation of its target. The 2-cycle at each atom, \( A \vdash \neg A \), will force choice of one element from each such pair, representing valuation of atoms.

The universal and existential quantifiers \((\exists x \text{ defined as } \neg \forall x \neg)\) are represented by the following branchings to instantiations of the quantified variable by all elements \( a, b, c, \ldots \) of the domain. (As explained in Section 2.1, a double edge \( x \rightarrow y \rightarrow z \), where \( x \) has no other out-neighbours and \( y \) no other neighbours, can be contracted by removing \( y \) and identifying \( x = z \). This is done for \( \exists \)-pattern to the right. Overlined \( X \) stands for \( \neg X \) )

\[
\begin{align*}
\forall x D(x) & \quad \text{ not } \quad \exists x D(x) \\
D(a) & \quad \text{ not } \quad D(a) \\
D(b) & \quad \text{ not } \quad D(b) \\
D(c) & \quad \text{ not } \quad D(c) \\
\ldots & \quad \text{ not } \quad \ldots \\
\forall x \neg D(x) & \quad \text{ not } \quad \exists x \neg D(x) \\
D(a) & \quad \text{ not } \quad D(a) \\
D(b) & \quad \text{ not } \quad D(b) \\
D(c) & \quad \text{ not } \quad D(c) \\
\ldots & \quad \text{ not } \quad \ldots
\end{align*}
\]

Quantifier prefix is converted to the graph by successively performing such branchings, until no quantified variables remain. For instance, \( \forall x \exists y D(x, y) \) becomes

\[
\begin{align*}
\exists y D(a, y) & \quad \text{ not } \quad \forall y D(c, y) \\
\forall y D(a, y) & \quad \text{ not } \quad \exists y D(c, y) \\
\exists y D(b, y) & \quad \text{ not } \quad \forall y D(c, y) \\
\forall y D(b, y) & \quad \text{ not } \quad \exists y D(c, y) \\
\exists y D(c, y) & \quad \text{ not } \quad \forall y D(c, y) \\
\forall y D(c, y) & \quad \text{ not } \quad \exists y D(c, y) \\
\end{align*}
\]

Definition 2.2 applies to all sentences but, wlog., we will address mostly sentences in prenex normal form with matrix in disjunctive normal form, \( \text{PDNF} \). For such a matrix \( D(\bar{x}) = C_1(\bar{x}) \lor \ldots \lor C_d(\bar{x}) \),
each instance \(D(\bar{a})\) has an edge to \(\circ_{D(\bar{a})}\), representing \(\neg C_1(\bar{a}) \land ... \land \neg C_d(\bar{a})\), and branches to each disjunct of \(D(\bar{a})\), i.e., conjunction \(C_i(\bar{a}), 1 \leq i \leq d\). Each \(C_i(\bar{a})\) has single edges to its negative literals and double edges to the positive ones. A single edge represents negation of its target, and a double edge, as double negation, the positive atom. These are not distinguished on the generic picture to the left, but are on the example to the right for \(D(x, y) = (P(x) \land \neg R) \lor (\neg Q(x, y) \land R)\):

\[
\begin{align*}
D(\bar{a}) & \xrightarrow{\circ_{D(\bar{a})}} D(a, b) \\
C_1(\bar{a}) \quad C_1(\bar{a}) \quad C_d(\bar{a}) & \quad C_1(a, b) \quad C_2(a, b) \\
L_1(\bar{a}) \cdots R_1 \quad L_1(\bar{a}) \cdots R_i & \quad L_d(\bar{a}) \cdots R_d \quad P(a) \xrightarrow{P(a)} R \xleftrightarrow{\bar{R}} Q(a, b)
\end{align*}
\]

\(R_i\) mark possible ground atoms containing none of the variables quantified in \(A\). Each sentence \(A\) yields thus a tree \(T_M(A)\) with the root \(A\), followed by quantifier branchings with all substitutions for the quantified variables, (2.4), down to the roots of \(\text{DNF-feet} (2.5)\), called \(D\)-level vertices \((D(\bar{a}), D(a, b))\), and respective instances of conjuncts \(C_i\), the \(C\)-level vertices.

Atoms refer to the leaves of trees: negative literals (with a single edge from a \(C\) vertex) and positive ones (with a double edge from \(C\)). Atomic sentences, called \(s\)-atoms or \(\mathcal{L}\)-atoms (for an appropriate language \(\mathcal{L}\)), occur in a graph only as its atoms. Atoms of \(T_M(A)\) are now only \(\mathcal{L}^\neg\)-atoms, but this will change as trees will cease to be trees and leaves, ceasing to be leaves, end up as atoms lying on cycles. On the example to the right, atoms are \(P(a), R\) and \(Q(a, b)\). Atoms, shared by distinct trees, break the tree property, having also 2-cycles by Definition 2.2.2. Still, we retain the tree language, as trees form the spine of the structure of language graphs.

All subtrees of \(T_M(A)\), rooted at the same level above \(D\)-level, are isomorphic. The height of such trees, counted from the root to the \(C\)-level, is always even.

2.1. Kernels as semantics. Kernels provide an equivalent way for defining semantics: for propositional logic, [2, 8], for FOL [10], and for other logics, possessing graph normal form, Definition 5.5. To make the paper self-contained, we repeat the definition of a kernel and present some basic facts about kernel existence, used later on.

Given a digraph \(G = (V_G, E_G)\), with \(E_G \subseteq V_G \times V_G\), by \(E_G^\neg\) we denote the converse of \(E_G\). For a binary relation \(E\), we let \(E(x) = \{y \mid E(x, y)\}\) and extend function applications pointwise to sets, \(E(X) = \bigcup_{x \in X} E(x)\). A kernel (or solution, introduced in [7]) of \(G\) is a \(K \subseteq V_G\), which is

(a) independent, i.e., \(E_G^\neg(K) \subseteq V_G \setminus K\), and

(b) absorbing, i.e., \(V_G \setminus K \subseteq E_G^\neg(K)\),

in short, such that \(E_G^\neg(K) = V_G \setminus K\). Equivalently, it is an assignment \(\kappa \in 2^V_G\) such that

\[
(2.6) \quad \forall x \in V_G : \kappa(x) = 1 \iff \forall y \in E_G(x) : \kappa(y) = 0.
\]

The set \(\{x \in V_G \mid \kappa(x) = 1\}\) satisfies then (a) and (b), while for \(K\) satisfying (a), (b), assignment \(\kappa\) given by \(\kappa(x) = 1 \iff x \in K\) satisfies (2.6). We therefore do not distinguish the two and by \(\text{sol}(G)\) denote the set of kernels (solutions, or such assignments) for \(G\). \(G\) is solvable when \(\text{sol}(G) \neq \emptyset\).

By the first result about kernels from [7], every tree without infinite branches has a unique kernel. In particular, a sink, i.e., a vertex \(v\) with \(E_G(v) = \emptyset\), belongs to every kernel. Trees in language graph have atomic 2-cycles, \(A \models \bar{A}\), instead of sinks. Exactly one element from such a cycle can be in any kernel. Rephrasing the result, every tree, with such 2-cycles instead of sinks, has exactly one kernel for every selection from these cycles. Using the right tree from (2.5), inclusion of \(P(a)\) and \(\bar{R}\) in a kernel \(K\) forces, by independence, \(\bar{P}(a)\) and \(R\) out of it. This, in turn, forces \(C_i(a, b) \in K\) by absorption, so that \(\bar{\bar{P}(a, b)} \notin K\) and \(D(a, b) \in K\). The implication from \(\{P(a), \bar{R}\} \subseteq K\) to \(D(a, b) \in K\) reflects the implication from \(P(a) = 1, R = 0\) to \(D(a, b) = 1\). Kernel \(K\) of \(S_M(\mathcal{L}^\neg)\) represents exactly the satisfied formulae (with their satisfying assignments) under valuation of atoms given by the selection from atomic 2-cycles, i.e., by \(K\big|_{A_M}\).

There is thus a bijection mapping a FOL structure \((M, \rho)\) to the the language graph \(S_M(\mathcal{L}^\neg)\) with its kernel \(K_\rho\), where \(A \models K_\rho \iff \rho(A) = 1\) for \(\mathcal{L}_M^\neg\)-atoms \(A\). Then also, for all \(S \in S_M\),

\[
(2.7) \quad (M, \rho) \models S \iff S \in K_\rho
\]
In Section 5 we will represent axiomatic theories as language graphs modified so that theory’s contradicting independence of \( K \) and this correspondence underlies the generalization of FOL semantics in what follows. A kernel for a language graph determines boolean values of all sentences, amounting to absence of paradoxes. More background details can be found in the works cited at the beginning of this section.

The two facts below imply equisolvability of graphs, making a stronger claim. If graph \( G' \) arises from \( G \) by contracting even paths and identifying vertices with identical out-neighbourhoods, then \( G \) and \( G' \) have essentially the same solutions, namely, each solution of \( G' \) can be expanded to a solution of \( G \), and each solution of \( G \), restricted to \( G' \), is a solution of \( G' \). These facts, applied implicitly on the drawings, justify also duplication of vertices \( S_M \) as \( \text{Aux} \), without affecting solutions. They are used explicitly only in Section 5.1. The first one is a trivial observation.

A path \( a_0 \ldots a_k \) is isolated if \( E_G(a_i) = \{a_{i+1}\} \) for \( 0 < i < k \) and \( E_G(a_i) = \{a_{i-1}\} \) for \( 0 < i < k \). A double edge, introduced earlier, is an isolated path of length 2. Contraction of such an isolated path amounts to identifying the first and the last vertex, joining their neighbourhoods and removing the intermediate vertices, i.e., obtaining graph \( G' \) where \( \text{V}_{G'} = \text{V}_G \setminus \{a_1 \ldots a_k\} \), \( E_{G'}(a_0) = E_G(a_k) \) and \( E_{G'}(a_k) = E_G(a_0) \cup E_G(a_k) \setminus \{a_1, \ldots, a_{k-1}\} \).

**FAC** 2.8. If \( G' \) results from \( G \) by contracting an isolated path of even length, then \( \forall K' \in \text{sol}(G') \exists K \in \text{sol}(G) : K' \subseteq K \), and \( \forall K \in \text{sol}(G) : K \cap \text{V}_{G'} \in \text{sol}(G') \).

The same holds if \( G' \) results from a transfinite number of such contractions, provided that no ray, i.e., an infinite outgoing path with no repeated vertex, is contracted to a finite path.

The second fact shows that identifying vertices with identical out-neighbourhoods preserves and reflects solutions. To define this operation, let \( R_G \subseteq \text{V}_G \times \text{V}_G \) relate two vertices in \( G \) with identical out-neighbourhoods, i.e., \( R_G(a, b) \Rightarrow E_G(a) = E_G(b) \). It is an equivalence, so let \( G^G \) denote the quotient graph over equivalence classes, \( [v] = \{u \in \text{V}_G \mid R_G(v, u)\} \), with edges \( E_{G^G}([v], [u]) \Leftrightarrow \exists w \in [v], u \in [u] : E_G(v, u) \). The operation can be iterated any number of times, denoted by \( G^{n+1} = G^G \) and \( G^{1+1} = (G^G)^G \). Vertices of \( G^n \) are taken as subsets of \( \text{V}_G \), i.e., \( [u]^n = \{v \in \text{V}_G \mid \exists i < n : R_G(\{u^i\}, [u^i])\} \). For limit ordinals \( \lambda \), \( G^{\lambda}\) is given by \( V_{G^{\lambda}} = \{x \in \text{V}_G \mid x \subseteq \lambda \} \) and \( E_{G^{\lambda}}([x]^\lambda, [u]^\lambda) \Leftrightarrow \exists n \in \lambda : E_{G^n}([x]^n, [u]^n) \).

**FAC** 2.9. For every ordinal \( n, \) (a) \( K \in \text{sol}(G) \Rightarrow \{v\} \cap \text{V} \subseteq \text{sol}(G^{\lambda}) \), and (b) \( K^{n+1} \in \text{sol}(G^{n+1}) \Rightarrow \bigcup K^{n+1} \in \text{sol}(G) \).

**PROOF.** (1) The proof for \( n = 1 \) shows the claim also for every successor \( n \).

(1) \( K\) is independent, for if \( E_G([v], [u]) \) for some \( [v], [u] \subseteq K \), then \( E_G(v, w) \) for some \( v \in [v], w \in [u] \). But then \( v, w \in K \) contradicting independence of \( K \) if \( x \in K \) then \( [x] \subseteq K \), since \( \forall x, y \in [v] : E_G(x, y) \), so \( \forall x \subseteq K \). Then \( [x] \subseteq \text{V}_G \backslash K \), then \( [x] \subseteq \text{V}_G \backslash K \subseteq E_G(K), \) so \( \forall x \in [x] \exists w \in K : E_G(x, w) \). Then \( [x] \subseteq K \) and \( [x] \subseteq E_G(K) \). Thus \( \forall x \in K \subseteq \text{V}_G \backslash K \subseteq \text{V}_G \backslash K \subseteq \text{sol}(G) \).

(2) We show the claim for limit \( \lambda \).

(2) \( K \subseteq \text{V}_G \backslash K \), then \( K^{\lambda} \subseteq \{v\} \subseteq K^{\lambda} \), if \( E_G([v]^\lambda, [u]^\lambda) \) for some \( [v]^\lambda, [u]^\lambda \subseteq K^{\lambda} \), i.e., \( v, u \subseteq K \), then for some \( n \in \lambda : E_G([v]^n, [u]^n) \), which means that \( K^{n+1} = \{x^\lambda \mid x \in K \} \) is not a kernel of \( G^n \), contrary to point (1). Hence \( K^{\lambda} \) is independent. If \( [v]^\lambda \not\subseteq \text{V}_G \backslash K \), then \( [v]^\lambda \subseteq \text{V}_G \backslash K \), so for any \( v \in [v] \), there is a \( u \in E_G(v) \cap K \). Then also \( [u]^\lambda \subseteq E_G([v]^\lambda) \cap K^{\lambda} \), hence \( \forall x \subseteq K^{\lambda} \subseteq \text{E}_G([v]^\lambda) \), and \( K^{\lambda} \subseteq \text{sol}(G^{\lambda}) \).

(2) For a kernel \( K^{\lambda} \) of \( G^{\lambda} \), let \( K = \bigcup K^{\lambda} = \{v \in \text{V}_G \mid [v]^\lambda \subseteq K^{\lambda}\} \). If \( v \not\subseteq \text{V}_G \backslash K \), then \( [v]^\lambda \not\subseteq \text{V}_G \backslash K, \) and \( [v]^\lambda \not\subseteq \text{V}_G \backslash K \), then \( [v]^\lambda \not\subseteq [v]^\lambda \subseteq K^{\lambda} \subseteq \text{E}_G([v]^\lambda) \cap K^{\lambda} \), contradicting independence of \( K^{\lambda} \). Thus \( \forall x \subseteq K^{\lambda} \subseteq \text{E}_G([v]^\lambda) \), so for some \( n < \lambda, [u]^n \subseteq \text{E}_G([v]^n) \), that is, for some \( u' \subseteq [u]^n, u' \subseteq E_G(v) \). Since \( [u]^\lambda \subseteq K^{\lambda} \), so \( [u]^n \subseteq [u]^\lambda \subseteq K, \) hence \( v \not\subseteq \text{E}_G(K) \) and \( K \subseteq \text{sol}(G) \). □
models and kernels of its graphs are in bijection. Until then, we are occupied with graphs for the whole language, which we now extend with sentential quantifiers.

§3. Sentential quantifiers. Language $\mathcal{L}^-$ is extended to language $\mathcal{L}$ with sentential quantifiers, $s$-quantifiers for short, by augmenting grammar (2.1) with three underlined productions.

\[
\begin{align*}
T_X & := X \mid \text{Const} \mid \text{Func}(T_X \ldots T_X) \\
A_X & := \Phi_1(T_X \ldots T_X) \mid C \\
F_X & := A_X \mid \Phi_X \mid F_X \land F_X \mid \forall X.F_X \mid \forall \Phi.F_X
\end{align*}
\]  

Sentence constants $C$, as new $s$-atoms, do not affect anything and will not be mentioned. Syntactically, the $s$-variables $\Phi$ are atomic formulae, but are excluded from $s$-atoms $A_X$ since they are not independently valued. $\Phi, C, C, F$ and $X$ are mutually disjoint. Concerned with truth, we restrict $\Phi$, and quantification over $\Phi$, to $\mathcal{L}$-sentences $S$, that is, formulae $F_X$ with no free variables $X$ or $\Phi$. Quantification into formulae, like $\forall x \exists \phi, \phi(x)$, is modeled by language graphs, but is not addressed. Even without it, the extension to $\mathcal{L}$ need not be trivial as $s$-quantifiers range over sentences of the full language $\mathcal{L}$ and not only of $\mathcal{L}^-$.

With $\mathcal{L}^- =$ FOL, the intended semantics is usual FOL semantics extended with substitutional $s$-quantifiers, by adding the following case to the standard definition

\[
(3.2) \quad M \models \forall \phi F[\phi] \iff \forall S \in \mathcal{S} : M \models F[S],
\]

for a FOL structure $M$. Among instances of the right side there are $F[\forall \phi F[\phi]], F[F[\forall \phi F[\phi]]]$, etc., causing troubles for inductive definitions. Graph-based semantics allows to formalize the meaning of (3.2), Definition 4.3, and to show that it coincides with quantification over boolean values in $\mathcal{L}$, Fact 4.5, but not when $\mathcal{L}$ is extended with sentential predicates in Section 5.1

**Definition 3.3.** The language graph $S_M(\mathcal{L})$, over domain $D$, is as $S_M(\mathcal{L}^-)$ in Definition 2.2, over $S_M \supset S_{M^+}$, with additional edges in point 3: (d) $\forall \phi F[\phi] \rightarrow S_M(-F[Q])$, for each $Q \in S_M$.

$S$-quantifiers yield the same branching patterns as object quantifiers, adding one child for each sentence (instead of domain element), including the very sentence causing the branching. Letting $A, B, C, \ldots$ abbreviate all $\mathcal{L}_M$-sentences, we obtain the plain counterparts of (2.3) – (2.5):

\[
\begin{align*}
\forall \phi D[\phi] & \quad \exists \phi D[\phi] \\
D[A] & \quad D[B] \quad D[\forall \phi D[\phi]] \quad \ldots \\
\forall \psi \exists \psi D[\phi, \psi] & \quad \exists \psi D[B, \psi] \quad \exists \psi D[C, \psi] \quad \ldots \\
\exists \psi D[A, \psi] & \quad \exists \psi D[B, \psi] \quad \exists \psi D[C, \psi] \quad \ldots \\
\forall \psi \neg D[A, \psi] & \quad \forall \psi \neg D[B, \psi] \quad \forall \psi \neg D[C, \psi] \quad \ldots \\
D[A, A] & \quad D[A, B] \quad \ldots \\
D[B, A] & \quad D[B, B] \quad \ldots \\
D[C, A] & \quad D[C, B] \quad \ldots
\end{align*}
\]

---

1Such predicates, like truth, motivate substitutional $s$-quantifiers but do not require substitutional term quantifiers, which would bring us beyond FOL. Keeping usual object quantifiers, $\mathcal{L}$ stays within FOL, Section 4.1.
In $\mathcal{L}$, s-variables occur only in sentential contexts, i.e., as subformulae, but not in nominal contexts, as arguments to predicates (considered in Section 5). Square brackets, $F[\phi]$, signal such an occurrence of $\phi$ as a subformula in the context $F[\phi]$, as in literals $L[\phi]$ on (3.6). The remaining literals $R_i$ have no s-variables (are closed or have instantiated object variables).

S-variables lead to new cycles, besides the atomic 2-cycles from $S_M(\mathcal{L}^-)$. Assuming that s-variable $\phi$ occurs in $F[\phi]$ and $A = \forall \phi F[\phi]$, sentences substituted for $\phi$ occur as atoms ((grand)children) under $C$ vertices of $T_M(A)$. When the substituted sentence is $A$, we obtain a cycle back to the root of $T_M(A)$. Cycles occur also, for instance, whenever an s-quantified sentence $Q$ is substituted for $\phi$, because $Q$ has, in turn, $A$ substituted for its s-variable with paths back to $A$. As an example, we draw parts of the subgraphs for $A = \forall \phi \phi, A' = \forall \phi \neg \phi$, and $E = \forall \phi \neg (\neg \phi \land \phi)$ (with dotted edges showing the general pattern of DNF-feet for each of these three sentences):

\[
\text{\includegraphics[width=0.8\textwidth]{diagram.png}}
\]

A has a double edge to every sentence (shown to $B, \neg B, A', \neg B \land A'$, and to $A$ itself) and $A'$ a single one, resulting from contracting 3-paths. A more general situation is exemplified by $E$, with each DNF-foot consisting of a $C$-vertex with single and double edge to one atom $\phi$, instantiated by distinct sentences in distinct feet, shown for $B, \neg B, \neg E \land \neg B$ and $E$ itself.

We now describe closer the structure of the resulting graph $S_M(\mathcal{L})$, central for the paper.

3.1. Sentence subgraphs. Patterns (3.4)-(3.6) form, for each $A \in S_M$, a tree $T_M(A)$, underlying sentence subgraph $S_M(A)$. For $A \in S_M$, $S_M(A) = T_M(A)$, but $S_M(A)$ is not a tree when $A \in S_M \setminus S_M$. Still, we retain the language of trees, which serves well in structuring the description, e.g., $A$ is the root of $S_M(A)$. The subgraph denoted by $S_M(\mathcal{L} \setminus \mathcal{L}^-)$, comprising all subgraphs $S_M(A)$, for $A \in S_M \setminus S_M$, and paths between them, is a strongly connected component, with (single and double) edges to the subgraph $S_M(\mathcal{L}^-)$, but no incoming edges thence.

For the most, we ignore object quantifiers. Although affecting truth of sentences, they do not affect the general structure of the language graph or the results. For instance, the two sentences

\[ A = \forall \phi \exists x ((P(x) \land \phi) \lor \neg P(x) \land \neg \phi) \]

are not equivalent, have different branching patterns, and cycles in both graphs occur in different ways (e.g., from substitutions of $A$ and $\overline{A}$ in $S_M(A)$, and of $B$ and $\overline{B}$ in $S_M(B)$). Still, the resulting DNF-feet in both graphs are identical, since substitutions for $x$ and $\phi$ do not interact, and cycles in both graphs represent the same pattern, depending only on the s-variables. We will therefore assume that branching patterns, affected by the possible presence of object quantifiers, are taken into account, and will not mention object quantifiers nor object variables, unless necessary.

1. Let $\exists$ stand for $\forall$ or $\exists$. Each root sentence $A = \exists \xi_1 \phi_1 ... \exists \xi_n \phi_n D[\phi_1 ... \phi_n] \in S_M \setminus S_M$ can be seen as a function $A : S_M \rightarrow S_M$ from sequences of sentences with length $\leq n$ to sentences, resulting from substituting the argument sentences $S_1 ... S_k$ for $\phi_1 ... \phi_k$, $k \leq n$, while removing the corresponding quantifiers $\exists \xi_1 ... \exists \xi_k$. (Object quantifiers are treated analogously.) Each path from the root $A$ is thus determined by a substitution sequence $\pi$ of sentences (and objects) substituted at each step, which yields sentence $A(\pi)$ occurring at the $|\pi|$-th position on the resulting path.
Sometimes, the following notation may be useful. To emphasize that $\pi$ is a sequence of sentences which are substituted into $A$, we may write it as $\pi /$, denoting by $/\pi$ the resulting sequence of sentences which are actual vertices in the graph on the path identified by $\pi$ and arising from successive substitutions of sentences from it. At each vertex, $S/\pi$ denotes the sentence substituted (for the outermost quantified $s$-variable) into the parent sentence and $/\pi$ the resulting sentence (after removal of the outermost $s$-quantifier). Each vertex, after a path $\pi /$, is thus a pair $S/\pi$, where $R = A(\pi S)$, which might be also written $R = A((\pi/)S/)$. We avoid this slashed notation primarily by identifying $\pi = \pi /$, but at some places it will prove useful. $D[\pi] = A(\pi)$ denotes than the sentence at the root of the DNF-foot (3.6) resulting from substituting sentences (and possibly, objects) specified by $\pi$ for all quantified variables of $A$.

Sentence $A$ that is not in PDNF is processed in basically the same way, only adding an edge for $\neg$ and branching with double edges for $\land$, whenever these are encountered. Each substitution for an $s$-variable introduces an atom into the resulting tree $T_M(A)$.

2. The resulting subgraph $T_M(A)$, for $A \in S_M \setminus S^-_M$, has atoms which are:
   (i). sentences $S^-_M$, and
   (ii). sentences $S_M \setminus S^-_M$, including $A$.

Atoms (i), that are complex sentences, are decomposed further into $s$-atoms $A^-_M \subset S^-_M$ in $S_M(\mathcal{L}^-)$. As there are no paths from subgraph $S_M(\mathcal{L}^-)$ to $S_M(\mathcal{L} \setminus \mathcal{L}^-)$, all $\mathcal{L}^-_M$-sentences play identical role for the latter, and we do not distinguish between atomic and non-atomic $\mathcal{L}^-_M$-sentences, unless this becomes relevant.

Atoms (i) are among the leaves of $T_M(A)$ and sinks of $S_M(A) \supset T_M(A)$. Atoms (ii), depending on the substituted sentences, yield cycles either directly to vertices of $T_M(A)$ or indirectly via roots of other sentence subgraphs. The subgraph $S_M(A)$ is formed by the tree $T_M(A)$, together with the double and single edges from its atoms $S, \bar{S}$ to vertex $S$ when it occurs in $T_M(A)$ (up to renaming of bound variables). For instance, starting with the root $A = \forall \phi \forall \psi(\psi \lor \phi)$, and substitution sequence beginning with $\pi_1 = \forall \theta \theta$, yields $A(\pi_1) = B = \forall \psi(\psi \lor \forall \theta \theta)$.

At next step, each branch substitutes a sentence for $\psi$ and terminates the process, having replaced all quantified variables from $A$ and reaching its DNF-foot. For instance, the substitution sequence:

1. $(\forall \theta \theta, A)$ yields DNF-foot $D_1 = \forall \forall \forall \psi(\psi \lor \phi) \lor \forall \theta \theta$,
2. $(\forall \theta \theta, \neg A)$ yields DNF-foot $D_2 = \neg \forall \forall \forall \psi(\psi \lor \phi) \lor \forall \theta \theta$,
3. $(\forall \theta \theta, B)$ yields DNF-foot $D_3 = \forall \psi(\psi \lor \forall \theta \theta) \lor \forall \theta \theta$,
4. $(\forall \theta \theta, \neg B)$ yields DNF-foot $D_4 = \neg \forall \psi(\psi \lor \forall \theta \theta) \lor \forall \theta \theta$, etc..

Atoms of $S_M(A)$, like the two disjuncts in each case above, are connected to the roots of their subgraphs by extra edges. Besides double edges from atoms $\forall \theta \theta$ to the root of $S_M(\forall \theta \theta)$ (dotted edges on Figure 1, which can be contracted by Fact 2.8), there is a double (single) edge from the other atom under $D_1$ ($D_2$) back to $A$, and from under $D_3$ ($D_4$) to $B$. These edges, marked with the double lines, belong to $S_M(A)$ but not to $T_M(A)$, which is marked with the single lines. In the same way, single/double edges connect atoms of one branch to their occurrences on others.

3. This introduces a distinction between two kinds of atoms (ii) in $S_M(A)$. An internal atom is a sentence $S$ which occurs in $T_M(A)$ as both its atom (leaf) and internal vertex (e.g., $A, B$ on Figure 1). The atom $S$ has then a (single or double) edge back to the internal vertex $S$ in $T_M(A) \subset S_M(A)$. External atoms, $ext(S_M(A))$, are sentences in $S_M \setminus S^-_M$ which occur in $S_M(A)$ only as atoms but not as internal vertices, like occurrences of $\forall \theta \theta$ among the atoms of $S_M(A)$. Together with atoms $S^-_M$ they form sinks of $S_M(A)$, having no children in this subgraph.

4. Although the roots $S_M$ of sentence graphs form only a proper subset of vertices of $S_M(\mathcal{L})$, any $\alpha \in 2^{S_M}$ determines a unique assignment $\alpha^*$ to all vertices of $S_M(\mathcal{L})$. This follows because every vertex of $S_M(\mathcal{L})$ represents (is equivalent to) an $\mathcal{L}$-sentence but also, in graph terms, because $S_M$ forms a feedback vertex set, i.e., its removal leaves acyclic graph which, moreover, has no rays (infinite simple outgoing paths). Consequently, a valuation $\alpha \in 2^{S_M}$ induces values to all remaining vertices. “Wrong” $\alpha$ may cause conflict $\alpha(S) \neq \alpha^*(S)$ for some $S \in S_M$, while it determines a
solution of $S_M(\mathcal{L})$ iff it is a fixed point of such an inducing, i.e., iff $\alpha^+|_{S_M} = \alpha$. Therefore, solutions of $S_M(\mathcal{L})$ are identified with such valuations of roots $S_M$ of sentence subgraphs.

3.2. The language of QBS. A special case is obtained when $\mathcal{L}$ and the set of sentence constants $\mathbf{C}$ are empty. The only atomic expressions are $s$-variables $\Phi$, giving the language of quantified boolean formulae, QBF. We consider only quantified boolean sentences, $\mathbf{QBS} \subseteq \mathbf{QBF}$.

According to Definition 3.3, there is then only one language graph, denoted $S(\mathbf{QBS}) = S_0(\emptyset)$, as the set of sentences is the same for each domain $M$, including $M = \emptyset$. $S$-variables can be substituted only by such sentences, so $C$-level vertices of $S_0(A)$, for each $A \in \mathbf{QBS}$, have edges only to the roots of this and other sentence subgraphs $S_0(B)$, $B \in \mathbf{QBS}$. Graph $S(\mathbf{QBS})$ is thus one strongly connected component, which is an induced subgraph of every $S_M(\mathcal{L})$.

The difference from the standard QBS is that we interpret $s$-quantifiers substitutionally over sentences, and not referentially over $\{1,0\}$ (or over arbitrary boolean algebra). Moreover, our semantics evaluates sentences by finding a kernel of the graph $S(\mathbf{QBS})$. As one might wish, the two semantics coincide, which follows from Theorem 4.2. It guarantees a unique solution for $S(\mathbf{QBS})$, containing exactly the true QBS, Fact 4.5.

3.3. Logical and graph equivalences. We introduce some notions of equivalence in terms of graphs. Two $\mathcal{L}_M$-sentences are equivalent (in $S_M(\mathcal{L})$) if they belong to the same kernels of $S_M(\mathcal{L})$, and $\mathcal{L}$-sentences are (logically) equivalent if they are so in every language graph:

$$\begin{align*}
\text{for a graph } G \text{ and } A, B \in V_G: & \quad A \xrightarrow{G} B \iff \forall K \in \text{sol}(G): A \in K \iff B \in K \\
\quad (3.7) & \quad \text{for } A, B \in S_M: \quad A \xrightarrow{S_M(\mathcal{L})} B \iff A \xrightarrow{S_M(\mathcal{L})} B \\
\quad & \quad \text{for } A, B \in S: \quad A \xrightarrow{S} B \iff \forall M: A \xrightarrow{S} B.
\end{align*}$$

A more specific equivalence will be used, corresponding to prenex operations. With the usual definition of $\exists \phi = \neg \forall \neg \phi$, and avoidance of name clashes for variables, each sentence can be written in prenex normal form using standard prenex transformations. Assuming propositional equivalences allowing to write every quantifier-free formula in DNF, every $\mathcal{L}_M$-sentence $A$ has a PDNF, i.e., a sentence in PDNF obtained from $A$ by prenex operations and propositional equivalences. $\mathcal{L}_M$-sentences $A, B$ are PDNF-equivalent, denoted by $A \xleftrightarrow{\mathcal{P}D} B$, if they have (also) identical PDNFs. As can be expected, PDNF-equivalence implies $\mathcal{L}$-equivalence, but to show this, we will use a more structural notion of equivalence in a graph $G$.

By $E_G^*$ we denote the reflexive and transitive closure of $E_G$ and by $E_G^*(S)$, for $S \in V_G$, the subgraph of $G$ induced by all vertices reachable from $S$. A common cut of $A, B \in V_G$ is a set of vertices $C \subseteq E_G^*(A) \cap E_G^*(B)$, such that every path leaving $A$ and prolonged sufficiently far crosses $C$ and so does every path leaving $B$. (C may intersect $A$ and $B$ and contain vertices on various cycles intersecting $A$ and $B$.) We say that $A$ and $B$ are cut equivalent, $A \xleftrightarrow{c} B$, if there is a common cut $C$ such that for every correct (not falsifying (2.6)) valuation of $C$, every correct

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example_diagram.png}
\caption{Example Diagram}
\end{figure}
extension to \( \{A, B\} \) forces identical value of \( A \) and \( B \). Obviously, if \( A \equiv B \) in a graph \( G \), then also \( A \equiv B \), as each \( K \) in \( \text{sol}(G) \) determines a correct valuation of every common cut of \( A \) and \( B \).

**FACT 3.8.** For \( A, B \in S_M \) in \( S_M(\mathcal{L}) \), if \( A \equiv B \) then \( A \equiv B \), hence \( A \equiv B \).

**Proof.** Let \( S = S_M(\mathcal{L}) \). If \( \text{sol}(S) = \emptyset \) then all sentences are equivalent in \( S \), so assume \( \text{sol}(S) \neq \emptyset \) and consider only \( s \)-quantifiers, as object quantifiers can be treated in the same way.

The claim holds trivially for \( B \) obtained by renaming bound \( s \)-variables (avoiding name clashes) in \( A \), as the two have the same subgraph. This is also the case for the subgraphs of \( A = \neg \exists \phi D[\phi] \) and \( B = \exists \phi \neg D[\phi] \).

We show \( A = (\forall \phi D[\phi]) \land C \equiv \forall \phi (D[\phi] \land C) = B \), with no free occurrences of \( \phi \) in \( C \). On the schematic subgraph, \( X_i, X_j \ldots \) stand for all \( S_M \) and common cut is marked by the waved line.

Inspecting the graph, we see that, for any kernel \( K \):

\[
B \in K \iff ((D[X_i] \land C) \in K \text{ for all } X_i) \iff (C \in K \land (D[X_i] \in K \text{ for all } X_i)) \iff A \in K.
\]

For the last case, \( A = \neg \exists \phi D[\phi] \equiv \forall \phi \neg D[\phi] = B \), the schematic subgraph is as follows:

Obviously, for any kernel \( K : A \in K \iff \circ \in K \iff (D[X_i] \not\in K \text{ for all } X_i) \iff B \in K \). \( \square \)

Thus, every sentence in \( \mathcal{L} \) has an \( \equiv \)-equivalent PDNF sentence. A useful consequence is that, considering now solvability of any \( S_M(\mathcal{L}) \), we can limit attention to \( \mathcal{L}_M \)-sentences in PDNF.

**§4. Solvability of \( S_M(\mathcal{L}) \).** Given a language \( \mathcal{L} \) and domain \( M \), we show unique solvability of \( S_M(\mathcal{L}) \) for each solution of \( S_M(\mathcal{L}^-) \). Thus, when all \( \mathcal{L}_M \)-sentences obtain correct boolean values, i.e., \( \mathcal{L}_M \) is free for paradoxes, then so is its extension to \( \mathcal{L} \). For \( \mathcal{L}_M = \text{FOL} \), and typically, valuation of \( \mathcal{L}_M \)-atoms suffices, but since the results in this section are independent of the structure of \( \mathcal{L}^- \), we mention only valuations of all \( S_M \). When \( \mathcal{L} \) has PDNF, it suffices to show solvability of the subgraph induced by all \( \mathcal{L} \)-sentences in PDNF, and we do this.

Main Theorem 4.2 is a corollary of Lemma 4.1, which shows that just like each valuation of \( s \)-atoms \( A_M \) determines typically a unique value of each sentence in \( S_M \), so each valuation of \( S_M \) determines a unique value of each \( s \)-quantified sentence in \( S_M \). The proof rests on the fact that every odd cycle in a sentence graph \( S_M(A) \) has a “corresponding” even one, such that any valuation of external atoms or \( S_M \), breaking the even cycle, breaks also the odd one. A simple form of this correspondence can be seen on the following examples for \( A_\forall = \forall \phi \phi \) and \( A_\exists = \exists \phi \phi \):

In the graph \( S(\forall \phi \phi) \), any \( B_i = 0 \) forces \( \forall \phi \phi = 0 \), but this is forced even if all \( B_i = 1 \), because the only possibility for the pair of cycles through the atoms \( A_\forall \) and \( A_\forall \) is \( A_\forall = 0 \) making \( \circ = 1 \) and \( \forall \phi \phi = A_\forall = 0 \). Similarly, the two cycles in \( S(\exists \phi \phi) \) force \( \bullet = 0 \), so that \( A_\exists = \exists \phi \phi = 1 \), even
if all $B_i = 0$. Although typical witnesses for $\forall \phi, \phi = 0$ and $\exists \phi, \phi = 1$ come from $\mathcal{L}^-$, these two sentences witness to their own falsity and truth independently of $\mathcal{L}^-$.

In each case, the sentence’s value is independent of the values of external atoms. This holds generally. Solution of $\pi(M(A))$—denoting, for $A \in S_M \setminus S_M^-$, vertices of $S_M$ without those in its DNF-feet—depends on the valuation of $S_M^-$, but not of $\text{ext}(S_M(A))$, as the second part of lemma below states. Valuation of $\text{ext}(S_M(A))$ affects, of course, values in DNF-feet, where they occur (like for $B_i$ above).

**Lemma 4.1.** $\forall A \in S_M \setminus S_M^- \ \forall \rho \in 2^{S_M \setminus \text{ext}(S_M(A))} \ \exists! \rho_A \in \text{sol}(S_M(A)) : \rho_A|_{S_M \setminus \text{ext}(S_M(A))} = \rho$.

For each pair $\rho, \sigma \in 2^{S_M \setminus \text{ext}(S_M(A))}$, if $\rho|_{S_M^-} = \sigma|_{S_M^-}$ then $\rho_A = \sigma_A|_{S_M(A)} = \sigma_A|_{S_M(A)} = \sigma_A$.

**Proof.** By Fact 3.8, we can limit attention to sentences in PDNF. For $A$ with the number of (sentential) quantifiers $q(A) = n + 1 \geq 1$, and for $\pi \in S_M^-$, roots of all feet, $A(\pi S) = D[\pi S]$, $S$ in $S_M$, are grandchildren of vertex $A(\pi) = \mathcal{F} \phi D[\pi \phi]$. (On the drawing, $\mathcal{F} = \exists$ and all feet have the common parent $\bullet$; when $\mathcal{F} = \forall$, they have distinct parents, all children of $A(\pi_i)$.) Each foot $A(\pi S)$ represents an application of the same boolean function $d^\pi(\phi) = D[\pi \phi]$, evaluating $D[\pi \phi]$ given valuation of its parameters $\pi, \phi$ and, possibly, some atoms $L_A \subset S_M^-$ occurring in the original matrix $D[...]$. For any $\rho \in 2^{S_M^+}$, $L_A$ obtain fixed values so, considering $d^\pi$, we assume that effects of $\rho(L_A)$ have been taken into account.

![Diagram](attachment:image.png)

- The internal vertices of $\pi$ are sentences occurring on the path after substitutions, $\text{int}(\pi) = /\pi$, and external ones are those which do not, $\text{ext}(\pi) = S_M \setminus \text{int}(\pi)$. The atoms, ‘sinks’ of the feet, have single or double edges to vertices from $\pi$, which may be among $\text{int}(\pi)$, including $\pi_0 = A$ and $\mathcal{F} \phi D[\pi \phi]$ (which can be substituted for $\phi$ in $D[\pi \phi]$). As branches from $\bullet$ instantiate $\phi$ with every sentence $S \in S_M$, all sentences from $\text{int}(\pi)$ do occur in some feet.

- Depending on whether $\mathcal{F}$ is $\forall$ or $\exists$, we adjust the general dependence of the value at vertex $\mathcal{F} \phi D[\pi \phi]$ on valuation of its grandchildren as either

  $$(*) \ \exists! \phi D[\pi \phi] = \bigvee_{S \in S_M} d^\pi(S)$$

or $\forall \phi D[\pi \phi] = \bigwedge_{S \in S_M} d^\pi(S)$.

We consider first the case when $|\pi| = q(A) - 1$, i.e., $A(\pi) = \mathcal{F} \phi D[\pi \phi]$ is the grandparent of the completely substituted (roots of) feet ($D[\pi A], D[\pi B]$, etc., on the drawing).

Every valuation of sentences from $\pi$, abbreviated as $\alpha \in 2^\mathcal{F}$, specializes function $d^\pi(\phi)$ to a unary boolean function $d^{A(\pi)}(\phi) = D[\alpha(\pi) \phi]$, and to

$$(**) \ \exists! \phi D[\alpha(\pi) \phi] = \bigvee_{S \in S_M} d^{A(\pi)}(S)$$

or $\forall \phi D[\alpha(\pi) \phi] = \bigwedge_{S \in S_M} d^{A(\pi)}(S)$.

- As a boolean function of one variable, $d^{A(\pi)}(\phi)$ is either constant or not. If it is constant, i.e., $d^{A(\pi)}(\phi) = d^{A(\pi)}(-\phi)$, then $\mathcal{F} \phi D[\alpha(\pi) \phi]$ obtains the same value in each case of $A(\pi)$. Otherwise, $d^{A(\pi)}(-\phi) = -d^{A(\pi)}(\phi)$ and, since for each $S \in S_M$ both $d^{A(\pi)}(S)$ and $d^{A(\pi)}(-S)$ enter evaluation of $A(\pi)$, this yields constant $0$ at their least common predecessor $\bullet$ when $\mathcal{F} = \exists$ and $A(\pi)$, when $\mathcal{F} = \forall$. In this way, for every $\alpha \in 2^\mathcal{F}$, $A(\pi)$ obtains a unique value $\alpha(A(\pi))$, induced from all $D[\alpha(\pi) S]$ by $A(\pi)$, but determined already by $d^{A(\pi)}(\phi)$, independently from

(i) valuation $\alpha(A(\pi))$, i.e., if $\alpha_0, \alpha_1 \in 2^\mathcal{F}$ differ only at $A(\pi)$, then $\alpha_0^\phi(A(\pi)) = \alpha_1^\phi(A(\pi))$, and

(ii) independently from valuation of $\text{ext}(\pi)$, since each external vertex $S$ enters both evaluation of $d^{A(\pi)}(S)$ and of $d^{A(\pi)}(-S)$, with jointly constant contribution to $A(\pi)$ as just explained.
Point (i) means that cycles from the feet to \(A(\pi)\) admit a unique solution \(\rho_{A(\pi),\alpha}\) to the subgraph \(S_M(A(\pi))\) of \(S_M(A)\), given any \(\rho \in 2^{S_M^{\pi} \cup \text{ext}(\pi)}\) and \(\alpha \in 2^{\pi^*}\), where \(\pi^* = \pi\) without its last element \(A(\pi)\). By point (ii), \(\rho_{\text{ext}(\pi)}\) is inessential, so if \(\rho|_{S_M}\) = \(\sigma|_{S_M}\) then \(\rho_{A(\pi),\alpha}(A) = \sigma_{A(\pi),\alpha}(A)\).

iv. This yields basis for the claim that for each \(\rho \in 2^{S_M^{\pi}}\), each \(A\) with \(q(A) \geq 1\), and each path \(\pi\) from the root \(A\) with \(|\pi| < q(A)\), vertex \(V = A(\pi)\) (above the roots of the feet) obtains for each \(\alpha \in 2^{\pi^*}\) a unique value \(\alpha^*(V)\), which depends at most on valuation of vertices on \(\pi^*\) (above \(V\)), but neither on the value (i) of \(\alpha(V)\) nor (ii) of \(\alpha(X)\), for any \(X \in \text{ext}(\pi)\).

The claim is shown by induction on \(h - l\), where \(h \geq 1\) is the distance of the root \(A\) from the roots of the feet and \(l\), with \(h > l \geq 0\), is the distance of \(V\) from the root \(A\). Point iii gives the basis for \(h - l = 1\), i.e., \(l = h - 1\).

v. The argument from iii works also in the induction step. For \(0 \leq |\pi| = l < h - 1\), we have the following counterpart of the drawing from iii, with \(A(\pi) = \exists \phi \exists \psi D[\pi \phi \psi]\), where \(\exists \psi\) is the sequence of remaining quantifiers, and \(\psi_1, \psi_2\) at the bottom signal various substitutions for \(\psi\).

Given \(\alpha \in 2^{\pi^*}\), IH applied to the lowest triangles, i.e., subgraphs \(S_M(A(\pi S))\) with roots \(A(\pi S)\) for \(S \in S_M\), gives to each \(A(\pi S)\) a unique value, independent of valuation of \(\text{ext}(\pi S)\). Consequently \(A(\pi \phi)\) is a function of only \(\pi\) and \(\phi\), so that for any \(\alpha \in 2^{\pi^*}\), it represents a function \(d^{\pi^*}(\phi)\) of \(\phi\). The same argument and cases for \(d^{\pi^*}(\phi)\) as in iii show that the value \(\alpha^*(A(\pi))\), induced to the common grandparent of all \(A(\pi S)\) under valuation \(\alpha \in 2^{\pi^*}\), is equal whether \(\alpha(A(\pi)) = 1\) or \(\alpha(A(\pi)) = 0\), giving point (i) of induction. As for each \(A(\pi S)\) its value under \(\alpha^*\) is independent from valuation of \(\text{ext}(\pi S)\) by IH, the induced value \(\alpha^*(A(\pi))\) is independent from \(\bigcap_{S \in S_M} \text{ext}(\pi S) = \text{ext}(\pi)\), giving point (ii) of induction. Consequently, \(\alpha^*(A(\pi))\) is unique and independent of valuations of \(\text{ext}(\pi)\) and of \(A(\pi)\), which establishes the induction step.

vi. Thus, the value of the root \(A\) is determined, for each \(\rho \in 2^{S_M^{\pi}}\), independently from valuation of \(\text{ext}(S_M(A))\). Starting now from \(A\) and using claim iv downwards, the value of \(A(S)\), for each \(S \in S_M\), is determined by \(\rho\) and value of \(A\) (independently from valuation of \(\text{ext}(S_M(A))\)). Since \(A\) is determined by \(\rho\), so is the value of \(A(S)\). Proceeding inductively down the tree \(T_M(A)\), valuation \(\rho_A^*\) of \(T_M(A)^*\) is seen determined by \(\rho\), independently from valuation of \(\text{ext}(S_M(A))\).

The latter determines values in all feet of \(S_M(A)\), yielding a unique solution \(\rho_A\) of \(S_M(A)\), with \(\rho_A < \rho_A^*\) and \(\rho|_{S_M^{\pi} \cup \text{ext}(S_M(A))} = \rho\). □

**Theorem 4.2.** For each \(S_M(\mathcal{L})\) and \(\rho \in 2^{S_M^{\pi}}\), there is a unique \(\hat{\rho} \in \text{sol}(S_M(\mathcal{L}))\) with \(\hat{\rho}|_{S_M^{\pi}} = \rho\).

**Proof.** Graph \(S_M(\mathcal{L})\) consists of \(S_M(\mathcal{L} \setminus \mathcal{L}^-) = \bigcup_{A \in S_M \setminus S_M^-} S_M(A)\) and \(S_M(\mathcal{L}^-) = \bigcup_{A \in S_M} T_M(B)\), with no edges from the latter to the former, and (single or double) edges from each \(V \in \text{ext}(S_M(A))\), \(A \in S_M \setminus S_M^-\), to the root of \(S_M(V)\) (of \(T_M(V)\) when \(V \in S_M^-\)). By Lemma 4.1, valuation \(\rho\) of \(S_M = V_{S_M(\mathcal{L}^-)}\), determines a solution \(\rho_A^*\) of each \(S_M(A)\), compatible with every valuation of \(\text{ext}(S_M(A))\). Hence, these can be combined into \(\rho \cup \bigcup_{A \in S_M \setminus S_M^-} \rho_A^*\) forcing value \(\rho \upharpoonright \mathcal{L}\) at each \(V \in \text{ext}(S_M(A))\), and thus determining solutions of all DNF-feet. Each \(S_M(A)\) obtains thus a
solution \( \rho_A \supseteq \rho_A \), yielding a unique \( \hat{\rho} = (\rho \cup \bigcup_{A \in \mathcal{S}_M \setminus \mathcal{S}_M^-} \rho_A) \in \text{sol}(\mathcal{S}_M(\mathcal{L})) \) which extends \( \rho \). \( \square \)

This central result implies that any language \( \mathcal{L}^- \), possessing PDNF, that is free for paradoxes remains so when extended with s-quantifiers to \( \mathcal{L} \).

Truth of sentences \( \mathcal{S}_M \setminus \mathcal{S}_M^- \) does not have any generally accepted definition, which was merely suggested by (3.2). By Theorem 4.2, their valuation \( \hat{\rho} \) is determined by \( \rho \in 2^{\mathcal{S}_M} \), just as such \( \rho \) valuating sentences \( \mathcal{S}_M^- \) is typically determined by a valuation of atoms \( A_M \). Existence and uniqueness of \( \hat{\rho} \) ensure that (3.2) is well-defined.

**DEFINITION 4.3.** An \( \mathcal{L}_M \)-sentence \( A \) is true in an \( \mathcal{L}^- \) domain \( M \) under valuation \( \rho \) of \( \mathcal{S}_M^- \), \( (M, \rho) \models A \), iff \( \hat{\rho}(A) = 1 \) for the unique solution \( \hat{\rho} \in \text{sol}(\mathcal{S}_M(\mathcal{L})) \) with \( \hat{\rho}|_{\mathcal{S}_M^-} = \rho \).

Consequently, the class of \( \mathcal{L}^- \)-structures \( \text{Mod}(\Gamma) = \{(M, \rho) \mid \forall A \in \Gamma : (M, \rho) \models A \} = \bigcap_{A \in \Gamma} \text{Mod}(A) \) is well-defined for every \( \Gamma \subseteq \mathcal{S} \). Definition 4.3 extends only the relation of \( \mathcal{L}^- \)-sentences to \( \mathcal{L}^- \)-sentences by, so to speak, converting the information from the graph \( \mathcal{S}_M(\mathcal{L}) \), to the relation between \( M \) and \( \mathcal{L} \)-sentences. The bijection between \( \text{FOL} \) structures and \( \mathcal{L}^- \)-graphs with kernels, mapping \( (M, \rho) \) to \( (\mathcal{S}_M(\mathcal{L}^-), K_{\rho}) \) at (2.7), extends to \( \mathcal{L}^- \)-graphs by mapping \( (M, \rho) \) to \( (\mathcal{S}_M(\mathcal{L}), \hat{\rho}) \).

**4.1. Expressive power of \( \mathcal{L} \).** S-quantification, interacting with \( \mathcal{L}^- \), does not yield merely QBS-tautologies, e.g., if \( \Gamma \models \forall x(\exists x \phi \rightarrow \neg Bx \land \neg Cx) \), then \( \Gamma \models \forall \phi, \psi \forall x(\exists x \phi 
rightarrow Bx \lor \psi) \). This, however, is still close to trivial, suggesting the question about possible contribution of sentential quantification. Theorem 4.2 implies that, as far as expressive power is concerned, it is nil. The theorem trivializes this quantification in \( \mathcal{L} \), allowing to view it as a complex form of quantification over boolean values. In models of \( A = \forall \phi F[\phi] \), \( F \) is true for all sentences \( \phi \), including \( A \) itself. Guaranteeing a well-defined boolean value for each sentence (in each structure), the theorem makes this “including itself” harmless, reducing the claim to the elementary boolean semantics. To verify \( A \) it suffices to verify \( F[\phi] \) for \( \phi = 1 \) and \( \phi = 0 \). This follows provided that every sentential context \( F[\phi] \) (having only \( \phi \) free), is a congruence preserving equivalence of sentences, i.e., such that for each pair of sentences \( A, B \),

\[
A \Leftrightarrow B \text{ implies } F[A] \Leftrightarrow F[B].
\]

(4.4)

Given an internal equivalence \( A \leftrightarrow B \Leftrightarrow (A \wedge B) \lor (\neg A \land \neg B) \), it suffices that for every structure \( M \) (abbreviating \( (M, \rho) \)), if \( M \models A \leftrightarrow B \) then \( M \models F[A] \leftrightarrow F[B] \). Let \( \top / \bot \) stand for an arbitrary tautology/contradiction in \( \mathcal{L} \).

**FACT 4.5.** For every \( \mathcal{L} \)-formula \( F[\phi] \) with only \( \phi \) free and for every \( \mathcal{L}^- \)-structure \( M \):

\[
M \models \forall \phi F[\phi] \text{ iff } M \models F[\top] \land F[\bot], \text{ and } M \models \exists \phi F[\phi] \text{ iff } M \models F[\top] \lor F[\bot].
\]

**PROOF.** If \( M \models \forall \phi F[\phi] \) then, in particular, \( M \models F[\top] \) and \( M \models F[\bot] \), so \( M \models F[\top] \land F[\bot] \). Conversely, assuming \( M \models F[\top] \land F[\bot] \), let \( S \) be an arbitrary \( \mathcal{L} \)-sentence. If \( M \models S \) then also \( M \models S \leftrightarrow \top \), hence \( M \models F[S] \) by (4.4), since \( M \models F[\top] \). If \( M \models S \) then also \( M \models S \leftrightarrow \bot \), hence \( M \models F[S] \), since \( M \models F[\bot] \). In either case \( M \models F[S] \), and since \( S \) was arbitrary, \( M \models \forall \phi F[\phi] \).

If \( M \models \exists \phi F[\phi] \), let \( S \) be a sentence for which \( M \models F[S] \). Either \( M \models S \lor M \not\models S \), i.e, \( M \models \neg S \). In the first case \( M \models S \leftrightarrow \top \) and in the latter \( M \models S \leftrightarrow \bot \). Thus either \( M \models F[\top] \) or \( M \models F[\bot] \), hence \( M \models F[\top] \lor F[\bot] \). Conversely, if \( M \models F[\top] \lor F[\bot] \) then either \( M \models F[\top] \) or \( M \models F[\bot] \). In either case \( M \models \exists \phi F[\phi] \). \( \square \)

In particular, the unique solution of \( S(QBS) \) contains exactly true QBS, whose semantics is given by the right sides of the equivalences in Fact 4.5. This opens also the possibility of inductive arguments on the number of \( s \)-quantifiers, in spite of the underlying circularity.

By Theorem 4.2, values of \( \mathcal{L}^- \)-sentences determine values of all \( \mathcal{L} \)-sentences. Consequently, if structures \( M, N \) are elementarily equivalent in \( \mathcal{L}^- \), \( M \equiv N \), they are so also in \( \mathcal{L} \), \( M \equiv N \).

**FACT 4.6.** For any \( \mathcal{L}^- \)-structures \( M \) and \( N \), \( M \equiv N \) iff \( M \equiv N \).
PROOF. The non-obvious implication to the right follows by induction on the number of s-quantifiers. Let \( M \models N \) denote that \( M \) and \( N \) model the same \( \mathcal{L} \)-sentences with up to \( k \) s-quantifiers, so that \( M \models \mathcal{L}^k \) corresponds to \( M \models \mathcal{L}_0 \), giving the induction basis. Consider first PDNF sentence \( A = \forall \phi \exists \psi[D(\phi, \psi)] \), where \( |\psi| = k \geq 0 \). Suppose that

\( (m) \) \( M \models A \), i.e., for every \( F \in S \) : \( M \models \exists \psi D[F, \psi] \), while

\( (n) \) \( N \not\models A \), i.e., for some \( F_0 \in S \) : \( N \not\models \exists \psi D[F_0, \psi] \).

\( F_0 \) has some \( s \)-quants, as otherwise \((m), (n)\) contradict \( IH \), \( M \models \mathcal{L}^k \). Taking any \( \mathcal{L}^k \)-sentence \( P_0 \) such that \( N \notmodels P_0 \iff N \notmodels F_0 \), yields \( N \not\models \exists \psi D[P_0, \psi] \) by (4.4). This last sentence has \( k \) sentential quantifiers so, by \( IH \), \( M \not\models \exists \psi D[P_0, \psi] \), which contradicts \((m)\). An analogical argument shows the induction step for \( A = \exists \phi \exists \psi D[\phi, \psi] \). \( \square \)

For any theory in \( \mathcal{L} \), Fact 4.5 makes it straightforward to construct a theory in \( \mathcal{L}^- \) with the same model class. For any \( \mathcal{L}^- \)-sentence \( A \) in PDNF, an \( \mathcal{L}^- \)-sentence \( A^- \), with \( Mod(A) = Mod(A^-) \), is obtained replacing \( \forall \phi F[\phi] \) by \( F[\top] \land F[\bot] \) and \( \exists \phi F[\phi] \) by \( F[\top] \lor F[\bot] \). E.g., starting with \( A = \forall \phi \exists \psi(C \land \phi) \lor (D \land \psi) \lor (\phi \land \psi) \), with \( C, D \in S^- \), one application of Fact 4.5 yields

\[ \exists \psi((C \lor \top) \lor (D \land \psi) \lor (\top \lor \psi)) \lor \exists \psi((C \lor \bot) \lor (D \land \psi) \lor (\bot \land \psi)) \]

which can be simplified to

\[ \exists \psi(C \lor (D \land \psi) \lor \psi) \land \exists \psi(D \land \psi) \iff \exists \psi(D \land \psi) \]

Fact 4.5 applied to the last sentence yields the first sentence below

\[ (D \land \top) \lor (D \land \bot) \iff D \]

so \( Mod(A) = Mod(D) \). Proceeding thus by induction on the number of \( s \)-quantifiers (in PDNF \( \mathcal{L}^- \)-sentences), Fact 4.5 yields \( \forall A \in S \exists A^- \in S^- : Mod(A) = Mod(A^-) \), establishing

**THEOREM 4.7.** For every \( \Gamma \subseteq \mathcal{L} \) there is a \( \Gamma^- \subseteq \mathcal{L}^- \) with \( Mod(\Gamma) = Mod(\Gamma^-) \).

In particular, for \( \mathcal{L}^- = \text{FOL} \), quantification over sentences, extending FOL apparently as far as possibility of self-reference, reduces to boolean quantification and does not allow to define model classes undefinable in FOL. We have, indeed, worked hard to achieve nothing.

Still, with respect to truth, changing nothing seems a virtue rather than a disadvantage, ensuring also that no revenge paradoxes arise. Besides, \( \mathcal{L} \) is only an intermediate station. The extension to \( \mathcal{L} \) allows turning axiomatizations with schematic sentences into finite axiomatizations, e.g., schema \( T \phi \leftrightarrow \phi \) becomes sentence \( \forall \phi(T \phi \leftrightarrow \phi) \). This involves sentential predicate \( T \). Such predicates can break equivalences from Fact 4.5, making \( s \)-quantification distinct from boolean quantification.

**§5. Sentential predicates.** Language \( \mathcal{L} \) from (3.1) is extended to \( \mathcal{L}^+ \) with predicates \( P_2 \):

\[
\begin{align*}
T_X & := V | \text{Const} | \text{Func}(T_X \ldots T_X) \\
A_X^+ & := P_1(T_X \ldots T_X) | C | P_2(F_X^+ \ldots F_X^+, T_X \ldots T_X) \\
F_X^+ & := A_X | \neg F_X^+ | F_X^+ \land F_X^+ | \forall V F_X^+ | \Phi | \forall \Phi F_X^+
\end{align*}
\]

As declared earlier, predicates \( P_2 \) are not applied to names but to sentences themselves, being therefore what is typically called “operators”. However, they need not respect logical equivalence, admitting e.g. \( P(A) \not\leftrightarrow P(\neg \neg A) \), which makes a big difference from operators like conjunction or negation. The grammar restricts their application to formulae, but extensions to arbitrary pieces of syntax can easily be envisioned. We restrict them even further defining their semantics only in sentential contexts, primarily, when applied to sentences., so they appear only as sentential predicates, \( s \)-predicates. Terms can be their additional arguments, but we continue without focusing on terms. For \( P, R \in P_2 \), \( A \in P_1 \), \( x, t \in T \) and \( \phi \in \Phi \), the new \( s \)-atoms may be \( P(\phi) \), \( P(P(A(x))) \), \( R(A(t), x) \), giving a distinction between

(a) object-level \( s \)-atoms, \( A : A(t_i) \) with \( t_i \in T \) and \( A \in P_1 \) (or \( S \in C \)), and

(b) meta-level \( s \)-atoms, \( A^+ \setminus A : P(\phi_i, t_j) \) with \( \phi_i \in F^+ \), \( P \in P_2 \), possibly \( t_j \in T \).

Definition 3.3 of language graph, denoted now by \( S_M(\mathcal{L}^+) \), remains unchanged, with applications of \( s \)-predicates treated as new atoms (b), i.e., with 2-cycles to their duality, as in point 2 of Definition 2.2. (Yet another reason for viewing \( P_2 \) as predicates.) For \( P \in P_2 \), sentence graph \( S_M(\forall \phi P(\phi)) \)
has thus only two levels with \( P(S) \), for all \( S \in S^L_M \), being s-atoms:

\[
\begin{array}{cccc}
P(S) & P(P(S)) & \ldots & P(P(\forall \phi.P(\phi))) \\
\downarrow & \downarrow & \ldots & \downarrow \\
P(S) & P(P(S)) & \ldots & P(P(\forall \phi.P(\phi))) \\
\end{array}
\]

The edge from \( \forall \phi.P(\phi) \) to \( P(\forall \phi.P(\phi)) \) does not create any new cycle because its target is coupled by a 2-cycle only with its dual s-atom \( P(\forall \phi.P(\phi)) \). There are no edges from \( \forall \phi.P(\phi) \), \( P(\forall \phi.P(\phi)) \), etc., occurring as arguments under \( P(.) \), to the root vertex of \( S_M(\forall \phi.P(\phi)) \). Such edges may appear only due to axioms defining \( P \).

5.1. Paradoxes. Predicates applied to sentences provide thus only fresh atoms, which can be defined as \( \mathcal{L} \)-atoms can be, so one might think that everything works unchanged. However, since valuation of these atoms depends on their argument sentences, and not boolean values of these sentences, things are very different. S-predicate \( P \in \mathcal{P} \) can violate some counterparts of semantic equivalences (4.4), e.g., \( P(A) \) and \( P(\neg \neg \neg A) \), being distinct atoms each with its own dual, may have different values. Consequently, s-predicates can break only even cycles, without breaking the corresponding odd ones, leading to paradoxes as illustrated by the following example.

Example 5.2. “Everything John says is true” becomes \( A = \forall \phi(\neg J(\phi) \lor \phi) \), where \( J(\phi) = 1 \) iff John says \( \phi \). The following schema captures \( S(A) \), with \( X \) marking all sentences other than \( A \):

\[
\begin{array}{cccc}
\neg A & J(\neg A) & A & J(A) \\
\swarrow & \searrow & \swarrow & \searrow \\
\swarrow & \searrow & \swarrow & \searrow \\
\neg A & X & \ldots & J(X) \\
\end{array}
\]

Things depend on what John says – in the ways one would expect:

(a) If he says nothing, then \( J(X) = 0 \) for all \( X \), \( \circ_X = 0 \) and \( A = 1 \). When he makes only some consistent statements, a model can be obtained with atoms making them, and \( A \), true.

(b) If he says something false, e.g., \( J(S \land \neg S) \), for a sentence \( S \), then \( \circ_{S \land \neg S} = 1 \) and \( A = 0 \).

(c) If John says “Everything John says is true” and nothing else, then only \( J(A) = 1 \) and there are two models, one where \( A = 1 \) and one with \( A = 0 \).

(d) If John says only \( \neg \neg \neg A = \neg A \), “Not everything I say is true”, then \( J(\neg A) = 1 \), from which paradox ensues, as all \( \circ_X = 0 \) except undetermined \( \circ_X \) in the remaining odd cycle \( A \sim \circ_X \sim \neg A \).

(e) Similar effect arises when John says everything equivalent to \( \neg A \), i.e.:

\[ J(\psi) \leftrightarrow (\psi \leftrightarrow \forall \phi(\neg J(\phi) \lor \phi)) \leftrightarrow ((\psi \land \neg \forall \phi(\neg J(\phi) \lor \phi)) \lor (\neg \psi \land \forall \phi(\neg J(\phi) \lor \phi))) \]

Now, there remains an unresolved odd cycle through \( \circ_{\psi} \) for every \( \psi \leftrightarrow \neg A \).

The truth-teller (c) shows that, unlike in \( \mathcal{L} \), valuation of all \( \mathcal{L}^+ \)-atoms does not always determine a unique valuation of all \( \mathcal{L}^+ \)-sentences, even when the graph is solvable. In cases (d) and (e) paradox is caused by the odd cycle remaining after unfortunate definition of \( J \). The problem is not that we are trying to evaluate \( A \leftrightarrow \neg A \), which is false in every language graph. The problem is that legal sentence \( A \), using available predicate \( J \), becomes impossible to evaluate with an unfortunate definition of \( J \). The culprit is not \( A \) but predicate \( J \), playing a game with the semantics of syntactic operators, in particular, negation. Semantic paradoxes seem to arise when such a game of syntax predicates goes counter to the language’s semantics.

The proof of Lemma 4.1, especially points iii and v, rely crucially on the absence of such games, namely, on evaluation of the root of each DNF-foot \( D[\phi] \) being a boolean function of \( \phi \)'s value: either constant or yielding complementary results on complementary instances, \( D[\neg \phi] = \neg D[\phi] \). Unlike this, DNF-foot \( D(\phi) \), like \( \neg J(\phi) \lor \phi \), depends on the sentence substituted for \( \phi \), not only on its boolean value. Consequently, while the cycles created by the former come only in pairs with opposite parities, the latter may leave unresolved odd cycles, as seen on the example. The proof of Lemma 4.1 goes unchanged for a graph \( S_M(\mathcal{L}^+) \) if each s-predicate \( P(\phi_1...\phi_n) \) is a boolean
function of each of its arguments, namely, if for each \(1 \leq i \leq n\) and each valuation \(\rho\) of \(S_M^L\) and of \(\{\phi_1...\phi_n\} \setminus \{\phi_i\}\):

\[
\begin{align*}
\text{either } & \forall \phi \in S_M^L \setminus S_M^{-} : \quad P(\rho(\phi_1)\ldots\phi_i\ldots\rho(\phi_n)) = P(\rho(\phi_1)\ldots\neg\phi_i\ldots\rho(\phi_n)) \\
\text{or } & \forall \phi \in S_M^L \setminus S_M^{-} : \quad \neg P(\rho(\phi_1)\ldots\phi_i\ldots\rho(\phi_n)) = P(\rho(\phi_1)\ldots\neg\phi_i\ldots\rho(\phi_n)).
\end{align*}
\]

(5.3)

This condition is violated in Example 5.2.(d), where \(J(\neg A) = \neg J(A)\), while \(J(B) = 0\) for \(B \neq A\). Similarly, in (e). It is trivially satisfied when John makes only factual statements (from \(S_M\)).

Reading edges as negations and branchings as conjunctions, every graph represents a theory, if not in FOL, then in infinitary propositional logic, [8]. Unsolvability, representing then inconsistency or paradox, occurs due to some structural patterns. In finitary graphs (having no infinite branching or no rays), it requires an odd cycle, [6], representing negative self-reference. Kernel theory provides several results, specifying often conditions on odd cycles, ensuring solvability of finite graphs, [3]. No general condition for arbitrary graphs is known but it is natural to conjecture that, besides self-negation, one should exclude an appropriately defined Yablo-like pattern. A special case of such a conjecture is shown in [9] for a wide class of graphs. Now, condition (5.3) ensures solvability of our very special language graphs, by restricting definitions of s-predicates. A similar restriction, given in Section 5.3 below, can be enforced by axioms, so we show first how to incorporate axioms into graph representation.

5.2. Theories. A theory means here an arbitrary set of sentences. Given a theory \(\Gamma \subseteq L^+\) and an \(L^-\)-domain \(M\), a graph for \(\Gamma\) can be obtained extending the language graph \(S_M^L(\Gamma)\) to \(S_M^L(\Gamma)\) by adding, for each axiom \(S \in \Gamma\), a fresh vertex \(\omega_S\) with a loop and an edge to vertex \(S\) in \(S_M^L(S)\). For instance, axiom \(S = P(t) \leftrightarrow (P(t) \rightarrow Q(s))\), written in PNF as \(P(t) \land Q(s)\), results in extending the language graph with vertex \(\omega_S\), having a loop and edge to \(S\):

\[
\begin{align*}
S_M^L(\Gamma) \ni & \omega_S \quad \text{loop} \\
S_M^L(\Gamma) \ni & \begin{array}{c}
p(t) \xrightarrow{P(t)} S \xrightarrow{Q(s)} \equiv Q(s) \\
\end{array}
\end{align*}
\]

(5.4)

Each kernel of so extended graph must contain each \(S \in \Gamma\), since independence excludes \(\omega_S\) with the loop from any kernel. Here, the only kernel is \(\{P(t), Q(s), S\}\), reflecting valuation \(P(t) = 1 = Q(s)\). This representation amounts to selection of models by identifying valuations of s-atoms making the axioms true. Although possible and close to the classical standard, it does not make the structure, in particular, possible circularities implied by axioms, explicit.

We therefore choose another way of modifying the underlying language graph \(S_M^L(L^+)\). Theories are assumed given in graph normal form, GNF, [2, 10], which relies on sufficiency of negation and relevant variant of conjunction (e.g., binary, infinitary, universal quantification).

**Definition 5.5.** A formula of a language \(L^+\) is in GNF if it is an equivalence where

- the left side is an s-atom, \(LS \in A_X^+\),
- the right side, \(RS\), is a (universally quantified) conjunction of negated s-atoms,
- free variables of \(RS\) occur freely in \(LS\) and all free variables are universally quantified outside the whole formula.

A theory \(\Gamma\) is in GNF if each \(F \in \Gamma\) is in GNF.

Atomic formulae are special cases of GNF, consisting of \(LS\) with empty \(RS\). Negations of s-atoms, represented by equivalences to contradictions in \(RS\), can be abbreviated by plain negations. Every theory \(\Gamma\) in FOL (also with equality) has a theory \(\Delta\) in GNF such that models of both are in bijection, [10]. \(\Delta\) is not unique and requires typically introduction of fresh predicate symbols.

Given a GNF theory \(\Gamma \subseteq L^+\), we extend \(S_M^L(L^+)\) to the graph \(S_M^L(\Gamma)\), by adding edges which capture the semantic requirements imposed by the axioms. The extension is given in the following definition, where uppercase \(B, B_i\) stand for predicate symbols in \(P_1 \cup P_2\) (dropping the formula arguments, gives special case for \(P_1\)), while \(F, E\) for sentences of \(L^+\) and \(d, e\) for elements of \(M\):

**Definition 5.6.** For a GNF theory \(\Gamma \subseteq L^+\) and a domain \(M\), the language (or theory) graph \(S_M^L(\Gamma)\) is obtained by modifying the language graph \(S_M^L(L^+)\) as follows:
1. For each axiom \( \forall \phi \forall x (B(\phi, x) \leftrightarrow \forall \psi \forall y (\bigwedge_{i \in I} \neg B_i(\phi, x, \psi, y))) \) and each \((F, d) \in (S^+_M \cup M)^{ar(B)}\), add an edge \( B(F, d) \rightarrow B_i(F, d, E, e), \) for each \( i \in I \) and \((E, e) \in (S^+_M \cup M)^{ar(B, i)(F, d)}\).

2. For each atom \( B(F, d) \), that is an instance of LS of any axiom, remove the edge going out of \( B(F, d) \) in its 2-cycle to its dual \( \overline{B(F, d)} \).

In point 1, the substituted \((F, d) \in (S^+_M \cup M)^{ar(B)}\) and \((E, e) \in (S^+_M \cup M)^{ar(B, i)(F, d)}\) are, of course, of appropriate type. In point 2, only the edge from the positive atom to its negative dual is removed, as GNF has only positive atoms in LSs.²

Axiom \( S \) from (5.4) in GNF becomes \( P(t) \leftrightarrow \neg \bullet \) and \( \bullet \leftrightarrow \neg P(t) \land \neg Q(s) \), giving graph \( S_M(S) \):

\[
\begin{align*}
P(t) &\rightarrow \bullet & Q(s) &\equiv \neg Q(s) \\
\downarrow \quad & & \Downarrow \\
P(t) &\rightarrow \\
\end{align*}
\]

As graph \( S^+_M(S) \) from (5.4), this GNF graph has only one kernel \{\( Q(s), P(t) \)\}, representing the valuation \( Q(s) = 1 = P(t) \) making \( S \) true.

In the following, referring to “all language graphs”, we mean all graphs which can be obtained according to Definition 5.6, in particular, graphs for empty theories.

Let \( GMd(\Gamma) \) denote graph models of \( \Gamma \), namely, all pairs \((S_M(\Gamma), K)\), over all domains \( M \), with \( K \in sol(S_M(\Gamma)) \). For \( \Gamma \) in FOL, or in \( \mathcal{L} \) over FOL, each such graph model determines a unique FOL model of \( \Gamma \) and vice versa. (The first statement is shown in [10], and the second one follows from it by Definition 4.3 and Theorems 4.2 and 4.7.) We do not attempt to extend the notion of truth in usual FOL structures to arbitrary \( \Gamma \subseteq L^+ \), since valuation of atoms may leave values of some sentences undetermined, as in Example 5.2.(c). It seems plausible to simply define models of \( \Gamma \) to be \( GMd(\Gamma) \). In case of definitional extensions, which are of primary interest below and avoid paradoxes and indeterminacy, \( GMd(\Gamma) \) remain in bijection to usual FOL models.

### 5.3. Definitional extensions and truth predicate.

Condition (5.3) is sufficient to ensure that language \( L^+ \) with s-predicates is free for paradoxes. Each constant s-predicate satisfies it, so each language graph \( S_M(L^+) \) is solvable, but such a constant interpretation is hardly satisfactory.

Verification of solvability for more specific definitions of s-predicates may be hampered by the semantic character of condition (5.3). We identify now a special syntactic form which implies this condition along with some additional properties.

As a special case of Definition 5.6, a given language graph (typically, \( S_M(\Gamma) \), for \( \Gamma \subseteq L \)) can be extended with additional predicates \( P \) and their GNF axiomatization \( Ax \). The resulting graph \( S_M(\Gamma \cup Ax) \), over the extended language \( L^P \), has new vertices and subgraphs for all new sentences \( S^+_M \cup S_M \), edges for all \( Ax \) as in Definition 5.6 and, in addition, extends each old subgraph \( S_M(A) \), \( A \in S_M \setminus S^+_M \), with all substitution sequences involving new sentences \( S^+_M \setminus S_M \). The following restriction on \( Ax \) ensures that such an expansion of \( S_M(\Gamma) \) has essentially unchanged solutions.

Starting with a single s-predicate \( P \), we require it to be defined over language \( L \) only, without reference to any s-predicates, in particular, without any self-reference beyond that arising from s-quantifiers. It can be defined only by a sentence of the following form:

\[
(5.7) \quad \forall \phi (P(\phi) \leftrightarrow \mathcal{F} \psi F[\phi, \psi]),
\]

where \( F \) is an \( L \)-formula (\( \phi, \psi \) may be sequences of s-variables). This special case of GNF corresponds to explicit definition which preserves and reflects solvability, allowing to eliminate symbol \( P \). This follows directly from Lemma 4.1, giving \( \mathcal{F} \psi F[\phi, \psi] \) a unique value under each \( \rho \in 2^{S^+_M \cup \phi} \) and making \( P(\phi) \) a boolean function of \( \phi \), as required by (5.3).

Lemma 5.12 below gives a stronger, and hardly unexpected, claim: an extension of a solvable graph \( G \) with \( P \) axiomatized by (5.7) is not only solvable, but has essentially the same solutions as \( G \). Its proof amounts to elimination of symbol \( P \), replacing each \( P(S) \) by its definitions \( \mathcal{F} \psi F[S, \psi] \). This operation, trivial in FOL, has to be performed recursively (e.g., \( P(P(S)) \) needs repeated

²In general, when \( T_X \neq X \), Definition 5.6 must be adjusted to cater for unifiable LSs of distinct axioms and for possibly identical interpretation of distinct terms. We ignore this here, referring to [10], where this is handled by extending the graph with equality predicate, covering also FOL with equality.
replacements) on a cyclic graph and involves some technicalities. These can be distracting so, skipping them on the first reading, one can continue with the paragraph before Lemma 5.12.

The proof assumes a language graph $G$ in which no two vertices have equal out-neighbourhoods. (If $G$ contains such vertices, as language graphs typically do, their identification preserves essentially the solutions by Fact 2.9, and we apply the construction and fact below to the so quotiented $G$.) The graph $H$ stands for $G$'s extension with $P$. We map $\gamma : H \rightarrow G$, performing a sequence of identifications $\gamma_i : H_{i-1} \rightarrow H_i$, for $0 < i \leq \omega$ and $H_0 = H$. Each $\gamma_i$ is identity on the subgraph $G$ of $H_i$, identifying some vertices from $V \setminus V_G$ with some in $V_G$. First, $\gamma_1(P(S)) = \exists \psi F[S, \psi]$, removing the double edge and the intermediate vertex $•$ $G$ partially the solutions by Fact 2.9, and we apply the construction and fact below to the so quotiented $H$. The graph $H_i$ contains such vertices, as language graphs typically do, their identification preserves essence of $H$. The proof assumes a language graph $G$.

**Example 5.9.** Let $P(\phi) \leftrightarrow \exists \psi (\phi \land \psi)$ and, for some $S \in S_M$, consider vertex $P(P(S)) \in V_H$. The relevant parts of the graph $H$ are sketched on Figure 2, with $A/L$, $B/L$, $A/R$, $B/R$, denoting vertices with the respective sentence substituted for the $3$-quantified $\psi$. The subscripts $L/R$ mark these instantiations in the respective subgraphs, e.g., $A/L = P(S) \land A$ and $A/R = \exists \phi (S \land \phi) \land A$. Sentences $A, B, ...$ (and $A, B, ...$) are duplicated in both subgraphs to increase readability, but they are actually the same vertices.

**Figure 2. Example 5.9.**

1. $P(P(S)) \sim P(\exists \psi (P(S) \land \psi))$ and $P(S) \sim P(\exists \psi (S \land \psi))$, hence $\mathbf{E}_S(P(S)) = \{\gamma_1(P(S))\} = \{P(\exists \psi (S \land \psi))\}$ and, consequently,

2. $P(S) \sim \exists \psi (S \land \psi)$. Then, for each $A \in S_M^+$, $\mathbf{E}_S(A/L) = \{\exists \psi (S \land \psi), A\} = \mathbf{E}_S(A/R)$, so that
3. $A/L \sim A/R$, for every $A \in S^+_M$.
4. Consequently, $\mathbf{L} \sim \mathbf{R}$ and then
5. $\exists \phi (\exists \phi (S \land \phi) \land \psi) \sim \exists \phi (P(S) \land \psi) \sim_5 P(P(S))$, leaving only the subgraph of $G$ to the right.

The equivalence $\sim$ is a congruence on $V_u$ in the sense that if all out-neighbours of $A$ and $B$ are $\sim$-equivalent then $A \sim B$, i.e., for $E_u(A) = \{ A_i \mid i \in I \}$ and $E_u(B) = \{ B_i \mid i \in I \}$:

\[(5.10)\]
if $\forall i \in I : A_i \sim B_i$ then $A \sim B$.

This holds since each sentence subgraph $S(A)$ (tree $T(A)$) has finite height $h(A)$, in particular distance from the root $A$ to atoms $P(S)$ of $S(A)$ is at most $h(A)$. Hence, if $\forall i \in I : A_i \sim B_i$ then $\exists n \leq \max \{ h(A), h(B) \} \forall i \in I : A_i \sim B_i$.

The equality $\gamma_n(A_i) = \gamma_n(B_i)$ implies, in turn, that $E_n(A) = \{ \gamma_n(A_i) \mid i \in I \} = \{ \gamma_n(B_i) \mid i \in I \} = E_n(B)$, which yields $A \sim B$.

**Fact 5.11.** (a) $\forall S \in S^+_M \setminus S_M : \exists Q \in S_M : Q \sim S$, hence $\gamma(H) = G$,

(b) $H$ and $G$ have essentially the same solutions.

**Proof.** Point (a) is shown by induction on the number $p$ of $Ps$ in a sentence $S \in S^+_M \setminus S_M$.

1. If $p = 1$ and $S$ is atomic, then $S = P(R)$ for some $R \in S_M$, so $S \sim_1 \exists \psi F[R, \psi] \in S_M$.

2. If $p = 1$ and $S$ is not atomic, we proceed by structural induction on $S$, with point 1 providing the basis and induction hypothesis $IH_2$:

i. $\land_{i \in I} S_i$, for finite $I$. By $IH_2$, for each $S_i$ there is $Q_i \in S_M$ with $S_i \sim Q_i$, so $\land_{i \in I} S_i \sim_2 \land_{i \in I} Q_i$ by (5.10), and $\land_{i \in I} Q_i \in S_M$.

ii. $\neg A$. By $IH_2$, $A \sim Q$ for some $Q \in S_M$, so $\neg A \sim \neg Q$ by (5.10), while $\neg Q \in S_M$.

iii. $S = \exists \phi A[\phi]$, where $\phi$ does not occur under $P$, so that $S = \exists \phi A[\phi, P(R)]$, for some $R \in S_M$ and context $A[\phi, \_]$ with no $P$. Since $P(R) \sim \exists \psi F[R] \in S_M$, taking $Q = \exists \phi A[\phi, \exists \psi F[R]] \in S_M$, we obtain $A[T, P(R)] \sim A[T, \exists \psi F[R]]$ for every $T \in S^+_M$ by (5.10), i.e., for all grandchildren of $S$ and $Q$. By (5.10), this yields $S \sim Q$.

iv. $S = \exists \phi A[P(C[\phi])]$, i.e., $S$ contains quantification into $P$, for some contexts $A[\_], C[\_]$ without any $P$, as $p = 1$. For grandchildren of $S$, namely, $A[P(C[T])]$ for all $T \in S^+_M$, the equivalence $P(C[T]) \sim_1 \exists \psi F[C[T], \psi]$ gives $A[P(C[T])] \sim \exists \psi F[C[T], \psi]$ by (5.10). Sentences on the left, for all $T \in S^+_M$, comprise all grandchildren of $S$, and those on the right all grandchildren of $Q = \exists \phi A[\exists \psi F[C[\phi], \psi]] \in S_M$, so $S \sim Q$ by (5.10).

3. For the induction step for $p > 1$, the two cases depend on whether $P$ is nested or not.

i. If the number of $Ps$ not nested under others is $n > 1$, consider all these highest $Ps$ in $T_M(S)$, i.e., $S = C[P(A_1), \ldots, P(A_n)]$, where $C[\_]$ contains no $Ps$. For $R = C[\exists \psi F[A_1, \psi], \ldots, \exists \psi F[A_n, \psi]]$, $S \sim R$ by (5.10). $R$ has $p - n < p$ $Ps$ so, by $IH$, $R \sim Q$ for some $Q \in S_M$, hence $S \sim Q$.

ii. If all $Ps$ are nested under each other, then $S = C[P(A)]$ for some context $C[\_]$ without any $Ps$, and with $p - 1$ occurrences of $P$ in $A$. $P(A) \sim_1 \exists \psi F[A, \psi]$ and, by $IH$, $\exists \psi F[A, \psi] \sim R$ for some $R \in S_M$, so that also $P(A) \sim R$. Then $C[P(A)] \sim C[R]$ by (5.10) and $C[R] \in S_M$, as required.

The equality $\gamma(H) = G$ follows since each $S \in V_u \setminus V_o$ represents a sentence in $S^+_M \setminus S_M$.

(b) For $i \geq 0$, $H_i$ is the quotient of $H$ by $\sim_i$, $\ldots$, $\sim_i$. By Fact 2.8, $H_1$ has essentially the same solutions as $H$. (No ray is contractible to a finite path, because the case $P(S) \sim_1 \exists \psi F[S, \psi]$ is applied at most finitely many times along each path under each sentence $Q$, since $Q$ contains at most finitely many nested $Ps$.) By Fact 2.9, the same holds for $H_1$ and every $H_i$, $i > 1$, including limits $H_\infty$. Thus, $H$ and $\gamma(H) = G$ have essentially the same solutions.

Let *definitional extension* refer to any finite chain starting with any theory $\Gamma_0 \subseteq L_0 = \mathbb{L}^+$ and adding, at step $i + 1$, axiom (5.7) with a fresh predicate $P \not\in L_i$ and $F[\psi, \phi] \in L_i$, for language $L_i$ of theory $\Gamma_i$ obtained at step $i$. The following counterexample of model theoretic conservativity of usual definitional extensions holds.

---

3This implication fails in general graphs for $\sim$ defined by (5.8) from some basis $\sim$, when $I$ is infinite and distance from $A_i, B_i, i \in I$, to relevant pairs $X \sim Y$ is unbounded.
LEMMA 5.12. Each language graph and the graph of its definitional extension have essentially the same solutions.

PROOF. Fact 5.11 gives the claim for an extension with a single predicate. By IH, definitional extension $G_i$ of $G = S_M(\Gamma_0)$ with $P_1, ..., P_i$, has essentially the same solutions as $G$. Graph $G_{i+1}$, obtained now by adding $P_{i+1}$, whose definiens $F_{i+1}$ can utilize $P_j, j \leq i$, has by Fact 5.11 essentially the same solutions as $G_i$, and hence as $G$.

A non-paradoxical language $L^+$ is one having a solvable graph $S_M(L^+)$ so, by this lemma, its definitional extension remains non-paradoxical. The lemma, concerning solvable language graphs in general, applies in particular to solvable theory graphs $S_M(\Gamma)$, showing that definitional extensions ensure conservativeness when applied to theories.

THEOREM 5.13. For every $\Gamma \subseteq L^+$ and its definitional extension $F$, theories $\Gamma$ and $\Gamma \cup F$ have essentially the same graph models.

Definition (T) $\forall \phi(T \phi \leftrightarrow \phi)$ satisfies trivially (5.7), so a simple corollary follows.

COROLLARY 5.14. Every theory $\Gamma \subseteq FOL$ and $\Gamma \cup \{(T)\}$ have essentially the same models.

Unrestricted convention (T) can be thus added conservatively to every FOL theory $\Gamma$. Since, by Lemma 5.12, reduct of each model of this extension is a model of $\Gamma$, the extension does not increase expressive power of FOL.

Tarski’s compositionality equivalences for sentences follow, e.g., $\forall \phi(T(\neg \phi) \leftrightarrow S_M(T) \neg \phi)$, etc. (in any graph $S_M(T)$ for theory (T), Definition 5.6, over $L^T$, denoting $L^+$ for $L$ extended with $T$.)

Restriction to sentential predicates makes the RS of the equivalence $T(\forall x \phi(x)) \leftrightarrow S_M(T) \forall x \phi(x)$ syntactically dubious. Still, both sides are sentences and it does hold, schematically in $\phi(x)$. By Definition 5.6, graph $S_M(T)$ replaces edge $T(S) \rightarrow T(S)$ of $S_M(L^T)$ by a double edge $T(S) \rightarrow S$ for every $S \in S_M(T)$, in particular, for $T(\phi(m_i))$ with each $m_i \in M$. Solution of $S_M(T)$, under any valuation of $L_{M^+}$-atoms, valuates correctly all $\phi(m_i)$, which form a common cut yielding the same value at both sides of the equivalence:

$\Gamma(\forall x \phi(x)) \leftrightarrow \forall x \phi(x)$

As illustrated by this equivalence, which one would like to write as $\forall \phi(T(\forall x \phi(x)) \leftrightarrow \forall x T(\phi(x))$, language graphs provide semantics also for quantification over open formulae that become closed at places where they are used, but such a quantification requires further investigation.

§6. Open issues. We have addressed only semantic aspects so syntactic issues, like the one just mentioned, await addressing. Concerning reasoning, complete systems for restricted cases should result from natural extensions of FOL: for $L$ directly from Fact 4.5, for the mere (T) by inserting $T$ arbitrarily into FOL-provable sentences, and by similar means for other definitional extensions. Genuine complications arise for the general case of $L^+$ with arbitrary s-predicates. Substitutional interpretation of s-quantifiers, combined here with s-quantification into nominal positions, destroys compactness. A unary s-predicate $P$ and $\Gamma = \{P(S) \mid S \in S^+\}$ make $\Gamma \models \forall \phi P(\phi)$, but no finite means yield $\Gamma \vdash \forall \phi P(\phi)$. Each finite sub-theory of $\Gamma$ is consistent with $\forall \phi P(\phi)$, but $\Gamma$ is not. For completeness, some infinitary rule seems necessary, allowing to derive $\forall \phi P(\phi)$ from $\Gamma$. Such a rule can easily lead to circular proofs, with $\Gamma \vdash \forall \phi F(\phi)$ reappearing higher up in the proof tree. Reasoning in this general case needs closer investigation.

As self-reference is modeled by graph cycles, all sentences with sentential quantifiers, i.e., the whole subgraph $S_M(L \setminus L^-)$, form one component of indirect self-references. Results from kernel theory give precise grasp of many informal intuitions, e.g., that in finite discourses only negative self-reference, namely odd cycles, can be vicious causing paradoxes, because a finitary graph without odd cycles is solvable. If one is missing self-reference in a more direct logical form,
sentence names $C$ from grammar (3.1) can be used. In language graphs, such undefined names become atoms with 2-cycles to their duals and, like other $s$-atoms, can also be defined. One can define the liar ($L$) as $L \leftrightarrow \neg TL$, although the purpose of doing this in the presence of $(T)$ would be unclear. Literally, $(L)$ is rather the strengthened liar, the liar being plain $L \leftrightarrow \neg L$. Both are simply false in any language graph ($(L)$, if we assume $(T)$), leading to unsolvability when posited as axioms. As each FOL language with sentential quantifiers and definitional extensions (5.7) remains free from paradoxes, usual paradoxes are expressible at most as inconsistencies.

REFERENCES


DEPARTMENT OF INFORMATICS
UNIVERSITY OF BERGEN
PBOX 7803, 5020 BERGEN, NORWAY
E-mail: michal@ii.uib.no