Bulletin of TICMI
Vol. 22, No. 1, 2018, 3–10

Skolemization in Unranked Logics

Lia Kurtanidze[a*] and Mikheil Rukhaia[b]

[a] *Faculty of Informatics, Georgian University*
[b] *I. Vekua Institute of Applied Mathematics, I. Javakhishvili Tbilisi State University*
(*Received March 30, 2018; Revised June 8, 2018; Accepted June 12, 2018*)

In this paper we study skolemization for unranked logics with classical first-order semantics. Skolemization is a transformation on first-order logic formulae, which removes all existential quantifiers from a formula. This technique is vital in proof theory and automated reasoning, especially for refutation based calculi, like resolution, tableaux, etc. Here we extend skolemization procedure to unranked formulae and prove that the procedure is sound and complete.

Keywords: Skolemization, Unranked logics; Proof complexity, Normal forms.

AMS Subject Classification: 68T15, 03F07, 03B70, 68T27, 03F05.

1. Introduction

Skolemization is a well known technique in refutational theorem proving to eliminate existential quantifiers. It is sometimes called an extension method, because it introduces new symbols in the signature of a formula. A very important feature of skolemization is that it looses logical equivalence, but preserves sat- or validity-equivalence.

Skolemization procedure is well studied for classical first-order logic (see e.g. [1, 5, 20]), Constrained Logic (see e.g. [2]), Intuitionistic Logic (see e.g. [3, 4]), Fuzzy Logics (see e.g. [6]), Łukasiewicz Logic (see e.g. [7]), Probabilistic Logic (see e.g. [27]) and the like.

The aim of this paper is to define different kinds of skolemization procedures for unranked logics and discuss the effects in the proof complexities of skolem normal forms.

Unranked languages are based on unranked alphabet, where function and/or predicate symbols do not have a fixed arity. Thus, in these languages, inside a term or a formula, it is possible to have several different occurrences of the same function/predicate symbol with different number of arguments. Additional strength to these languages are given by the notions of sequence variables and sequence functions. Sequence variables can be instantiated with finite sequences of terms, whereas sequence functions are interpreted as multi-valued functions. These constructs seems to be higher-order, but they have a precisely defined first-order semantics (see e.g. [16, 18, 22]).

In recent years, the usefulness of sequence variables and unranked symbols has been illustrated in practical applications related to XML [11–15, 25], knowledge

*Corresponding author. Email: lia.kurtanidze@gmail.com

ISSN: 1512-0082 print

representation [16, 18], automated reasoning [8, 19, 21, 23, 26], just to name a few. In this extent, it is well known that on the one hand, tableaux method is widely used for reasoning in Semantic Web and Knowledge Representation. On the other hand, skolemization procedure is crucial for the tableaux method. In [17] a tableaux calculus for unranked logics was presented, but the skolemization procedure itself was not defined. With the present paper we intend to fill the gap.

2. Preliminaries

We define first-order unranked predicate logic in the similar way as it was given in [17, 22]. We consider an alphabet $\mathcal{A}$ consisting of the following pairwise disjoint sets of symbols:

- the set $\mathcal{V}_i$ of individual variables, denoted by $x^i, y^i, z^i, \ldots$,
- the set $\mathcal{V}_s$ of sequence variables, denoted by $\overline{x}, \overline{y}, \overline{z}, \ldots$,
- the set $\mathcal{F}_i^r$ of fixed arity (ranked) individual function symbols, denoted by $f^r, g^r, \ldots$,
- the set $\mathcal{F}_i^u$ of flexible arity (unranked) individual function symbols, denoted by $f^u, g^u, \ldots$,
- the set $\mathcal{F}_s^r$ of fixed arity (ranked) sequence function symbols, denoted by $\overline{f}^r, \overline{g}^r, \ldots$,
- the set $\mathcal{F}_s^u$ of flexible arity (unranked) sequence function symbols, denoted by $\overline{f}^u, \overline{g}^u, \ldots$,
- the set $\mathcal{P}^r$ of ranked predicate symbols, denoted by $p, q, \ldots$,
- the set $\mathcal{P}^u$ of unranked predicate symbols, denoted by $p^u, q^u, \ldots$,
- logical connectives and quantifiers $\neg, \wedge, \vee, \rightarrow, \exists, \forall$,
- auxiliary symbols: parentheses and the comma.

Each ranked symbol has a unique arity (rank) associated with it. If the rank of a function symbol is 0, then it is called a *constant*. Unranked symbols do not have a fixed arity. Each set of variables, function symbols, and predicate symbols is countably infinite. The letter $\mathcal{V}$ denotes the set $\mathcal{V}_i \cup \mathcal{V}_s$ and its elements are denoted by $x, y, z, \ldots$.

The *terms* are defined as *individual* and *sequence* terms over $\mathcal{A}$ in the following inductive way:

$$t ::= x^i \mid f^r(t_1, \ldots, t_n) \mid f^u(\overline{s}_1, \ldots, \overline{s}_n) \qquad \text{individual terms}$$

$$\overline{s} ::= t \mid \overline{x} \mid \overline{f}^r(t_1, \ldots, t_n) \mid \overline{f}^u(\overline{s}_1, \ldots, \overline{s}_n), \quad n \geq 0 \qquad \text{sequence terms}$$

An *atom* is a formula of the form $p(t_1, \ldots, t_n), n \geq 0$ and $p^u(\overline{s}_1, \ldots, \overline{s}_m), m \geq 0$, where $p \in \mathcal{P}^r$ is an n-ary predicate symbol, and $p^u \in \mathcal{P}^u$ is a flexible arity predicate symbol. *Formulas* are built as usual from atomic formulas and logical connectives $\neg, \wedge, \vee, \rightarrow, \exists$, and $\forall$. Quantifications are allowed on both, individual as well as on sequence variables. We use the letters $A, B, \ldots$ to denote formulas.

A *substitution* is a finite set of distinct variable bindings, where variable binding is either expression $x^i \mapsto t$ or $\overline{x} \mapsto \overline{s}$. Substitutions are denoted by σ, θ and the empty substitution is denoted by ϵ. Application of a substitution σ on term t and formula A is defined in the similar way as it is given in [24] and is denoted by $t\sigma$ and $A\sigma$, respectively.

The semantics of our language is assumed to be classical first-order semantics, defined in a similar way as it is given in [22, 24]. The notions of interpretation, evaluation, etc., are standard. Note that, as usual, $\forall$ and $\exists$ are dual to each other in our setting too; i.e., $\forall^d \equiv \exists$ and $\exists^d \equiv \forall$.

The following definitions provide some terminology, vital for the next section.

Definition 2.1: (Strong and weak quantifiers). The polarity of a quantifier is defined in the following inductive way:

- (Qx) is positive in $(Qx)A$.
- if (Qx) is positive (negative) in A or in B, then it is positive (negative) in $A \wedge B$ and in $A \vee B$.
- if (Qx) is positive (negative) in B, then it is positive (negative) in $A \rightarrow B$.
- if (Qx) is positive (negative) in A, then it is negative (positive) in $\neg A$ and in $A \rightarrow B$.

Positive universal (existential) quantifiers and negative existential (universal) quantifiers are called *strong* (*weak*) quantifiers.

Definition 2.2: (Quantifier scope). Let A be a formula and let $(Qx)B$ be its subformula. We say, that B is in the scope of (Qx) and for every subformula $(Q'y)C$ of B, the quantifier $(Q'y)$ is in the scope of the quantifier (Qx).

Let A be a formula and x a variable occurring in A. We say that an occurrence of x is bound in A, if it occurs in the scope of the quantifier (Qx), otherwise its occurrence is free. If all occurrences of x are bound in A, then we say that x is a bound variable in A. Please note, that it can be the case, that one variable has free and bound occurrences in a formula, but we can avoid such cases by renaming bound variables. In the similar way we avoid cases when one variable is bound by two or more quantifiers. We say that A is a closed formula, if it does not contain free variables.

Definition 2.3: (Quantifier omission). Let A be a formula. $A_{-(Qx)}$ denotes the formula A, where an occurrence of the quantifier (Qx) is omitted.

Remark 1: Note that the bound variable x of A becomes free in $A_{-(Qx)}$.

Example 2.4 Let A be the formula $B \wedge (\forall x^i)(\exists \overline{x})p^u(x^i, \overline{x})$. Then $A_{-(\forall x^i)}$ is the formula $B \wedge (\exists \overline{x})p^u(x^i, \overline{x})$ and $A_{-(\exists \overline{x})}$ is $B \wedge (\forall x^i)p^u(x^i, \overline{x})$.

We say, that a formula is in the prenex form, if it has the form $(Q_1 x_1) \cdots (Q_n x_n)A$, where A is a quantifier-free formula. Every formula can be transformed into a prenex form using the following quantifier shifting rules:[1]

$$\begin{aligned}
\neg(Qx)A &\Longleftrightarrow (Q^d x)\neg A \\
(\exists x)(A \vee B) &\Longleftrightarrow (\exists x)A \vee (\exists x)B \\
(\forall x)(A \wedge B) &\Longleftrightarrow (\forall x)A \wedge (\forall x)B \\
(\exists x)(A \rightarrow B) &\Longleftrightarrow (\forall x)A \rightarrow (\exists x)B
\end{aligned}$$

[1]To transform a formula into prenex form, the quantifier shifting rules are used from right to left direction. The other direction is used to transform a formula into antiprenex form.

and if x is not free in B, then

$$\begin{aligned}
(Qx)(A \vee B) &\Longleftrightarrow (Qx)A \vee B\\
(Qx)(A \wedge B) &\Longleftrightarrow (Qx)A \wedge B\\
(Qx)(A \to B) &\Longleftrightarrow (Q^d x)A \to B\\
(Qx)(B \to A) &\Longleftrightarrow B \to (Qx)A
\end{aligned}$$

3. Skolemization

In a refutational calculus skolemization is a removal of the existential quantifiers from formulas. There are various ways to define skolemization:

Prenex: the traditional way to get skolem normal form of a formula. First, the formula is transformed to prenex normal form and then existential quantifiers are removed by replacing the corresponding bound variables by new n-ary function symbols, $n \geq 0$, where n is the number of universal quantifiers, preceding the existential one.
Structural: this method does not need transformation to the prenex normal form. It is a bit more general, because it can eliminate strong quantifiers from a formula. The rule is similar – strong quantifier (Qx) depends on the weak quantifiers having (Qx) in their scope. It is possible to remove weak quantifiers in the same way, but it is called *Herbandization* in the literature (see e.g. [9, 10]).
Antiprenex: this method is similar to structural skolemization, but contrary to the prenex normal form, quantifiers are shifted deep inside the formula using quantifier shifting rules, to minimize the range of quantifiers. This leads to smaller skolem terms in general.

Below we consider structural skolemization, because it is more general and does not need any preprocessing steps like prenex or antiprenex. We extend definitions from [1] to first-order unranked formulae.

Definition 3.1: (Structural skolemization). Let A be a closed unranked formula, where no variable is bound by two quantifiers. We define a structural skolemization operator $sk()$ in the following way:

- If A does not contain strong quantifiers, then $sk(A) = A$.
- Let (Qx^i) be the first strong quantifier occurring in A. If (Qx^i) is in the scope of weak quantifiers $(Q_1x_1), \ldots, (Q_nx_n)$, $n \geq 0$, then we distinguish the following cases:
  - if $n > 0$ and $x_j \in \mathcal{V}_s$ for any $j = 1, \ldots, n$, then take a fresh flexible arity individual function symbol f^u not occurring in A and $sk(A) = sk(A_{-(Qx^i)}[x^i \mapsto f^u(x_1, \ldots, x_n)])$;
  - otherwise, take a fresh n-ary individual function symbol f^r not occurring in A and $sk(A) = sk(A_{-(Qx^i)}[x^i \mapsto f^r(x_1, \ldots, x_n)])$; note that, if $n = 0$, then f^r is a constant.
- Let $(Q\overline{x})$ be the first strong quantifier occurring in A. If $(Q\overline{x})$ is in the scope of weak quantifiers $(Q_1x_1), \ldots, (Q_nx_n)$, $n \geq 0$, then we distinguish the following cases:
  - if $n > 0$ and $x_j \in \mathcal{V}_s$ for any $j = 1, \ldots, n$, then take a fresh flexible arity sequence function symbol $\overline{f}^u$ not occurring in A and $sk(A) = sk(A_{-(Q\overline{x})}[\overline{x} \mapsto$

$\overline{f}^u(x_1, \ldots, x_n)]$);
- otherwise, take a fresh n-ary sequence function symbol $\overline{f}^r$ not occurring in A and $sk(A) = sk(A_{-(Q\overline{x})}[\overline{x} \mapsto \overline{f}^r(x_1, \ldots, x_n)])$; note that, if $n = 0$, then $\overline{f}^r$ is a constant.

Remark 1 : Note that, if we exchange places of words "strong" and "weak" in Definition 3.1, then we get Herbandization of a formula – exact dual to Skolemization. Also, note that weak quantifiers become strong ones when formula is negated in refutational calculus.

It is easy to see, that $sk()$ operator is well-defined: there is a finite number, say n strong quantifiers in a formula, thus a fixed point of $sk()$ operator will be reached after n steps of the recursive definition. Moreover, $sk()$ is a function: it produces the same skolem normal form modulo function symbol names.

Definition 3.2: Let A be an unranked formula and let A' be its prenex form; then $sk(A')$ is a prenex skolem form of A. Analogously we define an antiprenex skolem form of A.

In the definition above we say "a" instead of "the", because a formula might have several prenex forms, thus different skolem normal forms. This is illustrated in the following example.

Example 3.3 Let us consider the formula:

$$A\colon (\exists x^i)\left((\forall y^i)p^u(x^i, y^i) \wedge (\forall \overline{x})q^u(\overline{x})\right) \wedge (\exists \overline{y})\,(p^u(\overline{y}) \vee q^u(\overline{y}))$$

and compute its structural, anti-prenex and prenex skolem forms. According to Definition 3.1, we get:

$$(\exists x^i)\left(p^u(x^i, f^r(x^i)) \wedge q^u(\overline{f}^r(x^i))\right) \wedge (\exists \overline{y})\,(p^u(\overline{y}) \vee q^u(\overline{y}))$$

Note, that f^r and $\overline{f}^r$ are different function symbols, individual and sequence function symbols, respectively, both with arity 1.

If we apply quantifier shifting rules, we can obtain the anti-prenex form of A:

$$A'\colon (\exists x^i)(\forall y^i)p^u(x^i, y^i) \wedge (\forall \overline{x})q^u(\overline{x}) \wedge (\exists \overline{y})\,(p^u(\overline{y}) \vee q^u(\overline{y}))$$

Again, by Definition 3.1, we get:

$$(\exists x^i)p^u(x^i, f^r(x^i)) \wedge q^u(\overline{f}^r) \wedge (\exists \overline{y})\,(p^u(\overline{y}) \vee q^u(\overline{y}))$$

Note, that here $\overline{f}^r$ (with arity 0) is a constant and does not depend on x^i.

There are several prenex forms of A, namely, the order of quantifiers can be $(\exists x^i)(\forall y^i)(\forall \overline{x})(\exists \overline{y})$, $(\exists x^i)(\forall \overline{x})(\forall y^i)(\exists \overline{y})$, $(\exists \overline{y})(\exists x^i)(\forall y^i)(\forall \overline{x})$, or $(\exists \overline{y})(\exists x^i)(\forall \overline{x})(\forall y^i)$. The skolem terms will be similar to the one from A in the first two cases, so interesting is to consider either from the last two cases, e.g.:

$$A''\colon (\exists \overline{y})(\exists x^i)(\forall y^i)(\forall \overline{x})\left(p^u(x^i, y^i) \wedge q^u(\overline{x}) \wedge (p^u(\overline{y}) \vee q^u(\overline{y}))\right)$$

According to Definition 3.1, we get:

$$(\exists\overline{y})(\exists x^i)\left(p^u(x^i, f^u(\overline{y}, x^i)) \wedge q^u(\overline{f}^u(\overline{y}, x^i)) \wedge (p^u(\overline{y}) \vee q^u(\overline{y}))\right)$$

where, again, f^u and $\overline{f}^u$ are different function symbols, individual and sequence function symbols, respectively, both with the flexible arity.

Theorem 3.4: *(Soundness). If A is valid, then $sk(A)$ is also valid.*

Proof: Proceed by induction on the number of strong quantifiers in A. The base case is trivial: $A = sk(A)$.

Assume, that the theorem holds for all formulae with at most n strong quantifiers and A contains $n+1$ strong quantifiers. Then we select the first strong quantifier (Qx), and assume it is in the scope of weak quantifiers $(Q_1x_1), \dots, (Q_mx_m), m \geq 0$ in A. We shift those quantifiers to the front using quantifier shifting rules, then we get an equivalent formula of A, $(\exists x_1)\dots(\exists x_m)(\forall x)B$, where $B = A_{-(Qx)(Q_1x_1)\dots(Q_mx_m)}$.[1]

By Definition 3.1, $sk((\exists x_1)\dots(\exists x_m)(\forall x)B) = sk((\exists x_1)\dots(\exists x_m)B\sigma)$, where σ is a substitution built according to Definition 3.1 (case distinction on $x, m, x_1, \dots, x_m$, not relevant for this proof).

By the induction hypothesis, if $(\exists x_1)\dots(\exists x_m)B\sigma$ is valid, then $sk((\exists x_1)\dots(\exists x_m)B\sigma)$ is valid too. Thus, it is sufficient to show that $(\exists x_1)\dots(\exists x_m)(\forall x)B \rightarrow (\exists x_1)\dots(\exists x_m)B\sigma$, which is trivially valid. □

Theorem 3.5: *(Completeness). If $sk(A)$ is valid, then A is also valid.*

Proof: Proceed by induction on the number of strong quantifiers in A. The base case is trivial: $sk(A) = A$.

Assume, that the theorem holds for all formulae with at most n strong quantifiers and A contains $n+1$ strong quantifiers. Then we select the first strong quantifier (Qx) and distinguish the following cases:

- (Qx) is not in the scope of weak quantifiers in A. Then, by Definition 3.1 $sk(A) = sk(A_{-(Qx)}\sigma)$, where σ is either $[x \mapsto f^r]$ (if $x \in \mathcal{V}_i$) or $[x \mapsto \overline{f}^r]$ (if $x \in \mathcal{V}_s$), for f^r (or $\overline{f}^r$), with arity 0, not occurring in A. By the induction hypothesis, if $sk(A_{-(Qx)}\sigma)$ is valid, then $A_{-(Qx)}\sigma$ is valid too. Thus, it is sufficient to show that if $A_{-(Qx)}\sigma$ is valid, then A is also valid.

  Note that A is equivalent to A': $(\forall x)A_{-Q(x)}$ (using quantifier shifting rules (Qx) is shifted to front) and assume that A' (and thus A) is not valid. Then there is an interpretation Γ, that falsifies A'. By the standard definition of interpretations, there is also an interpretation Γ' that differs from Γ in the interpretation of x and falsifies A'. Since f^r (or $\overline{f}^r$) does not occur in A, we can extend Γ' to Γ'', where x is interpreted as f^r (or $\overline{f}^r$). Then Γ'' falsifies $A_{-(Qx)}\sigma$, implying that $A_{-(Qx)}\sigma$ is not valid either.
- (Qx) is in the scope of weak quantifiers $(Q_1x_1), \dots, (Q_mx_m), m > 0$ in A. We shift those quantifiers in front using quantifier shifting rules, then we get an equivalent formula of A, A': $(\exists x_1)\dots(\exists x_m)(\forall x)B$, where $B = A_{-(Qx)(Q_1x_1)\dots(Q_mx_m)}$. By Definition 3.1, $sk(A') = sk((\exists x_1)\dots(\exists x_m)B\sigma)$, where $\sigma = [x \mapsto f(x_1, \dots, x_m)]$,

[1]Note that weak quantifiers become existential ones and strong quantifiers become universal ones when shifted in front.

and f is either in $\mathcal{F}_i^r, \mathcal{F}_i^u, \mathcal{F}_s^r$ or $\mathcal{F}_s^u$ depending on $x, x_1, \ldots, x_m$. Without loss of generality we assume that $f \in \mathcal{F}_s^u$, i.e., $x \in \mathcal{V}_s$, and there is $k \in \{1, \ldots, m\}$, s.t., $x_k \in \mathcal{V}_s$; the other cases are similar.

By the induction hypothesis, if $sk((\exists x_1) \ldots (\exists x_m) B\sigma)$ is valid, then $(\exists x_1) \ldots (\exists x_m) B\sigma$ is valid too. Thus, it is sufficient to show that if $(\exists x_1) \ldots (\exists x_m) B\sigma$ is valid, then A' (and thus A) is also valid.

Again, assume that A' is not valid, then there is an interpretation Γ that falsifies A' and is a model of $\neg A'$: $(\forall x_1) \ldots (\forall x_m)(\exists x) \neg B$. By axiom of choice, there is a function $\overline{g}^u$ of $x_1, \ldots, x_m$ to assign to x, and since f does not occur in A, we can take $\overline{g}^u = f$ and construct an interpretation Γ', extension of Γ, that will be a model of $(\forall x_1) \ldots (\forall x_m) \neg B\sigma$, thus falsify $(\exists x_1) \ldots (\exists x_m) B\sigma$. This proves the theorem.

□

Corollary 3.6: *A is sat-equivalent to $\neg sk(\neg A)$, i.e., A is satisfiable iff $\neg sk(\neg A)$ is.*

Proof: Direct consequence of theorems 3.4 and 3.5. □

It is easy to see, that different skolemization methods produce formulae of the similar length (the number of symbols) and logical complexity (the number of logical connectives). In [5] it was shown, that in terms of proof complexity, the particular form of skolemization actually matters, since it might destroy some information encoded inside a formula. The complexity analysis is based on the notion of *Herbrand complexity* – the minimum size of Herbrand disjunction of a formula. The detailed discussion can be found in [1, 5]. The complexity results obtained in [1, 5] can be extended to unranked logics in a straightforward way.

To summarize the mentioned results, as it was stated in [1], if a given formula is already in the prenex form, then *"antiprenex skolemization may give a non-elementary speed-up over the structural one"* during the proof search. Thus, skolemization should be considered as *"an integral part of the inference process and not as a preprocessing step of minor importance"*.

4. Conclusion

We have investigated skolemization procedure for first-order unranked formulas. We defined the algorithm and proved that it is sound and complete. Although we did not give a formal proof, it is easy to see, that the complexity relationship between different kind of skolemization procedures hold for unranked formulas as well.

Aknowledgements

This work was supported by the Shota Rustaveli National Science Foundation project PhD_F_17_218.

References

[1] M. Baaz, U. Egly, A. Leitsch, *Normal form transformations. In A. Robinson and A. Voronkov, editors,* Handbook of Automated Reasoning, North Holland, **I**, 5 (2001), 273-333

[2] HJ. Bürckert, B. Hollunder, A. Laux, *On skolemization in constrained logics*, Annals of Mathematics and Artificial Intelligence, Springer, **18**, 2 (1996), 95-131

[3] M. Baaz, R. Iemhoff, *The Skolemization of existential quantifiers in intuitionistic logic.* Annals of Pure and Applied Logic, Elsevier, **142**, 1-3 (2006), 269-295

[4] M. Baaz, R. Iemhoff, *On Skolemization in constructive theories*, The Journal of Symbolic Logic, Cambridge University Press, **73**, 3 (2008), 969-998

[5] M. Baaz, A. Leitsch., *On skolemization and proof complexity*, Fundamenta Informaticae, **20**, 4 (1994), 353-379

[6] M. Baaz, G. Metcalfe, *Herbrand theorems and Skolemization for prenex fuzzy logics*, In Conference on Computability in Europe, Springer, Berlin, Heidelberg, 2008, 22-31

[7] M. Baaz, G. Metcalfe, *Herbrand's Theorem, Skolemization and Proof Systems for First-Order Lukasiewicz Logic*, Journal of Logic and Computation, Oxford University Press, **20**, 1 (2010), 35-54

[8] B. Buchberger, T. Jebelean, T. Kutsia, A. Maletzky, W. Windsteiger, *Theorema 2.0: Computer-Assisted Natural-Style Mathematics*, Journal of Formalized Reasoning, **9**, 1 (2016), 149-185

[9] S.R. Buss, *On Herbrand's theorem*, In Logic and Computational Complexity, LNCS, Springer, Berlin, Heidelberg, **960** (1995), 195-209

[10] S.R. Buss, (editor). *Handbook of Proof Theory*, Elsevier, 1998

[11] J. Coelho and M. Florido, *CLP(Flex): Constraint Logic Programming applied to XML processing*, In Robert Meersman and Zahir Tari, editors, CoopIS/DOA/ODBASE (2), Springer, **3291** (2004), of *LNCS*, 1098–1112

[12] J. Coelho and M. Florido, *Veriflog: A constraint logic programming approach to verification of website content*, In Heng Tao Shen, Jinbao Li, Minglu Li, Jun Ni, and Wei Wang, editors, Advanced Web and Network Technologies, and Applications, APWeb 2006 International Workshops: XRA, IWSN, MEGA, and ICSE, Harbin, China, January 16-18, 2006, Proceedings, Lecture Notes in Computer Science, Springer, **3842** (2006), 148-156

[13] J. Coelho, M. Florido, *XCentric: logic programming for XML processing*, In Irini Fundulaki and Neoklis Polyzotis, editors, 9th ACM International Workshop on Web Information and Data Management (WIDM 2007), Lisbon, Portugal, November, ACM, **9**, (2007), 1-8

[14] J. Coelho, M. Florido, T. Kutsia, *Sequence disunification and its application in collaborative schema construction*, In Mathias Weske, Mohand-Said Hacid, and Claude Godart, editors, WISE Workshops, Lecture Notes in Computer Science, Springer **4832** (2007), 91-102

[15] J. Coelho, M. Florido, T. Kutsia, *Collaborative schema construction using regular sequence types*, In Proceedings of the IEEE International Conference on Information Reuse and Integration, IRI 2009, 10-12 August 2009, Las Vegas, Nevada, USA, pages 290–295. IEEE Systems, Man, and Cybernetics Society, 2009

[16] H. Delugach, *Common Logic (CL) – A Framework for a Family of Logic-Based Languages*, ISO/IEC 24707 Information Technology Standard, 2007

[17] B. Dundua, L. Kurtanidze, M. Rukhaia. *Unranked Tableaux Calculus and its Web-related Applications*, IEEE Conference on Electrical and Computer Engineering, IEEE Xplore Digital Library, 2017, 1181-1184

[18] M. Genesereth, *Knowledge Interchange Format*, Draft proposed American National Standard, 1998, Available at `http://logic.stanford.edu/kif/dpans.html`

[19] M. Ginsberg, *The MVL theorem proving system*, SIGART Bull., **2**, 3 (1991), 57-60

[20] J. Goubault. *A BDD-based simplification and skolemization procedure*, Logic Journal of the IGPL, Oxford University Press, **3**, 6 (1995), 827-855

[21] I. Horrocks, A. Voronkov, *Reasoning support for expressive ontology languages using a theorem prover*, In Jürgen Dix and Stephen J. Hegner, editors, Foundations of Information and Knowledge Systems, 4th International Symposium, FoIKS 2006, Budapest, Hungary, February 14-17, 2006, Proceedings, Lecture Notes in Computer Science, Springer, **3861** (2006), 201-218

[22] T. Kutsia, B. Buchberger, *Predicate Logic with Sequence Variables and Sequence Function Symbols*, In: A. Asperti, G. Bancerek, and A. Trybulec, editors, Proceedings of the 3rd International Conference on Mathematical Knowledge Management, LNCS, Springer, **3119** (2004), 205-219

[23] T. Kutsia, *Equational prover of THEOREMA*, In Robert Nieuwenhuis, editor, *Rewriting Techniques and Applications, 14th International Conference, RTA 2003, Valencia, Spain, June 9-11, 2003, Proceedings*, Lecture Notes in Computer Science, **2706** (2003), 367-379

[24] T. Kutsia, *Solving Equations Involving Sequence Variables and Sequence Functions*, In: Buchberger B., Campbell J. (eds) Artificial Intelligence and Symbolic Computation, LNCS, Springer, Berlin, Heidelberg, **3249** (2004), 157-170

[25] T. Kutsia, M. Marin, *Can context sequence matching be used for querying XML?* In Laurent Vigneron, editor, Proceedings of the 19th International Workshop on Unification UNIF'05, Nara, Japan, 22 April, 2005, 77-92

[26] L.C. Paulson. *Isabelle: The next 700 theorem provers*, In Logic and Computer Science, Academic Press, 1990, 361-386

[27] G. Van den Broeck, W. Meert, A. Darwiche, *Skolemization for weighted first-order model counting.* In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR), 2014, 1-10