Reports of Enlarged Session of the
Seminar of I. Vekua Institute
of Applied Mathematics
Volume 29, 2015

ABOUT CORRESPONDENCE BETWEEN PROOF SCHEMATA AND UNRANKED LOGICS

Kurtanidze L., Rukhaia M.

Abstract. In this paper we study correspondence between proof schemata and unranked logics. Proof schemata is a new formalism, an alternative to inductive reasoning, where cut-elimination theorem holds. Unranked logics are very important formalisms used in knowledge representation and semantic web. We describe a transformation, how an unranked logic sentence can be encoded into a formula schema.

Keywords and phrases: Unranked logics, schemata, calculi

AMS subject classification (2000): 03B70, 68T15.

1. Introduction. The proof theory takes its roots from G. Gentzen, when he introduced a sequent calculus for first-order logic. Since then, proofs are heavily used in computer science. It is well known that first-order logic is undecidable, therefore all complete proof-search procedures are non-terminating.

The concept of *term schematization* was introduced in [2] to avoid non-termination in symbolic computation procedures and to give finite descriptions of infinite derivations. Later, *formula schemata* for propositional logic was developed [1] to deal with schematic problems (graph coloring, digital circuits, etc.) in a more uniform way. In [3,5] the language of formula schemata was extended to first-order logic and a sequent calculus was defined, introducing a notion of *proof schema*.

Other very expressive formalisms used in computer science are unranked languages, which have unranked alphabet, i.e. function and/or predicate symbols do not have a fixed arity. Since such languages can naturally model XML documents and operations over them, they are more and more often used for knowledge representation and semantic web. Thus increasing demand for designing and improving deduction methods that would permit to automatize reasoning in unranked languages.

It is easy to see similarities between schematic formulas and unranked formulas, defined in [4]. Therefore the question rises: whether it is possible to use proof schemata for knowledge representation. To tackle this problem we try to find correspondence between these two formalisms. It appears, that every unranked formula can be encoded as a formula schema. In this paper we present such an encoding.

2. Proof schemata. In [3,5] we presented a language of *first-order schemata*, which allows us to specify an (infinite) set of first-order formulas by a finite term. There are distinction between *constant function/predicate symbols* and *defined function/predicate symbols*. While the first defines usual first-order terms and predicates, the latter allows primitive recursion on them.

The sequent calculus LKS we consider is given in Figure 1. The *proof axioms* (the pax rule), which are called *proof links* in [3,5], may appear only at the leaves of a proof.

A proof axiom has similar meaning to induction hypothesis, leading to notion of *proof schema*: a tuple of pairs of LKS proofs corresponding to the base and recursive cases of inductive definition. For a formal definition of proof schemata we refer an interested reader to [3,5].

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}\neg_l \qquad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\neg_r \qquad \frac{\Gamma \vdash \Delta, A \quad B, \Gamma \vdash \Delta}{A \Rightarrow B, \Gamma \vdash \Delta}\Rightarrow_l \qquad \frac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \Rightarrow B}\Rightarrow_r$$

$$\frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\wedge_l \qquad \frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\wedge_r \qquad \frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta}\vee_l \qquad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}\vee_r$$

$$\frac{A(t), \Gamma \vdash \Delta}{\forall x A(x), \Gamma \vdash \Delta}\forall_l \qquad \frac{\Gamma \vdash \Delta, A(u)}{\Gamma \vdash \Delta, \forall x A(x)}\forall_r{}^1 \qquad \frac{A(u), \Gamma \vdash \Delta}{\exists x A(x), \Gamma \vdash \Delta}\exists_l{}^1 \qquad \frac{\Gamma \vdash \Delta, A(t)}{\Gamma \vdash \Delta, \exists x A(x)}\exists_r$$

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}c_l \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}c_r \qquad \frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}w_l \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}w_r$$

$$\frac{}{a \vdash a}\mathsf{ax} \qquad \frac{\varphi(k, \bar{t})}{S(k, \bar{t})}\mathsf{pax}\,{}^2 \qquad \frac{S[t]}{S[t']}\mathcal{E}\,{}^3$$

[1] u is a variable, called an eigenvariable, of appropriate sort not occurring in $\Gamma, \Delta, A(x)$.
[2] φ is a proof symbol and S is a sequent where free variables are replaced by terms $\bar{t}$.
[3] Given a finite set of equations $\mathcal{E}$ with $\mathcal{E} \models t = t'$.

Fig.1. The sequent calculus LKS

3. Unranked Logic. Unranked languages are based on the unranked alphabets where functional and predicate symbols do not have a fixed arity. Variable and function symbols are divided into two groups: *individual symbols* (denoted by small Latin symbols) and *sequence symbols* (denoted by Latin symbols with a bar). Both groups include a fixed and flexible arity (unranked) function symbols. There is also distinction between fixed and flexible arity predicate symbols.

The terms are built in a standard inductive way, using individual as well as sequence variable and function symbols. The only restriction is that the fixed arity function symbols can be applied only to the individual terms.

The atoms are built in a standard way using predicate symbols and terms. The same restriction applies here as well: the fixed arity predicate symbols can be applied only to the individual terms. Finally, formulas are built in a standard way from atoms, logical operators and quantifiers.

Following [4], we define a sequent calculus LKU, which is obtained from LK by adding rules for quantifiers over sequence variables. The rules are given in Figure 2.

4. Transformation. To encode an unranked formula into a formula schema, it is enough to encode sequence terms into term schemata. The rest of encoding is straightforward.

Let $\ulcorner t_1, \ldots, t_n \urcorner$ be a sequence term. Let g be a binary function symbol. Then $\ulcorner t_1, \ldots, t_n \urcorner \overset{def}{\equiv} g(\ldots g(t_1, t_2), \ldots), t_n)$ and we can construct a corresponding term schema:

$$\widehat{g}(1) \to t_1 \qquad \text{and} \qquad \widehat{g}(n+1) \to g(\widehat{g}(n), t_{n+1})$$

$\mathsf{LKU} = (\mathsf{LKS} \setminus \{\mathsf{pax}, \mathcal{E}\}) \cup \{\forall^{\mathsf{u}}_{\mathsf{l}}, \forall^{\mathsf{u}}_{\mathsf{r}}, \exists^{\mathsf{u}}_{\mathsf{l}}, \exists^{\mathsf{u}}_{\mathsf{r}}, \approx\}$, where:

$$\frac{A(\ulcorner t_1, \ldots, t_n \urcorner), \Gamma \vdash \Delta}{\forall \bar{x} A(\bar{x}), \Gamma \vdash \Delta}\forall^{\mathsf{u}}_{\mathsf{l}} \qquad \frac{\Gamma \vdash \Delta, A(\bar{v})}{\Gamma \vdash \Delta, \forall \bar{x} A(\bar{x})}\forall^{\mathsf{u}\,1}_{\mathsf{r}}$$

$$\frac{A(\bar{v}), \Gamma \vdash \Delta}{\exists \bar{x} A(\bar{x}), \Gamma \vdash \Delta}\exists^{\mathsf{u}\,1}_{\mathsf{l}} \qquad \frac{\Gamma \vdash \Delta, A(\ulcorner t_1, \ldots, t_n \urcorner)}{\Gamma \vdash \Delta, \exists \bar{x} A(\bar{x})}\exists^{\mathsf{u}}_{\mathsf{r}}$$

$$\frac{\Gamma, (s \approx t \wedge a[s]) \Rightarrow a[t] \vdash \Delta}{\Gamma \vdash \Delta}\approx$$

[1] $\bar{v}$ is a sequence variable, called an eigenvariable, not occurring in $\Gamma, \Delta, A(\bar{x})$.

Fig.2. The sequent calculus LKU

It is also possible to transform an LKU derivation into an LKS derivation. Unranked quantifier rules will match LKS quantifier rules after transforming sequence terms into term schemata. It remains to transform the $\approx$ rule into LKS derivation.

In [3,5] it was shown that LKS $\equiv$ LKS $\cup$ {cut}, where the cut rule is:

$$\frac{\Gamma \vdash \Delta, A \qquad A, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}$$

$\approx$ rule can be derived in LKS $\cup$ {cut}, thus, in LKS as well. The derivation of $\approx$ in LKS $\cup$ {cut} is given in Figure 3.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a[s] \vdash a[s]}{s \approx t, a[s] \vdash a[s]}\mathsf{w}_{\mathsf{l}}}{s \approx t, a[s] \vdash a[t]}\mathcal{E}}{s \approx t \wedge a[s] \vdash a[t]}\wedge_{\mathsf{l}}}{\vdash (s \approx t \wedge a[s]) \Rightarrow a[t]}\Rightarrow_{\mathsf{r}} \qquad \Gamma, (s \approx t \wedge a[s]) \Rightarrow a[t] \vdash \Delta}{\Gamma \vdash \Delta}\mathsf{cut}$$

Fig.3. Derivation of $\approx$ rule in LKS

5. Conclusion. In this paper we studied correspondence between proof schemata and unranked logics. A transformation from LKU derivation to LKS derivation was presented. Thus, every unranked formula can be represented as a formula schema. Although in [4] authors presented an induction rule for unranked formulas, still transforming a formula schema into an unranked formula is not feasible.

Acknowledgment. This work was supported by the project No. FR/51/4-102/13 of the Shota Rustaveli National Science Foundation.

R E F E R E N C E S

1. Aravantinos V., Caferra R., Peltier N. A schemata calculus for propositional logic. *In Tableaux'09, Lecture Notes in Computer Science*, **5607** (2009), 32-46.

2. Chen H., Hsiang J., Kong H.C. On finite representations of infinite sequences of terms. In Kaplan S., and Okada M., editors, *Conditional and Typed Rewriting Systems, Lecture Notes in Computer Science*, **516** (1991), 99-114.

3. Dunchev C., Leitsch A., Rukhaia M., Weller D. Cut-elimination and proof schemata. In Aher M., Hole D., Jeřábek E., and Kupke C., editors, *Logic, Language, and Computation, Lecture Notes in Computer Science*, **8984** (2015), 117-136.

4. Kutsia T., Buchberger B. Predicate logic with sequence variables and sequence function symbols. In Asperti A., Bancerek G., and Trybulec A., editors, *Mathematical Knowledge Management, Lecture Notes in Computer Science*, **3119** (2004), 205-219.

5. Rukhaia M. About Cut-Elimination in Schematic Proofs. A monograph. *Lambert Academic Publishing, Saarbrücken*, 2013.

Received 21.05.2015; revised 30.11.2015; accepted 16.12.2015.

Authors' addresses:

L. Kurtanidze
Faculty of Informatics, Mathematics and Natural Sciences of
Georgian University
53a, I. Chavchavadze Av., Tbilisi 0179
Georgia
E-mail:lia.kurtanidze@gmail.com

M. Rukhaia
I. Vekua Institute of Applied Mathematics of
Iv. Javakhishvili Tbilisi State University
2, University St., Tbilisi 0186
Georgia
E-mail: mrukhaia@logic.at