



Ukrainian Logic Society

2ND WORLD LOGIC DAY

JANUARY 14, 2020

Logic and its Applications

The workshop

Book of Abstracts

Kyiv
2020

Ukrainian Logic Society

2nd World Logic Day – January 14, 2020. Logic and its Applications, The workshop. Book of Abstracts. Kyiv, 2020. – 25 p.

Українське логічне товариство

2-ий Всесвітній день логіки – 14 січня 2020 року. Логіка та її застосування. Семінар. – Збірка тез. Київ, 2020. – 25 с.

Українське логічне товариство проводить науковий семінар з логіки та її застосувань 14 січня кожного року, оскільки на цю дату за рішенням ЮНЕСКО припадає відзначення Всесвітнього дня логіки. За результатами семінару готується збірка тез. У збірку тез семінару 2020 року включено тези 14 доповідей, що стосуються математичних, філософських, історичних та дидактичних аспектів логіки. Для фахівців у галузі логіки та суміжних дисциплін.

Contents • Зміст

З. С. Х. Аль-Хіلالі, В. П. Шевченко. Логічні засади управління змінами у навчальному процесі	4
Ігор Дуцяк. Види істинності	5
Ievgen Ivanov. On generalized well-founded induction	7
Yaroslav Kokhan. Inductive Definitions via the Single Formula	8
Kryvyi S.L., Grinenko O.O. Logical approach to the research of properties of software engineering ecosystem	9
Alexander Lyaletski. Automated Theorem Proving in Kyiv: Historical Notes	10
Володимир Навроцький. Прийняття рішень і правдоподібні аргументи	11
Нікітченко Микола Степанович. Логіки Флойда–Хоара для рекурсивних програм та часткових умов	12
М. С. Нікітченко, С. С. Шкільняк. Секвенційні числення першопорядкових логік часткових предикатів з композицією предикатного доповнення	13
Омельчук Людмила Леонідівна, Шишацька Олена Володимирівна. Номінативні дані та розв'язність формул	15
Nataliia Rusina. The examples of natural language formalization in the course “mathematical logic”	16
О. С. Шкільняк. Відношення логічного наслідку в логіках часткових предикатів з композицією предикатного доповнення	18
С. С. Шкільняк. Першопорядкові логіки часткових предикатів з розширеними реномінаціями та предикатами рівності	20
Юркевич Олена Миколаївна, Павленко Жанна Олександрівна. Формування нової навчальної програми з логіки для юристів з урахуванням впровадження ТЗНПК	21

Логічні засади управління змінами у навчальному процесі

З. С. Х. Аль-Хіلالі, В. П. Шевченко

Київський національний університет імені Тараса Шевченка

Сучасний світ характеризується великою кількістю змін, що відбуваються щільно в часі. Разом з тим, навчання залишається важливим компонентом становлення освіченої людини, адаптованої до викликів сучасності. Тому врахування повсякчасних змін та адаптація до них навчального процесу є важливою складовою менеджменту, особливо — у e-Learning.

Щодо класичного навчання з учителем чи наставником (тобто, offline-навчання у класах), основні зміни стосуються розкладу (в рамках періоду навчання — семестру, чверті, півріччя тощо) чи набору та часових характеристик предметів (у порівнянні років, між періодами). Перші зміни керуються логікою несуперечливості правил, які накладають обмеження на множину допустимих розкладів, якщо розглядати зміну як функцію з одного розкладу в інший. Щодо змін між періодами — важливо, аби логіка змін наслідувала повноту (покриття матеріалу, матриці навичок та компетентностей), несуперечливість (мінімізація перетинів матеріалу, тощо), та послідовність (міжпредметні зв'язки). Особливо останнє важливо для предметів вільного вибору студентів.

Що стосується управління навчанням (learning curve) для заочних чи інших віддалених форм взаємодії — тут важливим є також слідування логіці несуперечливості, повноти та послідовності. Повнота особливо важлива у випадках наборів курсів (nano degrees, що стають вкрай популярними сьогодні), які дають уяву про певну предметну галузь, що, як правило, знаходиться на перетині кількох класичних дисциплін.

У випадку змін у таких (або більш довгих, класичних) курсах та їх наборах, важливо слідувати, знов-таки, логіки несуперечливості та повноти, враховуючи заздалегідь обумовлену послідовність подання знань.

У сучасних системах управління навчанням (Learning Management Systems) не часто робиться акцент на ці логічні аспекти, не зважаючи на їх значущість. Натомість, розроблена авторами і впроваджена система e-Learning враховує означені засади, роблячи акцент на дотриманні зафіксованих правил, згаданих вище. Так, одна з важливих функцій — “безпечна” заміна у розкладі, одноразова або на регулярній основі, виконана з урахуванням всіх необхідних обмежень та застосованих правил.

Оскільки система вже пройшла дослідну експлуатацію, впроваджена у кількох школах, та має позитивні відгуки про досвід користування, можемо стверджувати, що було розв'язано важливу задачу управління навчальним процесом з мінімізацією людського втручання. Підґрунтям цієї розробки постав логічний апарат, який допоміг:

- 1) автоматизувати відповідальні ділянки роботи,
- 2) як наслідок, зменшити людський фактор у прийнятті рішень,
- 3) систематизувати деякі аспекти управління навчальним процесом і зробити саме управління більш зручним і менш підвласним помилкам (у першу чергу, логічного характеру).

Отже, практична складова роботи підкреслює важливість та застосовність окреслених логічних аспектів та засад в управлінні навчальним процесом, зокрема — змінами у ньому.

Види істинності

Ігор Дуцяк

Національний університет "Львівська політехніка"

iduntsyak@gmail.com

Для детального логічного аналізу знань потрібні такі види істинності:

1. Об'єктивна фактична істинність — ознака вислову, яка проявляється в тому, що зафіксоване в ньому знання узгоджене з дійсністю (це об'єктивно фактично істинний вислів), або не узгоджене з дійсністю (це об'єктивно фактично хибний вислів). Наприклад, вислів "Спостережуваний рух Сонця є обертання навколо Землі" є об'єктивно фактично істинним. Об'єктивно фактично істинний вислів позначатимемо символом $F_o^t(p)$, тобто: фактично (F) об'єктивно (індекс o) істинний (індекс t) вислів p . Об'єктивно фактично хибний вислів позначатимемо символом $F_o^f(p)$, де індексом f позначено хибність вислову p .

2. Подана суб'єктом (декларована) фактична істинність вислову. Суб'єкт може подавати вислів з іншим значенням істинності аніж він приймає (це називаємо брехнею), і, крім того, подане мовцем значення фактичної істинності не пов'язане з об'єктивною фактичною істинністю. Суб'єкт може подати фактичну істинність трьома способами:

2.1. Логічними словами чи словосполученнями (такими як “і”, “або”, “неправда, що” та ін.), які в логіці висловів формально записують булевими функціями.

2.2. Стверджувальною формою вислову. Кожне стверджувальне речення має в письмовому мовленні останнім символом крапку, якою позначають факт ствердження, тобто подання вислову як істинного. Подібним знаком в усному мовленні є спадна інтонація. Якщо зміст слів “неправда, що” ми позначаємо булевою функцією “заперечення” (негація), то, подібно, зміст слів “правда, що” (тобто “істинно, що”) також доцільно позначати булевою функцією. Якщо зміст логічного терміна “неправда, що” записуємо так: $\neg p: p = 1, \neg p = 0; p = 0, \neg p = 1$, то зміст логічного терміна “правда, що” запишемо $\vdash p: p = 1, \vdash p = 1; p = 0, \vdash p = 0$.

2.3. Змістом вислову. Той факт, що змістом (поданого як фактично істинний) вислову p_i є фактична істинність вислову p_j позначимо квадратними дужками: $\vdash p_i [\vdash p_j]$. Парадокс брехуна: $\vdash p_i [\neg p_j]$ — другим і третім способами вислів p_i позначено з різними значеннями істинності.

3. Прийнята суб'єктом фактична істинність — це оцінка фактичної істинності вислову, яку суб'єкт мовлення приймає як коректну. Ця оцінка фактичної істинності не пов'язана з об'єктивною фактичною істинністю — вислів може бути об'єктивно фактично істинним, а суб'єкт сприйматиме його як об'єктивно фактично хибний (це, власне, випадок омани). Наприклад, суб'єкт вважає, що дійсним рухом Сонця є обертання навколо Землі, тоді як насправді це є не так. Може бути також навпаки — суб'єкт може сприймати якийсь вислів як фактично істинний, тоді як він є фактично хибним. Вислів, який суб'єкт приймає як фактично істинний, позначимо $F_s^t(p)$. Подібно, вислів, який суб'єкт приймає як фактично хибний, позначимо $F_s^f(p)$. У такому разі оманливі вислови, які людина не оприлюднила (мовлені подумки), можна записати так: 1) $F_o^f F_s^t(p)$, тобто, те, що насправді (об'єктивно) є фактично хибним (вислів p), суб'єкт подумки приймає як фактично істинне, або 2) $F_o^t F_s^f(p)$, тобто, навпаки, — те, що є об'єктивно фактично істинним (вислів p), суб'єкт сприймає як фактично хибне. Приклади запису об'єктивно істинного вислову промовленого як брехня: $F_o^t F_s^t \neg(p)$ — прийняте суб'єктом як істинне (F_s^t), він подає як хибне (\neg); і навпаки — $F_o^f F_s^f \neg(p)$.

On generalized well-founded induction

Ievgen Ivanov

Taras Shevchenko National University of Kyiv

ivanov.eugen@gmail.com

Software specification and verification using interactive proof assistants, such as Isabelle and Coq, frequently depends on inductive definitions and proofs. In this context, inductive proofs can usually be considered as instances of application of the well-founded induction principle. However, induction principles, different from well-founded induction, appear in some domain-specific logics for analysis and verification of cyber-physical systems, mathematical models of which combine discrete and continuous components. They can also be applied to formalization and analysis of such systems in interactive proof assistants.

An important principle of this sort is Raoult's open induction [1]. Although a formalization of this principle is available e.g. in Isabelle/HOL library, practical application of it to discrete-continuous system formalization and verification in proof assistants is much less obvious and straightforward than, e.g. application of inductive proofs for inductive data types to formalization and verification of regular (discrete) sequential programs. Besides, no specific mechanism for constructing types (concrete dcpos) to which it is applicable and unfolding its formulation for such types is present.

Thus investigation of type systems that ensure convenience of construction of types important for discrete-continuous dynamical systems formalization to which the open induction principle is applicable, and convenience of application of this principle to such types, is a relevant research question.

In this talk we consider this question on semantic level and propose some categories which can serve as semantics domains for such type systems, and show that for such categories, open induction can be associated with certain weak forms of the notion of a well-founded coalgebra [2, 3].

References

1. Raoult J.-C. Proving open properties by induction / J.-C. Raoult // Information Processing Letters 29. – 1988. – P. 19-23.
2. Taylor P. Practical foundations of mathematics / P. Taylor. – Cambridge University Press. – 1999.

3. Adamek J. Well-pointed coalgebras / J. Adamek, S. Milius, L.S. Moss, L. Sousa // Logical Methods in Computer Science. – Vol. 9. – 2013. – P. 1-5.

Inductive Definitions via the Single Formula

Yaroslav Kokhan

Institute of Philosophy, National Academy of Sciences, Kyiv, Ukraine

yarkaen@gmail.com

There is still no comprehensive theory of induction in logic. As a result, definitions by induction (inductive or recursive definitions) are described incorrectly in some works, including handbooks (authors do not formulate the inductive finalization).

The structure of the inductive definition (F) of an adequate predicate $F(x_0, x_1, \dots, x_n)$ is following:

The *inductive basis*:

$$(F_0 1) A_1(x_0, x_1, \dots, x_n) \rightarrow F(x_0, x_1, \dots, x_n);$$

⋮

$$(F_0 k) A_k(x_0, x_1, \dots, x_n) \rightarrow F(x_0, x_1, \dots, x_n).$$

The *inductive clause*:

$$(F_1 1) \exists y_1 \dots \exists y_l \left(\bigwedge_{1 \leq i \leq l} F(y_i, x_1, \dots, x_n) \wedge R_1(x_0, y_1, \dots, y_l) \right) \rightarrow F(x_0, x_1, \dots, x_n);$$

⋮

$$(F_1 m) \exists y_1 \dots \exists y_l \left(\bigwedge_{1 \leq i \leq l} F(y_i, x_1, \dots, x_n) \wedge R_m(x_0, y_1, \dots, y_l) \right) \rightarrow F(x_0, x_1, \dots, x_n).$$

The *inductive finalization*:

$$(F_\infty) \bigwedge_{1 \leq i \leq k} \neg A_i(x_0, x_1, \dots, x_n) \wedge \forall y_1 \dots \forall y_l \left(\bigwedge_{1 \leq j \leq l} F(y_j, x_1, \dots, x_n) \rightarrow \bigwedge_{1 \leq e \leq m} \neg R_e(x_0, y_1, \dots, y_l) \right) \rightarrow \neg F(x_0, x_1, \dots, x_n).$$

Variables x_1, \dots, x_n in predicates $A_1^{(n+1)}, \dots, A_k^{(n+1)}$ in inductive basis (F₀) can be apparent; in this case, we have unary predicates, which specify kinds of initial objects. Predicates $R_1^{(n+1)}, \dots, R_m^{(n+1)}$ in inductive clause (F₁) mean the rules, which determine the obtaining new objects by induction from any appropriate specified objects.

It is obvious that the inductive finalization (F_∞) is the contraposition of the conversion of the formula, equivalent to the conjunction of all the formulas of the inductive basis and the inductive clause of (F). Thus the whole inductive definition (F) can be formulated as the following equivalence:

$$F(x_0, x_1, \dots, x_n) \leftrightarrow \left(\bigvee_{1 \leq i \leq k} A_i(x_0, x_1, \dots, x_n) \vee \exists y_1 \dots \exists y_l \left(\bigwedge_{1 \leq j \leq l} F(y_j, x_1, \dots, x_n) \wedge \bigvee_{1 \leq e \leq m} R_e(x_0, y_1, \dots, y_l) \right) \right).$$

This means that

(MAIN THESIS) *Every inductive definition can be formulated via the single formula.*

The last fact allows us

(i) to formalize proofs by induction in logical calculi,

(ii) to solve the problem, whether the proofs by induction require the separate rule of induction or not (i. e., whether the rule of induction can be derived from logical rules or not).

Logical approach to the research of properties of software engineering ecosystem

Kryvyi S. L.

Taras Shevchenko National University of Kyiv

sl.krivoi@gmail.com

Grinenko O. O.

National Aviation University

Elena.Grinenko@livenau.net

High level ecosystem (ES) model is proposed, which is oriented on software engineering domain.

In this scientific paper the following definition of ecosystem is taken into account, which models its computing active part.

Definition 1. Ecosystem (ES) is called a triple $ES=(E, B, S_{BE})$, where

– E is autonomous family (set) of objects, which interact with each other and with environment (kernel or core of ES);

– B is (ontological) knowledge base, where the evolution of family's development is saved as population (macro-knowledge of ES);

– S_{BE} is a server to provide interaction between E, B and environment.

Kernel of ES is presented as an ordered tuple $SYS_E=(S, \Omega, o, f)$, where $S=(S_1, \dots, S_n)$ is finite set, elements of which are called servers; Ω is finite set of operations, which are realized by servers; $o: S \rightarrow B(\Omega)$ is a mapping of the set of servers in the set of subsets (power set $B(\Omega)$) of operations; $f: S \times \Omega \rightarrow N^+$ is a function, values of which are time of execution of certain operations on the certain server, where N^+ is a set positive natural numbers.

Let $\Omega_i = \{C_1^i, C_2^i, \dots, C_{|\Omega_i|}^i\}$ defines a set of operations, which are executed on the i -th server $C_1^i, C_2^i, \dots, C_{|\Omega_i|}^i \in \Omega$ and $f(S_i, K_k^i) = \tau_k^i, i=1, \dots, n, k=1, \dots, |\Omega_i|$. Because the system allows to perform the same operation on different servers, then a time of execution of operation may be different on different servers. In this formulation the planning task will have a large number of implementations in the system *SYS* [1].

It is considered logical approach to analysis of planning task of resources in [2]. While is considered as multi-agent system (MAC). To describe the behavior of agents the following logics are used: temporal logic (TL), epistemic propositional logic (EPL) and dynamic propositional logic (DPL). Semantics of these logics is described using marked transitive system $A = (W, Act, \{R, \{\rightarrow_{p_i}, \sim_i\}\}, I, AP, L)$.

References

1. Telenyk S., Rolik A., Bukasov M., Halushko D. Models and methods of resource management VPS hosting. Technical transaction. Automatic control. Politechnika Krakowska. – 2013. – v.4–AC. – P. 41 – 52.
2. Kryvyi S. L., Pogorilyy S. D., Boyko Y. V. Network Model of IT-infrastructure Resource Management. *Proced. Intern. Conf. 2019 – ATIT IEEE. Tr.2. P.1–5.*

Automated Theorem Proving in Kyiv: Historical Notes

Alexander Lyaletski

Kyiv

a.lyaletski@nubip.com.ua

A brief historical sketch of carrying out the Kyiv's investigations in the automated theorem proving, beginning from 1962, the time of starting the investigations, and ending today is given. The main results in this field and researchers obtained them are listed in the chronological order. At that, connections with some of the similar investigations of other researchers are indicated.

Прийняття рішень і правдоподібні аргументи

Володимир Навроцький

Інститут філософії ім.Г. С. Сковороди НАН України

navrotsk@gmail.com

Значна частина рішень приймається людьми за не повністю визначених обставин. Обґрунтування таких рішень здійснюється зазвичай правдоподібними аргументами, у склад яких входять твердження, які оцінюються не як істинні чи хибні, а як певною мірою прийнятні. Засновкам аргументу приписуються певні міри правдоподібності, наприклад, числові ступені. А висновок аргументу отримує певний ступінь обґрунтованості.

Прийняття висновку правдоподібного аргументу залежить від аргументаційного контексту, наприклад, від взаємодії аргументу із контраргументами, бо такий аргумент є потенційно під питанням і може бути атакований. У ході такої взаємодії усі три структурні компоненти аргументу можуть бути уражені. Наприклад, якщо атаковане правило, тим самим кинутий виклик успішності переходу від прийнятності засновків аргументу до прийнятності його висновку. Тоді виникає потреба обґрунтувати застосування правила виводу.

Отже, успішність обґрунтування висновку аргументу (і відповідного рішення) і раціональність його прийняття залежать від того, як аргумент відповів на такі атаки, тобто, від його сили. Свою остаточну силу аргумент отримує саме через змагання із контраргументами, бо у ході такого змагання його засновки і правило виводу отримують певну вагу. Формально сила аргументу може бути представлена функцією із ваги засновків і ваги правила виводу у ступінь обґрунтованості висновку.

Логіки Флойда–Хоара для рекурсивних програм та часткових умов

Нікітченко Микола Степанович

доктор фіз.-мат. наук, професор

КНУ імені Тараса Шевченка, Київ, Україна

mykola.nikitchenko@gmail.com

Abstract. *Classical Floyd–Hoare logics are not valid for partial conditions (partial predicates). Several methods to solve this problem are proposed. Here we construct Floyd–Hoare logics with partial predicates for recursive programs over hierarchical data. Obtained results can be useful for program analysis and verification.*

Логіки Флойда–Хоара є природним засобом опису властивостей програм та, зокрема, доведення їх коректності. Ці логіки широко використовуються в різних методах розробки програм, наприклад, у контрактному програмуванні, у мовах Dafny, B, RAISE, VDL та інших.

Логіки Флойда-Хоара ґрунтуються на понятті трійки Хоара. Ця трійка складається з передумови, програми та післяумови і означає таку вимогу: коли вхідні дані задовольняють передумові, то результат програми повинен задовольняти післяумову, якщо програма завершується. Класична логіка Флойда-Хоара надає аксіоми та правила виведення для конструкцій простої імперативної мови програмування. Також були розроблені правила для інших мовних конструкцій, зокрема, паралелізму, процедур, переходів, вказівників тощо [1].

У класичній логіці Флойда-Хоара, як і у її розширеннях, перед- та післяумови (предикати над даними, які обробляються програмою) вважаються тотальними (визначеними на всіх даних), а програми можуть бути частковими (у тому сенсі, що якщо програма не завершується, її отримане значення не визначене). Здатність працювати з частковістю є важливим аспектом, оскільки часткові операції часто виникають у програмуванні. Але проблема полягає в тому, що перехід до часткових предикатів порушує коректність класичної логіки Флойда-Хоара [2, 3]. Для розв'язання цієї проблеми було запропоновано декілька підходів [4]:

- 1) введення обмеженого поняття істинності трійки Хоара;
- 2) накладання додаткових обмежень на правила виведення;
- 3) розширення мови предикатів композицією доповнення.

В доповіді розглядаються такі питання:

4) розробка композиційно-номінативних моделей рекурсивних програм над ієрархічними даними;

5) побудова логік Флойда-Хоара з частковими предикатами для таких програм.

Отримані результати можуть бути корисними для аналізу та верифікації програм.

Література

1. Apt, K.: Ten years of Hoare's logic: a survey - part I, ACM Trans. Program. Lang. Syst. 3(4), pp. 431–483 (1981)

2. Kryvolap, A., Nikitchenko, M., Schreiner, W.: Extending Floyd-Hoare logic for partial pre- and postconditions. In: Communications in Computer and Information Science, vol. 412, pp. 355–378. Springer, Cham (2013)

3. Nikitchenko, M., Shkilniak, S.: Algebras and logics of partial quasiary predicates, Algebra and Discrete Mathematics, vol. 23. No 2, pp. 263–278 (2017)

4. Ivanov, I., Nikitchenko, M.: Inference Rules for the Partial Floyd-Hoare Logic Based on Composition of Predicate Complement. In Communications in Computer and Information Science, vol. 1007, pp. 71-88. Springer, Cham (2019)

Секвенційні числення першопорядкових логік часткових предикатів з композицією предикатного доповнення

М. С. Нікітченко, С. С. Шкільняк

Київський національний університет імені Тараса Шевченка

В роботі запропоновано числення секвенційного типу для нових програмно-орієнтованих логічних формалізмів — першопорядкових логік часткових квазіарних предикатів з предикатним доповненням. Характерною особливістю цих логік є наявність спеціальної немонотонної операції (композиції) предикатного доповнення \sim . Операції подібного типу використовуються в різних варіантах логік Флойда-Хоара [1] з частковими перед- та післяумовами. Ми розглядаємо чисті першопорядкові логіки з розширеними реномінаціями та предикатами-індикаторами наявності значення для предметних імен (змінних); такі логіки, збагачені композицією предикатного доповнення, названо L_{\perp}^{QC} . Для випадку *однозначних* часткових предикатів такі логіки названо L_{\perp}^{QCP} .

Числення секвенційного типу формалізують відношення логічного наслідку для множин формул, властивості цих відношень є семантичною основою побудови відповідних числень. Для L_{\perp}^{QC} можна ввести два різновиди таких відношень:

– традиційні відношення $^{Pc}|=_{\mathcal{T}}, ^{Pc}|=_{\mathcal{F}}, ^{Pc}|=_{\mathcal{TF}}, ^{Rc}|=_{\mathcal{T}}, ^{Rc}|=_{\mathcal{F}}, ^{Rc}|=_{\mathcal{TF}}$ ($^{Pc}|=_{\mathcal{IR}}$ та $^{Rc}|=_{\mathcal{IR}}$ некоректні);

– відношення логічного наслідку за умов невизначеності $|=_{\mathcal{IR}}^{\perp}, ^P|=_{\mathcal{T}}^{\perp}, ^P|=_{\mathcal{F}}^{\perp}, ^P|=_{\mathcal{TF}}^{\perp}, ^R|=_{\mathcal{T}}^{\perp}, ^R|=_{\mathcal{F}}^{\perp}, ^R|=_{\mathcal{TF}}^{\perp}$.

Для відношень типу $|=_{\mathcal{T}}$ та типу $|=_{\mathcal{F}}$ маємо різні властивості декомпозиції формул вигляду $\sim\Phi$, які не можна подати як спільну властивість для відношень типу $|=_{\mathcal{TF}}$. Тому секвенційне числення для відношення типу $|=_{\mathcal{TF}}$ є поєднанням двох числень: для відношення типу $|=_{\mathcal{T}}$ та для відношення типу $|=_{\mathcal{F}}$. Згідно теореми про елімінацію умов невизначеності відношення $^P|=_{\mathcal{T}}^{\perp}, ^P|=_{\mathcal{F}}^{\perp}, ^R|=_{\mathcal{T}}^{\perp}, ^R|=_{\mathcal{F}}^{\perp}$ можна звести до відношень $^{Pc}|=_{\mathcal{T}}, ^{Pc}|=_{\mathcal{F}}, ^{Rc}|=_{\mathcal{T}}, ^{Rc}|=_{\mathcal{F}}$.

Таким чином, фактично треба побудувати секвенційні числення для відношення $|=_{\mathcal{IR}}^{\perp}$ в L_{\perp}^{QCP} , для відношень $^{Pc}|=_{\mathcal{T}}$ і $^{Pc}|=_{\mathcal{F}}$ в L_{\perp}^{QCP} та для відношень $^{Rc}|=_{\mathcal{T}}$ і $^{Rc}|=_{\mathcal{F}}$ в L_{\perp}^{QC} .

Першопорядкові секвенційні числення для цих відношень відповідно названо C^{QCP-U} та $C^{QCP-T}, C^{QCP-F}, C^{QC-T}, C^{QC-F}$.

Для кожного із зазначених числень наведено базові секвенційні форми та умови замкненості секвенцій.

Для побудованих секвенційних числень доведено теореми коректності та повноти.

Література

1. C. Hoare. An axiomatic basis for computer programming, Comm. 1969. ACM. 12(10). P. 576–580.
2. M. Nikitchenko, O. Shkilniak, S. Shkilniak. Program Logics Based on Algebras with the Composition of Predicate Complement, in 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, pp. 285–288 (2019).
3. M. Nikitchenko, O. Shkilniak, S. Shkilniak, T. Mamedov. Completeness of the First-Order Logic of Partial Quasiary Predicates with the Complement Composition. Computer Science Journal of Moldova, vol. 27, No 2 (80), 162–187 (2019).

Номінативні дані та розв'язність формул

Омельчук Людмила Леонідівна

*кандидат фіз.-мат. наук, доцент кафедри теорії та технології програмування
Київського національного університету імені Тараса Шевченка*

l.omelchuk@knu.ua

Шишацька Олена Володимирівна

*кандидат фіз.-мат. наук, асистент кафедри теорії та технології програмування
Київського національного університету імені Тараса Шевченка*

shyshatska@knu.ua

Композиційно-номінативний підхід надає єдину методологічну основу для формалізації поняття специфікації програм, доведення їх властивостей, а також для проведення їх подальшої конкретизації до мов програмування більш низького рівня.

На номінативному рівні дані трактуються як номінативні дані. Клас номінативних даних ND побудований на основі множин імен V та значень W можна задати наступним рекурсивним визначенням: $ND = W \cup (V \overset{m}{\underset{-}{ND}}$, де $V \overset{m}{\underset{-}{ND}}$ — клас часткових багатозначних (недетермінованих) функцій.

Для представлення номінативних даних можна використовувати форму $d = [v_i \mapsto a_i \vee i \in I]$, де I деякий набір індексів. Система даних номінативного рівня може бути задана як кортеж $NDS = (V, W, D)$, де $D \subseteq ND$. Кожен із рівнів розгляду множини даних D індукує відповідні йому функції та композиції, які будемо позначати індексом D . Функції мають тип $\overset{m}{\underset{-}{D}}$, а композиції трактуються як n -арні композиції. В [1, 2] побудовано та досліджено теорію номінативних даних. Зокрема, досліджено номінативну обчислюваність та продемонстровано, що будь-яка ЧРФ може бути представлена номінативно обчислюваними функціями над множиною натуральних чисел у номінативних даних.

Для побудови методів верифікації програм важливим завданням є вирішення задачі розв'язності для логічних формул з урахуванням теорій, що лежать в їх основі (Satisfiability Modulo Theories (SMT)). Велика кількість SMT-вирішувачів використовують уніфіковану мову опису задач SMT-LIB, що дозволяє достатньо просто замінити один SMT-вирішувач на інший. SMT-LIB — це міжнародна ініціатива, започаткована в 2003 році, спрямована на сприяння науково-дослідним розробкам та дослідженням SMT-теорій. Наявність загальних стандартів і бібліотек для SMT полегшило оцінку і порівняння

SMT-систем, а також стимулює їх подальший розвиток. У SMT-LIB теорія визначається як клас структур з однаковою сигнатурою. Σ -теорія — це клас Σ -структур. Кожна з цих структур є моделлю теорії. Типові теорії SMT-LIB складаються з єдиної моделі (наприклад, цілих чисел, дійсних чисел) або з класу всіх структур, які відповідають аксіомам теорії.

Незважаючи на те, що теорія номінативних даних представляє інтерес для моделювання та верифікації програмного забезпечення наразі відсутня її підтримка у форматі SMT-LIB. Тому, авторами запропоновано та досліджено теорію номінативних даних для стандарту SMT-LIB 2.6. Теорія номінативних даних може бути представлена як поєднання базових теорій. Проте, авторами запропоновано визначити цю теорію як базову, оскільки вона містить специфіку, яку неможливо отримати модульним поєднанням теорій, а використання спеціально побудованої SMT-теорії дозволить ефективніше здійснювати перевірку виконуваності формул.

The examples of natural language formalization in the course “mathematical logic”

Nataliia Rusina

Taras Shevchenko National University of Kyiv

rusina@knu.ua

Natural languages are used as an effective way of communication. Teaching students to formalize natural language using logical laws and operations in practice is one of the objectives of the course "Mathematical logic" for students of the educational program "Applied Linguistics".

To construct natural language sentences in the expression logic, the algorithm is used: distinguish all simple statements and mark them with propositional variables; connect simple utterances to logical connectors; the resultant formula is written.

The above algorithm will be further used by students to complete a practical assignment.

Here are examples of formalization of sentences for students of philology departments.

Example 1. Write in the language of the logic of the utterances the following sentence: "The sun is shining or the snow is falling and the wind is whistling".

By structure, this sentence is complicated. It consists of several simple statements: "the sun is shining" - denote (a); "snow is falling" - denote (b); "wind is whistling" - denote (c).

Also in this sentence there are linguistic links (grammatical conjunctions): "i", "and" which are denoted by the conjunction " \wedge ", the conjunction "or" by the disjunction " \vee ". As a result of the sentence analysis, the formalized form of the given statement will be: $a \vee b \wedge c$.

Example 2. Formalize the statement in the language of the logic of the statements: "If the Pirates or Giants lose and the Lilliputians win, then Gulliver loses the first place and, in addition, Grogul loses money."

By structure, this sentence is complicated. It is made up of such simple statements that we denote by the propositional variables: "Pirates lose" - (a), "Giants lose" - (b), "Lilliputians win" - (c), "Gulliver loses first place" - (d), "Grogul loses money" - (t). The grammatical conjunctions are: conjunction " \wedge ", disjunction " \vee ", "If, ... then" - " \rightarrow ".

As a result, after analysis, the breakdown into simple statements, the initial complex sentence can be written in a formalized form: $((a \vee b) \wedge c) \rightarrow (d \wedge t)$.

Example 3. Formalize the statement in the language of the logic of the statements: "If the workers or the administration balk, the strike will be settled only when the government makes concessions but no arrest won't be imposed on the plant."

By structure, this sentence is complicated. It consists of several simple statements, namely: "workers balk" - denote (r); "administration balk" - denote (a); "strike will be settled" - (s); "the government makes concessions" - (u); "the arrest will be imposed on the plant" - (z). We find grammatical conjunctions: "or" - " \vee " "If, ... then" - " \rightarrow "; "If and only then" - " \leftrightarrow "; "But" is synonymous with the conjunction "i" - " \wedge "; "No" - " \sim ". After the analysis we write down the sentences by the means of logic statements: $(r \vee a) \rightarrow (s \leftrightarrow (u \wedge (\sim z)))$.

As a conclusion, the formalization of sentences in the language of utterance logic should promote understanding of the concepts of logical laws and operations by students of philology, which can later be used for linguistic analysis, such as textual documents.

References

1. Robert R. Stoll. (1961). Sets Logic and Axiomatic Theories Edition (Undergraduate Mathematics Books).
2. Nikitchenko M.S., Shkilnyak S.S., (2008) Mathematical Logic and Theory of Algorithms: A Textbook. Kyiv [in Ukrainian].

Відношення логічного наслідку в логіках часткових предикатів з композицією предикатного доповнення

О. С. Шкільняк

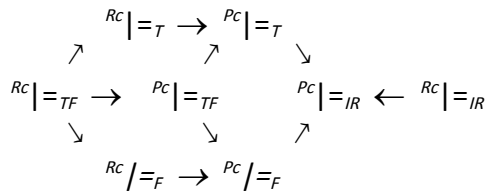
Київський національний університет імені Тараса Шевченка

me.oksana@gmail.com

Досліджено першопорядкові логіки часткових предикатів з композицією (операцією) предикатного доповнення \sim , такі логіки названо L^C . Подібні операції використовуються в розширеннях відомих [1] програмних логік Флойда-Хоара з частковими перед- та після-умовами. Описано композиційні алгебри і мови L^C . Композиція \sim не зберігає тотальність предикатів, тому коректним є розгляд L^C R -предикатів (часткових неоднозначних реляційного типу) та L^C P -предикатів (часткових однозначних), такі логіки названо L^{CP} .

На множинах формул мови L^C можна визначити низку відношень логічного наслідку. Поширення на мову L^C відомих відношень $^P|=_{IR}$ (неспростовнісний наслідок) $^P|=_T$ (істиннісний наслідок), $^P|=_F$ (хибнісний наслідок), $^P|=_{TF}$ та $^R|=_{TF}$ (сильний наслідок) дає в L^{CP} та в L^P відповідні відношення логічного наслідку $^{PC}=_{IR}$, $^{PC}=_T$, $^{PC}=_F$, $^{PC}=_{TF}$ та $^{RC}=_{IR}$, $^{RC}=_T$, $^{RC}=_F$, $^{RC}=_{TF}$. При цьому для відношень типу $^P|=_T$ та типу $^P|=_F$ маємо різні властивості декомпозиції формул вигляду $\sim\Phi$, які не можна подати як спільну властивість для відношень типу $^P|=_{TF}$, тому відношення $^{RC}=_{IR}$, $^{RC}=_T$, $^{RC}=_F$, $^{RC}=_{TF}$ в L^C виявляються різними. Водночас в традиційній логіці квазіарних предикатів маємо $^R|=_{IR} = \emptyset$ та $^R|=_T = ^R|=_F = ^R|=_{TF}$. Маємо $^{RC}=_{IR} \neq \emptyset$, проте відношення $^{RC}=_{IR}$ має вироджений характер: якщо формули множин Γ та Δ не мають входжень \sim , то завжди $\Gamma^{RC} \not\equiv_{IR} \Delta$, зокрема, $\Gamma^{RC} \not\equiv_{IR} \Gamma$, що засвідчує нереклексивність $^{RC}=_{IR}$.

Між зазначеними відношеннями логічного наслідку в L^C маємо такі співвідношення:



Показано, що для відношення $P^C|_{IR}$ неможливо коректним способом задати умови декомпозиції формул вигляду $\sim\Phi$. Це мотивує перехід від $P^C|_{IR}$ до більш загального відношення $|_{IR}^\perp$ неспростовнісного логічного наслідку за умов невизначеності [2]. Подібний перехід зроблено [3] для відношень типів T , F та TF . Отримуємо відношення $|_{IR}^\perp$, $P|_T^\perp$, $P|_F^\perp$, $P|_{TF}^\perp$, $R|_T^\perp$, $R|_F^\perp$, $R|_{TF}^\perp$, співвідношення між ними ідентичні співвідношенням між $P^C|_{IR}$, $P^C|_T$, $P^C|_F$, $P^C|_{TF}$, $R^C|_T$, $R^C|_F$, $R^C|_{TF}$. Для відношень типів $|_T$ та $|_F$ доведено теорему про елімінацію умов невизначеності, що дає змогу звести $P|_T^\perp$, $P|_F^\perp$, $R|_T^\perp$, $R|_F^\perp$ до відношень $P^C|_T$, $P^C|_F$, $R^C|_T$, $R^C|_F$. Водночас $|_{IR}^\perp$ неможливо звести до відношення $P^C|_{IR}$, яке некоректне.

Встановлені властивості відношень логічного наслідку в L^C засвідчують істотну відмінність L^C від традиційної логіки квазіарних предикатів.

Література

1. С. Хоар. An axiomatic basis for computer programming, Comm. 1969. ACM. 12(10). P. 576–580.
2. М. Нікіченко, О. Шкілняк, С. Шкілняк. Program Logics Based on Algebras with the Composition of Predicate Complement, in 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, pp. 285–288 (2019).
3. Відношення логічного наслідку в логіках часткових предикатів з композицією предикатного доповнення // Проблеми програмування. – 2019. – № 3. – С. 11–27.

Першопорядкові логіки часткових предикатів з розширеними реномінаціями та предикатами рівності

С. С. Шкільняк

Київський національний університет імені Тараса Шевченка

ttp@unicyb.kiev.ua

Вивчаються нові програмно-орієнтовані логічні формалізми — чисті першопорядкові композиційно-номінативні логіки квазіарних предикатів із розширеними реномінаціями, предикатами-індикаторами наявності значення для змінних, вони названі L_{\perp}^Q . В центрі дослідження — L_{\perp}^Q із спеціальними предикатами рівності. Можна виділити предикати слабкої рівності $=_{xy}$ та предикати строгої рівності \equiv_{xy} , це дає L_{\perp}^Q із предикатами слабкої рівності, які названі $L_{\perp}^{Q=}$, та з предикатами строгої рівності, які названі $L_{\perp}^{Q\equiv}$.

Досліджено властивості композицій пропонуваніх логік, описано відповідні композиційні алгебри R -предикатів A_{\perp}^Q , $A_{\perp}^{Q=}$, $A_{\perp}^{Q\equiv}$. Клас P -предикатів (однозначних R -предикатів) замкнений щодо базових композицій цих алгебр, тому в A_{\perp}^Q , $A_{\perp}^{Q=}$, $A_{\perp}^{Q\equiv}$ виділяємо підалгебри P -предикатів A_{\perp}^{QP} , $A_{\perp}^{Q=P}$, $A_{\perp}^{Q\equiv P}$. Основну увагу зосереджено на дослідженні властивостей, пов'язаних з предикатами рівності. Зокрема, показано, що предикати \equiv_{xy} можна подати через предикати $=_{xy}$ та предикати-індикатори Ez . З іншого боку, наявність всюди невизначеного предиката \wedge та предикатів-індикаторів Ez дає змогу подати предикати $=_{xy}$ через предикати \equiv_{xy} , при цьому маємо подання $\wedge = R_{\perp}^x(=_{xx})$.

Описано мови L_{\perp}^Q , $L_{\perp}^{Q=}$, $L_{\perp}^{Q\equiv}$. Введено та досліджено низку відношень логічного наслідку в цих логіках: неспростовнісний (тип IR), істиннісний (тип T), хибнісний (тип F), сильний (тип TF). Показано, що в L_{\perp}^Q та $L_{\perp}^{Q\equiv}$ маємо по 5 різних невідроджених відношень: $^P|_{=IR}$, $^P|_{=T}$, $^P|_{=F}$, $^P|_{=TF}$, $^R|_{=TF}$; $^{P\equiv}|_{=IR}$, $^{P\equiv}|_{=T}$, $^{P\equiv}|_{=F}$, $^{P\equiv}|_{=TF}$, $^{R\equiv}|_{=TF}$. Водночас в $L_{\perp}^{Q=}$ коректним залишається лише відношення $^{P=}|_{=IR}$, а для відношень $^{P=}|_{=T}$, $^{P=}|_{=F}$, $^{P=}|_{=TF}$, $^{R=}|_{=TF}$ порушена умова транзитивності. Зазначені відношення логічного наслідку поширено на пари множин формул. Виділено такі групи властивостей цих відношень:

- властивості спрощення;
- властивості еквівалентних перетворень;
- властивості декомпозиції формул;
- властивості елімінації кванторів;

– властивості, пов'язані з предикатами рівності (реномінації рівності, транзитивності, заміни рівних);

– умови наявності відповідного відношення логічного наслідку.

Властивості відношень логічного наслідку є семантичною основою побудови в L_{\perp}^Q , $L_{\perp}^{Q\neq}$, $L_{\perp}^{Q\equiv}$ відповідних першопорядкових числень секвенційного типу.

Література

1. Mykola S. Nikitchenko and Stepan S. Shkilniak. Algebras and logics of partial quasiary predicates // Algebra and Discrete Mathematics, Volume 23 (2017). Number 2, pp. 263–278.

2. С. С. Шкільняк. Першопорядкові композиційно-номінативні логіки з предикатами слабкої та строгої рівності // Проблеми програмування. – 2019. – № 3. – С. 28–44.

Формування нової навчальної програми з логіки для юристів з урахуванням впровадження ТЗНПК

Юркевич Олена Миколаївна

*доктор філософських наук, професор кафедри філософії
Національного юридичного університету імені Ярослава Мудрого*
olenayurkevych@ukr.net

Павленко Жанна Олександрівна

*кандидат юридичних наук, доцент кафедри філософії
Національного юридичного університету імені Ярослава Мудрого*
zhannapavlenko4@gmail.com

Починаючи з 2016 року за спеціальністю 081 «Право» для диференціації бакалаврів за рівнями сформованості в них загальних навчальних правничих компетентностей використовують тестовий інструмент за технологією ЗНО, що має скорочену назву «ТЗНПК», який передбачає вимірювання рівня сформованості таких компетентностей, як аналітичне мислення, критичне мислення та логічне мислення. Міністерство освіти України вказує на виділення даних трьох компонентів тому, що саме вони найбільшою мірою корелюють із важливими особливостями навчальної й професійної діяльності, пов'язаної з юриспруденцією. У зв'язку з цим виникає питання, як викладати логіку студентам юридичних факультетів, щоб вона стала у нагоді на практиці.

У вільному демократичному суспільстві на перший план висувається здатність і готовність оцінювати ситуацію критично, але без упередження, відділяти істину від хиб, самостійно знаходити рішення і захищати істину в суперечці. Традиційно викладання навчальної дисципліни «Логіка» визначають через призму форм (поняття, судження, умовиводу), законів мислення (тотожності, несуперечності, виключеного третього, достатньої підстави) та теорії доведення. Вивчення логіки за такого підходу дає змогу поглибити знання про структуру думки, але не удосконалити вміння мислити. Час вимагає побудови нової моделі викладання, що розрахована на розвиток не тільки теоретичних знань, а й практичних навичок мислити, аргументувати, письмово формувати та висловлювати точку зору, приймати рішення, розв'язувати суперечки тощо. Важливо продемонструвати, що різного роду проблемні ситуації, які потребують вирішення, можна типізувати і класифікувати. А знання процедури їхнього вирішення, ґрунтоване на евристичних принципах, дасть змогу практично виформувати фундаментальні навички логічного мислення.

Розробка логічних алгоритмів, що є методологічним підґрунтям побудови тестів з вимірювання розвитку інтелектуальних здібностей юристів (ТЗНПК), вперше в Україні знайшла втілення в роботі О. М. Юркевич і Ж. О. Павленко «Підготовка до складання ТЗНПК в завданнях та рішеннях: логічний підхід» (2018 р., 2-ге вид. у 2019 р.). Впровадження в курс логіки подібної методики має сприяти розвитку цієї дисципліни в практичному напрямку.

Отже, необхідна програма, в якій викладання логіки підпорядковане вказаній перспективі. Тому має сенс головну увагу при викладанні курсу приділити наступним інтелектуальним завданням: 1) послідовному та операціональному характеру формування інтелекту; 2) комбінаторному, імовірнісному, інформаційному і поведінковому аспектам інтелектуальних дій особи; 3) значній частині недедуктивних умовиводів в міркуваннях людини.

Прийняття до уваги цих особливостей надасть можливість по-новому викладати курс логіки для юристів, представивши його зміст як послідовно пов'язані один з одним рівні формування єдиного інтелекту.

Authors • Автори

- Zainab Saadi Hussein Al-hilali • **Аль-Хілалі Зайнаб Сааді Хусейн**
Київський національний університет імені Тараса Шевченка
- Olena Grinenko • **Гріненко Олена Олександрівна**
Національний авіаційний університет
- Ihor Dutsiak • **Дуцяк Ігор Зенонович**
д. філос. н., к. т. н., професор
Національний університет "Львівська політехніка"
- Ievgen Ivanov • **Іванов Євген Вячеславович**
к. фіз.-мат. н., доцент
Київський національний університет імені Тараса Шевченка
- Yaroslav Kokhan • **Кохан Ярослав Олексійович**
к. філос. н., м. н. с.
Інститут філософії ім. Г. С. Сковороди НАН України
- Serhiy Kryvyi • **Кривий Сергій Лук'янович**
д. фіз.-мат. н., професор
Київський національний університет імені Тараса Шевченка
- Alexander Lyaletski • **Лялецький Олександр Вадимович**
к. фіз.-мат. н., с. н. с.
Національний університет біоресурсів і природокористування України
- Volodymyr Navrotskyi • **Навроцький Володимир Вячеславович**
д. філос. н.
Інститут філософії ім. Г. С. Сковороди НАН України
- Mykola Nikitchenko • **Нікітченко Микола Степанович**
д. фіз.-мат. н., професор
Київський національний університет імені Тараса Шевченка
- Ludmila Omelchuk • **Омельчук Людмила Леонідівна**
к. фіз.-мат. н., доцент
Київський національний університет імені Тараса Шевченка
- Zhanna Pavlenko • **Павленко Жанна Олександрівна**
к. юр. н., доцент
Національний юридичний університет імені Ярослава Мудрого

- Nataliia Rusina** • **Русіна Наталія Геннадіївна**
к. пед. н.
Київський національний університет імені Тараса Шевченка
- Volodymyr Shevchenko** • **Шевченко Володимир Петрович**
к. фіз.-мат. н., доцент
Київський національний університет імені Тараса Шевченка
- Olena Shyshatska** • **Шишацька Олена Володимирівна**
к. фіз.-мат. н.
Київський національний університет імені Тараса Шевченка
- Oksana Shkilniak** • **Шкільняк Оксана Степанівна**
к. фіз.-мат. н., доцент
Київський національний університет імені Тараса Шевченка
- Stepan Shkilniak** • **Шкільняк Степан Степанович**
д. фіз.-мат. н., професор
Київський національний університет імені Тараса Шевченка
- Olena Yurkevych** • **Юркевич Олена Миколаївна**
д. філос. н., професор
Національний юридичний університет імені Ярослава Мудрого

Наукове видання

Українське логічне товариство
2-ий Всесвітній день логіки – 14 січня 2020 року.
Логіка та її застосування. Семінар. – Збірка тез.
(українською та англійською мовами)

Упорядник *М. С. Нікітченко*
Дизайнер *Я. О. Кохан*

Електронне видання
Випущено 24.01.2020 р.
Гарнітура Calibri. Формат 210×297 (A4). Ум. др. арк. 0,94.
Доступ до файла в мережі Інтернет:
<http://logic.net.ua/sites/default/files/WLD/World%20Logic%20Day%20Workshop%202020-Book%20of%20Abstracts.pdf>