

## Теория полисортных графов знаний-обучения

М. С. Полянская\*, В. А. Сухомлин

ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова», г. Москва, Российская Федерация

119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1

\* m.s.polyanskaya@mail.ru

### Аннотация

Стремительные темпы развития цифровой экономики ставят новые вызовы перед системами образования, делая сверх актуальной задачу развертывания в сжатые сроки новых технологий и процессов, обеспечивающих своевременную подготовку востребованных экономикой цифровых навыков. Это в свою очередь обуславливает необходимость поиска новых эффективных образовательных технологий и решений.

В статье рассматриваются теоретические основы аппарата специального вида графов, состоящих из вершин и направленных ребер нескольких сортов и называемых полисортными. Такой аппарат предназначен для создания на его основе инструментария системы развития цифровых навыков, способствующего повышению эффективности разработки и реализации образовательных процессов. Достоинством предлагаемого инструментария является возможность его использования для разработки, описания, конфигурирования образовательного контента, а также для управления реализацией персонализированных образовательных процессов. В статье описывается алгебра полисортных графов, основные операции над такими графами, приведены примеры использования этого аппарата.

**Ключевые слова:** граф знаний, графы обучения, цифровые навыки, своды знаний, куррикулы, результаты обучения, полисортные графы.

*Авторы заявляют об отсутствии конфликта интересов.*

**Для цитирования:** Полянская, М. С. Теория полисортных графов знаний-обучения / М. С. Полянская, В. А. Сухомлин. – DOI 10.25559/SITITO.16.202004.940-950 // Современные информационные технологии и ИТ-образование. – 2020. – Т. 16, № 4. – С. 940-950.

© Полянская М. С., Сухомлин В. А., 2020



Контент доступен под лицензией Creative Commons Attribution 4.0 License.  
The content is available under Creative Commons Attribution 4.0 License.



## Theory of Polysort Graphs of Knowledge-Learning

M. S. Polyanskaya\*, V. A. Sukhomlin

Lomonosov Moscow State University, Moscow, Russian Federation

1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation

\* m.s.polyanskaya@mail.ru

### Abstract

The rapid pace of development of the digital economy poses new challenges for education systems, making it extremely urgent to accelerate the deployment of new technologies and processes that ensure the timely development of digital skills demanded by the economy. This, in turn, necessitates the search for new effective educational technologies and solutions.

The article discusses the theoretical foundations of the apparatus of special type graphs, consisting of vertices and directed edges of several sorts and called multi-sorted. Such apparatus is intended to create on its basis a toolkit for a system for the development of digital skills contributing to an increase in the efficiency of the development and implementation of educational processes. The advantage of the proposed toolkit is the ability to use it for the development, description, and configuration of educational content, as well as for managing the implementation of personalized educational processes. The article describes the algebra of polysort graphs, the basic operations on such graphs, examples of using this apparatus are given.

**Keywords:** knowledge graph, learning graphs, digital skills, bodies of knowledge, curricula, learning outcomes, polysort graphs.

*The authors declare no conflict of interest.*

**For citation:** Polyanskaya M.S., Sukhomlin V.A. Theory of Polysort Graphs of Knowledge-Learning. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2020; 16(4):940-950. DOI: <https://doi.org/10.25559/SITITO.16.202004.940-950>



## 1. Введение

В декларации министерской конференции в Канкуне<sup>1</sup>, по существу давшей старт всемирному процессу построения цифровой экономики, одной из важнейших задач, стоящей перед современным социумом, определена задача своевременной подготовки востребованных экономикой цифровых навыков. Эта задача требует от образовательных систем поиска новых подходов и технологий к подготовке кадров, направленных на повышение эффективности и качества процессов развития навыков.

В статье рассматриваются теоретические основы инструментальных средств системы развития навыков [1, 2], основанных на теории графов и применяемых на этапах как разработки образовательных контентов, так и управления персонализированными процессами развития навыков обучающихся. Авторами развивается аппарат графов специального вида, состоящих из вершин и ребер нескольких сортов и называемых полисортированными (polysorted graphs – PGs). Теория графов представляет собой весьма гибкий и популярный инструмент для решения многих задач в области представления и обработки знаний, благодаря тому, что обеспечивает достаточно четкую формальную основу, допускающую многообразие топологических решений и легкость расширения семантическими средствами, и главным ее достоинством является наглядность представления с помощью графов моделей данных и процессов.

Применение теории графов в виде семантических сетей в задачах искусственного интеллекта имеет полувековую историю [3, 4, 5]. В наши дни возросший пик интереса к использованию теории графов в задачах обработки знаний связан с представлением компанией Google в 2012 году технологии «Граф знаний», добавленной к поисковой системе Google<sup>2</sup>. Обширный обзор современных технологий, основанных на графах знаний, представлен в работе [6].

Графы обучения представляют собой еще одно применение аппарата графов в задачах обработки знаний, точнее в управлении процессом обучения [7], [8]. Понятие графов обучения (Learning Graphs – LG) детально разработано в европейском проекте MaTHiSiS<sup>3,4</sup>, в котором графы обучения применяются в качестве нового образовательного структурного инструмента, поддерживающего нелинейное выполнение сценариев обучения, одновременно способствуя персонализированному и адаптивному обучению.

Вершинами таких графов являются умные обучающие атомы

(Smart Learning Atoms – SLA) и цели обучения (learning goals – LG), в то время как отношения между SLA и учебными целями обозначаются ребрами графов. SLA – это элементарные и полные знания, которые можно изучить и оценить за одну краткосрочную итерацию, нацеленную на конкретные проблемы. А вершины LG соответствуют составным целям, включающих один или несколько SLA, изучение которых приводит к данной цели обучения.

Подход к обучению MaTHiSiS опирается на описание обучающих сценариев с помощью графов обучения, которые инкапсулируют знания, навыки и / или компетенции, которые необходимо приобрести в процессе обучения для достижения конкретных целей обучения в виде взаимосвязанных вершин в структуре графа.

Кроме этого, благодаря интерфейсу с экосистемой MaTHiSiS, графы обучения позволяют фиксировать прогресс / освоение учебного сценария для каждого конкретного учащегося, принимающего участие в процессе обучения, на основе как персонализированных показателей, зафиксированных исторически во время опыта учащегося с экосистемой MaTHiSiS, так и на основе эмоциональной реакции учащегося и результативности его целенаправленной учебной деятельности. Для этого используется механизм персонализированных весов компетентности, связываемых с вершинами графов. Таким образом аппарат графов обучения, обогащенный дополнительными настраиваемыми атрибутами, является гибким инструментом адаптивного обучения, при котором применение учебных действий, связанных в элементами графа, применяется с учетом контекста учебного процесса, учитывающего многие объективные и субъективные факторы процесса обучения.

В развиваемом авторами подходе акцент делался на разработку удобной инструментальной основы, предназначенной для решения двух основных задач учебного процесса. Первая из них – разработка структурированного наглядного описания сложных сводов знаний курикулов (Body of Knowledge – BOK)<sup>5</sup> или учебных программ, определяющих содержание обучение и его организационную структуру. Вторая задача – трансформация графовых представлений сводов знаний в графовую модель сценария образовательного процесса, которая в результате обогащения ее семантическими параметрами, аналогичными в подходе MaTHiSiS и отражающими состояние контекста учебного процесса, могла бы стать управляющей программой для образовательного движка, реализующего интеллектуальный целенаправленный образовательный процесс.

<sup>1</sup> Ministerial Declaration on the Digital Economy: Innovation, Growth and Social Prosperity (“Cancún Declaration”). In: 2016 Ministerial Meeting The Digital Economy: Innovation, Growth and Social Prosperity. Cancun – Mexico, June 21-22, 2016. [Электронный ресурс]. URL: <https://www.oecd.org/sti/ieconomy/Digital-Economy-Ministerial-Declaration-2016.pdf> (дата обращения: 14.09.2020).

<sup>2</sup> *Singhal A.* Introducing the Knowledge Graph: things, not strings [Электронный ресурс] // Google. May 16, 2012. URL: <https://blog.google/products/search/introducing-knowledge-graph-things-not-strings> (дата обращения: 14.09.2020).

<sup>3</sup> Managing Affective-learning THrough Intelligent atoms and Smart InteractionS. D3.3 The MaTHiSiS Learning Graphs / Ed. by D. Tsatsou [et al.]. European Commission, Luxembourg, 2016. [Электронный ресурс]. URL: <http://mathisis-project.eu/sites/default/files/mathisis/public/content-files/deliverables/MaTHiSiS%20D3.3%20The%20MaTHiSiS%20Learning%20Graphs.pdf> (дата обращения: 14.09.2020).

<sup>4</sup> Managing Affective-learning THrough Intelligent atoms and Smart InteractionS. D3.2 The MaTHiSiS Smart Learning Atoms / Ed. by T. Techene [et al.]. European Commission, Luxembourg, 2016. [Электронный ресурс]. URL: [http://mathisis-project.eu/sites/default/files/mathisis/public/content-files/deliverables/MaTHiSiS%20D3.2%20The%20MaTHiSiS%20Smart%20Learning%20Atoms%20M24\\_v1.0.pdf](http://mathisis-project.eu/sites/default/files/mathisis/public/content-files/deliverables/MaTHiSiS%20D3.2%20The%20MaTHiSiS%20Smart%20Learning%20Atoms%20M24_v1.0.pdf) (дата обращения: 14.09.2020).

<sup>5</sup> *Rashid A. et al.* The Cyber Security Body of Knowledge. Version 1.0, 31st October, 2019 [Электронный ресурс]. URL: <https://www.cybok.org> (дата обращения: 14.09.2020).



В данной статье рассматриваются теоретические основы полисортовых графов, достаточные для решения первой задачи. В частности, в статье описывается алгебра полисортовых графов, основные операции над такими графами, приведены примеры их использования для описания образовательных контентов курсов.

Отдавая должное результатам проекта MaTHiSiS, в данной статье частично будут использованы терминология и идеи этого проекта.

## 2. Основные понятия и определения

Ключевое понятие рассматриваемой разработки – полисортовый граф или PG. Его назначение – наглядно изображать взаимосвязь и логическую последовательность дидактических единиц учебного контента, связывать с ними получаемые знания и навыки, а впоследствии служить управляющей структурой при автоматизации сценария учебного процесса.

Направленное ребро  $(x, y)$  в PG показывает отношение: "изучение  $x$  необходимо для изучения  $y$ ". Будем называть  $x$  *предшественником*  $y$ , а  $y$  *последователем*  $x$ .

Введем ряд определений для вершин PG.

**Определение.** *Атомы обучения (SLA, [smart] learning atoms)*<sup>6</sup> – это элементарные темы учебного материала, для которых не требуется предварительного обучения и с которых начинается изучение данного курса:  $\text{deg}^+(u) = 0$  для  $u$  из  $\{SLA\}$ , где  $\text{deg}^+(u)$  обозначает полустепень захода в вершину  $u$  [9].

**Определение.** *Цели обучения (LG, learning goals)* – это все последующие относительно SLA темы учебного материала.

**Определение.** *Результаты обучения (LO, learning outcomes)* – это навыки, получаемые непосредственно при изучении определенных LG или SLA.

При структурном анализе графа PG *результаты обучения* считаются свойством соответствующих LG и SLA.

**Определение.** *Вершины типа подграфа (SG, subgraph)* – это вершина, предназначенная для сжатого представления некоторого PG, входящего в PG более высокого уровня в качестве строительного блока. Свертка подграфа в вершину типа SG подробно описана в пункте 2.

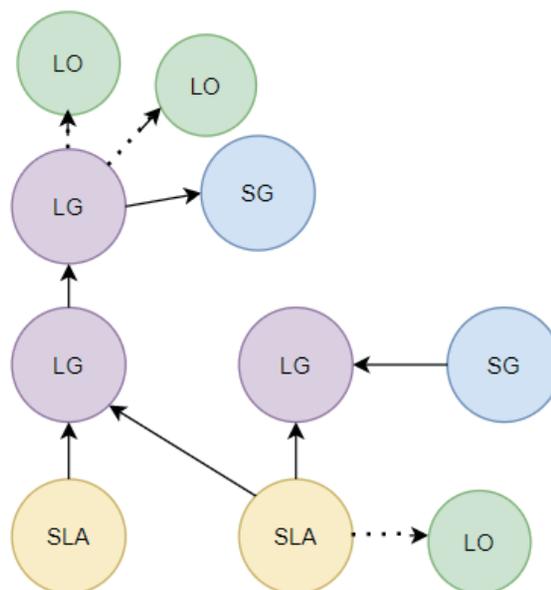
**Определение.** PG – это граф со следующими свойствами:

- ориентированный,
- ациклический,
- слабо связный (то есть связный как неориентированный граф),
- частично взвешенный,
- без кратных ребер,

$\{V \cup V', E \cup E', f\}$ , где  $V$  и  $V'$  – вершины,  $E$  и  $E'$  – ребра,  $f$  – весовая функция:  $E \rightarrow (0,1]$ .

$V = \{SLA\} \cup \{LG\} \cup \{SG\}$ .  $V' = \{LO\}$ . Ребра типа  $E$  соединяют между собой вершины типа  $V$ : их вес показывает приоритет предшественника для изучения последователя. Ребра типа  $E'$  идут от вершин типа  $V$  к вершинам типа  $V'$ .

Введенные понятия иллюстрируются на рисунке 1.



Р и с. 1. Пример PG с вершинами SLA, LG, LO и соответствующими им направленными ребрами

Fig. 1. Example PG with vertices SLA, LG, LO and their corresponding directed edges

**Определение.** Рекурсивно определим *глубину PG - d*:

- $d(PG) = 0$ , если в PG отсутствуют вершины типа SG;
- иначе  $d(PG) = 1 + \max(d(s))$ , где  $s$  берется по всем подграфам, свернутым в вершину типа SG.

**Определение.** Рекурсивно определим *завершенный PG*.

Если 1)  $\text{deg}^+(u) > 0$  для любого  $u$  из  $\{LG\}$ , 2) у любой вершины типа SG, в которую свернут незавершенный подграф, существуют предшественники, то PG называется *завершенным*. Иначе – *незавершенным*.

По завершенному подграфу можно прокладывать учебные траектории, которые обязаны начинаться SLA или завершенным SG.

**Определение.** *Учебная траектория* — последовательность  $U$  неповторяющихся вершин PG типа  $V$ , для которой верно: если  $u$  принадлежит  $U$ , то все предшественники  $u$  также принадлежат  $U$ .

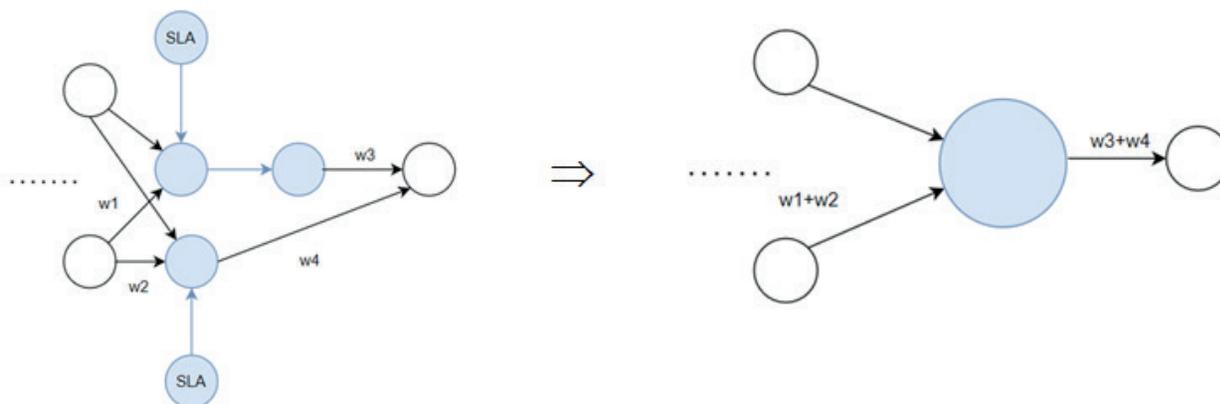
## 2. Свертка подграфа

Вершины типа подграф введены для удобства структурирования сводов знаний (BOK) курсов. В некотором роде введенный способ структурирования графов подобен способу, применявшемуся в работе [10].

Свертка графов иллюстрируется на рисунке 2.

<sup>6</sup> Термин *smart learning atom* позаимствован из документации MaTHiSiS, где вершины данного типа играют главную роль в динамическом планировании обучения, меняя свой вес. В данной работе эта тема опущена.





Р и с. 2. Иллюстрация корректной свертки вершин PG в левой части рисунка в вершину подграфа типа SG в правой части рисунка

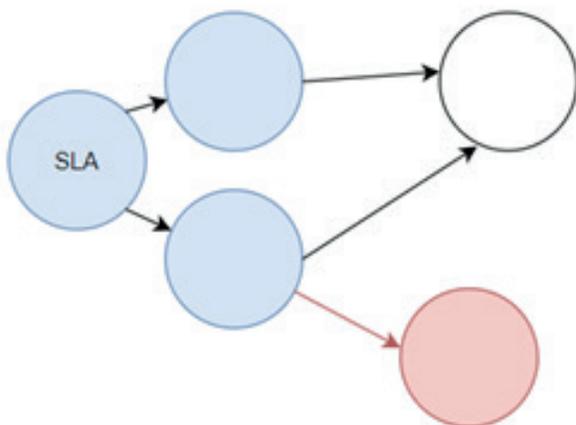
Fig. 2. An illustration of the correct convolution of the PG vertices on the left side of the figure to the vertex of the SG type subgraph on the right side of the figure

Рассмотрим подробно свертку подграфа в вершину подграфа типа SG:

Пусть подграф является PG. Пусть *старшие вершины* – те, у которых (в данном подграфе) нет последователей; *младшие вершины* – те, у которых (в данном подграфе) нет предшественников, кроме, может быть, SLA.

Сформулируем условия свертки подграфа *subgraph* в вершину типа подграфа *S*:

- 1) от предшественников *S* идут ребра во **все** младшие вершины *subgraph*;
- 2) в последователях *S* идут ребра от **всех** старших вершин *subgraph*;
- 3) все ребра, инцидентные вершинам, принадлежащим *subgraph*, принадлежат *subgraph* **либо** соединяют его с предшественниками/последователями *S*. То есть, на рисунке 3 нельзя свернуть подграф с вершинами, помеченными синим: «мешает» вершина, помеченная красным:



Р и с. 3. Пример невозможности свернуть в вершину типа SG подграф, содержащий вершины, помеченные синим цветом

Fig. 3. An example of the impossibility of collapsing a subgraph containing the vertices marked in blue to an SG vertex

При использовании весов ребер, в случае наличия более одного ребра, связывающего сворачиваемый граф с предшественником или последователем, такие ребра склеиваются в одно ребро с весом, равным сумме исходных ребер, как показано на рисунке 2.

Заметим, что полустепень захода ( $\text{deg}^+$ ) вершины типа подграф может быть любой (в отличие от вершин SLA и LG). Также вес ребра, инцидентного вершине типа подграф и образованного склейкой ребер, может быть больше 1.

### 3. Алгебра полисортированных графов

Введем замкнутые операции над PG глубины 0 (к которым может быть приведен любой PG путем развертывания вершин типа SG). Эти операции реализованы в редакторе графов. В основном это стандартные операции над графами (и их комбинации) с проверкой тех и только тех свойств из определения, которые могут быть нарушены при применении данной операции.

**Определение.** *Правильное добавление вершины типа  $V$  к графу  $G$*  – это добавление вершины такое, что:

1. Вершина описана как:
  - название, прежде не использованное среди вершин графа  $G$
  - подтип (SLA или LG)
  - список различных последователей в виде пар: (последователь, вес ребра)
  - список различных предшественников в виде пар: (предшественник, вес ребра)
  - список LO, ассоциированных с вершиной
2. Последователи и предшественники принадлежат  $V(G)$ . При этом LO могут как принадлежать  $V(G)$ , так и отсутствовать там, тогда они естественным образом добавляются.
3. Среди ее последователей нет SLA.
4. Если для вершины указан подтип SLA, то список ее предшественников пуст.
5. В графе не возникает циклов (проверка: нет пересечений множества предшественников со множеством последователей; в графе не существует направленных путей от какого-либо последователя к какому-либо предшественнику).



6. Граф сохраняет слабую связность (вершину без последователей и предшественников одновременно можно добавить лишь в пустой граф).

После успешного добавления вершины состояние графа ((не)завершенный) обновляется.

Обозначим результат добавления вершины  $u$  (с описанием из пункта 1) к графу  $G$ :  $G' = add(G, u)$ .

**Утверждение 1.** Если возможны операции  $add(G, u)$ ,  $add(add(G, u), v)$ ,  $add(G, v)$ ,  $add(add(G, v), u)$ , то  $add(add(G, u), v) = add(add(G, v), u)$ .

**Утверждение 2.** Множество PG замкнуто относительно операции правильного добавления вершины.

**Определение.** Правильное добавление ребра к графу  $G$  – это добавление ребра такое, что:

1. Ребро описано

- либо тройкой  $(u1, u2, \text{вес})$ , где  $u1, u2 \in V(G)$ , такие что  $(u1, u2) \notin E(G)$ ,  $\text{вес} \in (0; 1]$ ,

- либо парой  $(u1, u2)$ , где  $u1 \in V(G)$ ,  $u2 \in V'(G)$ , такие что  $(u1, u2) \notin E'(G)$ .

2. В первом случае,  $u2$  не является SLA.

3. В графе не возникает цикла.

После успешного добавления ребра состояние графа ((не)завершенный) обновляется.

Обозначим результат добавления к графу  $G$  ребра  $e$ :  $G' = add(G, e)$ .

**Утверждение 1.** Если возможны операции  $add(G, e)$ ,  $add(add(G, e), f)$ ,  $add(G, f)$ ,  $add(add(G, f), e)$ , то  $add(add(G, e), f) = add(add(G, f), e)$ .

**Утверждение 2.** Множество PG замкнуто относительно операции правильного добавления ребра.

**Определение.** Правильное удаление ребра из графа  $G$  – это удаление ребра такое, что:

1. Ребро принадлежит  $E(G) \cup E'(G)$  и описано соответствующей парой.

2. Граф сохраняет слабую связность.

После успешного удаления ребра состояние графа ((не)завершенный) обновляется.

Обозначим результат удаления из графа  $G$  ребра  $e$ :  $G' = sub(G, e)$ .

**Утверждение 1.** Если возможны операции  $sub(G, e)$ ,  $sub(sub(G, e), f)$ ,  $sub(G, f)$ ,  $sub(sub(G, f), e)$ , то  $sub(sub(G, e), f) = sub(sub(G, f), e)$ .

**Утверждение 2.** Множество PG замкнуто относительно операции правильного удаления ребра.

**Определение.** Правильное удаление вершины из графа  $G$  – это последовательное правильное удаление всех ребер, инцидентных данной вершине, с вычеркиванием ее из  $V(G) \cup V'(G)$  на последнем удалении перед проверкой слабой связности.

После успешного удаления вершины состояние графа ((не)завершенный) обновляется.

Обозначим результат удаления из графа  $G$  вершины  $u$ :  $G' = sub(G, u)$ .

**Утверждение 1.** Если возможны операции  $sub(G, u)$ ,  $sub(sub(G, u), v)$ ,  $sub(G, v)$ ,  $sub(sub(G, v), u)$ , то  $sub(sub(G, u), v) = sub(sub(G, v), u)$ .

**Утверждение 2.** Множество PG замкнуто относительно операции правильного удаления вершины.

**Определение.** Правильное объединение графов  $G$  и  $H$  – это такое объединение, которое можно свести к последовательному правильному добавлению к графу  $G$  вершин  $V(H) \cup V'(H)$ .

Заметим, что по условию сохранения слабой связности необходимо наличие у  $G$  и  $H$  общих вершин.

После успешного объединения устанавливается состояние нового графа ((не)завершенный).

Обозначим результат объединения:  $GH = U(G, H)$ .

**Утверждение 1 (коммутативность).** Если возможна операция  $U(G, H)$ , то возможна операция  $U(H, G)$  и  $U(G, H) = U(H, G)$ .

**Утверждение 2 (ассоциативность).** Если возможны операции  $U(G, H)$ ,  $U(U(G, H), J)$ ,  $U(H, J)$ , то возможна операция  $U(G, U(H, J))$  и  $U(U(G, H), J) = U(G, U(H, J))$ .

**Утверждение 3 (замкнутость).** Множество PG замкнуто относительно операции правильного объединения.

В редакторе графов при выполнении операций, которые сводятся к последовательному применению ранее введенных операций, – объединение графов и удаление вершины из графа — реализация операций применяется к копии графа, и если операции выполняются успешно, то граф заменяется этой копией. Это сделано потому, что корректность операции приходится проверять на каждом шаге.

Следующая операция уже не является стандартной операцией над графом с наложенными условиями, она переопределена.

**Определение.** Правильное прямое произведение графов  $G$  и  $H$  – прямое произведение  $G - \{SLA(H) \cup LO(H)\}$ ,  $H - \{SLA(H) \cup LO(H)\}$ , к которому естественным образом присоединены SLA и LO: соединяются с вершиной  $(u, v) \Leftrightarrow$  соединены с  $u$  в графе  $G$  либо с  $v$  в графе  $H$ .

После успешного произведения устанавливается состояние нового графа ((не)завершенный). Заметим, что новый граф является завершенным тогда и только тогда, когда хотя бы один из двух множителей — завершенный [13]-[20].

Обозначим результат произведения:  $GH = X(G, H)$ .

**Утверждение 1 (коммутативность).** Операция правильного прямого произведения коммутативна:  $X(G, H) = X(H, G)$ .

**Утверждение 2 (ассоциативность).** Операция правильного прямого произведения ассоциативна:  $X(X(G, H), J) = X(G, X(H, J))$ .

**Утверждение 3 (замкнутость).** Множество PG замкнуто относительно операции правильного прямого произведения.

## 4. Вершинные характеристики [11]

Будем рассматривать отдельно *основную долю* PG, то есть PG вида  $\{V, E, f\}$ . Выделим следующие понятия из теории графов, применимые к вершине:

- *Полустепень захода* – число входящих в вершину ребер.

Вершины с высокой полустепенью захода хорошо подходят для проверки знаний по младшим вершинам. Находясь на этой вершине, студент достаточно опытен, и им можно сопоставить сложные, творческие задания [21]-[25].

- *Полустепень исхода* – число исходящих из данной вершины ребер. С вершин с высокой полустепенью исхода удобно начинать курс/раздел, так как они приоткрывают доступ ко многим LG.



- *Степень посредничества* – суммарная доля кратчайших направленных путей между парами вершин, проходящих через данную вершину.

$$BC(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

где  $\sigma_{st}$  – общее число кратчайших путей из  $s$  в  $t$ , а  $\sigma_{st}(v)$  – число их, проходящих через вершину  $v$ .

Вершины с высокой степенью посредничества соединяют части графа: при их удалении он распадется на компоненты связности. В них удобно проверять знания по части курса.

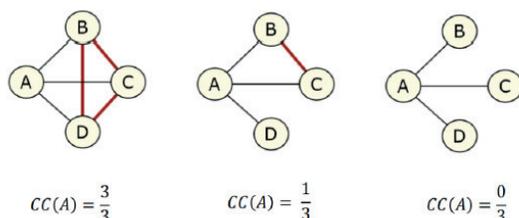
- *Степень влиятельности* – мера влиятельности вершины, связывающая ее с влиятельностью последователей:

Вероятно, манипулируя содержанием вершины с высокой степенью влиятельности, можно значительно повлиять на учебные показатели.

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t,$$

или  $Ax = \lambda x$ , где  $A$  – матрица смежности (возможно, взвешивания),  $\lambda$  – её наибольшее собственное значение.

- *Коэффициент кластеризации* – удельное число связей между ближайшими соседями данной вершины. Рисунок 4 иллюстрирует три примера подсчета коэффициента кластеризации вершины A в модельных графах.



Р и с. 4. Иллюстрация понятия: коэффициент кластеризации (CC).

В числителе – количество связей между ближайшими соседями вершины A (помечены красным цветом); в знаменателе – максимально возможное количество связей между этими соседями

Fig. 4. Concept illustration: Clustering Coefficient (CC). The numerator contains the number of links between the nearest neighbors of the vertex A (marked in red); the denominator is the maximum possible number of connections between these neighbors

Вершины с высоким коэффициентом кластеризации также можно считать влиятельными (в своем кластере), и на модификацию их содержания или объема изучения последует быстрый фидбек: изменения в учебных показателях учащихся, в качестве усвоения ими материала.

Применительно к PG данные характеристики, как отмечается в интерпретации каждой из них, рекомендуется соответствующим образом учитывать при построении учебных траекторий и модификации учебных материалов.

## 5. Примеры использования аппарата PG для представления контента куррикула по кибербезопасности

Для иллюстрации введенных понятий был выбран Cybersecurity Curricula 2017 (CSEC2017) [12]. В этом куррикуле представлена иерархия вложенных учебных разделов: Knowledge Area > Knowledge Unit > Topic > Subtopic.

На языке JavaScript в рамках проекта [2] создан редактор PG, выполняющий описанные выше операции над PG.

На рисунке 5 иллюстрируется выполнение в редакторе операции правильного добавления вершины, с визуализацией полученного графа и с возможностью хранения графа в формате json для его повторного использования.

Выберите файл | файл не выбран | save graph

### Добавить вершину

Название вершины:

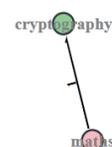
Тип вершины:  
 SLA  LG

Соседи:

родство	имя	вес ребра
<input checked="" type="radio"/> родитель <input type="radio"/> потомок	cryptography	1

Добавить поле | Удалить поле

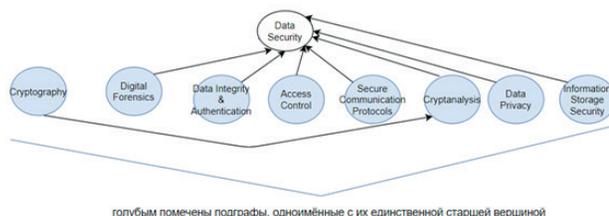
Готово | Сброс



Р и с. 5. Иллюстрация функционала редактора PG (скриншот).

В рассматриваемый PG последовательно добавлены вершины типа LG *cryptography* и *maths* (предшественник *cryptography* с весом ребра 1). Вершина типа LG без предшественников помечается красным цветом Fig. 5. Illustration of the functionality of the PG editor (screenshot). In the considered PG, vertices of the LG *cryptography* and *maths* type (the predecessor of *cryptography* with an edge weight of 1) are successively added.

The node of LG type without predecessors is marked in red



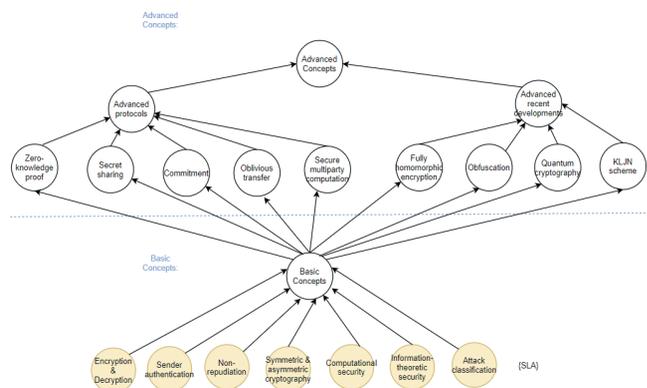
голубым помечены подграфы, одноименные с их единственной старшей вершиной

Р и с. 6. PG, составленный по фрагменту CSEC2017: Knowledge Area *Data Security* (вершина типа LG) и ее предшественники – вершины типа SG, в которые свернуты подграфы, одноименные с их единственной старшей вершиной уровня Knowledge Unit

Fig. 6. PG, compiled from the fragment CSEC2017: Knowledge Area *Data Security* (node of the LG type) and its predecessors - nodes of the SG type, into which subgraphs are folded, which have the same name with their only senior node of the Knowledge Unit level



С помощью описанного выше аппарата PG в рамках проекта выполнено представление свода знаний CSEC2017. Для иллюстрации возможностей PG для проектирования и описания сводов знаний курсикулов на рисунке 6 и рисунке 7 показаны фрагменты курсикула CSEC2017, представленные в виде PG.



Р и с. 7. PG, составленный по фрагменту CSEC2017: развернутая иллюстрация того, что подграф *Basic Concepts* (в который входят вершина *Basic Concepts* и все ее подразделы) является предшественником подграфа *Advanced Concepts* (в который входят вершина *Advanced Concepts* и все ее подразделы). *Basic Concepts* и *Advanced Concepts* находятся на уровне Topics

Fig. 7. PG based on CSEC2017 snippet: a detailed illustration of the *Basic Concepts* subgraph (which includes the *Basic Concepts* vertex and all its subsections) is the predecessor of the *Advanced Concepts* subgraph (which includes the *Advanced Concepts* vertex and all its subsections). *Basic Concepts* and *Advanced Concepts* are at the Topics level

## Заключение

В статье рассмотрены теоретические основы графов специального вида, состоящих из вершин и направленных ребер нескольких сортов и названных полисортными. Данный аппарат разработан с целью применения в качестве графического метаязыка для наглядного представления сводов знаний образовательных курсикулов. Рассмотрены алгебра полисортных графов, основные операции над полисортными графами и их свойства, приведены примеры использования таких графов. Предложенный аппарат полисортных графов и разработанный на его основе редактор графов используются для проектирования и документирования образовательного контента учебных программ и курсов в системе развития цифровых навыков [2]. Авторами также планируется использование аппарата полисортных графов, расширенных связанными с вершинами семантическими весовыми коэффициентами компетентности, аналогичными графам обучения проекта MaTHiSiS, для описания сценариев адаптивного учебного процесса с возможностью динамической прокладки через систему графов персонализированных учебных траекторий.

## Список использованных источников

- [1] Сухомлин, В. А. Методологические аспекты концепции цифровых навыков / В. А. Сухомлин, Е. В. Зубарева, А. В. Якушин. – DOI 10.25559/SITITO.2017.2.253 // Современные информационные технологии и ИТ-образование. – 2017. – Т. 13, № 2. – С. 146-152. – URL: <https://www.elibrary.ru/item.asp?id=30258665> (дата обращения: 14.09.2020). – Рез. англ.
- [2] Сухомлин, В. А. Система развития цифровых навыков ВМК МГУ & Базальт СПО. Методика классификации и описания требований к сотрудникам и содержанию образовательных программ в сфере информационных технологий / В. А. Сухомлин, Е. В. Зубарева, Д. Е. Намиот, А. В. Якушин. – М.: Базальт СПО; МАКС Пресс, 2020. – 184 с.
- [3] Попов, Э. В. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта / Э. В. Попов, Г. Р. Фирдман. – М.: Наука, 1976. – 456 с.
- [4] Поспелов, Д. А. Логико-лингвистические модели в системах управления. – М.: Энергоиздат, 1981. – 232 с.
- [5] Ceruzzi, P. E. A History of Modern Computing / P. E. Ceruzzi. – Second Edition. – Cambridge, MA, USA: MIT Press, 2003.
- [6] Гурин, В. С. Представление знаний в виде графа: основные технологии и подходы / В. С. Гурин, Е. В. Костров, Ю. Ю. Гавриленко, Д. Ф. Саада, Е. А. Ильюшин, И. В. Чижов. – DOI 10.25559/SITITO.15.201904.932-944 // Современные информационные технологии и ИТ-образование. – 2019. – Т. 15, № 4. – С. 932-944. – URL: <https://elibrary.ru/item.asp?id=43575614> (дата обращения: 14.09.2020). – Рез. англ.
- [7] Harmelen, F. Handbook of Knowledge Representation / F. Harmelen, V. Lifschitz, B. Porter. – 1st Edition. – Vol. 1. – Elsevier Science, 2007.
- [8] Пилецкий, И. И. Граф знаний и машинное обучение как базис методологии искусственного интеллекта в обучении / И. И. Пилецкий, М. П. Батура, Н. А. Волорова // BIG DATA и анализ высокого уровня: сборник научных статей VII Международной научно-практической конференции / редкол.: В. А. Богуш [и др.]. – Минск: Бестпринт, 2021. – С. 198-209. – URL: <https://libeldoc.bsuir.by/handle/123456789/43904> (дата обращения: 14.09.2020). – Рез. англ.
- [9] Харари, Ф. Теория графов / Ф. Харари. – М.: Мир, 1973. – 300 с.
- [10] Сухомлин, В. А. Алгоритмическая система для описания процессов трансляции / В. А. Сухомлин // Программирование. – 1975. – № 2. – С. 77-83.
- [11] Lightfoot, J. M. A Graph-Theoretic Approach to Improved Curriculum Structure and Assessment Placement / J. M. Lightfoot // Communications of the IIMA. – 2010. – Vol. 10, issue 2. – Article 5. – URL: <http://scholarworks.lib.csusb.edu/ciima/vol10/iss2/5> (дата обращения: 14.09.2020).
- [12] Joint Task Force on Cybersecurity Education. – DOI 10.1145/3184594 // Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity. – Association for Computing Machinery, New York, NY, USA, 2018.



- [13] Li, Z. Recalibration convolutional networks for learning interaction knowledge graph embedding / Z. Li, H. Liu, Z. Zhang, T. Liu, J. Shu. – DOI 10.1016/j.neucom.2020.07.137 // *Neurocomputing*. – 2021. – Vol. 427. – Pp. 118-130.
- [14] Wang, H. Exploring High-Order User Preference on the Knowledge Graph for Recommender Systems / H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo. – DOI 10.1145/3312738 // *ACM Transactions on Information Systems*. – 2019. – Vol. 37, issue 3. – Article 32.
- [15] Wang, Q. Knowledge Graph Embedding: A Survey of Approaches and Applications / Q. Wang, Z. Mao, B. Wang, L. Guo. – DOI 10.1109/TKDE.2017.2754499 // *IEEE Transactions on Knowledge and Data Engineering*. – 2017. – Vol. 29, issue 12. – Pp. 2724-2743.
- [16] Zhao, Y. Knowledge graph entity typing via learning connecting embeddings / Y. Zhao, A. Zhang, H. Feng, Q. Li, P. Gallinari, F. Ren. – DOI 10.1016/j.knosys.2020.105808 // *Knowledge-Based Systems*. – 2018. – Vol. 196. – Pp. 105808.
- [17] Le, N.-T. Mining weighted subgraphs in a single large graph / N.-T. Le, B. Vo, L.B.Q. Nguyen, H. Fujita, B. Le. – DOI 10.1016/j.ins.2019.12.010 // *Information Sciences*. – 2020. – Vol. 514. – Pp. 149-165.
- [18] Suchanek, F. M. Yago: a core of semantic knowledge / F. M. Suchanek, G. Kasneci, G. Weikum. – DOI 10.1145/1242572.1242667 // *Proceedings of the 16th international conference on World Wide Web (WWW '07)*. – Association for Computing Machinery, New York, NY, USA, 2007. – Pp. 697-706.
- [19] Steinmetz, N. Semantic Multimedia Information Retrieval Based on Contextual Descriptions / N. Steinmetz, H. Sack. – DOI 10.1007/978-3-642-38288-8\_26 // *The Semantic Web: Semantics and Big Data. ESWC 2013. Lecture Notes in Computer Science*; P. Cimiano, O. Corcho, V. Presutti, L. Hollink, S. Rudolph (ed.). Springer, Berlin, Heidelberg. – 2013. – Vol. 7882. – Pp. 382-396.
- [20] Zhang, M. Knowledge Graph Completion for the Chinese Text of Cultural Relics Based on Bidirectional Encoder Representations from Transformers with Entity-Type Information / M. Zhang, G. Geng, S. Zeng, H. Jia. – DOI 10.3390/e22101168 // *Entropy*. – 2020. – Vol. 22, issue 10. – Pp. 1168.
- [21] Бхатт, Ш. Графы знаний как средство улучшения искусственного интеллекта // Ш. Бхатт, Ц. Чжао, А. Сетх, В. Шалин // *Открытые системы.СУБД*. – 2020. – № 03. – С. 24-26. – URL: <https://elibrary.ru/item.asp?id=43925226> (дата обращения: 14.09.2020).
- [22] Pommellet, T. Feeding Machine Learning with Knowledge Graphs for Explainable Object Detection? / T. Pommellet, F. Lécué // *CEUR Workshop Proceedings*. – 2019. – Vol. 2456. – Pp. 277-280. – URL: <http://ceur-ws.org/Vol-2456/paper72.pdf> (дата обращения: 14.09.2020).
- [23] Berrendorf, M. Knowledge Graph Entity Alignment with Graph Convolutional Networks: Lessons Learned / M. Berrendorf, E. Faerman, V. Melnychuk, V. Tresp, T. Seidl. – DOI 10.1007/978-3-030-45442-5\_1 // *Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science*; J. Jose [et al.] (ed.). Springer, Cham. – 2020. – Vol. 12036. – Pp. 3-11.
- [24] St-Hilaire, F. A Comparative Study of Learning Outcomes for Online Learning Platforms / F. St-Hilaire [et al.]. – DOI 10.1007/978-3-030-78270-2\_59 // *Artificial Intelligence in Education. AIED 2021. Lecture Notes in Computer Science*; I. Roll, D. McNamara, S. Sosnovsky, R. Luckin, V. Dimitrova (ed.). Springer, Cham. – 2021. – Vol. 12749. – Pp. 331-337.
- [25] Giabbanelli, P. J. Learning Analytics to Support Teachers' Assessment of Problem Solving: A Novel Application for Machine Learning and Graph Algorithms / P. J. Giabbanelli, A. A. Tawfik, V. K. Gupta. – DOI 10.1007/978-3-319-64792-0\_11 // *Utilizing Learning Analytics to Support Study Success*; D. Ifenthaler, D. K. Mah, J. K. Yau (ed.). Springer, Cham. – 2019. – Pp. 175-199.

Поступила 14.09.2020; одобрена после рецензирования 25.10.2020; принята к публикации 17.11.2020.

#### Об авторах:

**Полянская Марина Сергеевна**, магистрант, программист лаборатории открытых информационных технологий, факультет вычислительной математики и кибернетики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), ORCID: <http://orcid.org/0000-0002-6848-0413>, m.s.polyanskaya@mail.ru

**Сухомлин Владимир Александрович**, заведующий лабораторией открытых информационных технологий, факультет вычислительной математики и кибернетики, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), доктор технических наук, профессор, ORCID: <http://orcid.org/0000-0001-9468-7138>, sukhomlin@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

## References

- [1] Sukhomlin V.A., Zubareva E.V., Yakushin A.V. Methodological Aspects of the Digital Skills Concept. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2017; 13(2):146-152. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.25559/SITITO.2017.2.253>
- [2] Sukhomlin V.A., Zubareva E.V., Namiot D.E., Yakushin A.V. *Sistema razvitiya cifrovyyh navykov VMK MGU & Bazal't SPO. Metodika klassifikacii i opisaniya trebovanij k sotrudnikam i sodержaniju obrazovatel'nyh programm v sfere informacionnyh tehnologii* [Digital Skills Development System]. MAKS Press: Basealt Publ., Moscow; 2020. (In Russ.)
- [3] Popov E.V., Fridman G.F. *Algorithmicheskie osnovy intellektualnykh robotov i iskusstvennogo intellekta* [Algorithmic Basics of Intelligent Robots and Artificial Intelligence]. Nauka Publ., Moscow; 1976. (In Russ.)
- [4] Pospelov D.A. *Logiko-lingvisticheskie modeli v sistemakh upravleniya* [Logical-linguistic models in control systems].



- Energoizdat Publ., Moscow; 1981. (In Russ.)
- [5] Ceruzzi P.E. A History of Modern Computing. Second Edition. MIT Press, Cambridge, MA, USA; 2003. (In Eng.)
- [6] Gurin V.S., Kostrov E.V., Gavrilenko Yu.Yu., Saada D.F., Ilyushin E.A., Chizhov I.V. Knowledge Graph Essentials and Key Technologies. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2019; 15(4):932-944. (In Eng.) DOI: <https://doi.org/10.25559/SITITO.15.201904.932-944>
- [7] Harmelen F., Lifschitz V., Porter B. Handbook of Knowledge Representation. 1st Edition, vol. 1. Elsevier Science; 2007. (In Eng.)
- [8] Piletski I.L., Batura M.P., Volarava N.A. System for Complex Analysis of Data from Internet Sources. In: Bogush V.A. (ed.) *Proceedings of the 7th International Conference on BIG DATA and Advanced Analytics*. Bestprint, Minsk; 2021. p. 198-209. Available at: <https://libeldoc.bsuir.by/handle/123456789/43904> (accessed 14.09.2020). (In Russ., abstract in Eng.)
- [9] Harary F. Graph Theory. Addison-Wesley Publishing Company, Boston; 1969. (In Eng.)
- [10] Sukhomlin V.A. *Algoritmicheskaja sistema dlja opisaniya processov transljaccii* [Algorithmic System for Describing Translation Processes]. *Programmirovanie* = Programming and Computer Software. 1975; (2):77-83. (In Russ.)
- [11] Lightfoot J.M. A Graph-Theoretic Approach to Improved Curriculum Structure and Assessment Placement. *Communications of the IIMA*. 2010; 10(2):5. Available at: <http://scholarworks.lib.csusb.edu/ciima/vol10/iss2/5> (accessed 14.09.2020). (In Eng.)
- [12] Joint Task Force on Cybersecurity Education. Cybersecurity Curricula 2017: Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity. Association for Computing Machinery, New York, NY, USA; 2018. (In Eng.) DOI: <https://doi.org/10.1145/3184594>
- [13] Li Z., Liu H., Zhang Z., Liu T. Shu J. Recalibration convolutional networks for learning interaction knowledge graph embedding. *Neurocomputing*. 2021; 427:118-130. (In Eng.) DOI: <https://doi.org/10.1016/j.neucom.2020.07.137>
- [14] Wang H., Zhang F., Wang J., Zhao M., Li W., Xie X., Guo M. Exploring High-Order User Preference on the Knowledge Graph for Recommender Systems. *ACM Transactions on Information Systems*. 2019; 37(3):32. (In Eng.) DOI: <https://doi.org/10.1145/3312738>
- [15] Wang Q., Mao Z., Wang B., Guo L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*. 2017; 29(12):2724-2743. (In Eng.) DOI: <https://doi.org/10.1109/TKDE.2017.2754499>
- [16] Zhao Y., Zhang A., Feng H., Li Q., Gallinari P., Ren F. Knowledge graph entity typing via learning connecting embeddings. *Knowledge-Based Systems*. 2018; 196:105808. (In Eng.) DOI: <https://doi.org/10.1016/j.knosys.2020.105808>
- [17] Le N.-T., Vo B., Nguyen L.B.Q., Fujita H., Le B. Mining weighted subgraphs in a single large graph. *Information Sciences*. 2020; 514:149-165. (In Eng.) DOI: <https://doi.org/10.1016/j.ins.2019.12.010>
- [18] Suchanek F.M., Kasneci G., Weikum G. Yago: a core of semantic knowledge. In: *Proceedings of the 16th international conference on World Wide Web (WWW '07)*. Association for Computing Machinery, New York, NY, USA; 2007. p. 697-706. (In Eng.) DOI: <https://doi.org/10.1145/1242572.1242667>
- [19] Steinmetz N., Sack H. Semantic Multimedia Information Retrieval Based on Contextual Descriptions. In: Cimiano P., Corcho O., Presutti V., Hollink L., Rudolph S. (ed.) *The Semantic Web: Semantics and Big Data. ESWC 2013. Lecture Notes in Computer Science*. 2013; 7882:382-396. Springer, Berlin, Heidelberg. (In Eng.) DOI: [https://doi.org/10.1007/978-3-642-38288-8\\_26](https://doi.org/10.1007/978-3-642-38288-8_26)
- [20] Zhang M., Geng G., Zeng S., Jia H. Knowledge Graph Completion for the Chinese Text of Cultural Relics Based on Bidirectional Encoder Representations from Transformers with Entity-Type Information. *Entropy*. 2020; 22(10):1168. (In Eng.) DOI: <https://doi.org/10.3390/e22101168>
- [21] Bhatt Ah., Zhao J., Sheth A., Shalin V. *Grafy znaniy kak sredstvo uluchshenija iskusstvennogo intellekta* [Knowledge graphs as a means of improving artificial intelligence]. *Open Systems.DBMS*. 2020; (03):24-26. Available at: <https://elibrary.ru/item.asp?id=43925226> (accessed 14.09.2020). (In Russ.)
- [22] Pommellet T., Lécué F. Feeding Machine Learning with Knowledge Graphs for Explainable Object Detection? *CEUR Workshop Proceedings*. 2019; 2456:277-280. Available at: <http://ceur-ws.org/Vol-2456/paper72.pdf> (accessed 14.09.2020). (In Eng.)
- [23] Berrendorf M., Faerman E., Melnychuk V., Tresp V., Seidl T. Knowledge Graph Entity Alignment with Graph Convolutional Networks: Lessons Learned. In: Jose J. et al. (ed.) *Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science*. 2020; 12036:3-11. Springer, Cham. (In Eng.) DOI: [https://doi.org/10.1007/978-3-030-45442-5\\_1](https://doi.org/10.1007/978-3-030-45442-5_1)
- [24] St-Hilaire F. et al. A Comparative Study of Learning Outcomes for Online Learning Platforms. In: Roll I., McNamara D., Sosnovsky S., Luckin R., Dimitrova V. (ed.) *Artificial Intelligence in Education. AIED 2021. Lecture Notes in Computer Science*. 2021; 12749:331-337. Springer, Cham. (In Eng.) DOI: [https://doi.org/10.1007/978-3-030-78270-2\\_59](https://doi.org/10.1007/978-3-030-78270-2_59)
- [25] Giabbanelli P.J., Tawfik A.A., Gupta V.K. Learning Analytics to Support Teachers' Assessment of Problem Solving: A Novel Application for Machine Learning and Graph Algorithms. In: Ifenthaler D., Mah D.K., Yau J.K. (ed.) *Utilizing Learning Analytics to Support Study Success*. Springer, Cham; 2019. p. 175-199. (In Eng.) DOI: [https://doi.org/10.1007/978-3-319-64792-0\\_11](https://doi.org/10.1007/978-3-319-64792-0_11)

Submitted 14.09.2020; approved after reviewing 25.10.2020;  
accepted for publication 17.11.2020.

#### About the authors:

**Marina S. Polyanskaya**, Master's student, Software Engineer of the Open Information Technologies Lab, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), ORCID: <http://orcid.org/0000-0002-6848-0413>, [m.s.polyanskaya@mail.ru](mailto:m.s.polyanskaya@mail.ru)



**Vladimir A. Sukhomlin**, Head of the Open Information Technologies Lab, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), Dr.Sci. (Technology), Professor, ORCID: <http://orcid.org/0000-0001-9468-7138>, [sukhomlin@mail.ru](mailto:sukhomlin@mail.ru)

*All authors have read and approved the final manuscript.*

