Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**РАСЧЕТНАЯ РАБОТА**

по дисциплине «Основы искусственного интеллекта»

на тему   
«Нахождение прямого произведения двух неориентированных графов»

Выполнил: Игнатович Эрик

Ст. гр. 777

Проверил: Майкл Джексон

Минск 2014

**Цель:** Получить навыки формализации и обработки информации с использованием семантических сетей

**Задача:** Найти прямое произведение двух неориентированных графов

* **Список понятий**
* Графовая структура (абсолютное понятие) - это такая одноуровневая реляционная структура, объекты которой могут играть роль либо вершины, либо связки:
* Вершина (относительное понятие, ролевое отношение);
* Связка (относительное понятие, ролевое отношение).

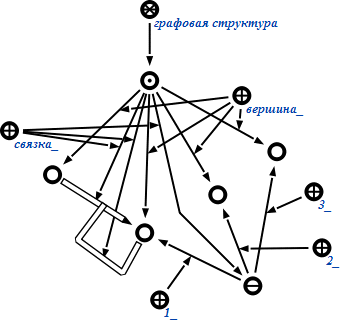


Рисунок 1.1 Пример графовой структуры

* Графовая структура с ориентированными связками (абсолютное понятие)
* Ориентированная связка (относительное понятие, ролевое отношение) –связка, которая задается ориентированным множеством.

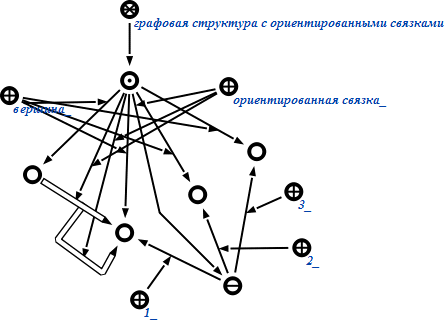


Рисунок 1.2 Пример графовой структуры с ориентированными связками

* Графовая структура с неориентированными связками (абсолютное понятие)
* Неориентированная связка (относительное понятие, ролевое отношение) –связка, которая задается неориентированным множеством.

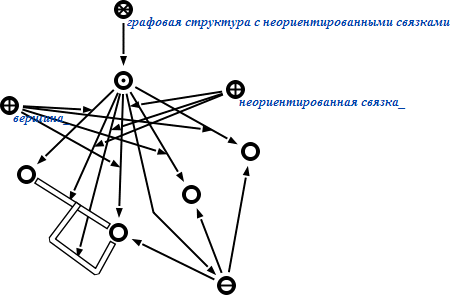


Рисунок 1.3 Пример графовой структуры с неориентированными связками

* Гиперграф (абсолютное понятие) – это такая графовая структура, в которой связки могут связывать только вершины:
* Гиперсвязка (относительное понятие, ролевое отношение);
* Гипердуга (относительное понятие, ролевое отношение) – ориентированнаягиперсвязка;
* Гиперребро (относительное понятие, ролевое отношение) – неориентированнаягиперсвязка.

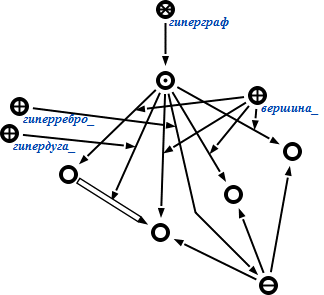


Рисунок 1.4 Пример гиперграфа

* Псевдограф (абсолютное понятие) – это такой гиперграф, в котором все связки должны быть бинарными:
* Бинарная связка (относительное понятие, ролевое отношение) –гиперсвязка арности 2;
* Ребро (относительное понятие, ролевое отношение) –неориентированнаягиперсвязка;
* Дуга (относительное понятие, ролевое отношение) – ориентированная гиперсвязка;
* Петля (относительное понятие, ролевое отношение) – бинарная связка, у которой первый и второй компоненты совпадают.

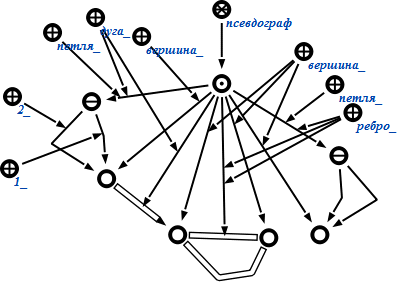


Рисунок 1.5 Пример псевдографа

* Мультиграф (абсолютное понятие) – это такой псевдограф, в котором не может быть петель:

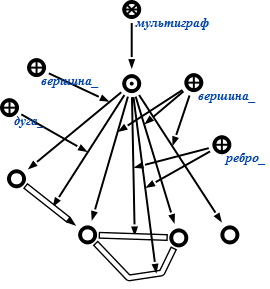


Рисунок 1.6 Пример мультиграфа

* Граф (абсолютное понятие) – это такой мультиграф, в котором не может быть кратных связок, т.е. связок у которых первый и второй компоненты совпадают:

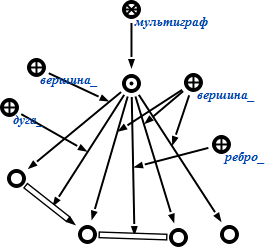


Рисунок 1.7 Пример мультиграфа

* Неориентированный граф (абсолютное понятие) –это такой граф, в котором все связки являются ребрами:

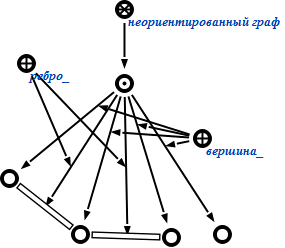


Рисунок 1.8 Неориентирофанный граф

* Ориентированный граф (абсолютное понятие) - это такой граф, в котором все связки являются дугами:

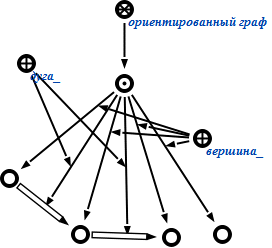


Рисунок 1.9 Ориентированный граф

* Маршрут (относительное понятие, бинарное ориентированное отношение) – это чередующаяся последовательность вершин и гиперсвязок в гиперграфе, которая начинается и кончается вершиной, и каждая гиперсвязка последовательности инцидентна двум вершинам, одна из которых непосредственно предшествует ей, а другая непосредственно следует за ней. В примере ниже показан маршрут *A*, *CON1*, *C*, *CON2*, *D*, *CON3*, *B*, *CON1*, *A*в гиперграфе.

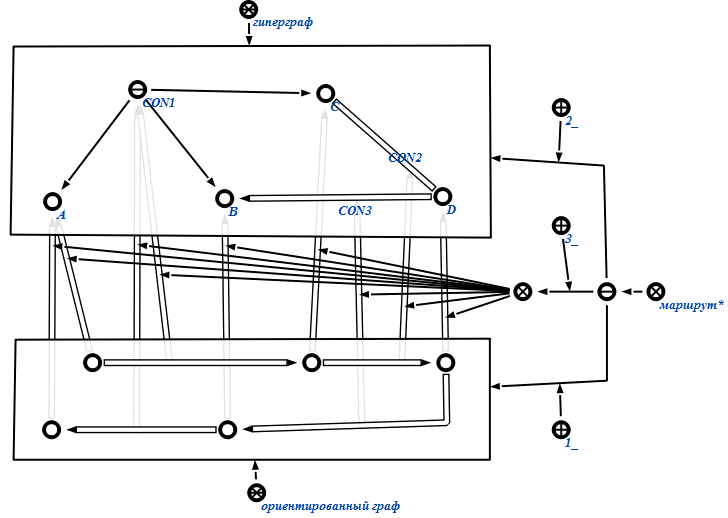


Рисунок 1.10 Маршрут в ориентированном графе

* Цепь (относительное понятие, бинарное ориентированное отношение) – это маршрут, все гиперсвязки которого различны. В примере ниже показана цепь *A*, *CON1*, *C*, *CON2*, *D*, *CON3*, *B*, *CON4*, *A*в гиперграфе.

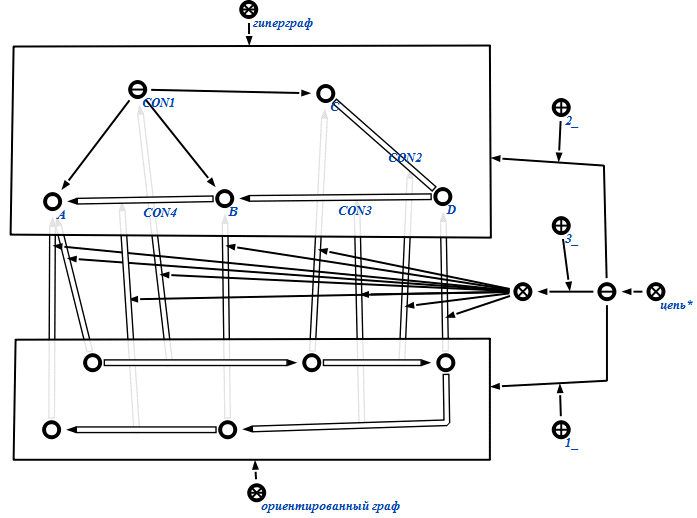


Рисунок 1.11 Цепь в ориентированном графе

* Простая цепь, путь (относительное понятие, бинарное ориентированное отношение) – это цепь, в которой все вершины различны. В примере ниже показан путь *A*, *CON1*, *C*, *CON2*, *D*, *CON3*, *B*в гиперграфе.

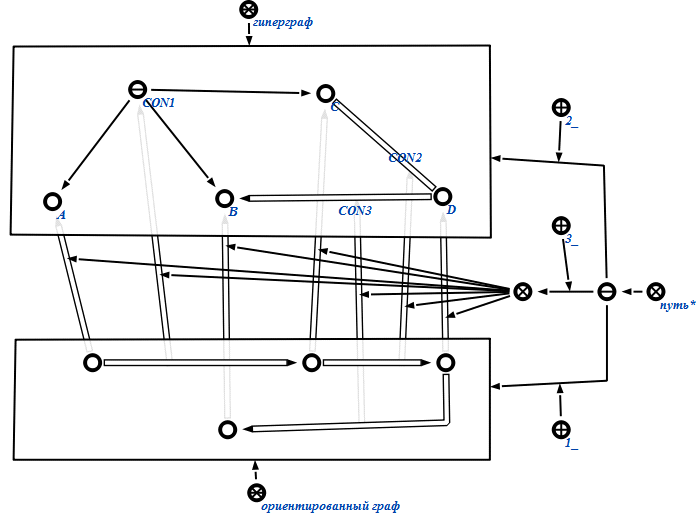


Рисунок 1.12 Путь в ориентированном графе

**Тестовые примеры**

**Тест 1**

**Вход:**

Необходимо найти прямое произведение 2-х неориентированных графов.

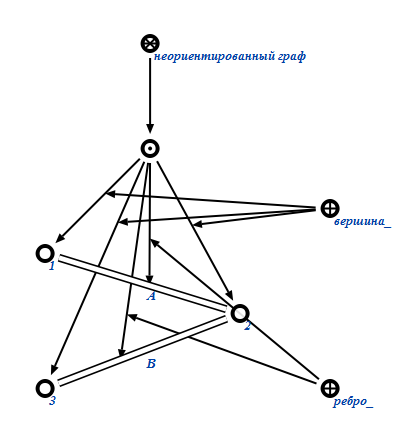
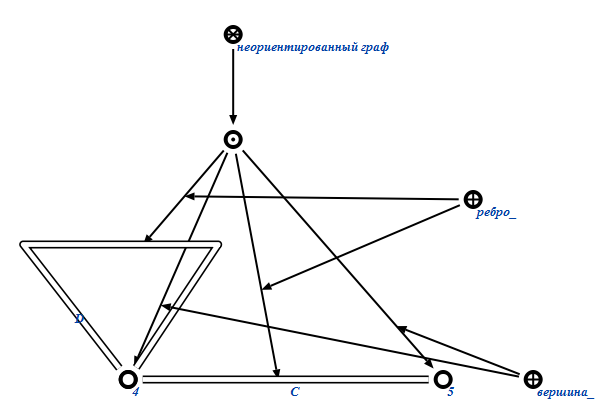
 

Рисунок 2.1. Первый исходный граф Рисунок 2.2. Второй исходный граф

**Выход:**

Будет найдено прямое произведение 2-х исходных графов:

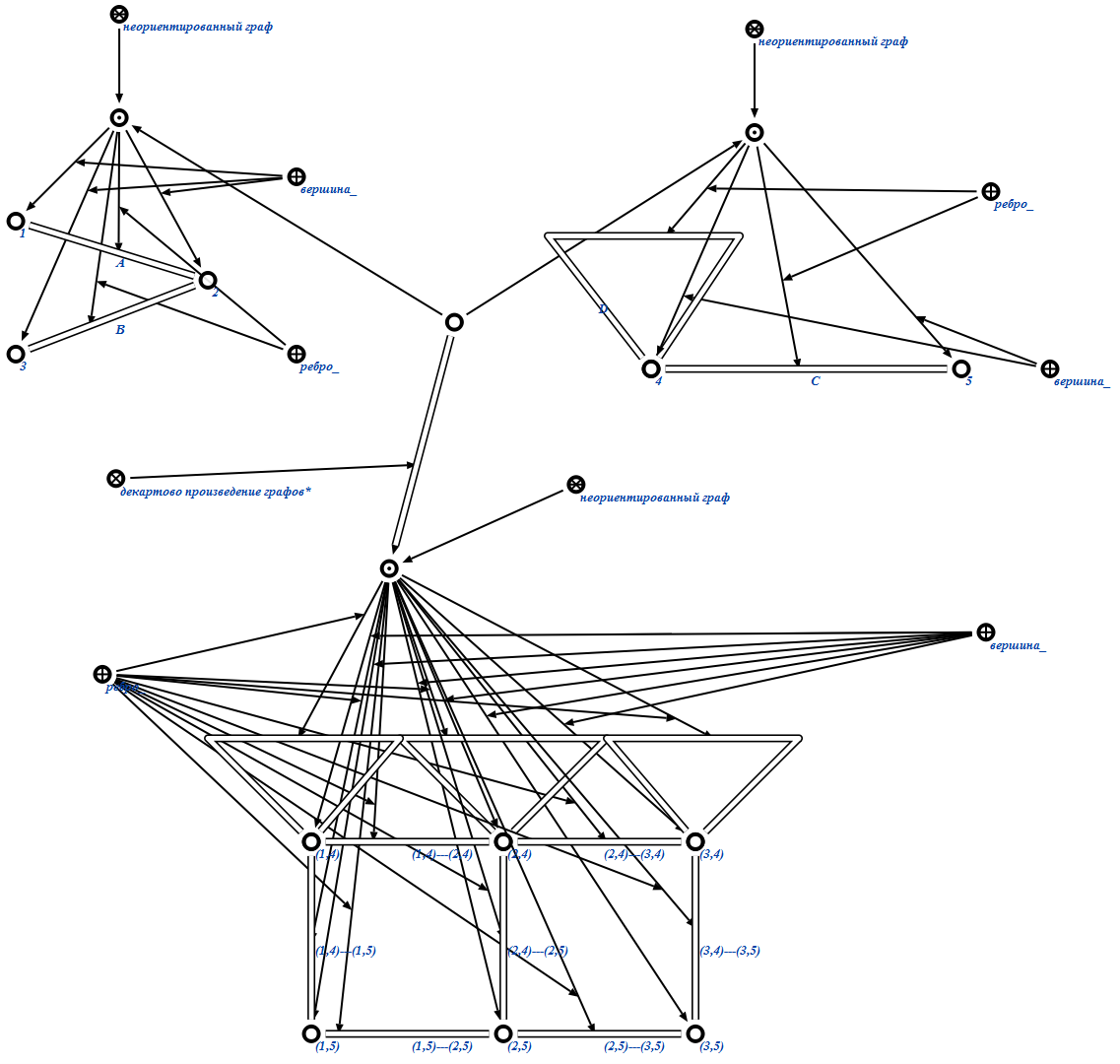


Рисунок 2.3 Полученное прямое произведение 2-х неориентированных графов

**Тест 2**

**Вход:**

Необходимо найти прямое произведение 2-х неориентированных графов.

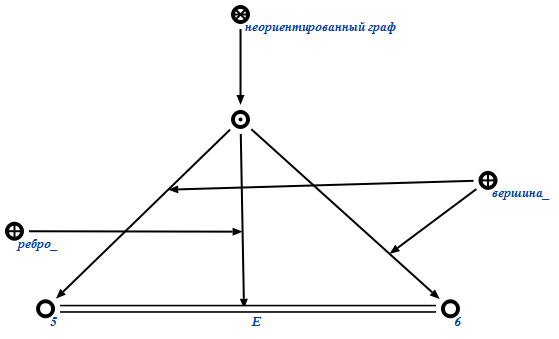
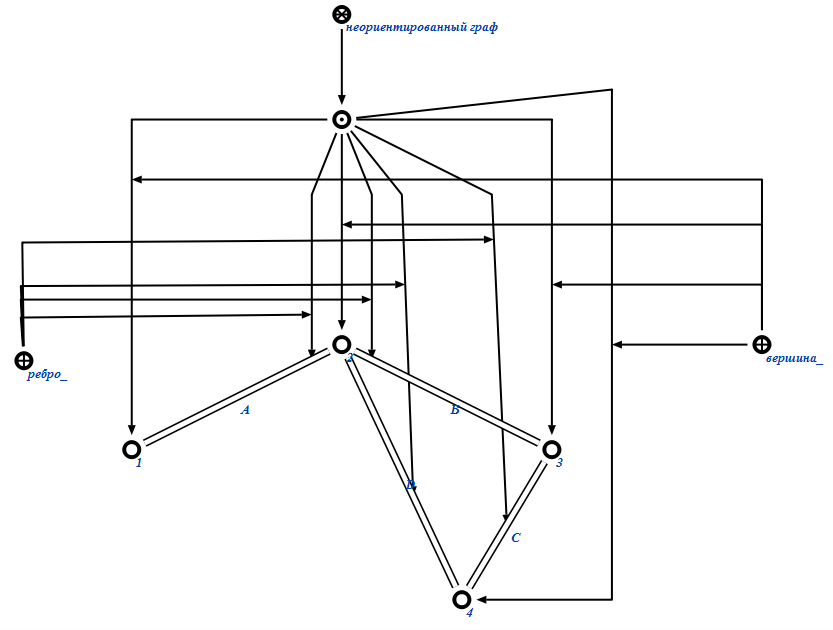


Рисунок 2.4 Первый исходный граф Рисунок 2.5 Второй исходный граф

**Выход:**

Будет найдено прямое произведение 2-х исходных графов:

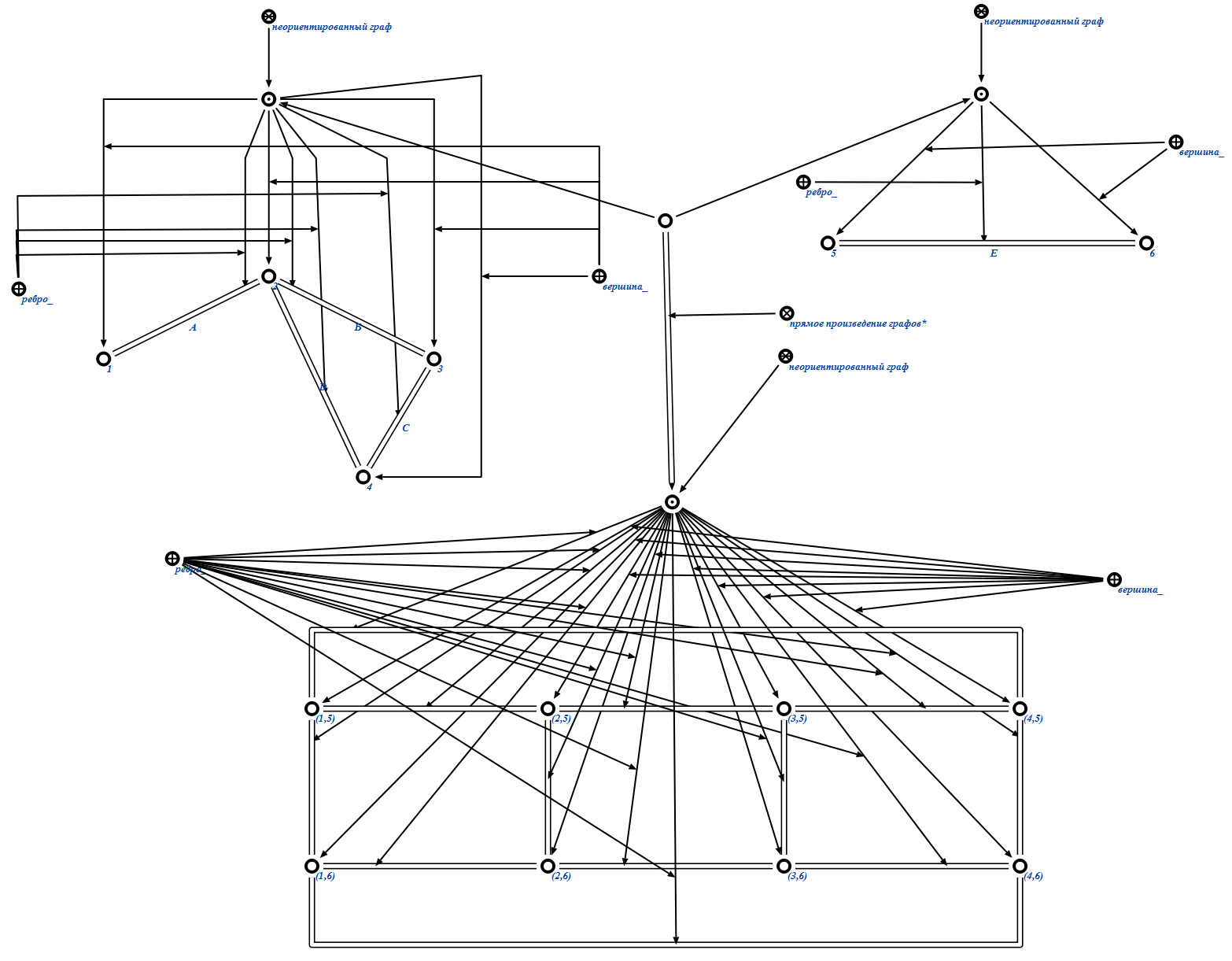
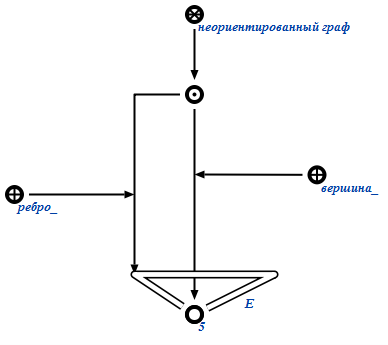


Рисунок 2.6 Полученное прямое произведение 2-х неориентированных графов

**Тест 3**

**Вход:**

Необходимо найти прямое произведение 2-х неориентированных графов.



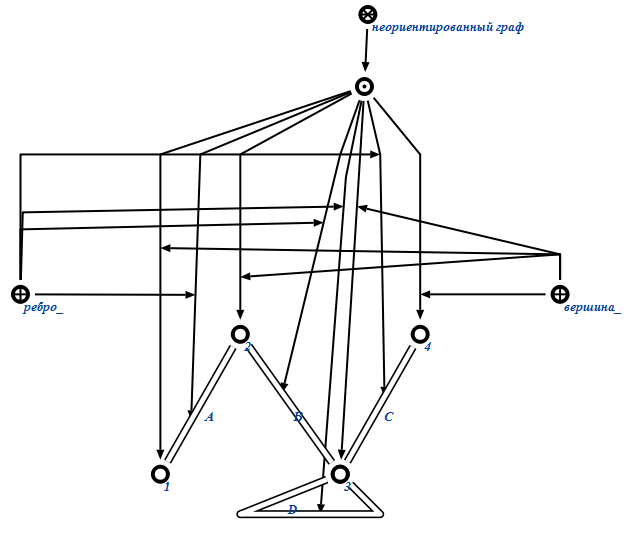


Рисунок 2.7 Первый исходный граф Рисунок 2.8 Второй исходный граф

**Выход:**

Будет найдено прямое произведение 2-х исходных графов:

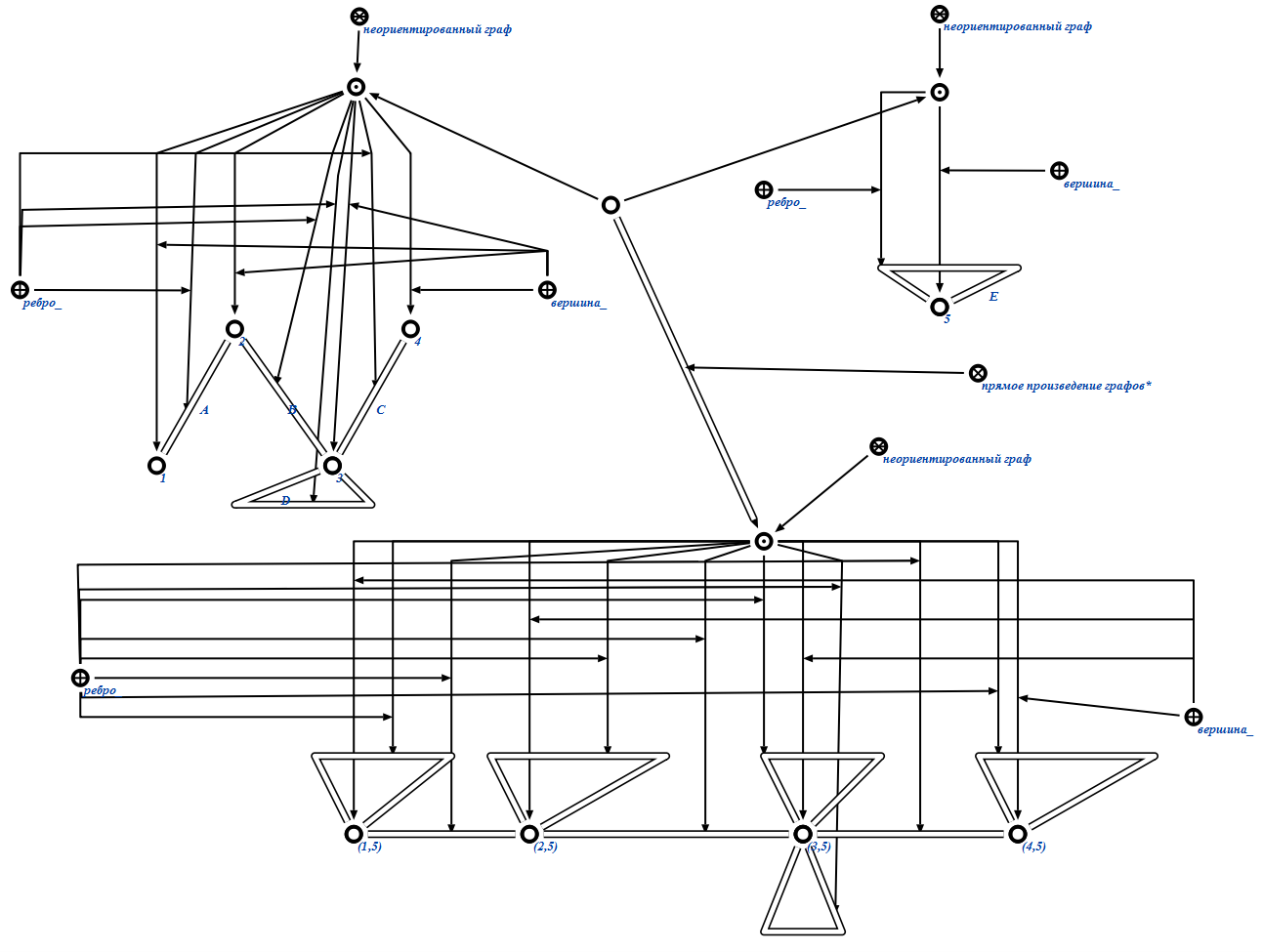


Рисунок 2.9 Полученное прямое произведение 2-х неориентированных графов

**Тест 4**

**Вход:**

Необходимо найти прямое произведение 2-х неориентированных графов.

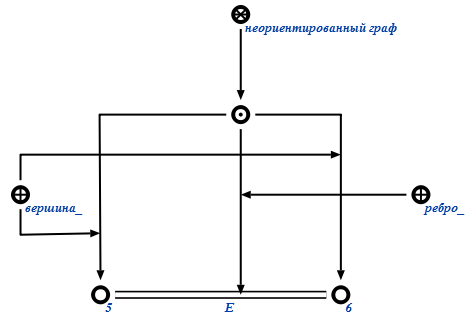
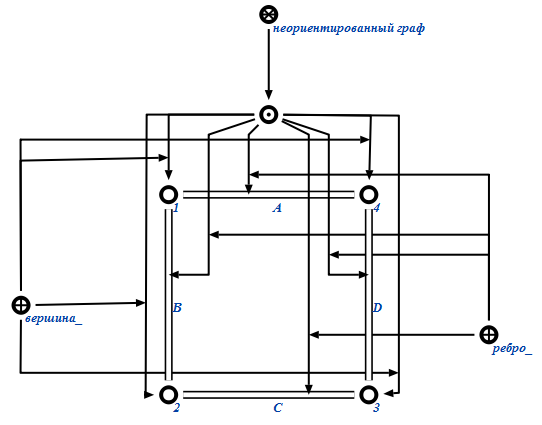


Рисунок 2.10 Первый исходный граф Рисунок 2.11 Второй исходный граф

**Выход:**

Будет найдено прямое произведение 2-х исходных графов:

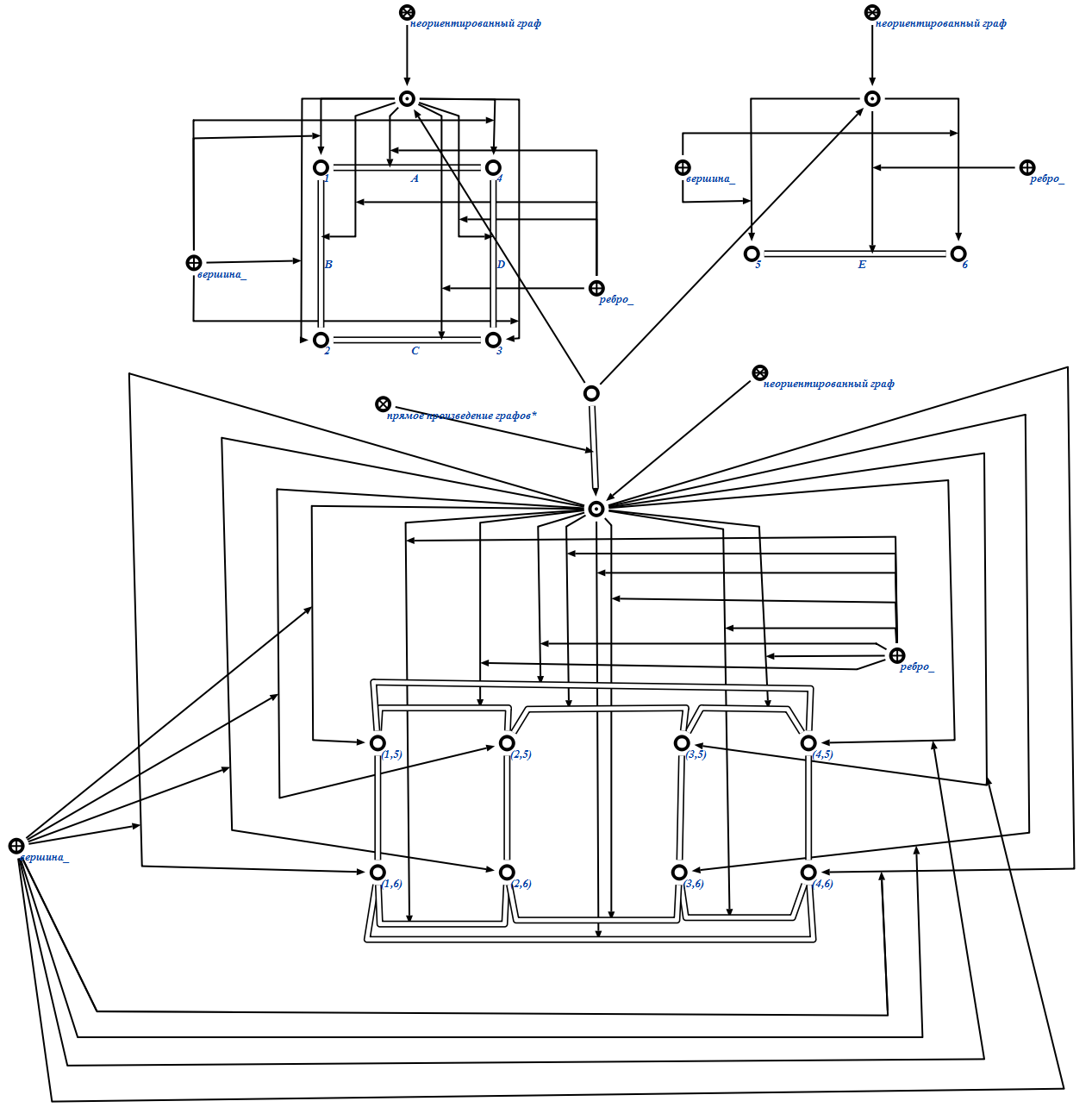
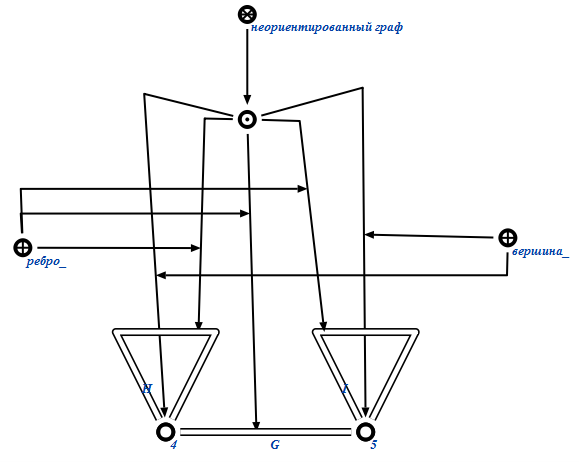


Рисунок 2.12 Полученное прямое произведение 2-х неориентированных графов

**Тест 5**

**Вход:**

Необходимо найти прямое произведение 2-х неориентированных графов.



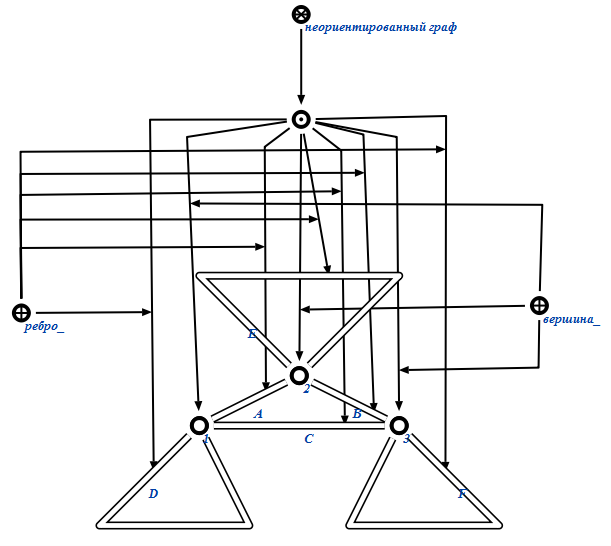


Рисунок 2.13 Первый исходный граф Рисунок 2.14 Второй исходный граф

**Выход:**

Будет найдено прямое произведение 2-х исходных графов:

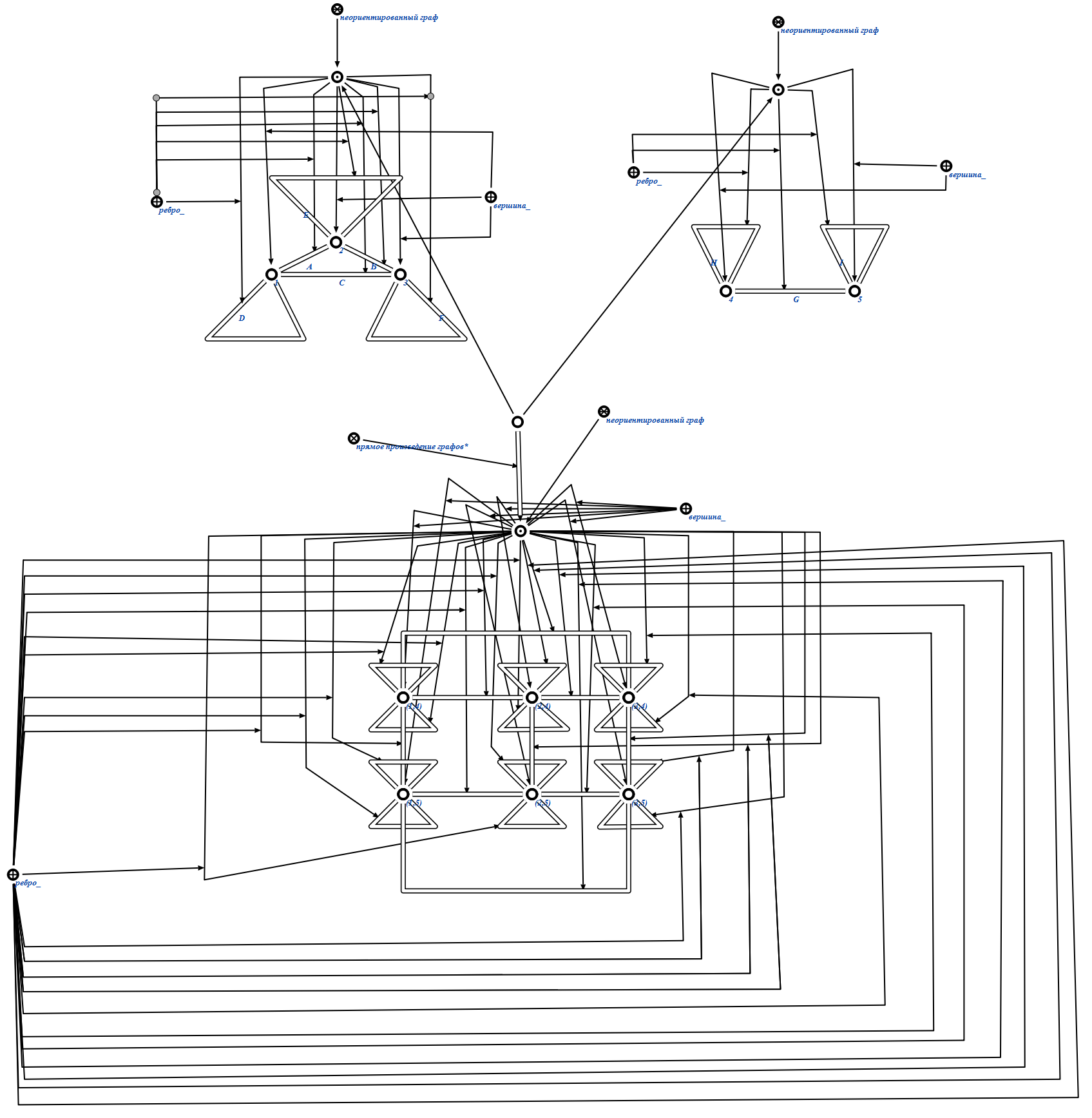


Рисунок 2.15 Полученное прямое произведение 2-х неориентированных графов

**3.Алгоритм**

**3.1.Основные понятия задачи**

Задачей является поиск прямого произведения 2-х неориентированных графов. Остановимся подробнее на определениях использованных понятий.

Графом называется множество вершин и связок, соединяющих вершины, в классических графах связки являются бинарными.

**3.2.Ключевые узлы**

Ключевыми узлами для реализуемой операции являются sc-элементы, с заданной семантикой.

***неориентированный граф*** – множество, элементами которого являются неориентированные графы.

***вершина\_*** – атрибутивное отношение, связывающее узел графа с его вершиной.

***ребро\_*** – атрибутивное отношение, связывающее узел графа с его ребром.

**3.3.Описание идеи алгоритма**

Функционирование любой операции можно разделить на следующие этапы:

1. Проверка появления задачной ситуации.
2. Выполнение алгоритма, приводящего к решению задачи, при появлении задачной ситуации.
3. Сборка мусора, т.е. удаление всех временных конструкций, созданных при выполнении алгоритма.
4. Разрушение задачной ситуации.

Опишем общую структуру алгоритма, приводящего к решению задачи. В алгоритме осуществляется прямое произведение вершин 2-х исходных графов, а впоследствии потроение между ними дуг до тех пор, пока не получится прямое произведение 2-х исходных графов.

В соответствии с вышеприведённым описанием, для разрабатываемой операции можно составить крупноблочную структуру алгоритма:

1. Выполняем прямое произведение вершин 2-х графов.
2. Количество вершин получается равным произведению вершин из 2-х исходных графов
3. В полученных вершинах выбираем группу вершин, образованных с помощью какой-то одной вершины Х в исходных графах.
4. Эта група вершин была также образована с помощью вторых каких-то вершин Y (т.е. полученные вершины образованы исходными вершинами <X,Y>, причём Х – одно значение, а Y может быть разным, к тому же Х принадлежит одному графу, а Y другому).
5. Если в графе, содержащем Y, между Y1 и Y2 есть дуга, то вершины <X, Y1 > и < X, Y2 > также связываются дугами.
6. После того как в выбранной группе вершин, образованных с помощью Х, рассмотрятся все пары Y, и будут построены все нужные дуги, то вершина Х обозначается пройденной, и иксом обозначается другая вершина.  
   Это происходит до тех пор, пока все вершины в 2-х исходных графах не будут иметь статус пройденных.
7. Если все вершины имеют статус пройденных, то завершаем работу.

**3.4.Переменные программы**

Переменными программы являются:

***\_current\_ties*** – значением этой переменной является sc-узел обозначающий вершины графов, к которым в данный момент мы обращаемся.

***\_visited\_ties***  – значением этой переменной является sc-узел, обозначающий вершины графов, к которым уже произошло обращение.

***\_current\_stick*** – значением этой переменной является sc-узел обозначающий ребро графа, к которому в данный момент мы обращаемся.

***\_visited\_stick*** – значением этой переменной является sc-узел, обозначающий ребра графов, к которым уже произошло обращение.

**3.5.Пример решения**

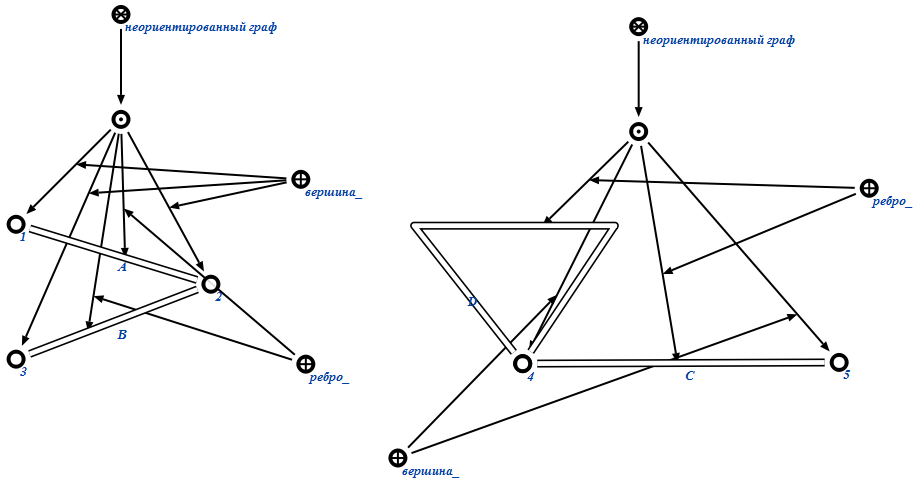


Рисунок 3.5.1.Исходные графы

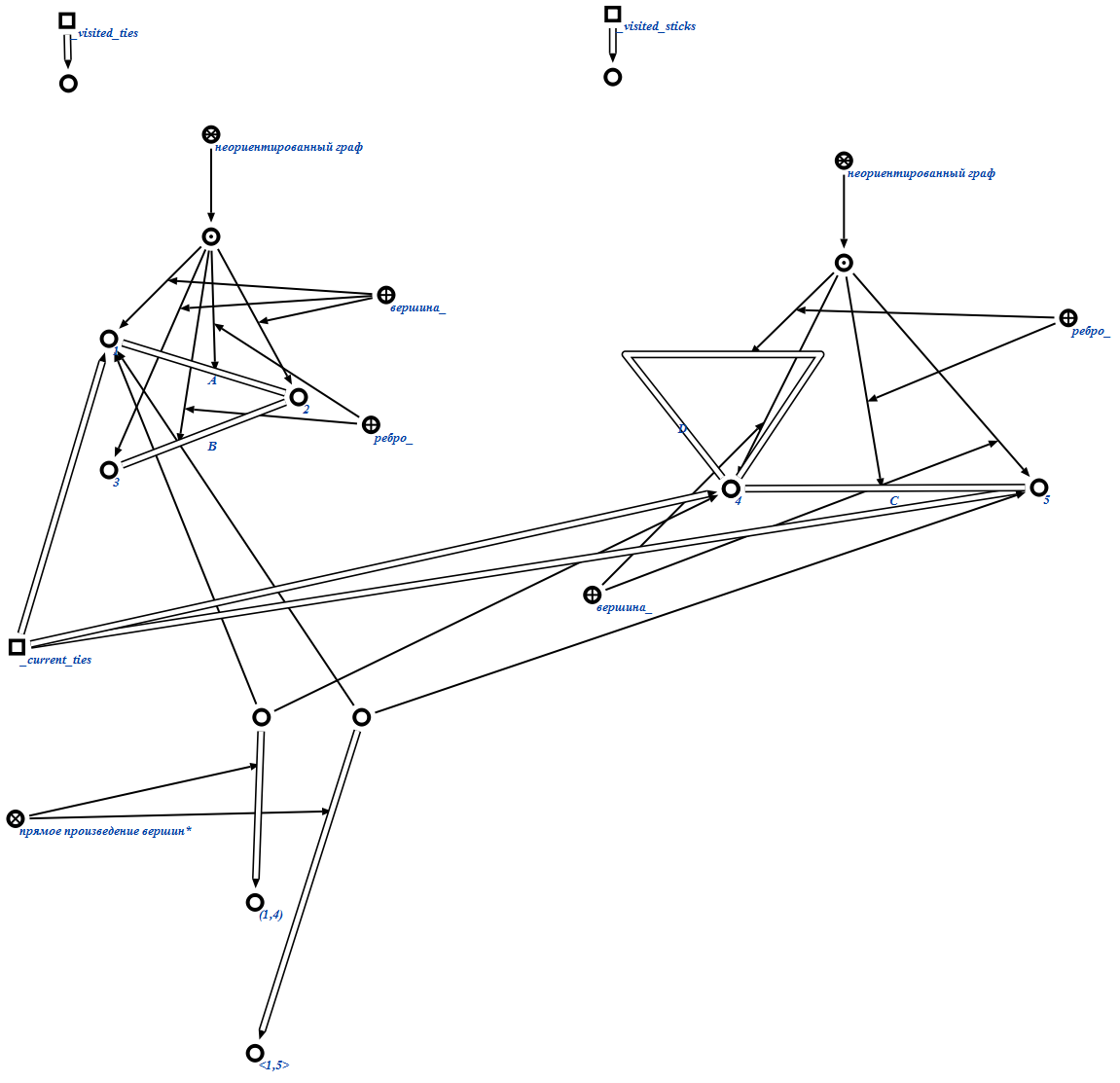


Рисунок 3.5.2.Прямое произведение вершин

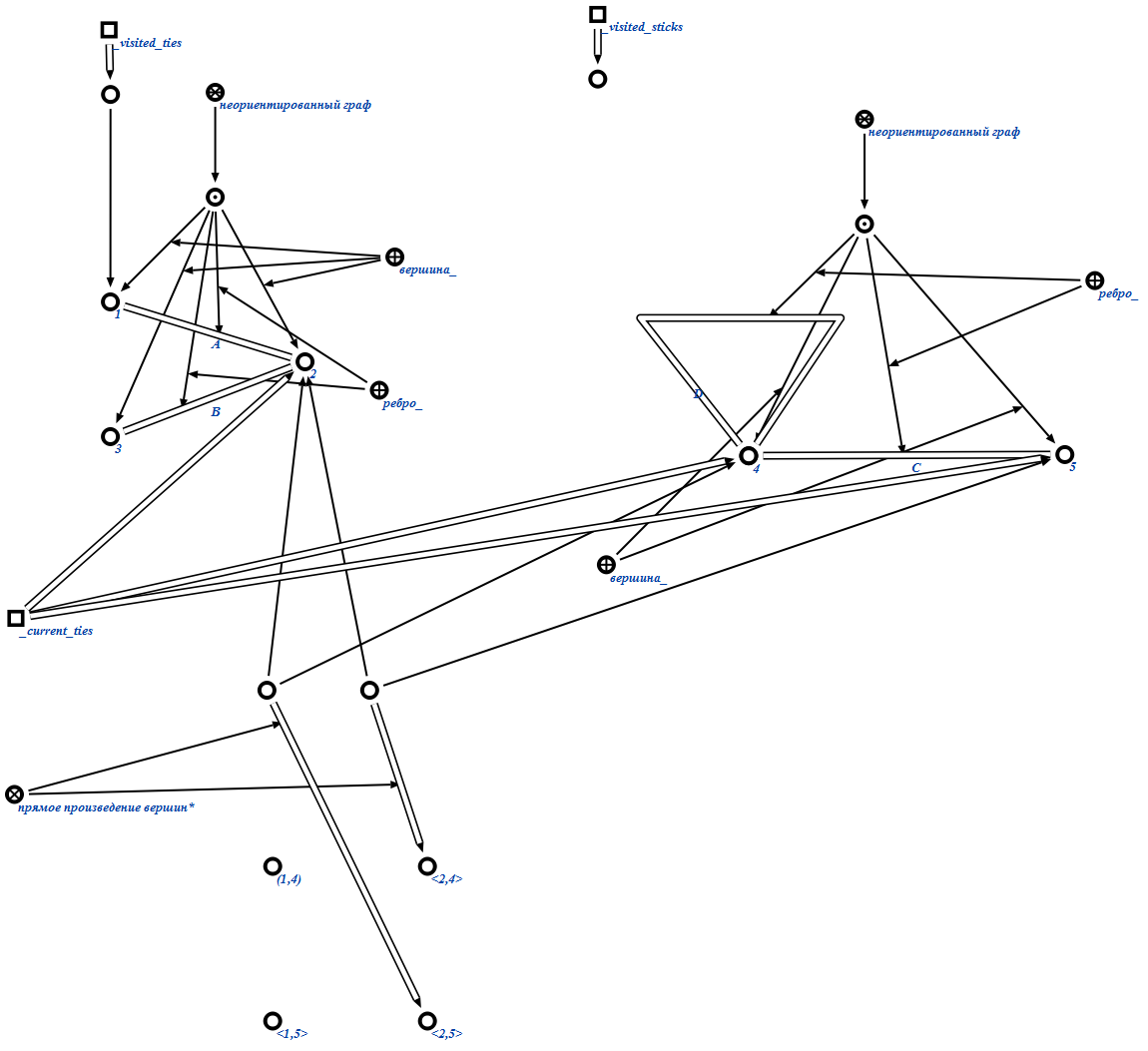


Рисунок 3.5.3.Прямое произведение вершин

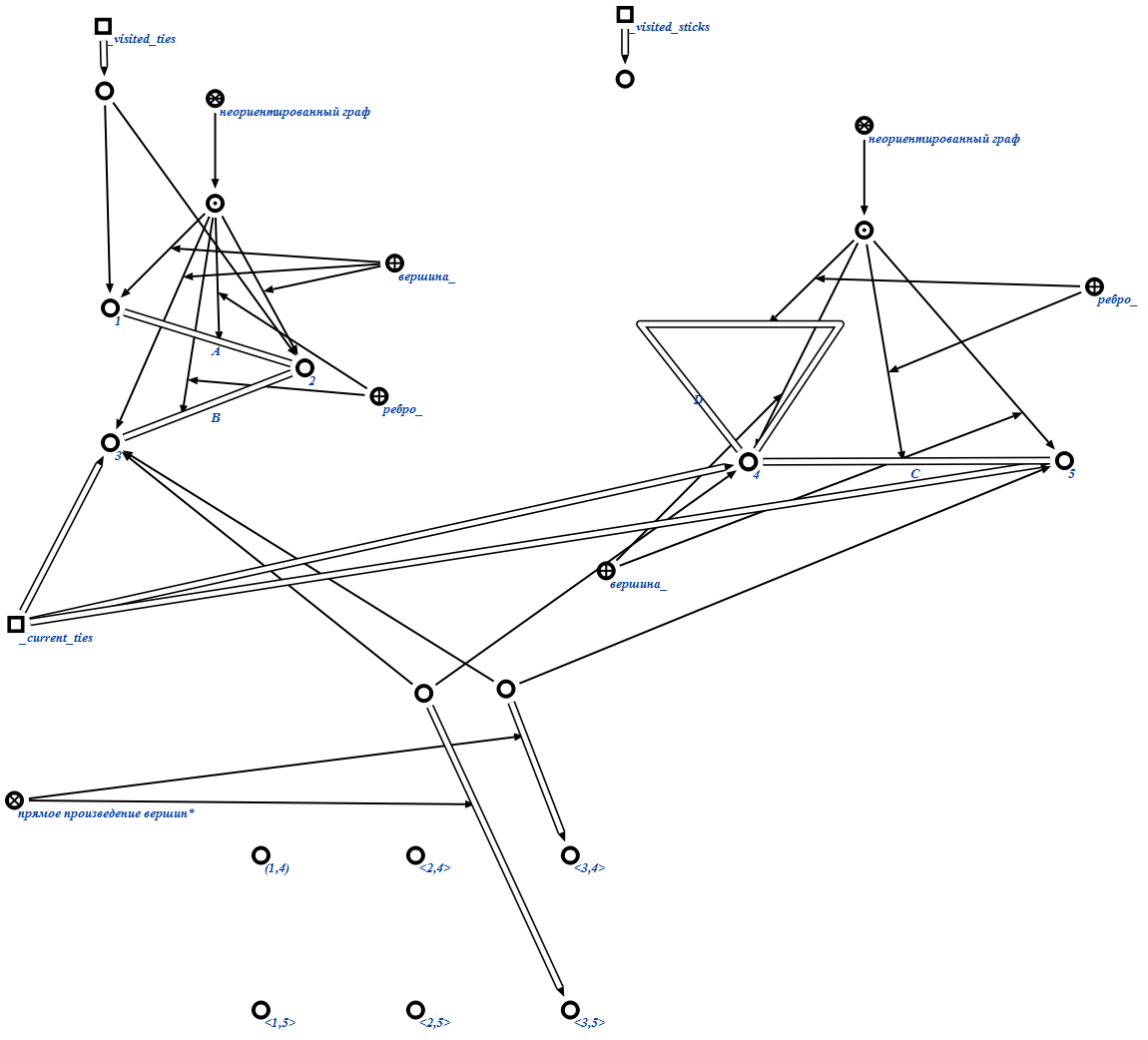


Рисунок 3.5.4.Прямое произведение вершин

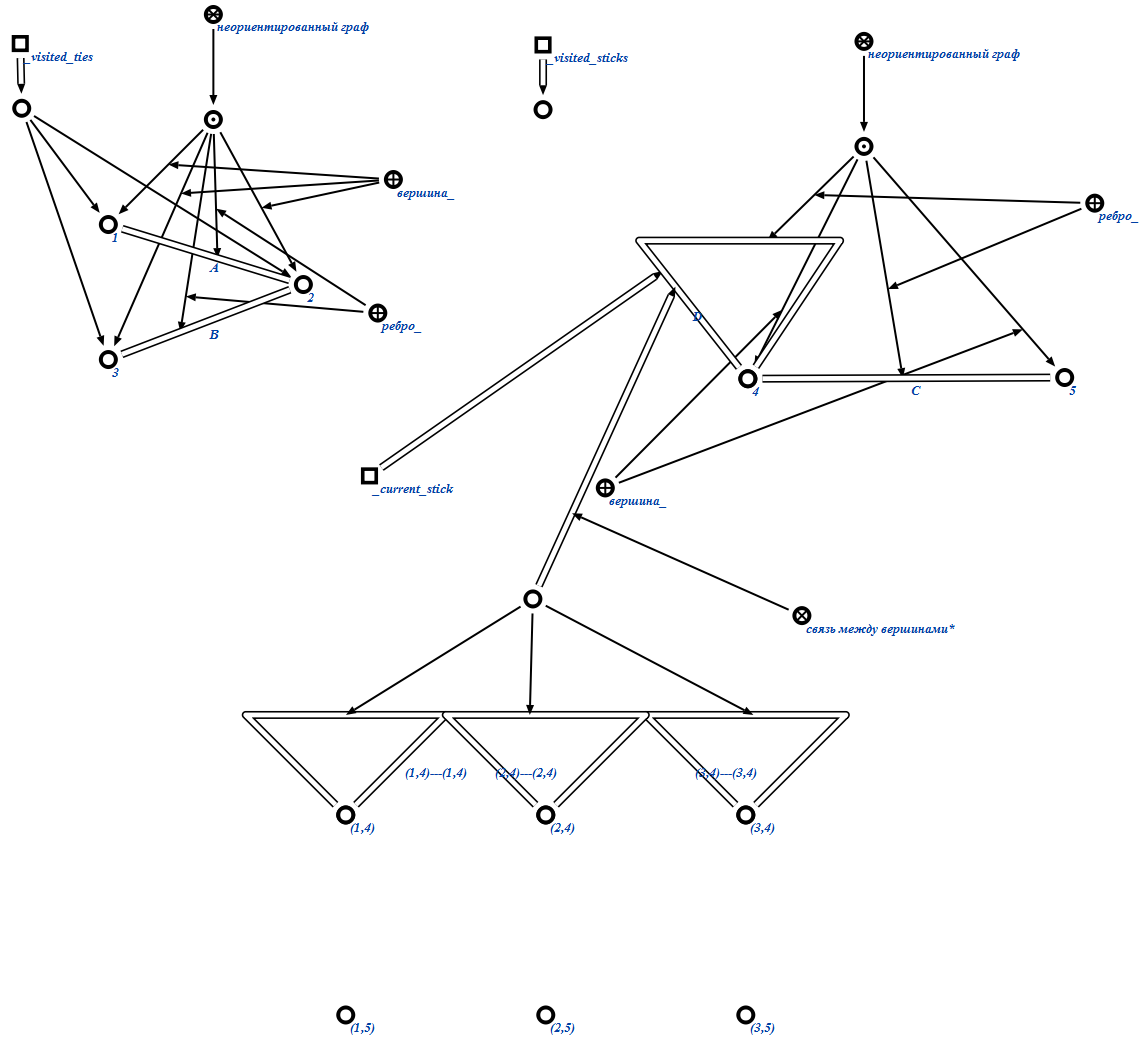


Рисунок 3.5.5. Построение рёбер между вершинами образованные с помощью узла “4”

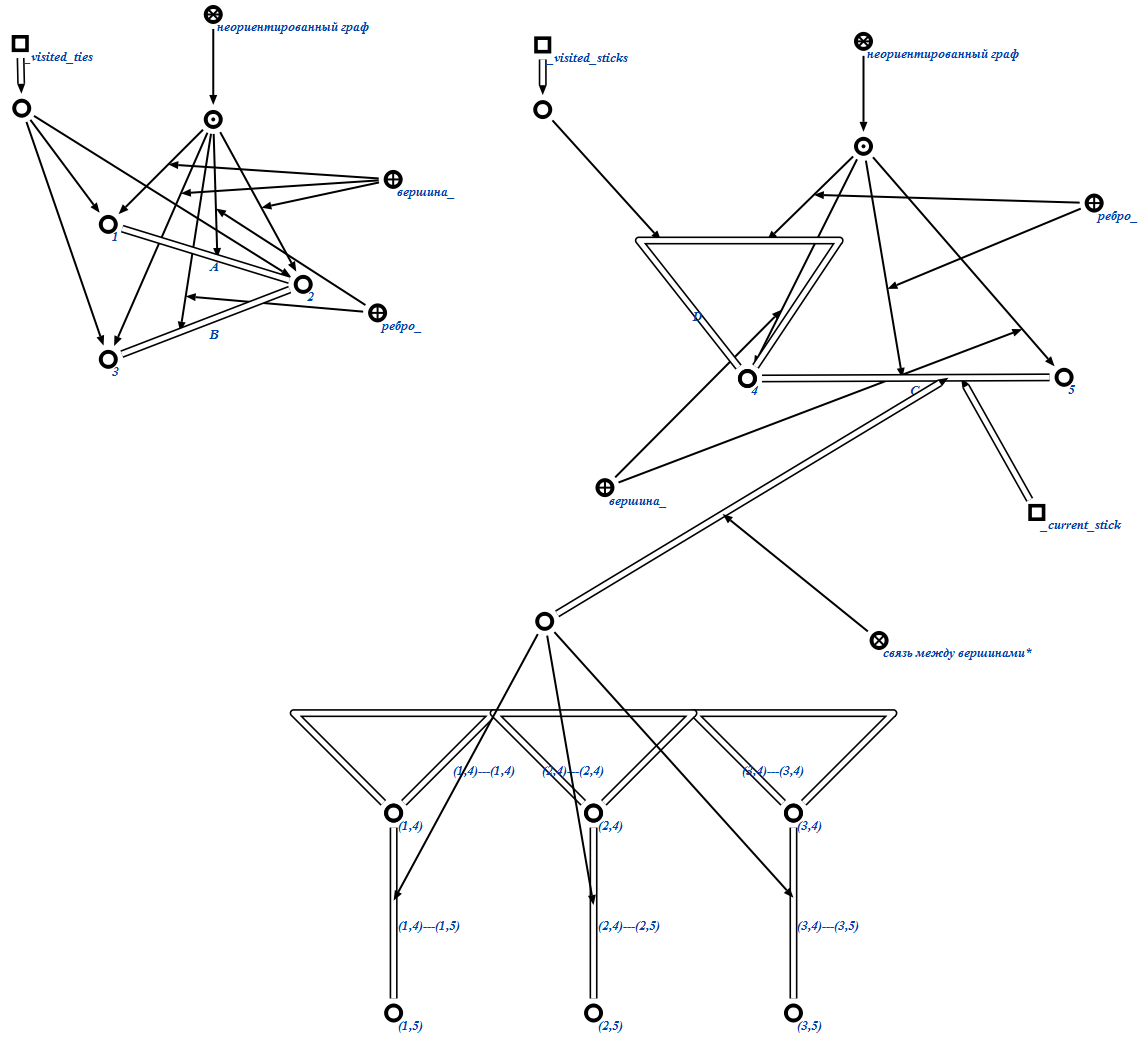


Рисунок 3.5.6. Построение рёбер между вершинами образованных с помощью узлов “4” и “5”

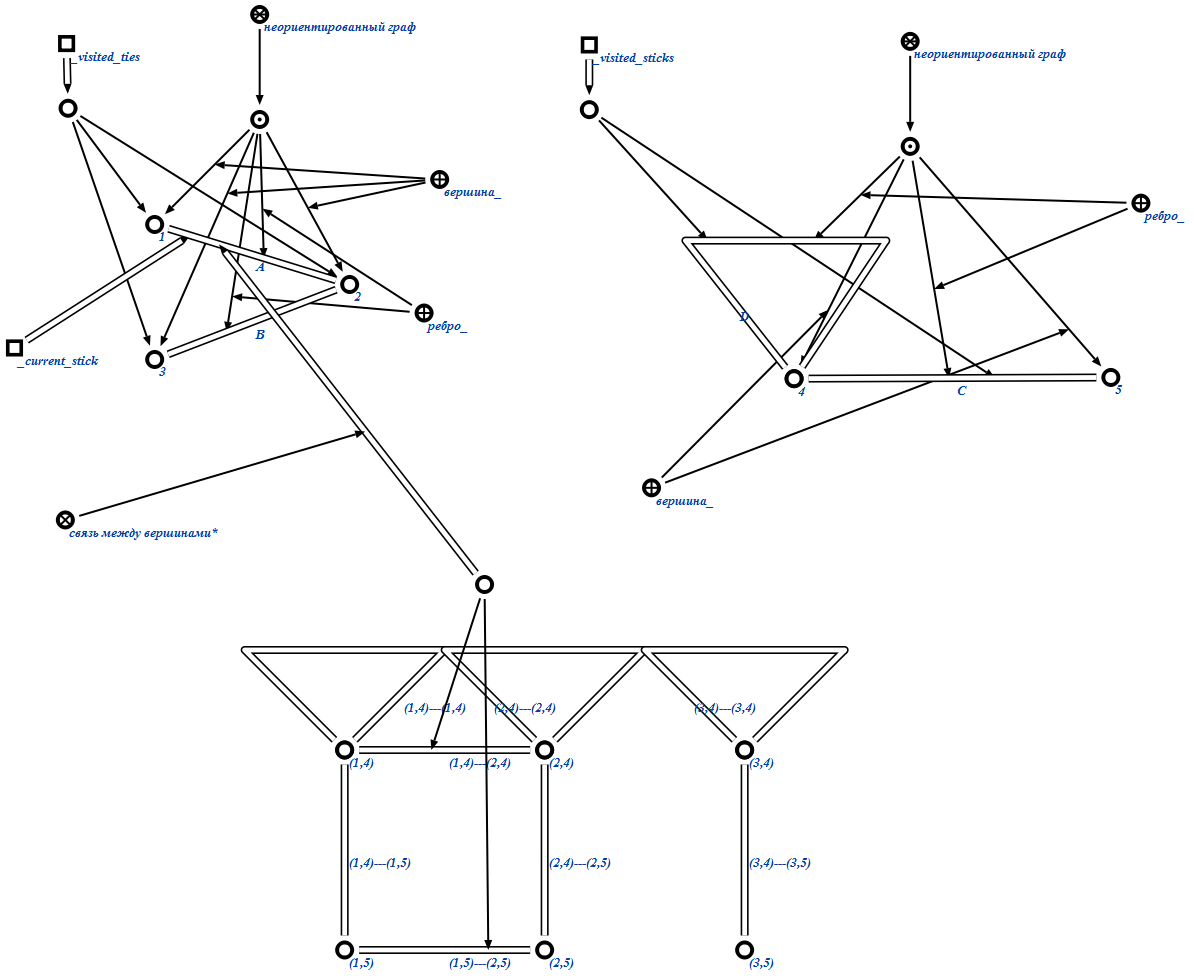


Рисунок 3.5.7. Построение рёбер между вершинами образованных с помощью узлов “1” и “2”

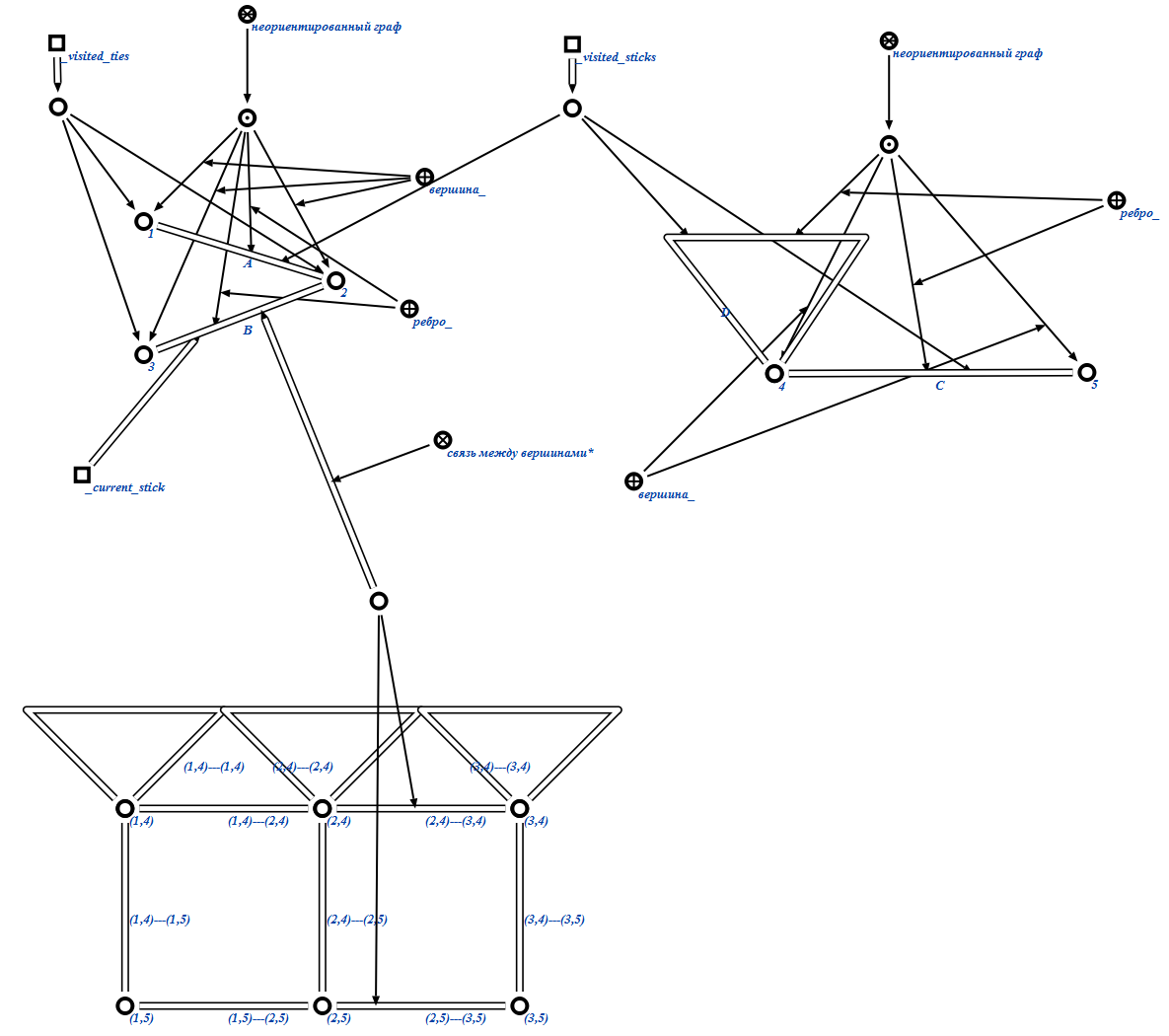


Рисунок 3.5.8. Построение рёбер между вершинами образованных с помощью узлов “2” и “3”

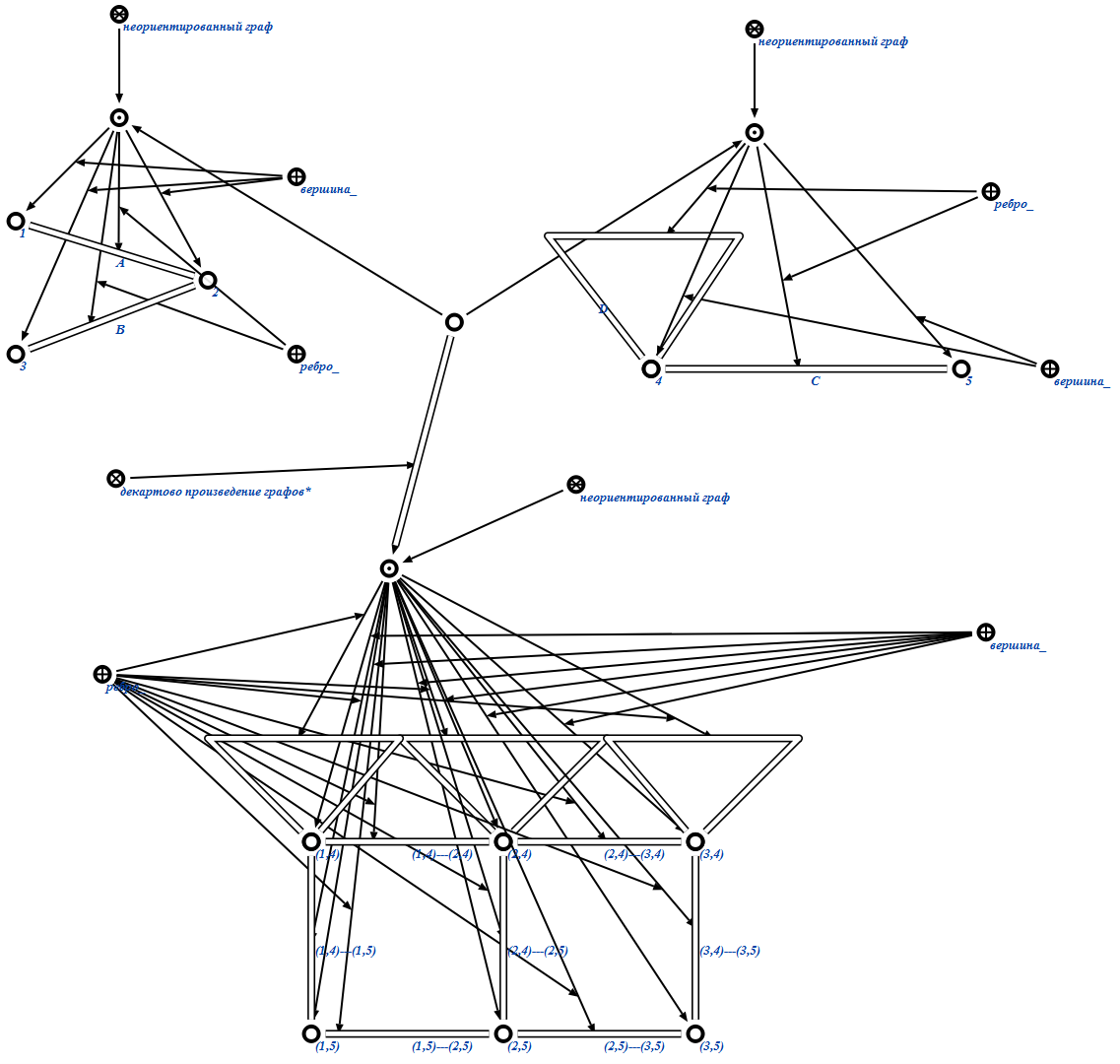


Рисунок 3.5.9. Полученное прямое произведение 2-х графов

Литература

* *Джозеф Джарратано, Гари Райли* **Глава 2. Представление знаний** // Экспертные системы: принципы разработки и программирование = Expert Systems: Principles and Programming. — 4-е изд. — М.: [«Вильямс»](http://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2006. — С. 1152. — [ISBN 978-5-8459-1156-8](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/9785845911568)
* Amaravadi, C. S., "Knowledge Management for Administrative Knowledge, " Expert Systems, 25(2), pp 53–61, May 2005.
* [Ronald J. Brachman](http://ru.wikipedia.org/w/index.php?title=Ronald_J._Brachman&action=edit&redlink=1); What IS-A is and isn’t. An Analysis of Taxonomic Links in Semantic Networks; IEEE Computer, 16 (10); October 1983
* Jean-Luc Hainaut, Jean-Marc Hick, Vincent Englebert, Jean Henrard, Didier Roland: Understanding Implementations of IS-A Relations. ER 1996: 42-57
* Hermann Helbig: *Knowledge Representation and the Semantics of Natural Language*, Springer, Berlin, Heidelberg, New York 2006
* Arthur B. Marкмan: *Knowledge Representation* Lawrence Erlbaum Associates, 1998
* Michael Negnevitsky: *Artificial Intelligence, A Guide to Intelligent Systems*, Pearson Education Limited, 2002
* John F. Sowa: *Knowledge Representation*: Logical, Philosophical, and Computational Foundations. Brooks/Cole: New York, 2000
* Adrian Walker, Michael McCord, John F. Sowa, and Walter G. Wilson: *Knowledge Systems and Prolog*, Second Edition, Addison-Wesley, 1990
* [Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. — Запоріжжя: ЗНТУ, 2008. — 341 с.](http://www.csit.narod.ru/subject/MOPZ/mopz_book.pdf)