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 «Традиционные и интеллектуальные информационные технологии»
на тему
Задача проверки ориентированного графа на ацикличность

Выполнил студент группы 021703 — Шереметьев М.А.

Проверил — Юрков А.А

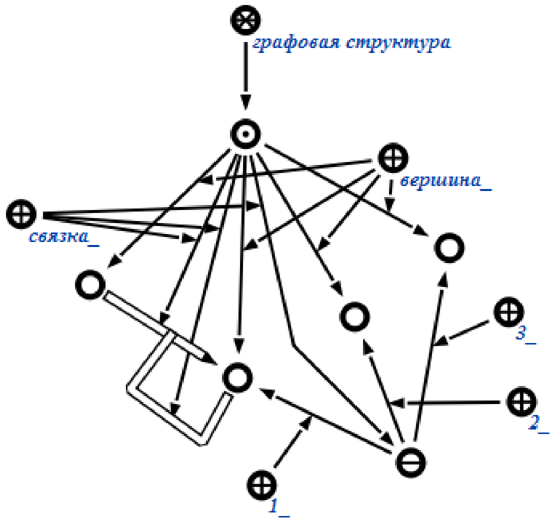
Минск 2021

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

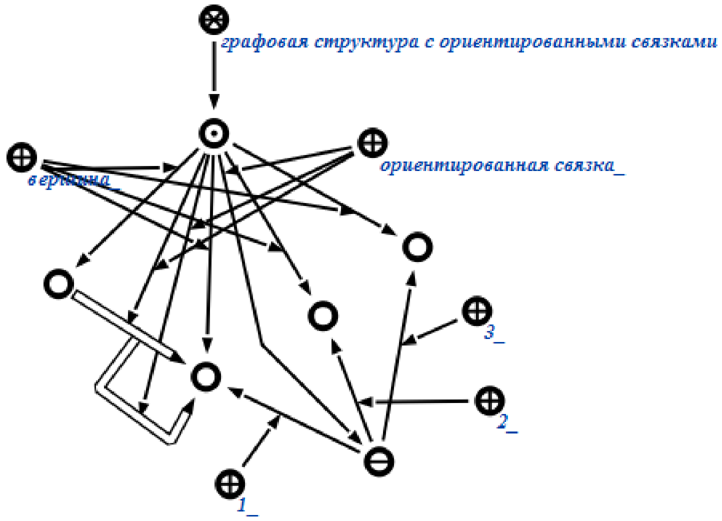
Задача: проверка графа на ацикличность

1 Список понятий

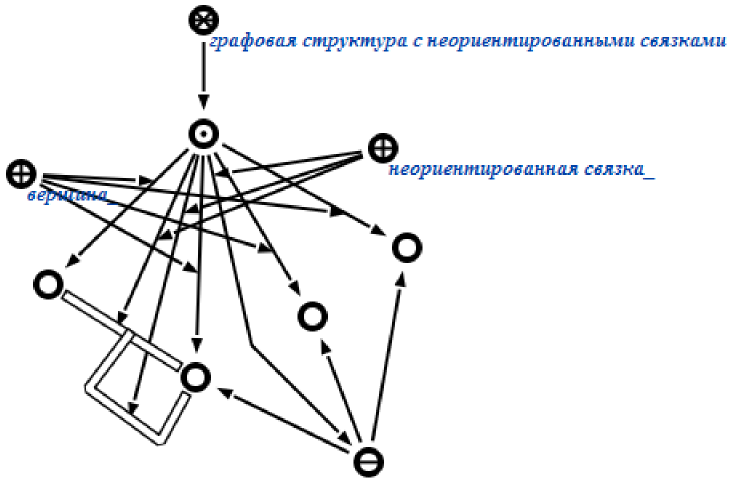
1. Графовая структура (абсолютное понятие) - это такая одноуровневая реляционная структура, объекты которой могут играть роль либо вершины, либо связки:
    a. Вершина (относительное понятие, ролевое отношение);
    b. Связка (относительное понятие, ролевое отношение).



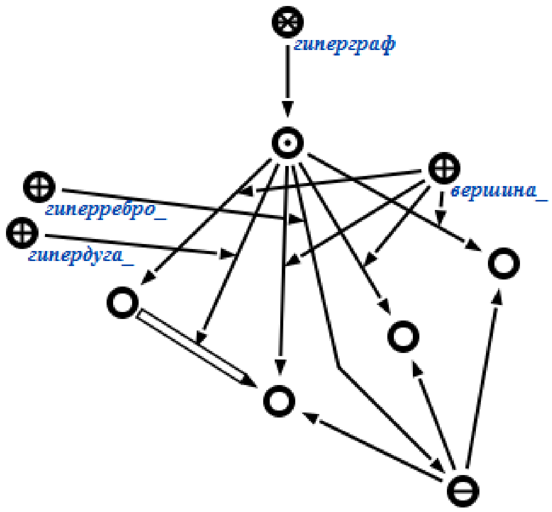
2. Графовая структура с ориентированными связками (абсолютное понятие)
    a. Ориентированная связка (относительное понятие, ролевое отношение) – связка, которая задается ориентированным множеством.



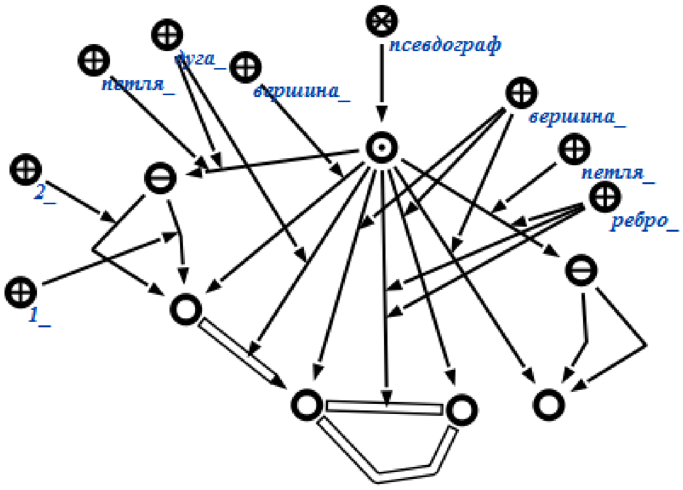
3. Графовая структура с неориентированными связками (абсолютное понятие)
    a. Неориентированная связка (относительное понятие, ролевое отношение) – связка, которая задается неориентированным множеством.



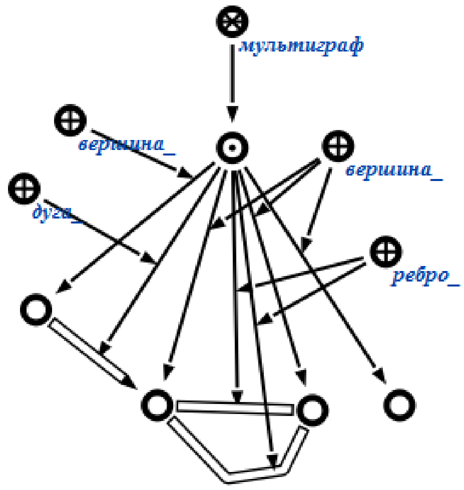
4. Гиперграф (абсолютное понятие) – это такая графовая структура, в которой связки могут связывать только вершины:
    a. Гиперсвязка (относительное понятие, ролевое отношение);
    b. Гипердуга (относительное понятие, ролевое отношение) – ориентированнаягиперсвязка;
    c. Гиперребро (относительное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



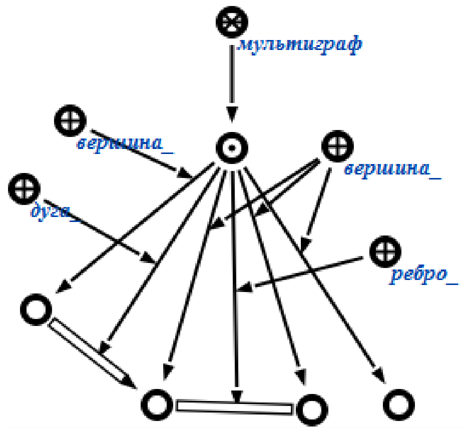
5. Псевдограф (абсолютное понятие) – это такой гиперграф, в котором все связки должны быть бинарными:
    a. Бинарная связка (относительное понятие, ролевое отношение) –гиперсвязка арности 2;
    b. Ребро (относительное понятие, ролевое отношение) – неориентированнаягиперсвязка;
    c. Дуга (относительное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



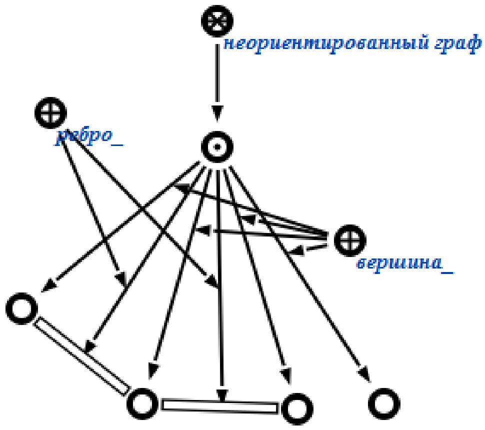
6. Мультиграф (абсолютное понятие) – это такой псевдограф, в котором не может быть петель:



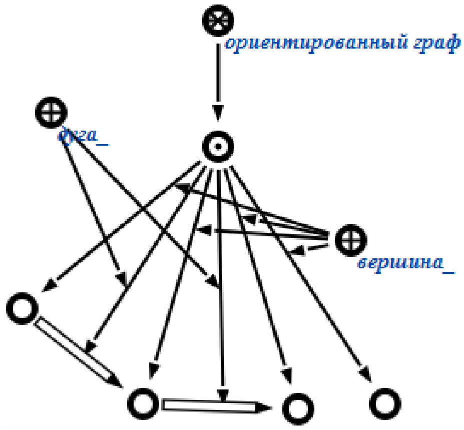
7. Граф (абсолютное понятие) – это такой мультиграф, в котором не может быть кратных связок, т.е. связок у которых первый и второй компоненты совпадают:



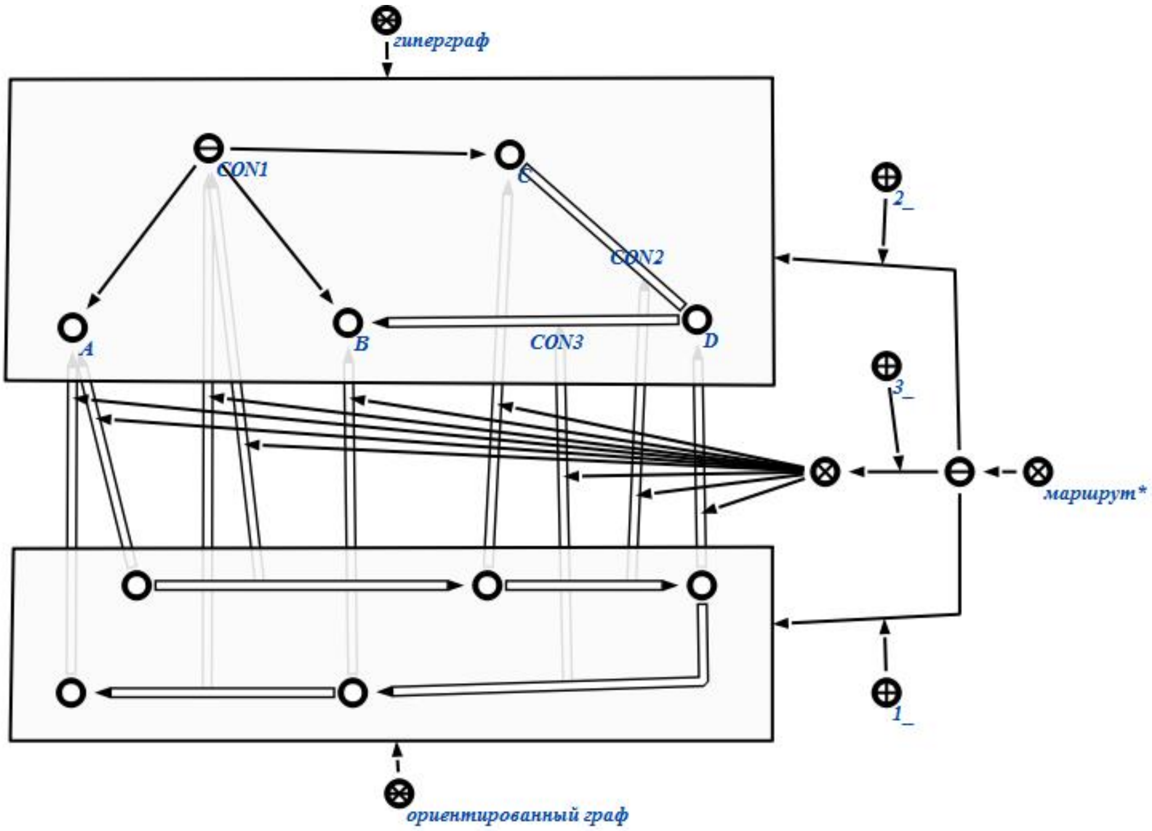
8. Неориентированный граф (абсолютное понятие) –это такой граф, в котором все связки являются ребрами:



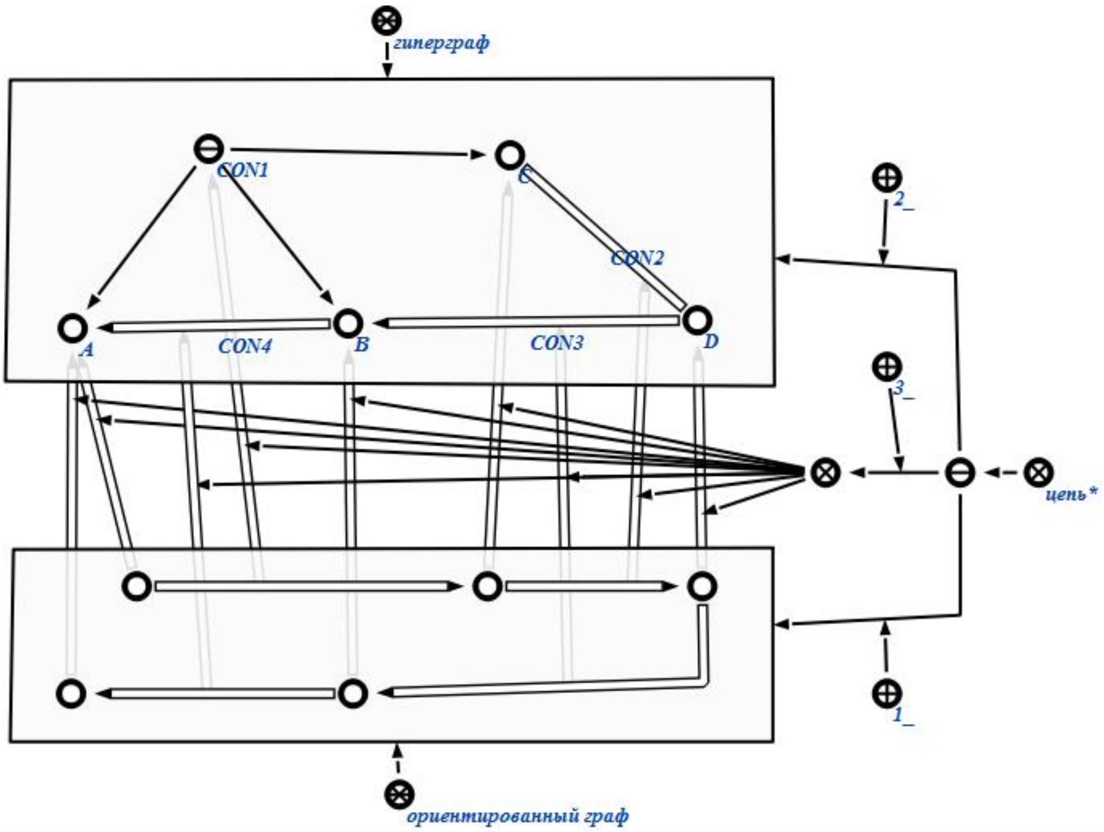
9. Ориентированный граф (абсолютное понятие) - это такой граф, в котором все связки являются дугами:



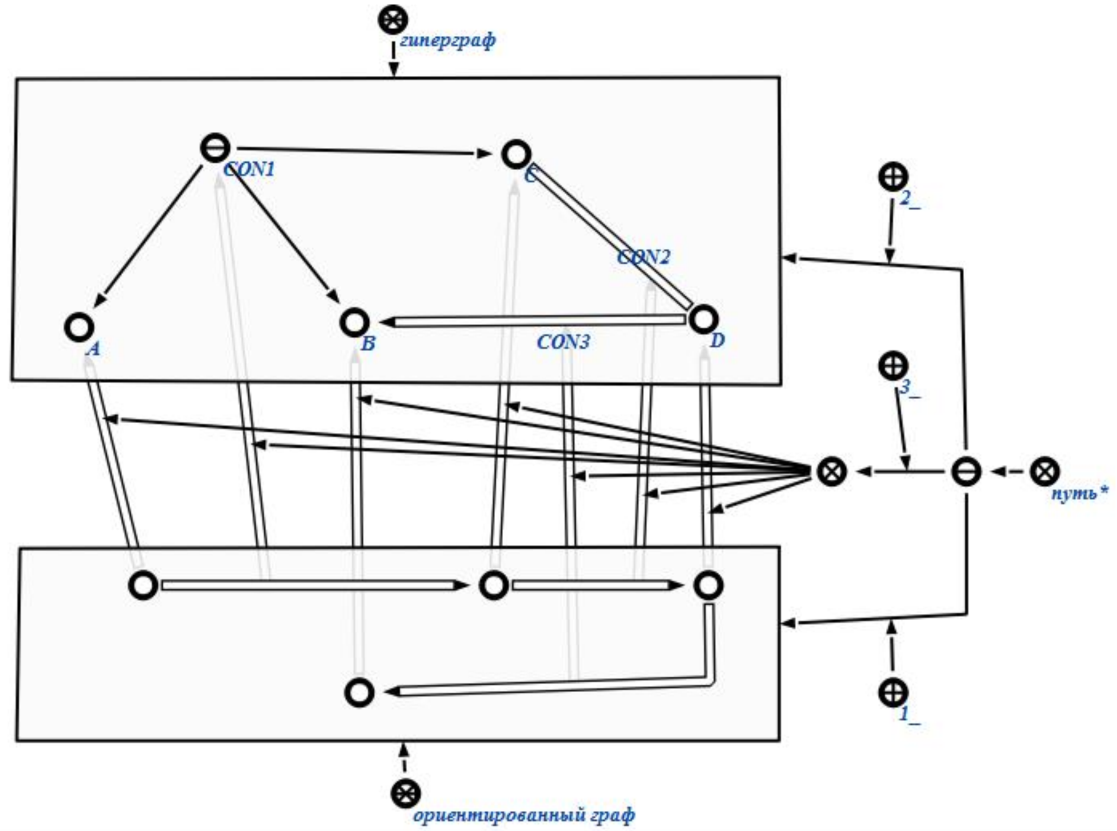
10. Маршрут (относительное понятие, бинарное ориентированное отношение) – это чередующаяся последовательность вершин и гиперсвязок в гиперграфе, которая начинается и кончается вершиной, и каждая гиперсвязка последовательности инцидентна двум вершинам, одна из которых непосредственно предшествует ей,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



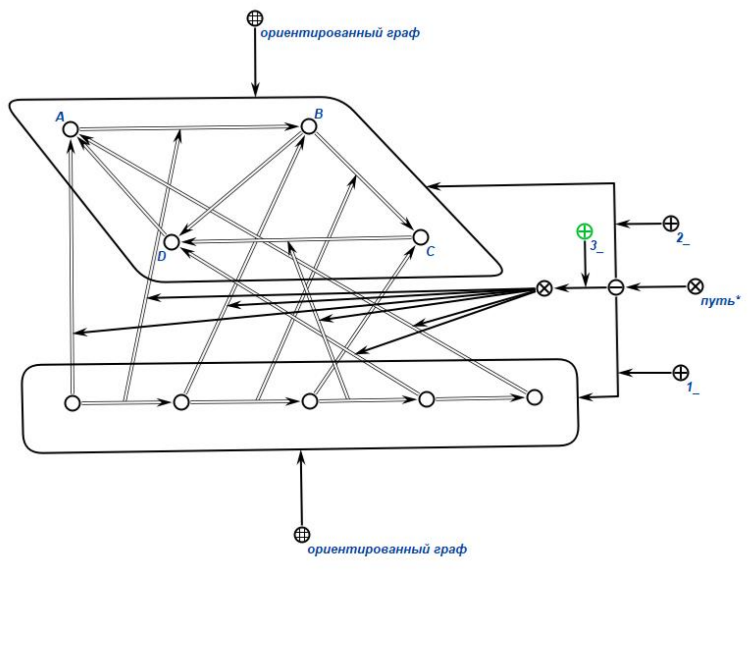
11. Цепь (относительное понятие, бинарное ориентированное отношение) – это маршрут, все гиперсвязки которого различны. В примере ниже показана цепь *A*, *CON1*, *C*, *CON2*, *D*, *CON3*, *B*, *CON4*, *A* в гиперграфе.



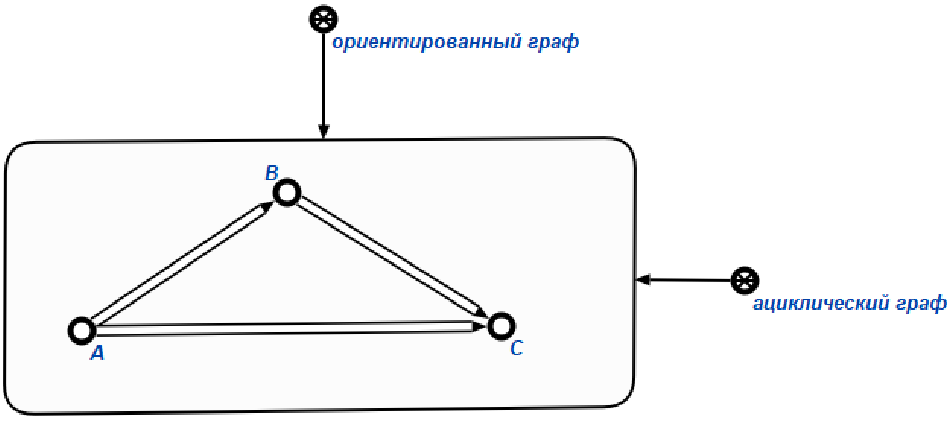
12. Простая цепь, путь (относительное понятие, бинарное ориентированное отношение) – это цепь, в которой все вершины различны. В примере ниже показан путь *A*, *CON1*, *C*, *CON2*, *D*, *CON3*, *B* в гиперграфе.



13. Цикл (относительное понятие, бинарное ориентированное отношение) – это замкнутый путь, который начинается и заканчивается в одной и той же вершине, и каждые две последовательные вершины в пути смежны. В примере ниже цикл: A, B, C, D, A.



14. Ациклический граф (относительное понятие, бинарное ориентированное отношение) – это граф, не имеющий циклов. В примере ниже ациклический граф:



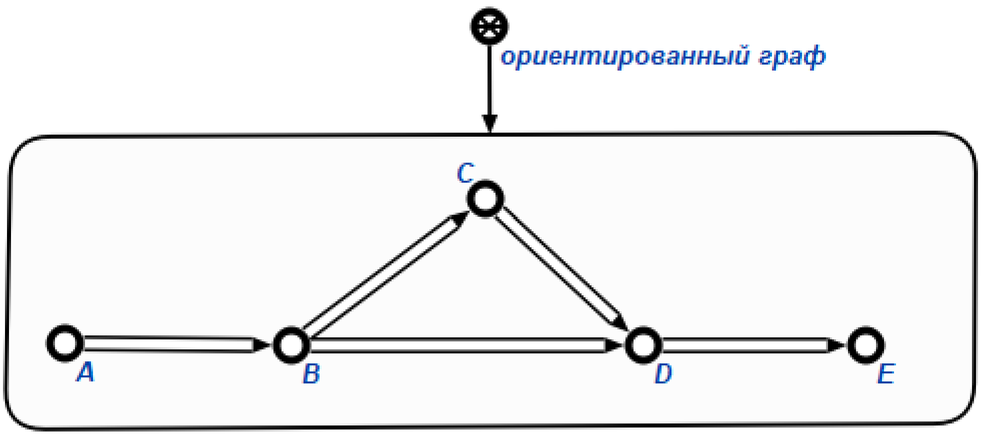
2 Тестовые примеры

Во всех тестах графы будет приведены в сокращенной форме со скрытыми ролями элементов графа.

2.1 Тест 1

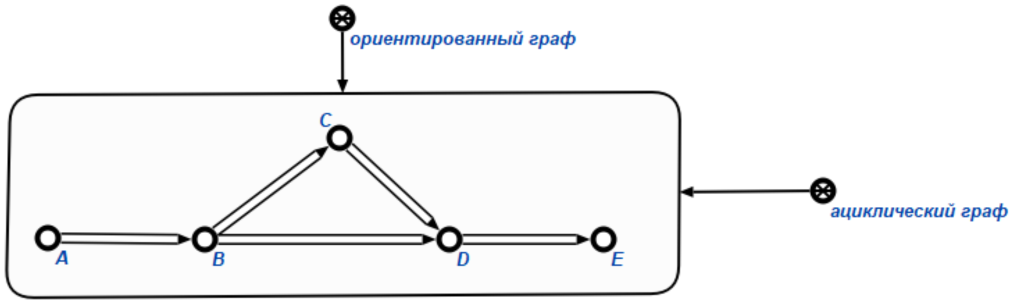
Вход:

Определить, является ли входной граф ациклическим.



Выход:

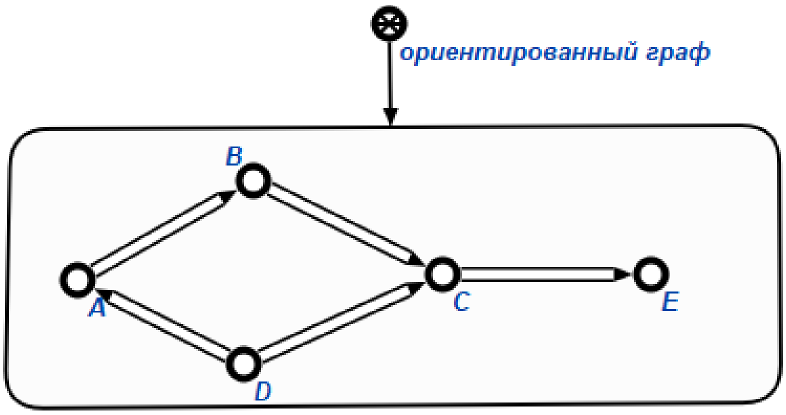
Входной граф ациклический:



2.2 Тест 2

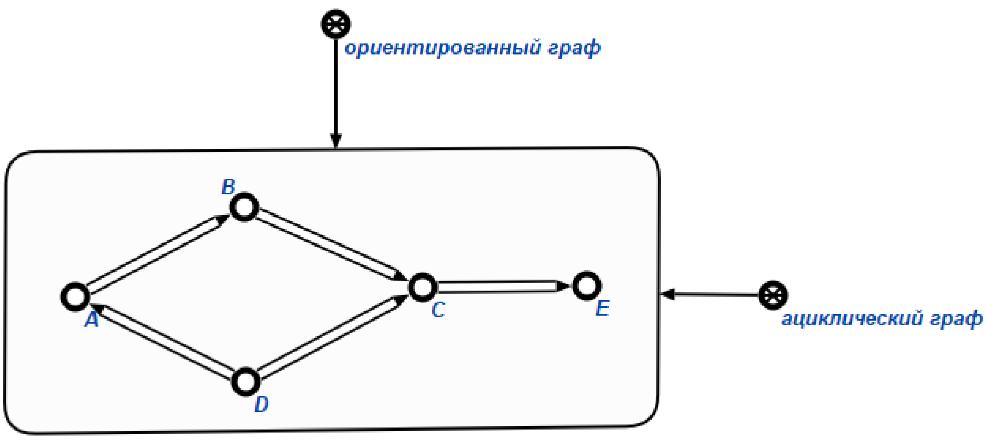
Вход:

Определить, является ли входной граф ациклическим.



Выход:

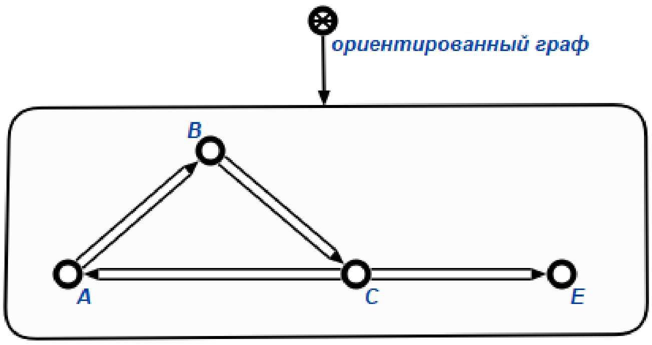
Входной граф ациклический.



2.3 Тест 3

Вход:

Определить, является ли входной граф ациклическим.



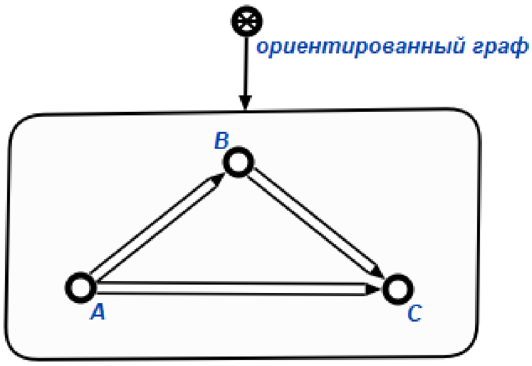
Выход:

Входной граф имеет цикл.

2.4 Тест 4

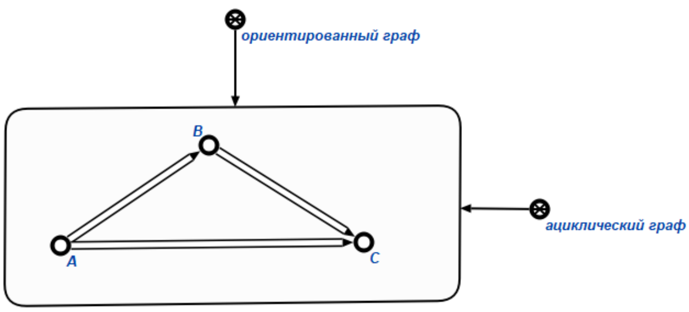
Вход:

Определить, является ли входной граф ациклическим.



Выход:

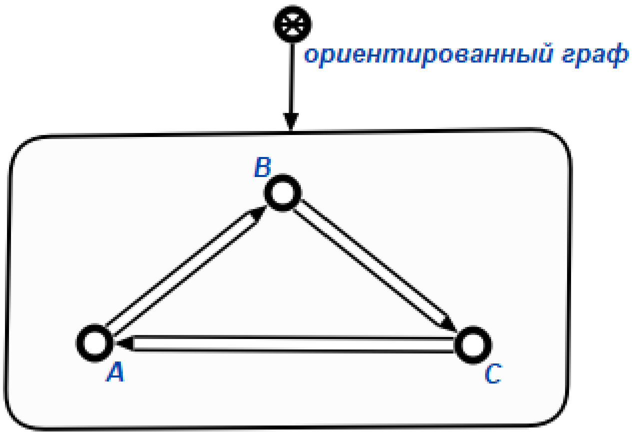
Входной граф ациклический:



2.5 Тест 5

Вход:

Определить, является ли входной граф ациклическим.

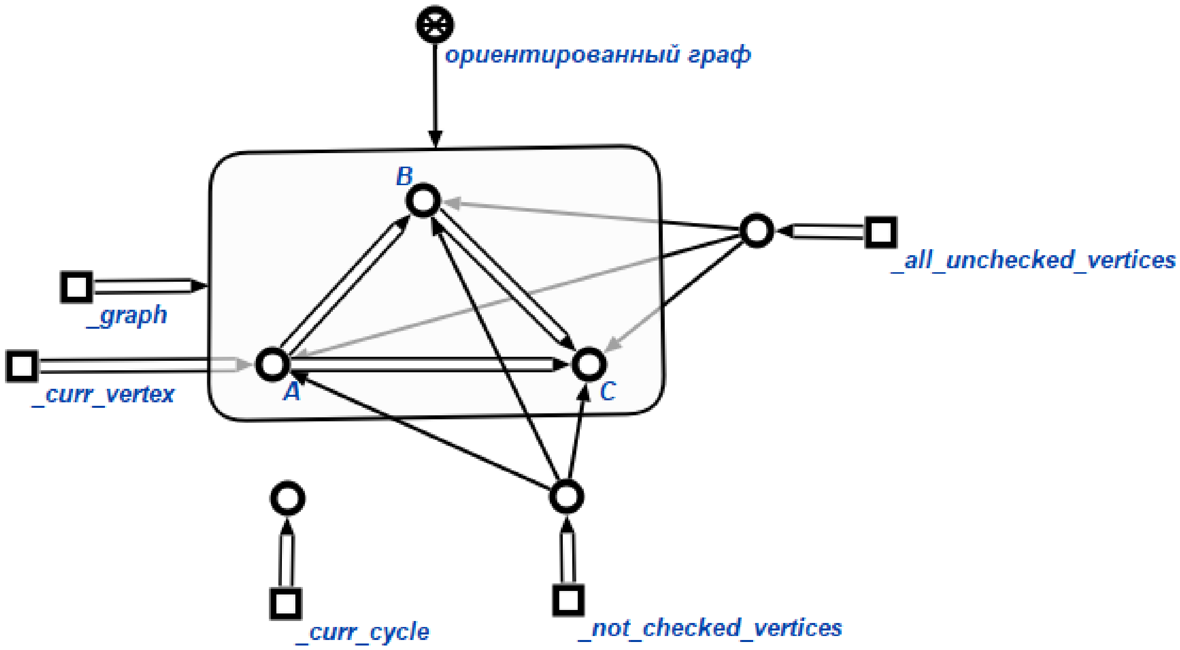


Выход:

Входной граф имеет цикл.

3 Описание алгоритма

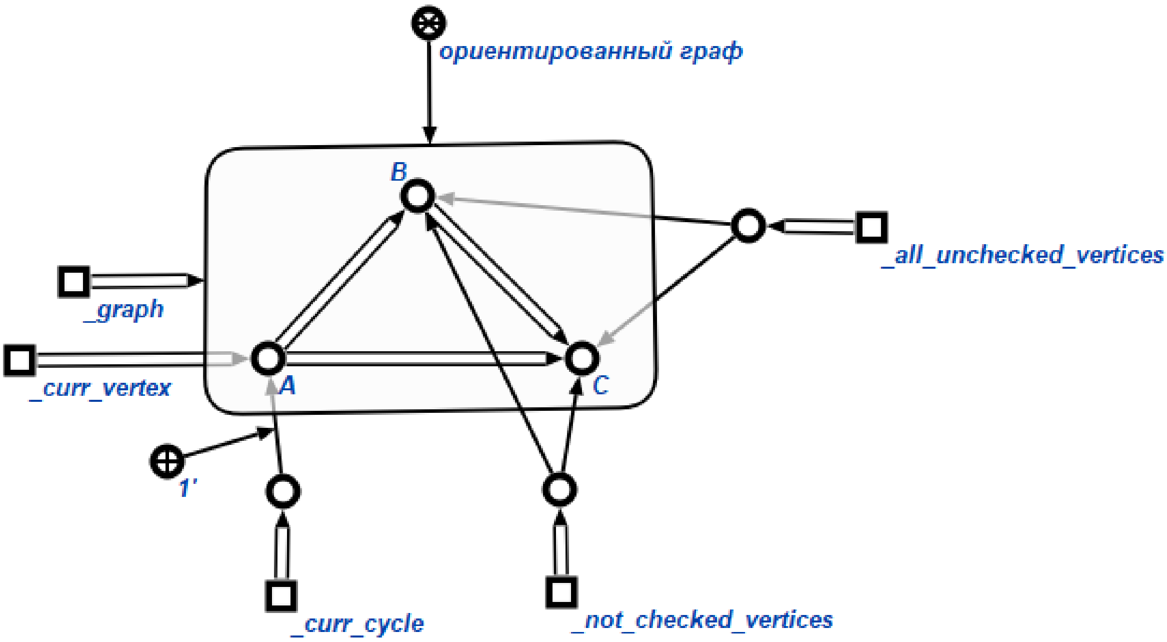
1. Задание входного графа, множества неиспользованных в цикле вершин, множества непроверенных вершин, ориентированного множества строящегося цикла



Переменные изменятся следующим образом:

- _graph получит в качестве значения sc-узел ориентированного графа;
- _curr_vertex получит в качестве значения вершину А, которая будет начальной для проверки на ацикличность;
- _curr_cycle получит в качестве значения ориентированное множество текущего строящегося цикла. Если цикл хоть для одной вершины получится построить, граф не будет ациклическим
- _not_checked_vertices получит в качестве значение множество неиспользованных в цикле вершин, в которое добавляем все вершины. При добавлении вершин во множество строящегося цикла будем удалять вершины из множества неиспользованных в цикле вершин
- _all_unchecked_vertices получит в качестве значение множество непроверенных вершин, в которое добавляем все вершины. Каждую вершину нам надо проверить на возможность построения цикла из нее. После неудачной (невозможность построить цикл) проверки удаляем вершины из множества непроверенных вершин.

2. Начало проверки на возможность построения цикла для вершины А



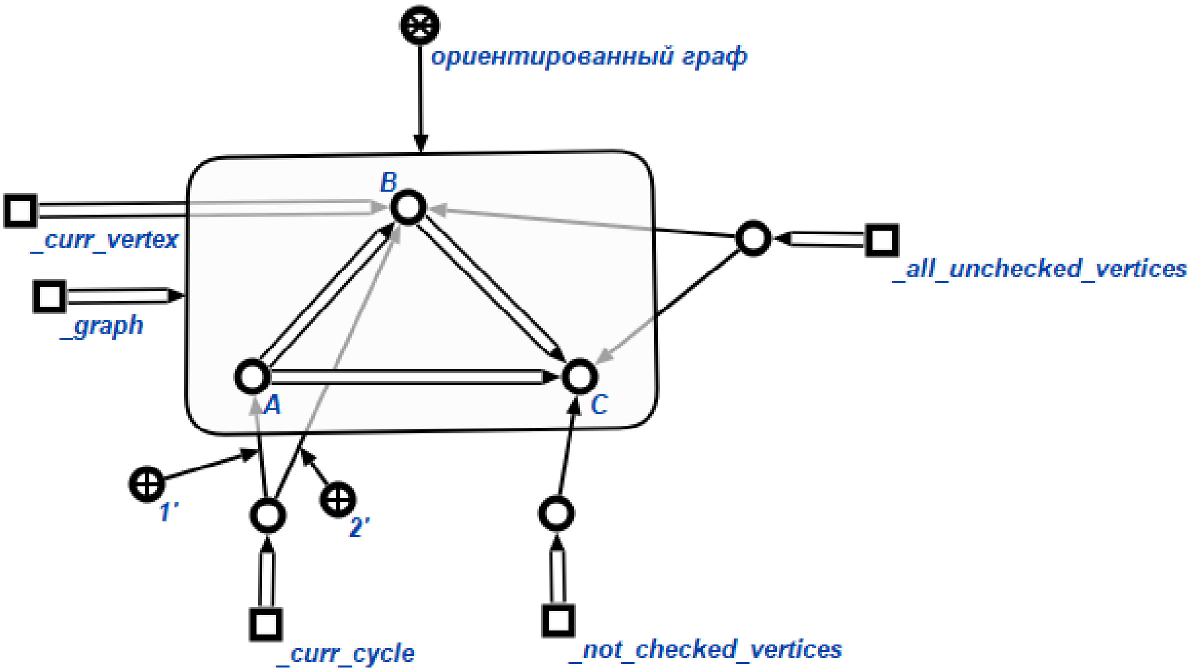
Начали проверку для вершины А. Удаляем ее из множества непроверенных вершин.

Добавляем вершину во множество строящегося цикла с порядковым номером 1.

Удаляем вершину А из множества неиспользующихся вершин.

По множеству неиспользованных вершин проверяем, с какими вершинами имеет связи вершина А. Это вершины С и В.

3. Добавление во множество текущего цикла вершины B

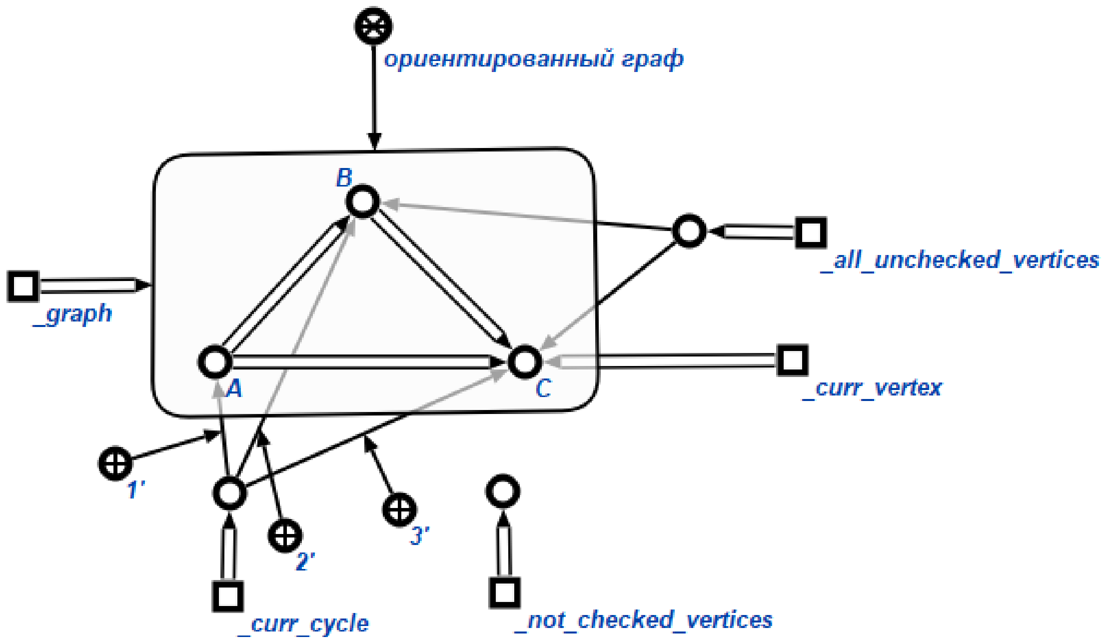


Из двух вершин, с которыми имеет связи выбираем первую - B. Передаем переменной _curr_vertex вершину B. Добавляем ее во множество строящегося цикла с порядковым номером 2.

Удаляем из множества неиспользованных вершин вершину B.

Вершина B имеет связь с единственной вершиной: C.

4. Добавление во множество текущего цикла вершины С



Передаем переменной _curr_vertex вершину С. Добавляем ее во множество строящегося цикла с порядковым номером 3.

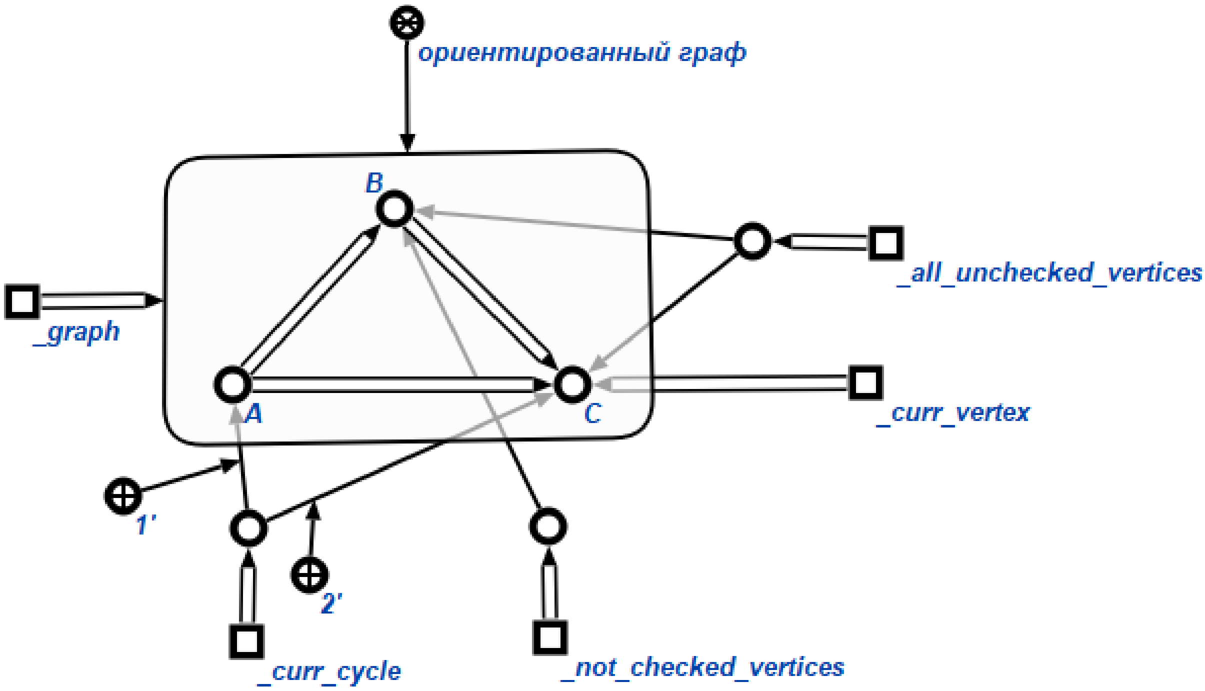
Удаляем из множества неиспользованных вершин вершину С.

Вершина С не имеет исходящих ребер. Построение цикла не удалось.

Возвращаемся к предыдущей вершине. Вершину С подаем во множество неиспользованных вершин и удаляем из множества строящегося цикла. Передаем переменной _curr_vertex вершину В. Она не имеет больше непроверенных исходящих ребер.

Возвращаемся к предыдущей вершине. Вершину В подаем во множество неиспользованных вершин и удаляем из множества строящегося цикла. Передаем переменной _curr_vertex вершину А. Она имеет вторую связь с вершиной С.

5. Добавление во множество текущего цикла вершины C



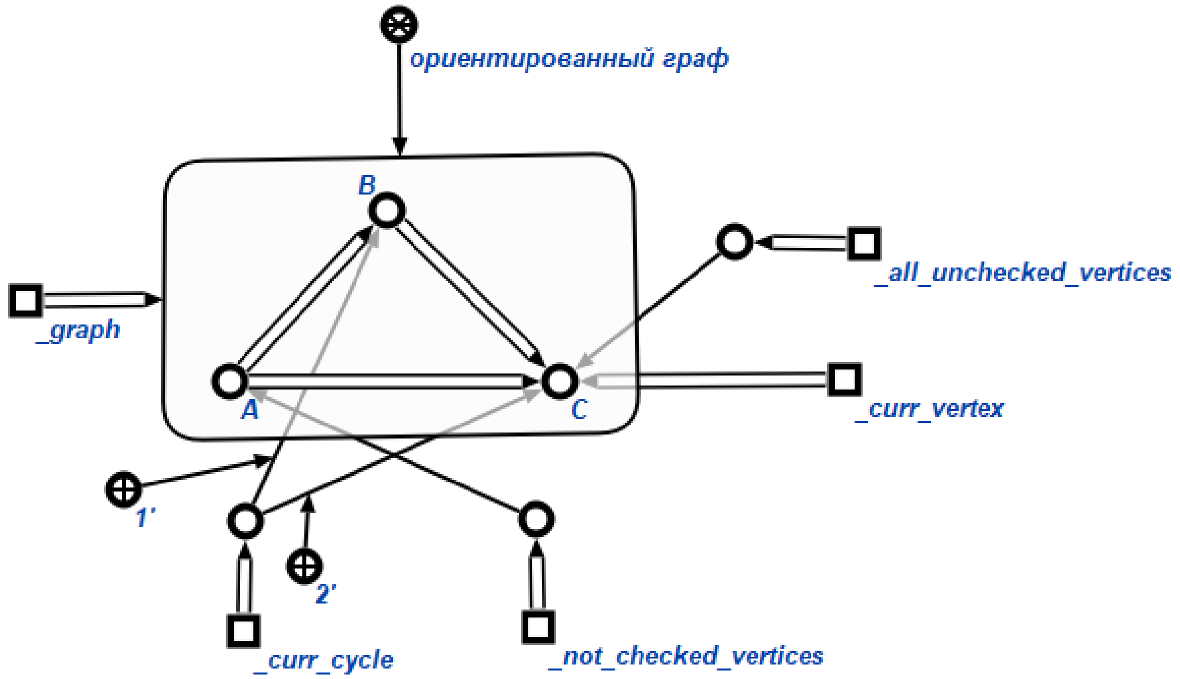
Передаем переменной _curr_vertex вершину C. Добавляем ее во множество строящегося цикла с порядковым номером 2.

Удаляем из множества неиспользованных вершин вершину C.

Вершина C не имеет исходящих ребер. Построение цикла не удалось.

Возвращаемся к предыдущей вершине. Вершину C подаем во множество неиспользованных вершин и удаляем из множества строящегося цикла. Передаем переменной _curr_vertex вершину A. Она не имеет больше непроверенных исходящих ребер. Вершина A прошла проверку на невозможность построения цикла, добавляем ее во множество неиспользованных вершин. Переходим к следующей вершине – B.

6. Проверка вершины B на возможность построения цикла



Начали проверку для вершины B. Передаем переменной _curr_vertex вершину B. Удаляем ее из множества непроверенных вершин.

Добавляем вершину во множество строящегося цикла с порядковым номером 1.

Удаляем вершину B из множества неиспользующихся вершин.

По множеству неиспользованных вершин проверяем, с какими вершинами имеет связи вершина B. Это вершина C.

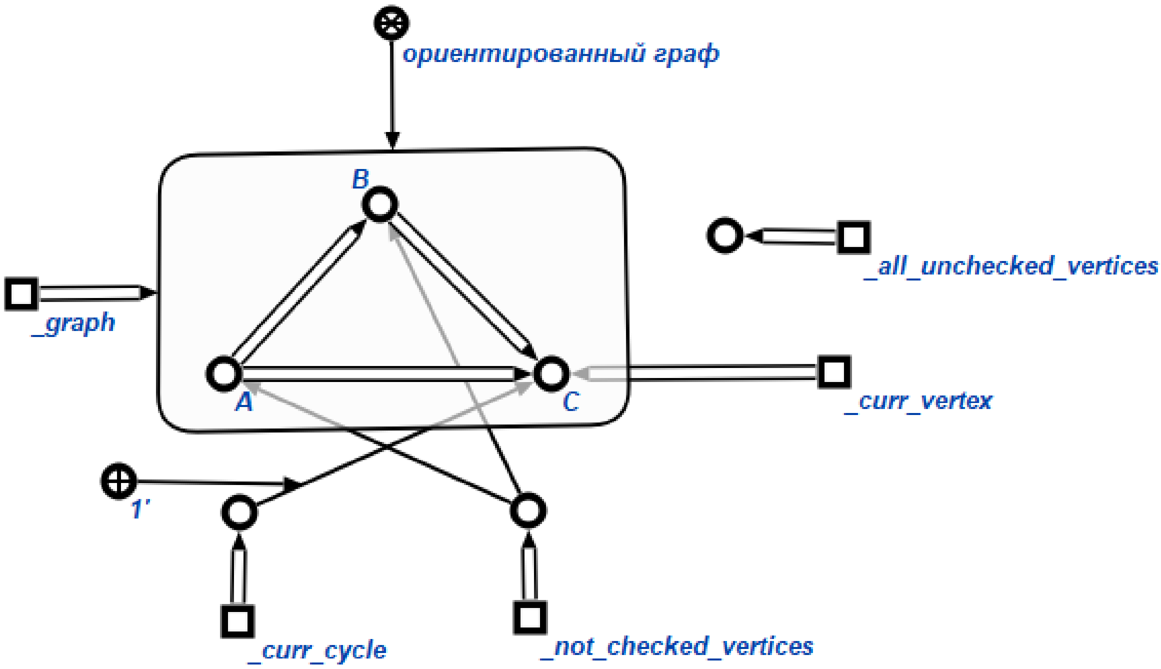
Передаем переменной _curr_vertex вершину C. Добавляем ее во множество строящегося цикла с порядковым номером 2.

Удаляем из множества неиспользованных вершин вершину C.

Вершина C не имеет исходящих ребер. Построение цикла не удалось.

Возвращаемся к предыдущей вершине. Вершину C подаем во множество неиспользованных вершин и удаляем из множества строящегося цикла. Передаем переменной _curr_vertex вершину B. Она не имеет больше непроверенных исходящих ребер. Вершина B прошла проверку на невозможность построения цикла, добавляем ее во множество неиспользованных вершин. Переходим к следующей вершине – C.

7. Проверка вершины С на возможность построения цикла



Начали проверку для вершины С. Передаем переменной _curr_vertex вершину С. Удаляем ее из множества непроверенных вершин.

Добавляем вершину во множество строящегося цикла с порядковым номером 1.

Удаляем вершину С из множества неиспользующихся вершин.

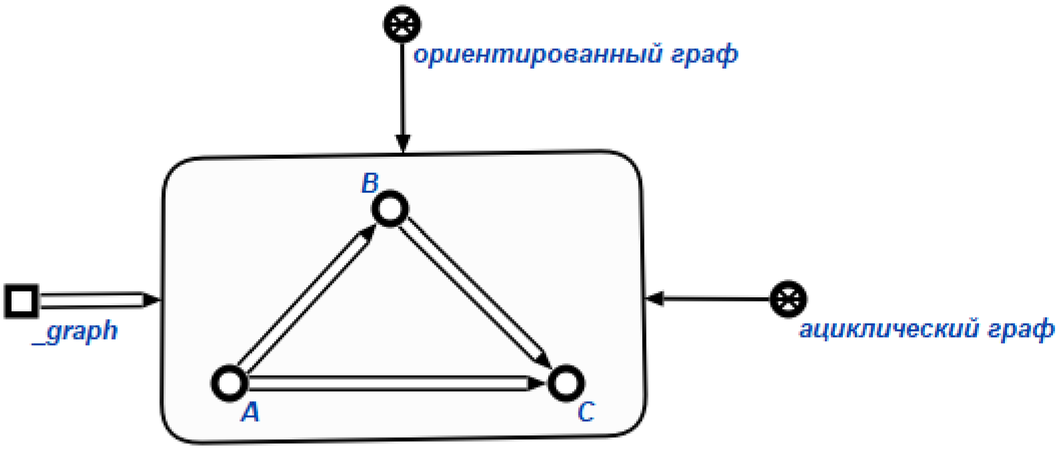
По множеству неиспользованных вершин проверяем, с какими вершинами имеет связи вершина С.

Вершина С не имеет исходящих ребер. Построение цикла не удалось.

Вершина С прошла проверку на невозможность построения цикла, добавляем ее во множество неиспользованных вершин. Множество непроверенных вершин осталось пусто, а значит, все вершины прошли проверку. Граф является ациклическим.

Удаляем все множества и переменную текущей вершины.

8. Результат работы алгоритма



4 Список литературы

OSTIS GT [В Интернете] // База знаний по теории графов OSTIS GT. - 2011 г.. - http://ostisgraphstheo.sourceforge.net/index.php/Заглавная_страница.

Хאון Ф. Теория графов [Книга]. - Москва : Едиториал УРСС, 2003.