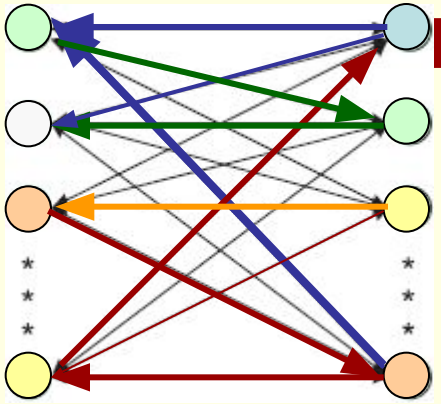
ализация рсивных запросов в динамической ивной реляционной



Л.Ю. Жилякова

ПИ ЮФУ,

г. Ростов-на-Дону

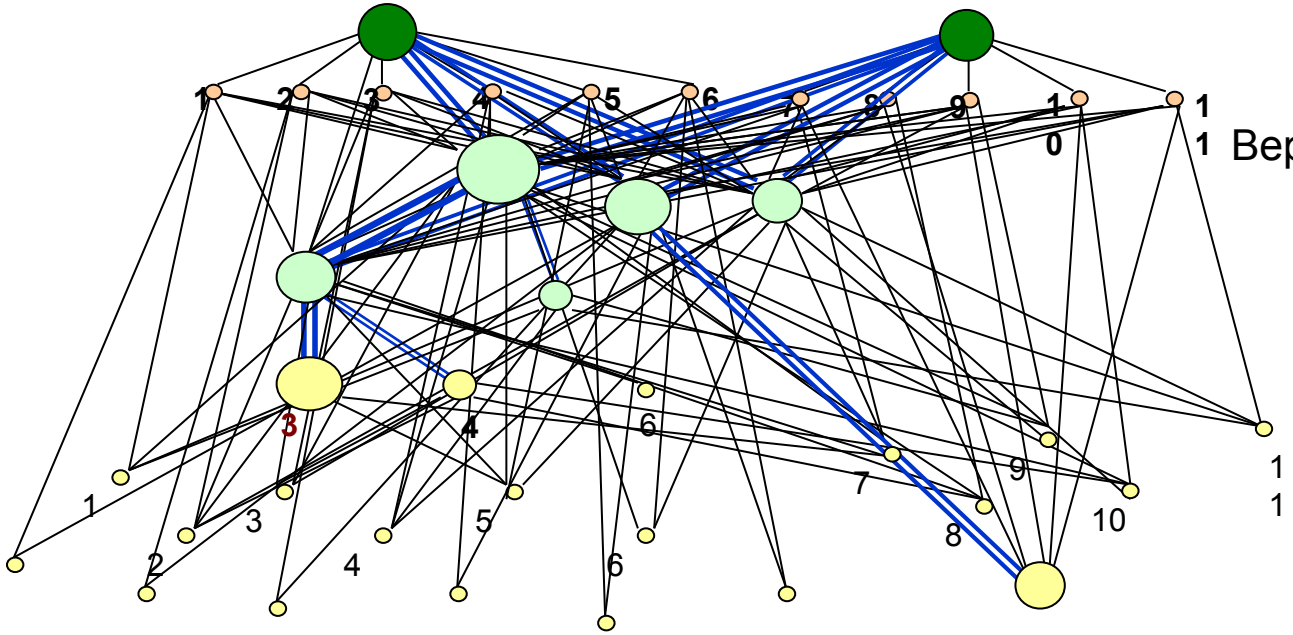
zhilyakov@aaanet.ru

ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Тверь, КИИ-2010

Ассоциативная ресурсная сеть

Ассоциативная ресурсная сеть представляет собой динамическую модель памяти, основанную на *неоднородной ресурсной сети.*



Вершины сети соответствуют сущностям предметной области, ребра — ассоциативным связям между ними.

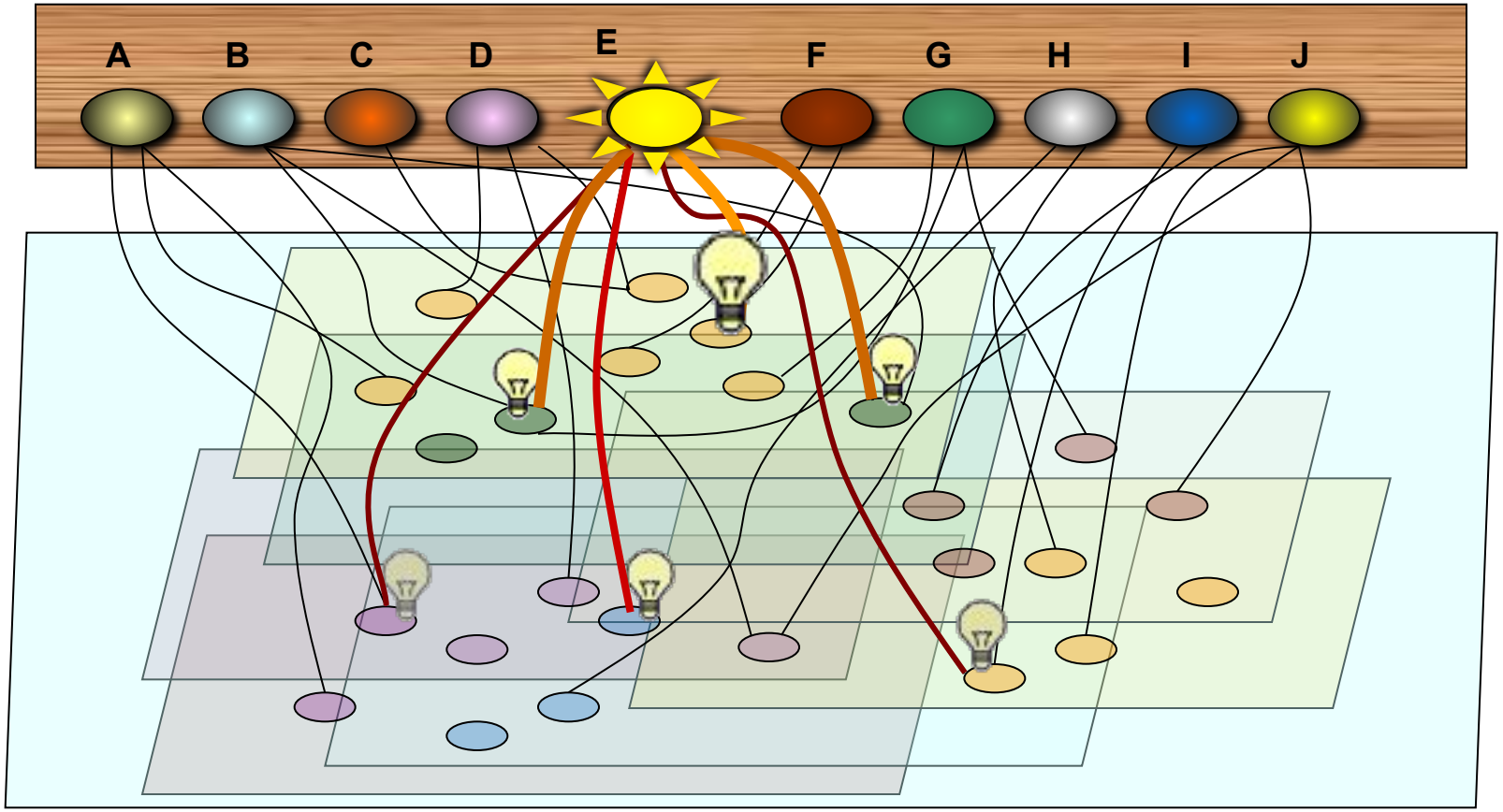
Способ хранения информации в ассоциативной сети таков, что наиболее часто используемые данные оказываются и наиболее доступными.

Чем данные используются реже, тем труднее их найти.

Обеспечение быстрого доступа к часто используемым данным реализуется благодаря двум свойствам сети.

Способ хранения информации в ассоциативной сети таков, что наиболее часто используемые данные оказываются и наиболее доступными.

Чем данные используются реже, тем труднее их найти.

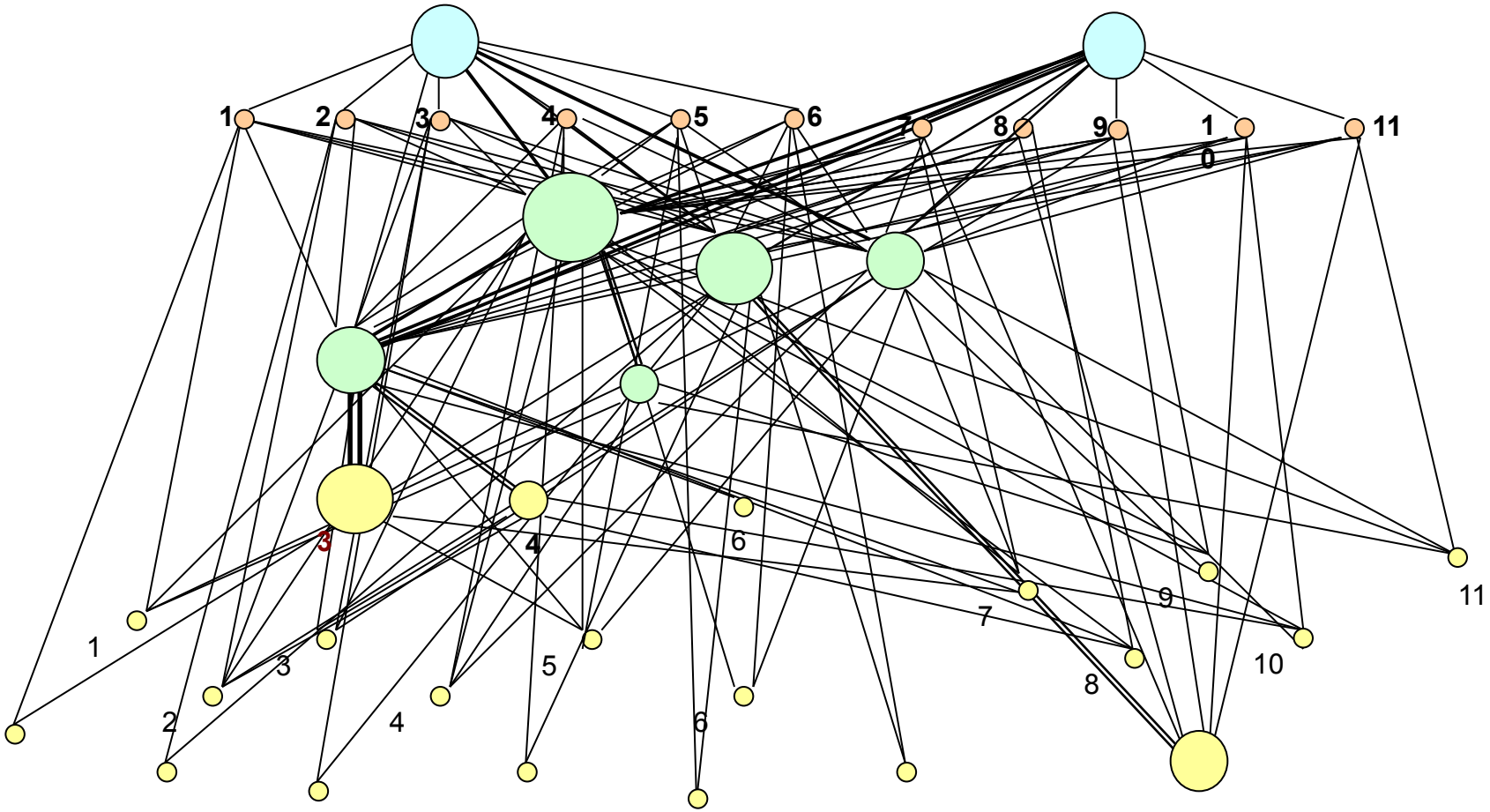


Обеспечение быстрого доступа к часто используемым данным реализуется благодаря двум свойствам сети.

СВОЙСТВА СЕТИ

I. ЯРКОСТЬ

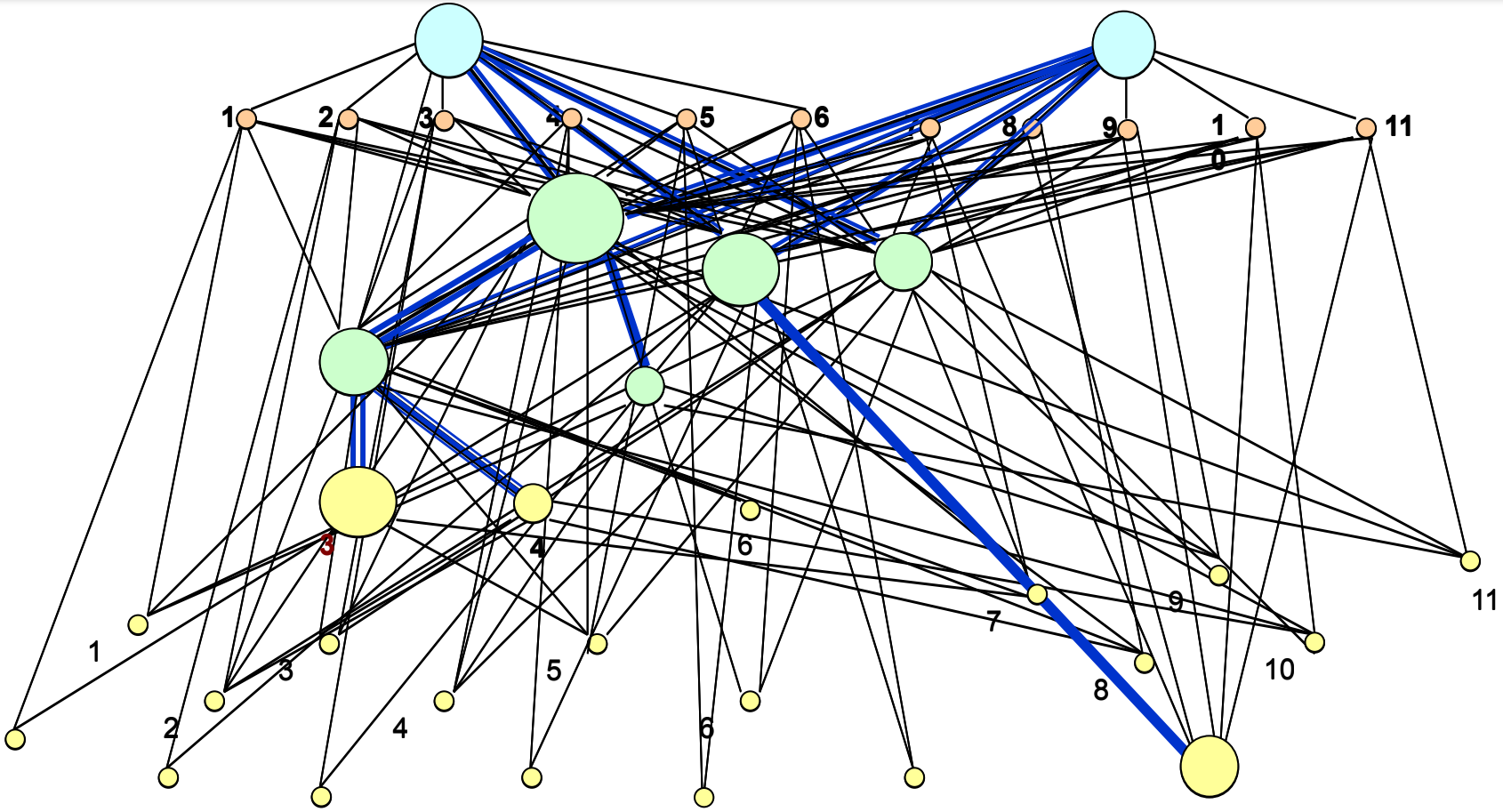
Каждая вершина обладает *яркостью*: доступность вершины тем выше, чем больше ее яркость, – тем эта вершина «виднее» при поиске.



II. ПРОПУСКНАЯ СПОСОБНОСТЬ (ПРОВОДИМОСТЬ)

Каждая дуга сети, имеет свою *проводимость*, которая отвечает за способность передавать яркость от одной вершины к другой.

Проводимость соответствует силе ассоциативной связи между сущностями: чем сильнее связь, тем выше проводимость.

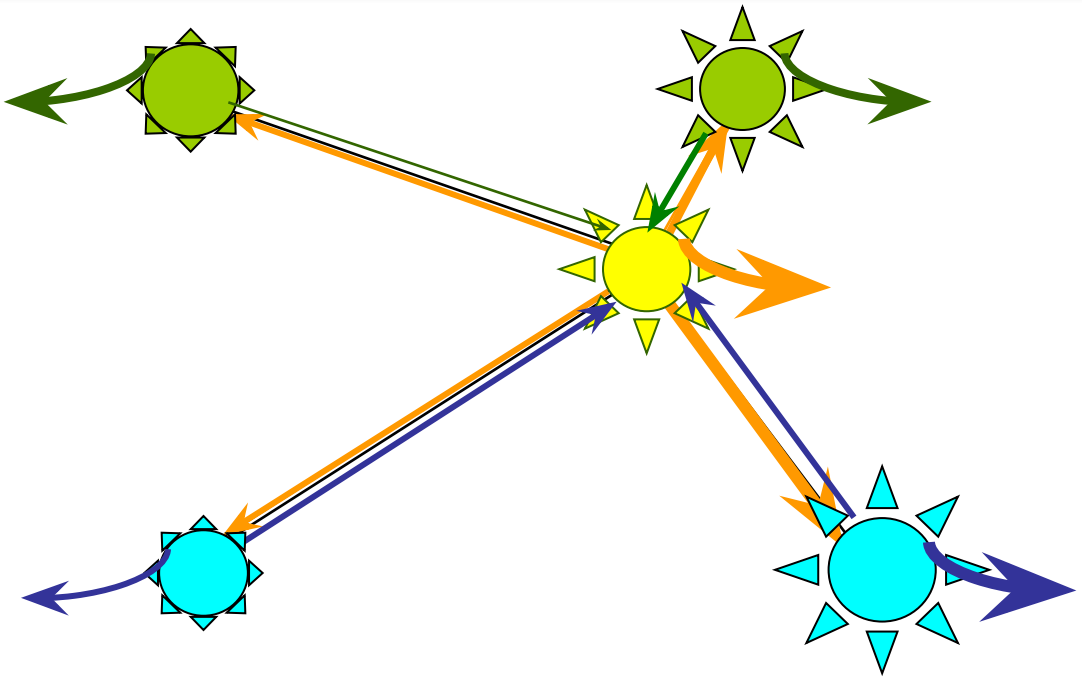


Ресурсная сеть

В качестве математического аппарата для такой модели используется **ресурсная сеть**.

Ресурсной сетью называется ***двусторонний граф с петлями***, вершинам которого приписаны неотрицательные числа, называемые ***ресурсами***, а рёбра способны доставлять ресурс от одной вершины к другим.

Каждому ребру графа приписано неотрицательное число, называемое ***проводимостью***, и характеризующее максимальное количество ресурса, передаваемое по нему за один такт времени.

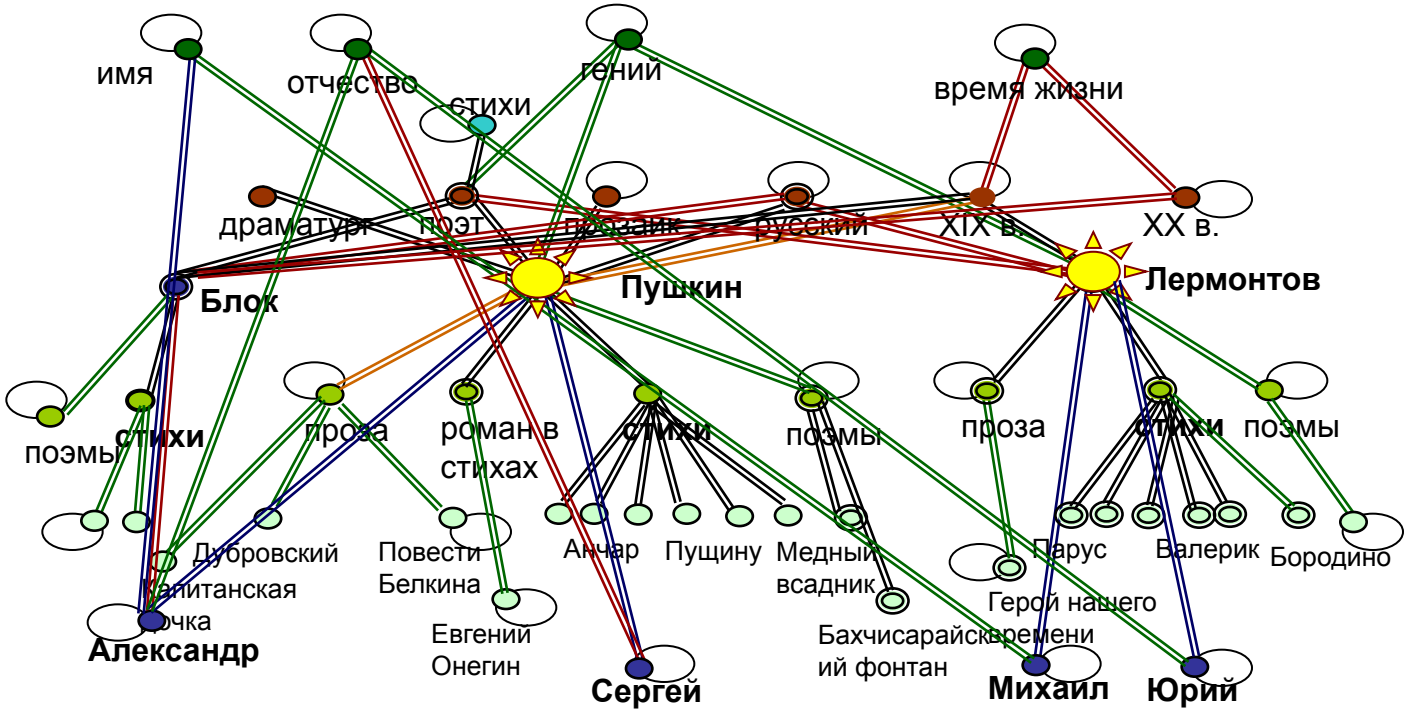


Ассоциативная ресурсная сеть

Ассоциативной ресурсной сетью называется ресурсная сеть, каждая вершина которой имеет имя из некоторого множества имен.

Ребра, соединяющие различные вершины, – отношения на именах, соответствующие ассоциативным связям между обозначаемыми понятиями.

Петли отвечают за автоассоциации.

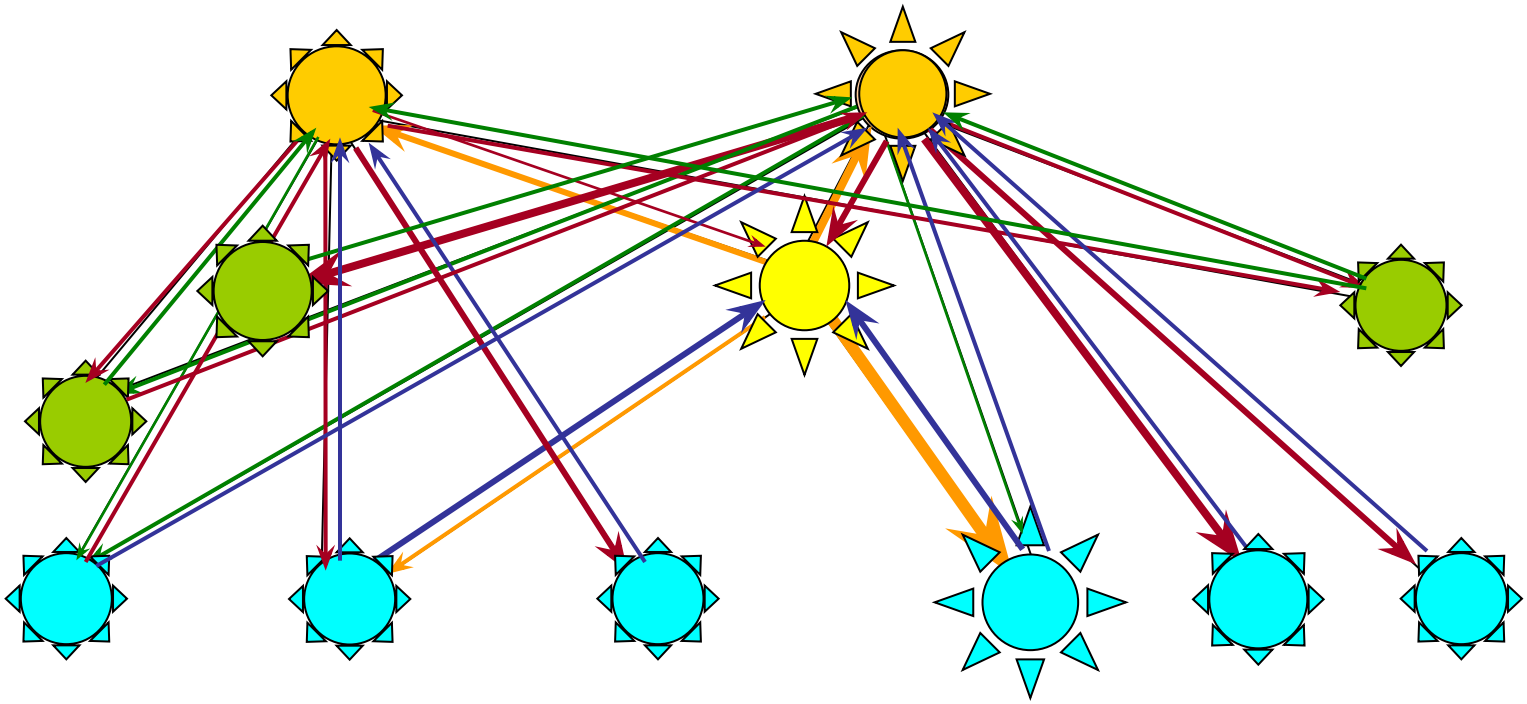


Распространение яркости

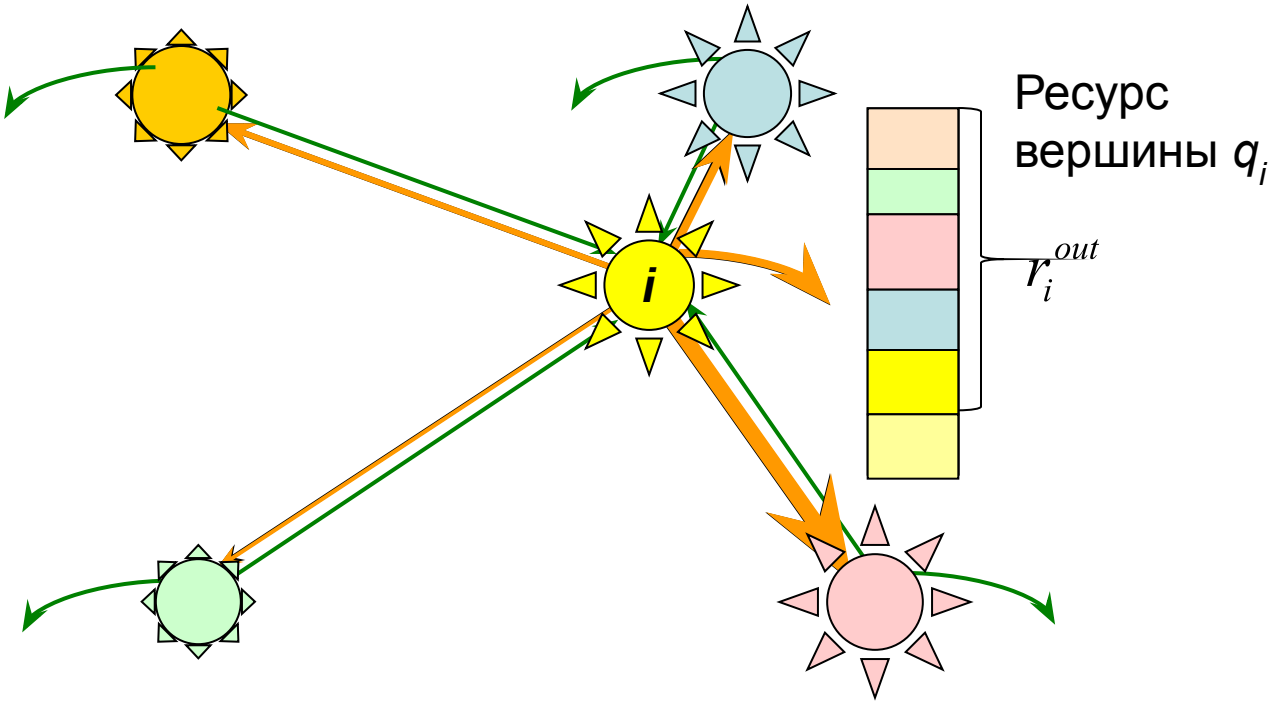
Ресурс вершины отвечает за *яркость* соответствующего ей понятия. Чем он больше, тем понятие ярче, тем оно доступнее в памяти.

Яркость попадает в вершины, участвующие в запросе, и передается по рёбрам от вершины к вершине.

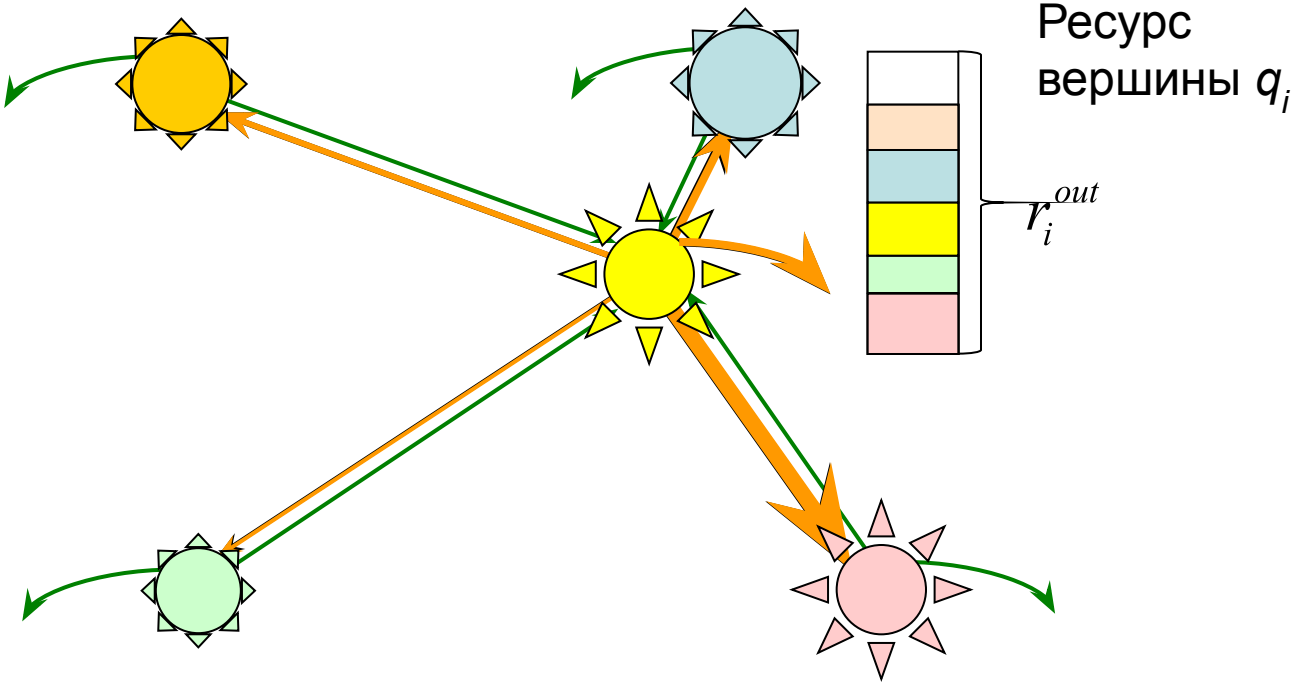
При перетекании яркости высвечиваются вершины, ассоциированные с данными.



Правила распространения яркости (правило 1)



Правила распространения яркости (правило 2)

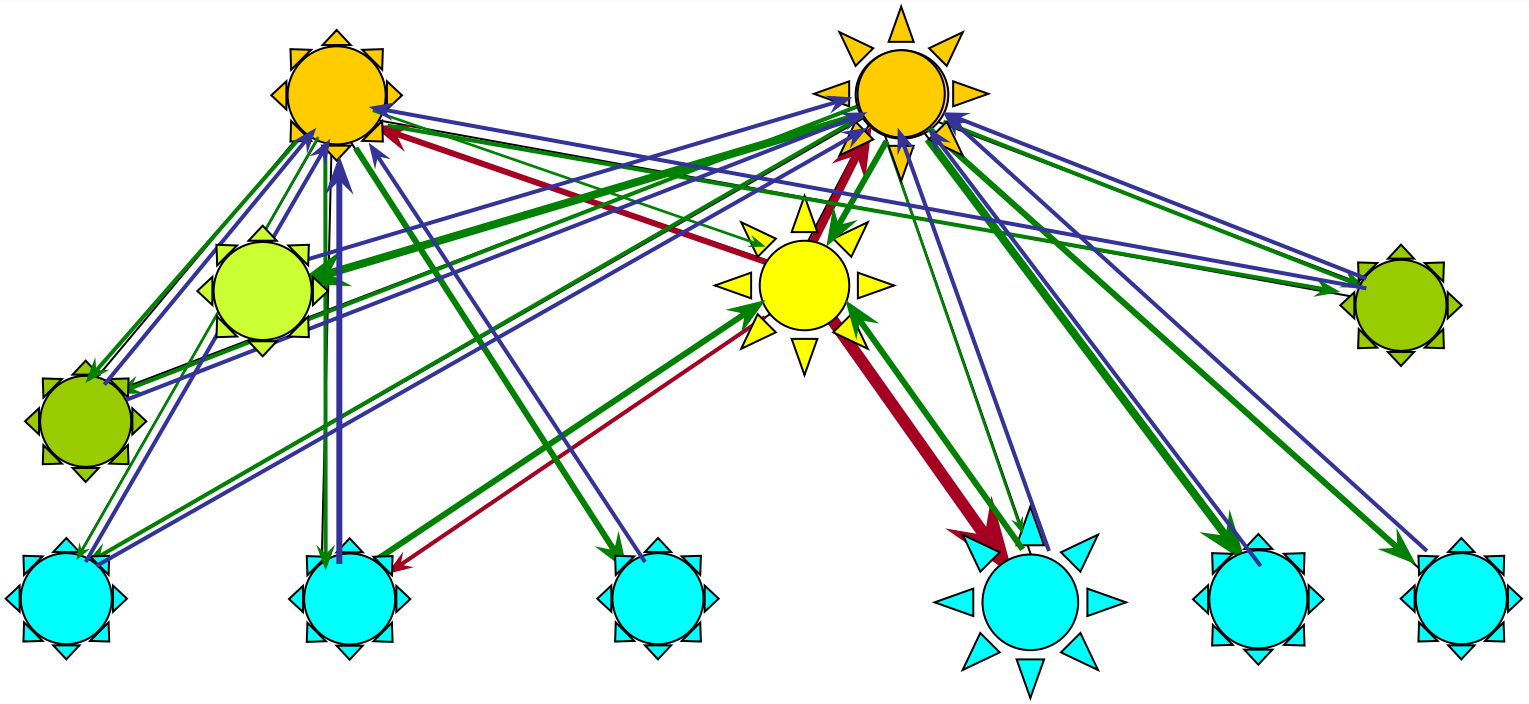


Распространение яркости

В каждый такт времени происходит перераспределение ресурса между вершинами.

Процесс завершается, когда ресурс в вершинах достигает постоянного предельного значения или асимптотически сходится к нему.

Это условие равносильно тому, что в каждой двусторонней паре навстречу друг другу начинает течь равное (или почти равное) количество ресурса.



Изменение топологии сети

Особенностью предложенной модели является динамическое изменение ее топологии всякий раз после того, как происходит обращение к сети с очередным запросом.

Если во время выполнения запроса ребро участвовало в перераспределении ресурса, оно увеличит свою проводимость.

Медленное и быстрое время

Пока в сеть не поступает запросов, она находится в неактивном состоянии.

В сети вводится время двух типов:

1) *медленное время τ* ;

2) *быстрое время t*.

Один такт медленного времени соответствует выполнению одного запроса.

Медленное время отвечает за изменение проводимостей ребер и создание новых ребер. За один такт τ у каждого ребра происходит не более одного изменения проводимости.

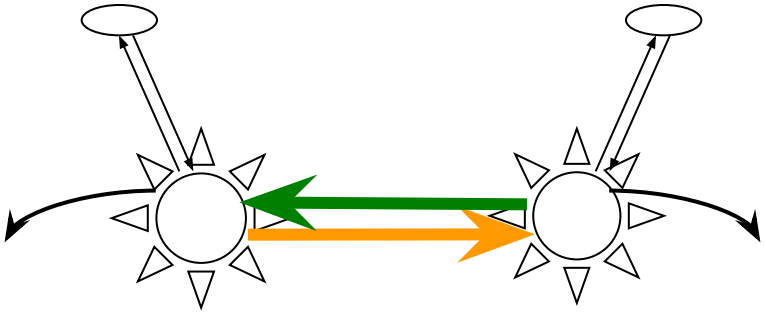
Быстрое время включается во время исполнения запроса.

Оно отвечает за распределение ресурса по вершинам.

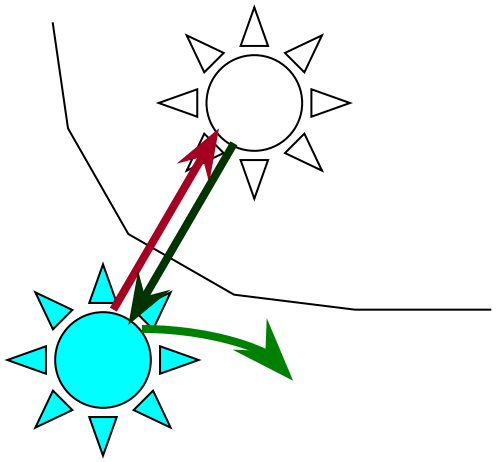
Алгоритм построения сети

Построение сети (наполнение ее информацией) и обращение к ней с запросами совершаются в одном и том же медленном времени т.

Информация заносится в сеть минимальными структурными единицами. Они могут быть двух типов.



двусторонняя пара, связывающая две вершины;

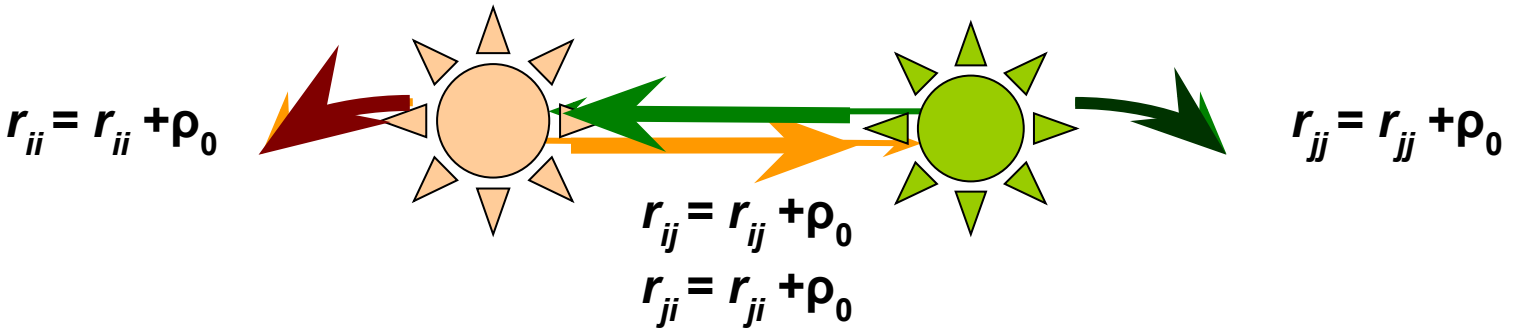


новая вершина с петлей и двусторонняя пара, связывающая эту вершину с уже имеющейся.

Изменение проводимостей

Проводимость петли увеличивается на заданный «*квант проводимости*» всякий раз, когда вершина связывается с новой вершиной, или упоминается при любом изменении структуры.

Проводимость связи увеличивается всякий раз, когда две соответствующие вершины упоминаются вместе.

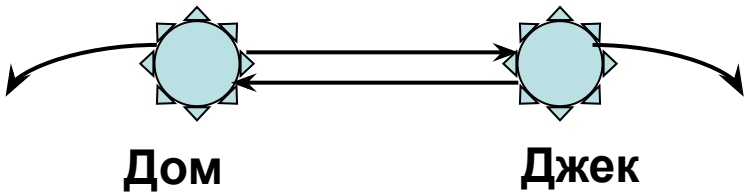


Сеть обязательно заполняется с самого начала. На нулевом шаге она пуста.

Пример построения сети

Дом, который построил Джек

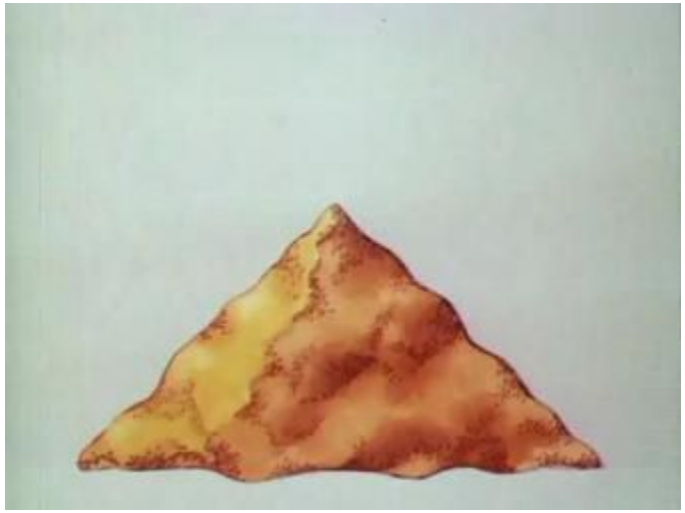
Вот дом,
Который построил Джек.

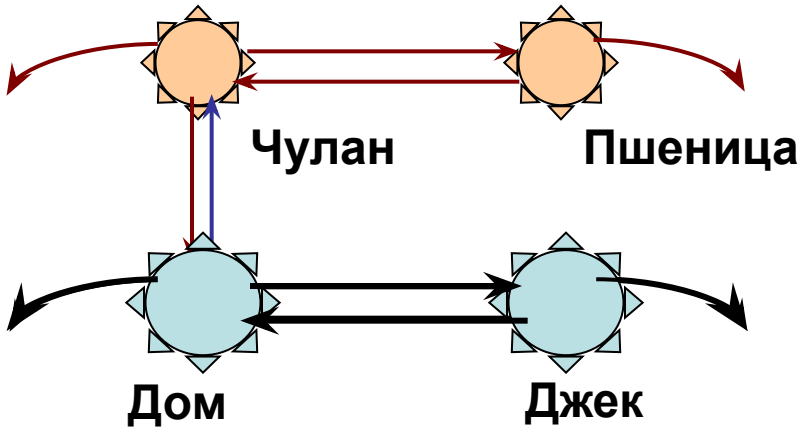


Пример построения сети

Дом, который построил Джек

А это пшеница,
Которая в темном чулане хранится
В доме,
Который построил Джек.

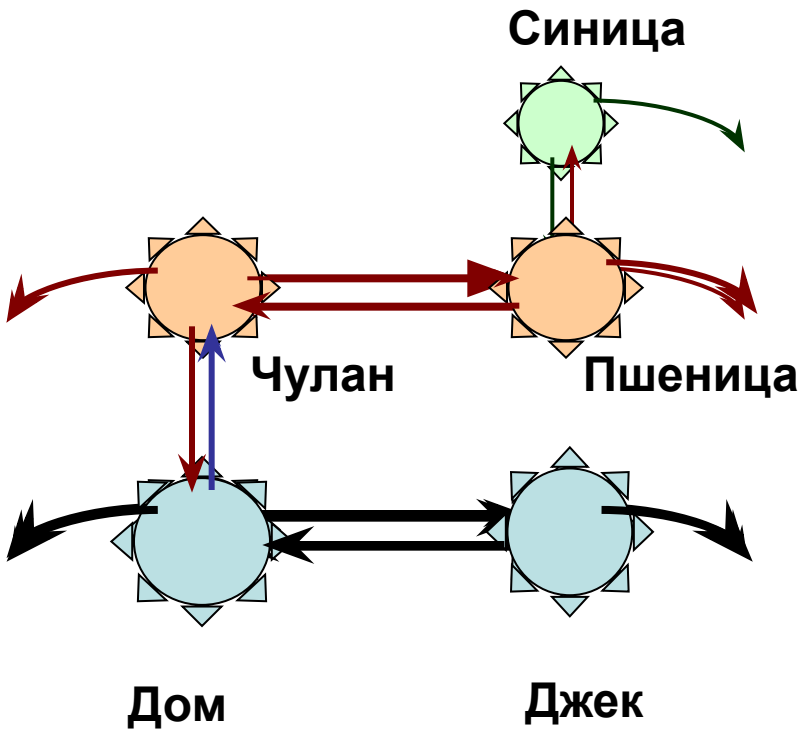




Пример построения сети

Дом, который построил Джек

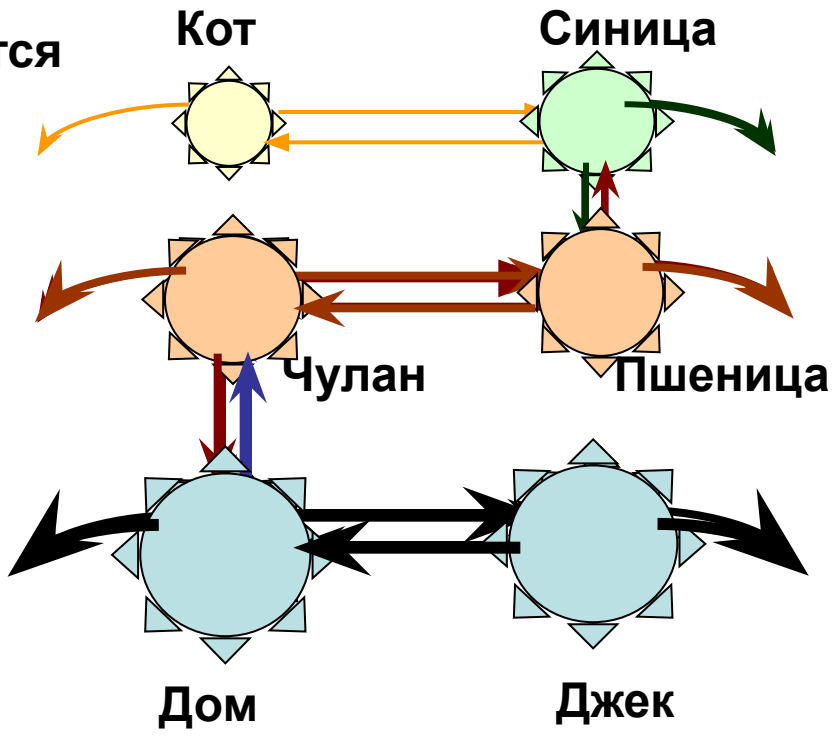
А это веселая птица-синица,
Которая часто ворует пшеницу,
Которая в темном чулане хранится
В доме,
Который построил Джек.



Пример построения сети

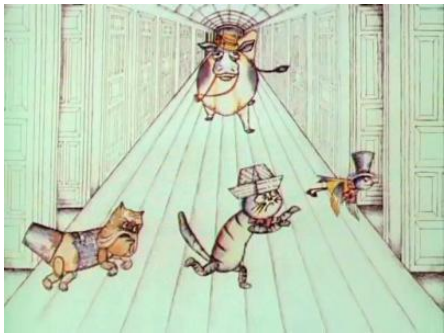
Дом, который построил Джек

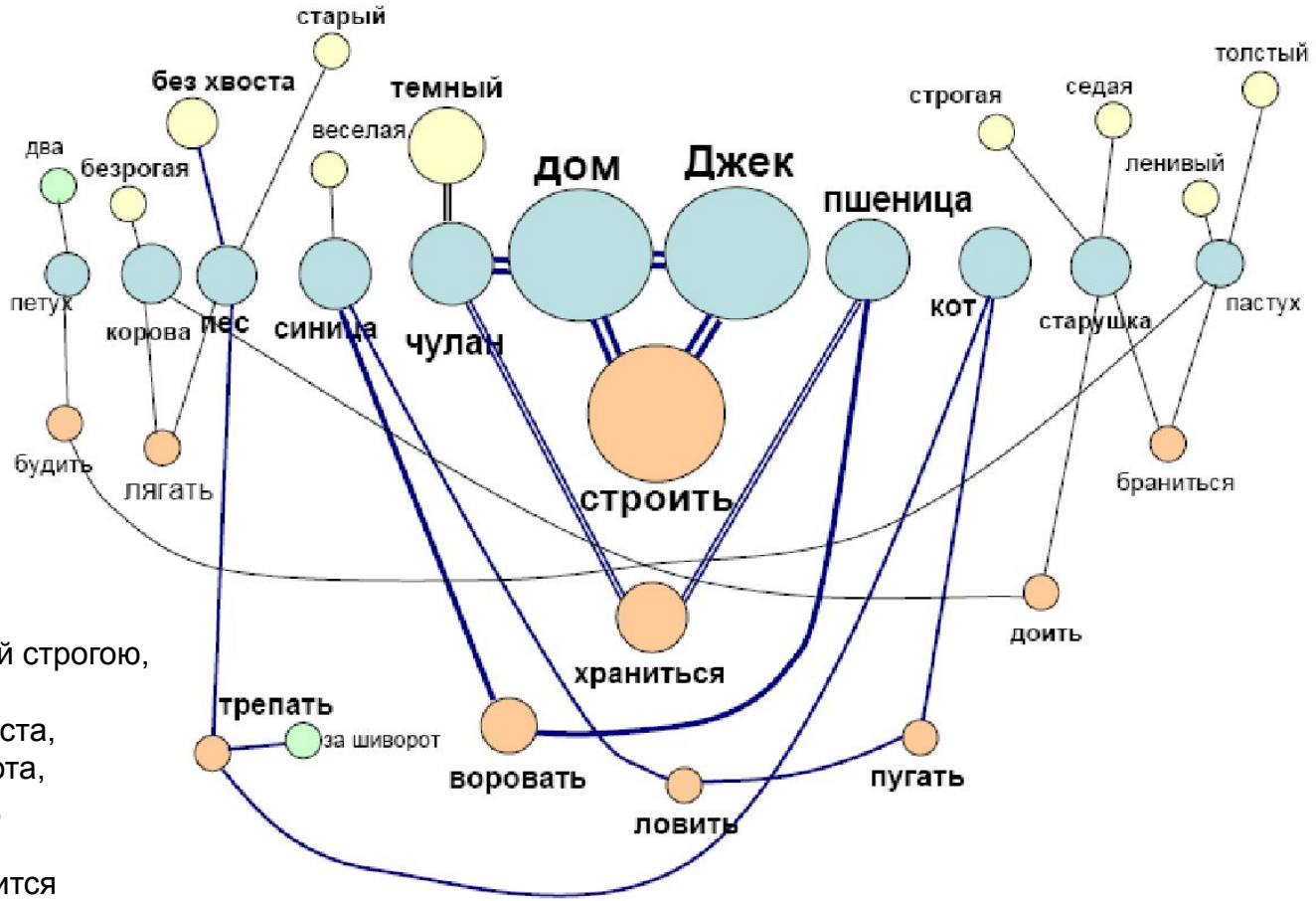
Вот кот,
Который пугает и ловит синицу,
Которая часто ворует пшеницу,
Которая в темном чулане хранится
В доме,
Который построил Джек.



Готовый фрагмент сети

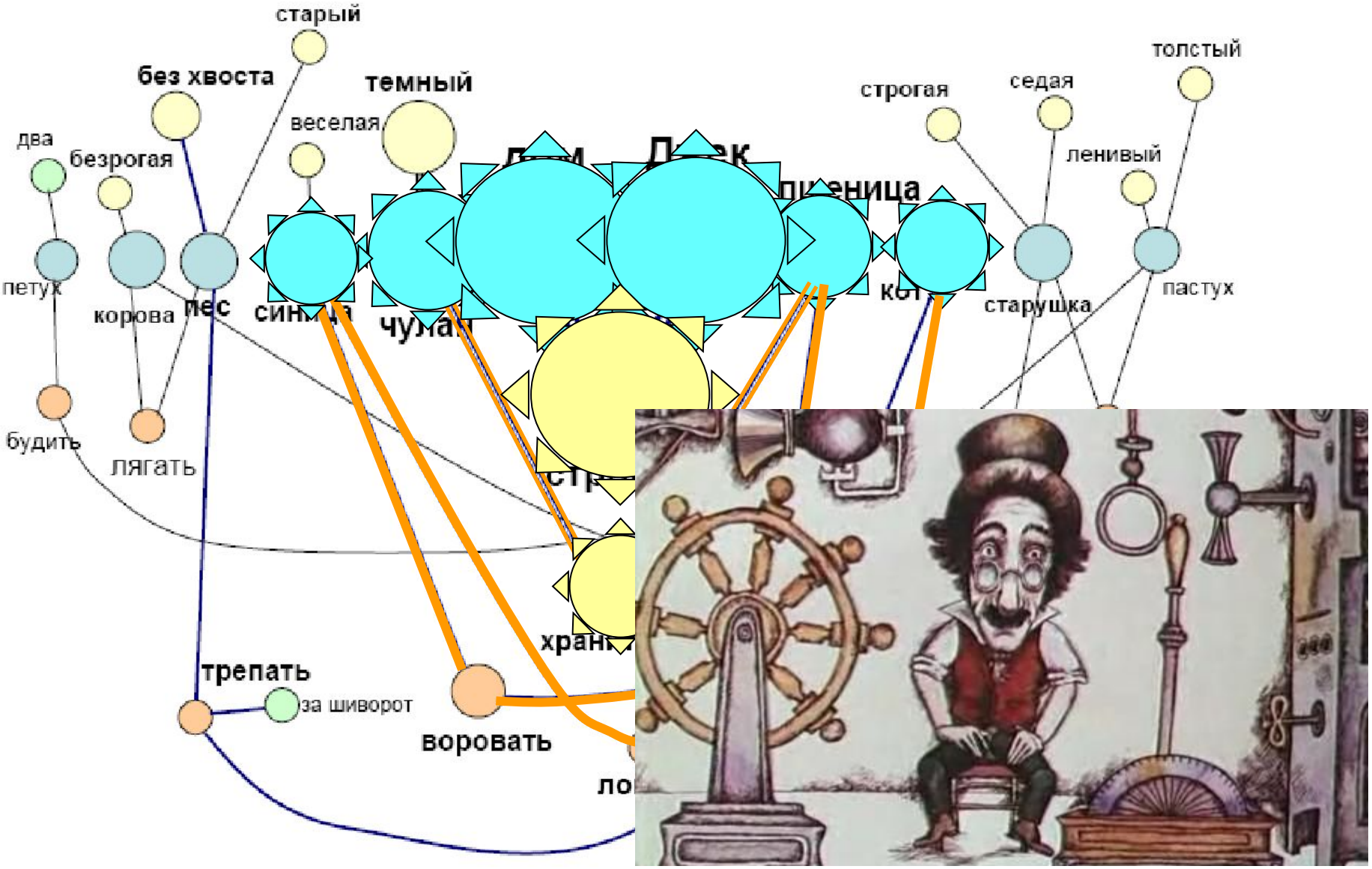
Дом, который построил Джек





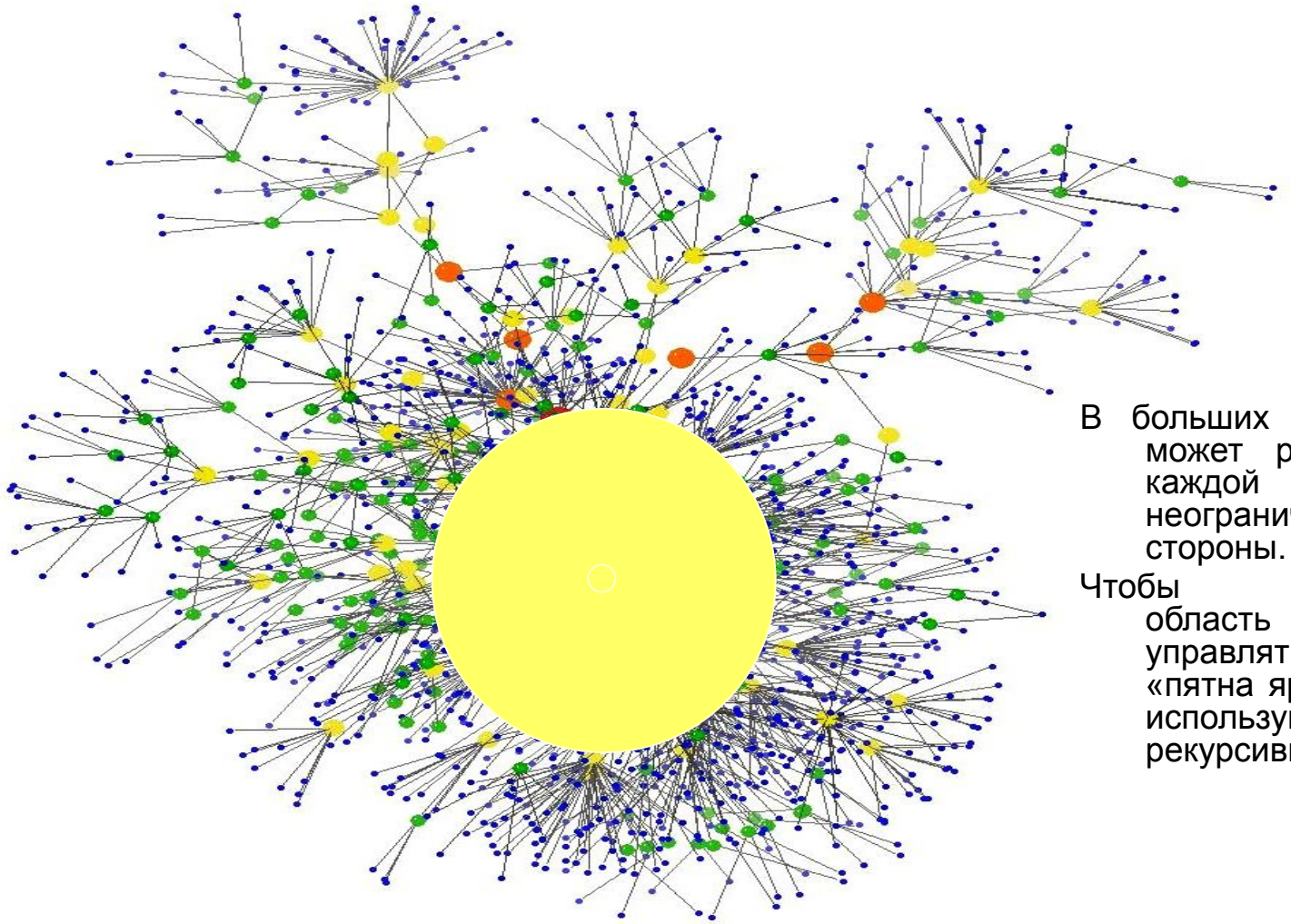
Вот два петуха,
Которые будят того пастуха,
Который бранится с коровницей строгою,
Которая доит корову безрогую,
Лягнувшую старого пса без хвоста,
Который за шиворот треплет кота,
Который пугает и ловит синицу,
Которая часто ворует пшеницу,
Которая в темном чулане хранится
В доме,
Который построил Джек.

Восстановление образа по его части



Управление движением яркости

Запрос к ассоциативной сети — входное множество вершин и количество яркости, которое им приписывается.



В больших сетях яркость может растекаться от каждой вершины неограниченно во все стороны.

Чтобы локализовать область поиска и управлять движением «пятна яркости» в сети, используются рекурсивные запросы.

Реализация рекурсивных запросов

Под *рекурсивным запросом* будем понимать многократный запрос, входное множество вершин которого изменяется в зависимости от выходного множества на предыдущем шаге по одному из наперед заданных правил.

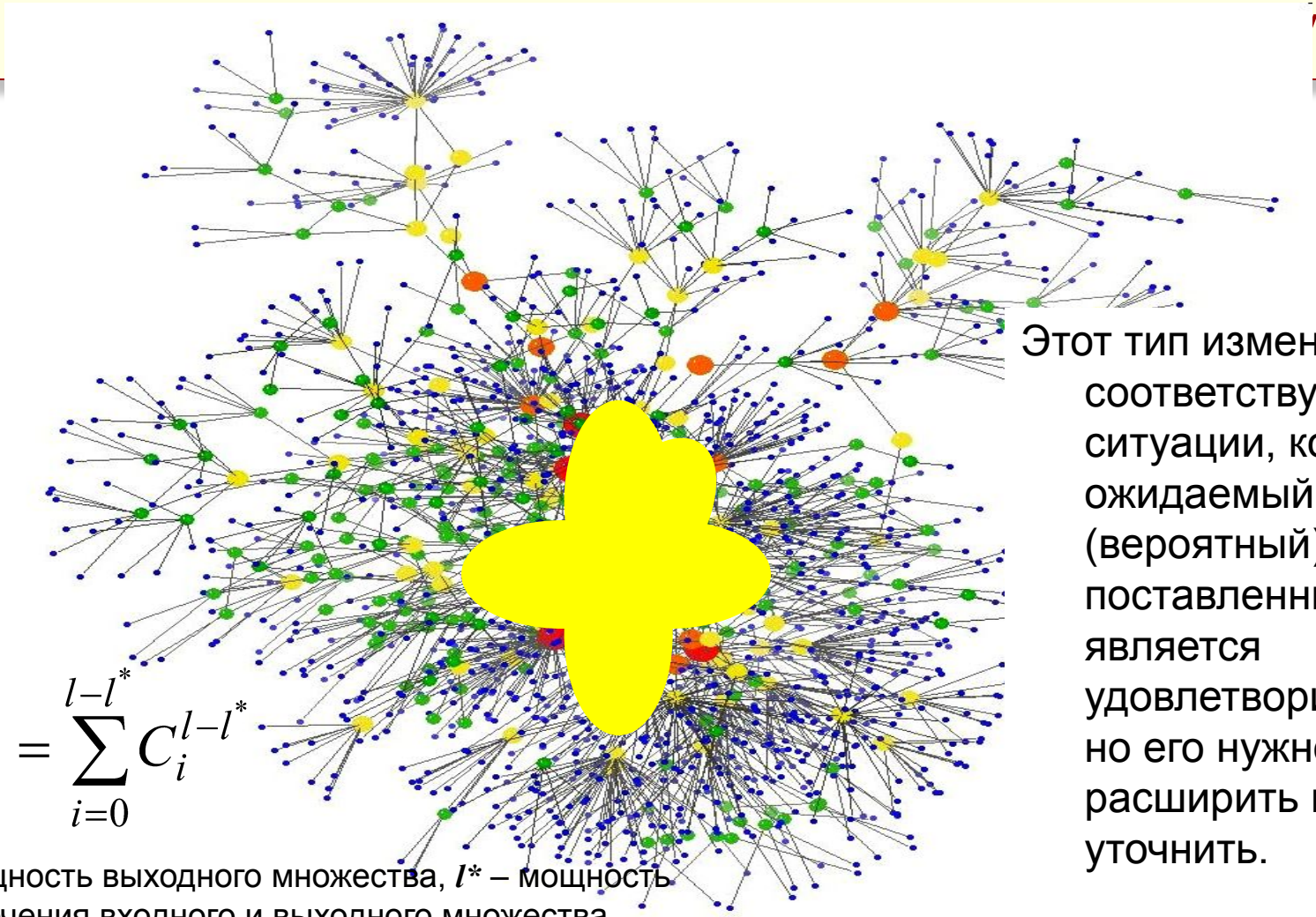
Алгоритм выполнения одного шага рекурсии

1. В начальное множество вершин поступает яркость;
2. Яркость начинает распространяться в соответствии с правилами 1-2 ресурсной сети t_R тактов быстрого времени t. (Величина t_R задана заранее.);
3. По окончании распределения из вершин, имеющих яркость, выбирается новое начальное множество.

И процесс повторяется.

Виды запросов

1. Добавление к входному множеству одной или нескольких
рсии



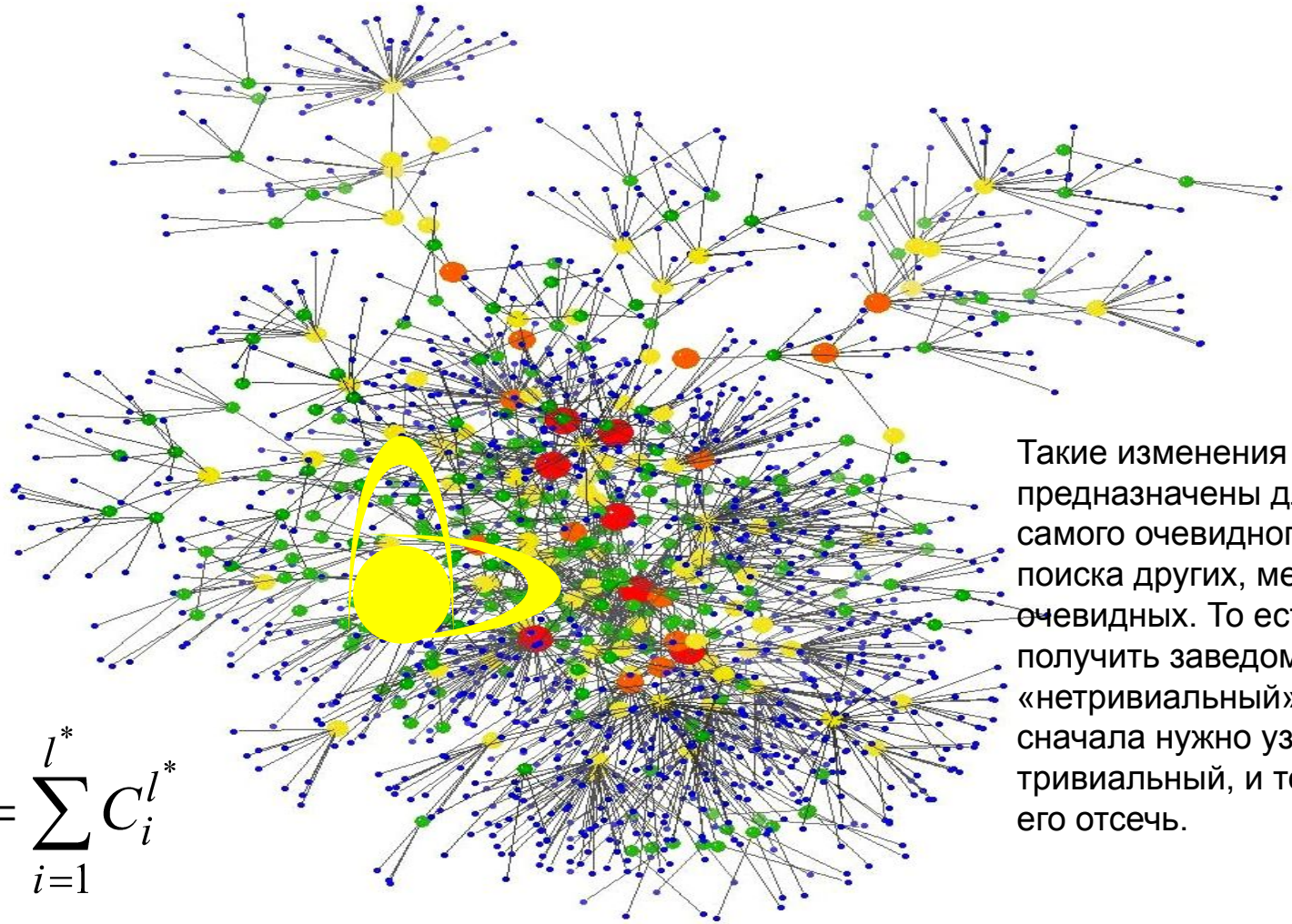
$$N^{+} = \sum_{i=0}^{l-l^{*}} C_{i}^{l-l^{*}}$$

l – мощность выходного множества, ***l**** – мощность пересечения входного и выходного множества.

Этот тип изменений соответствует ситуации, когда самый ожидаемый (вероятный) ответ на поставленный запрос является удовлетворительным, но его нужно расширить и/или уточнить.

2. Удаление одной или нескольких вершин из входного множества предыдущего запроса

2. а) Удаляются вершины из пересечения множеств вопрос-ответ, т.е.

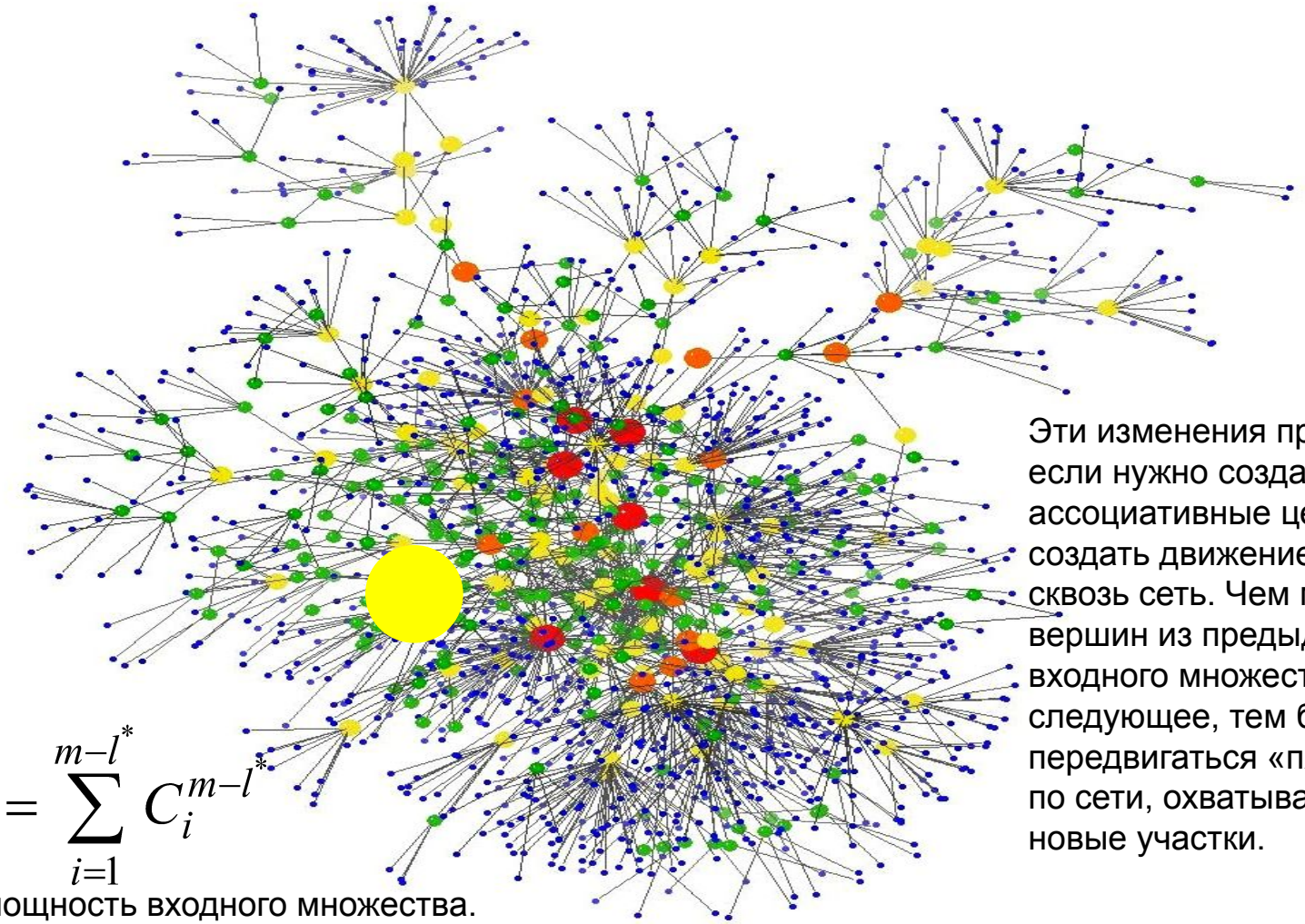


$$N_a^- = \sum_{i=1}^{l^*} C_i^{l^*}$$

Такие изменения предназначены для отсекания самого очевидного ответа и поиска других, менее очевидных. То есть, чтобы получить заведомо «нетривиальный» ответ, сначала нужно узнать ответ тривиальный, и только затем его отсечь.

2. Удаление одной или нескольких вершин из входного множества предыдущего запроса

2. b) Удаляются вершины из предыдущего входного множества, которых



$$N_b^- = \sum_{i=1}^{m-l^*} C_i^{m-l^*}$$

m – мощность входного множества.

Эти изменения производятся, если нужно создать длинные ассоциативные цепочки, – создать движение яркости сквозь сеть. Чем меньше вершин из предыдущего входного множества перейдет в следующее, тем быстрее будет передвигаться «пятно яркости» по сети, охватывая каждый раз новые участки.

Комбинируя все возможные сочетания добавления и удаления вершин, получим

$$N = \sum_{i=0}^{l-l^*} C_i^{l-l^*} \cdot \sum_{i=0}^{m} C_i^m$$

различных множеств, каждое из которых претендует на то, чтобы быть входным множеством запроса на следующем шаге рекурсии.

Операции над графами

1. *Оператор T(G)* – транзитивное замыкание графа G. Действует он следующим образом: для любого графа G $T(G)$ – такой граф, что для любых двух вершин верно: если есть путь любой длины из вершины v_i в вершину v_j, то есть и двусторонняя пара $<(v_i, v_j),(v_j, v_i)>$, связывающая эти вершины.

$$G_{InOut}(i) = T(G_{In}(i) \cup G_{Out}(i)).$$

Множество его вершин – это по-прежнему вершины графа $G_{In}(i) \cup G_{Out}(i)$. Проводимость каждого вновь созданного ребра рассчитывается как среднее геометрическое проводимостей ребер, составляющих цепочку.

2. Оператор *A* – добавление вершин. Запись: $A_{j_1,\ldots,j_k}(G_1, G_2)$ означает, что из графа G_2 в граф G_1 будет добавлено k вершин с номерами $j_1, \ldots, j_k$ вместе со всеми ребрами, соединяющими эти вершины с вершинами G_1.

Операции над графами

3. Оператор *E* (удаление вершин) применяется к одному графу.

Запись: $E_{j_1',\ldots,j_h'}(G)$ означает, что из графа G будет удалено h вершин с номерами j_1', …, j_h' вместе с их инцидентными ребрами.

Тогда на шаге $i + 1$ удаление из графа $G_{ln}(i)$ вершин с номерами j_1', …, j_h', где $h \le m$ (m – мощность множества вершин $G_{ln}(i)$), запишется в следующем виде:

$$G_{ln}(i+1) = E_{j_1',\ldots,j_h'}(G_{ln}(i)).$$

Операции над графами

Будем считать, что сначала к графу $G_{In}(i)$ применяется оператор E, а затем к результату – оператор A.

Операторы не коммутируют, порядок их применения важен.

Таким образом, на шаге $i + 1$ входной граф запроса находится по следующей рекуррентной формуле:

$$G_{In}(i+1) = A_{j_1,\ldots,j_k}(E_{j_1',\ldots,j_h'}(G_{In}(i))).$$

Непосредственно из этой формулы вытекает, что каждый новый входной подграф однозначно определяется входным и выходным подграфами на предыдущем шаге и парой последовательностей натуральных чисел переменной длины: ($\{j_1', \ldots, j_h'\}$; $\{j_1, \ldots, j_k\}$).

Заключение

Топология изменяется автоматически таким образом, что наиболее востребованная информация оказывается наиболее доступной. Ассоциативность сети заключается не только в адресации по содержанию, но и в структуре взаимосвязей моделируемой предметной области, в которой близость понятий определяется не только и не столько семантикой, сколько самим функционированием сети, т.е. пользовательскими запросами и ответами на них.

Управление движением ресурса сквозь сеть осуществляется как самой топологией сети, которая направляет ресурс по ребрам с большей проводимостью, так и рекурсивным заданием нового входного множества и продолжением поиска в заданном направлении.

Спасибо за внимание