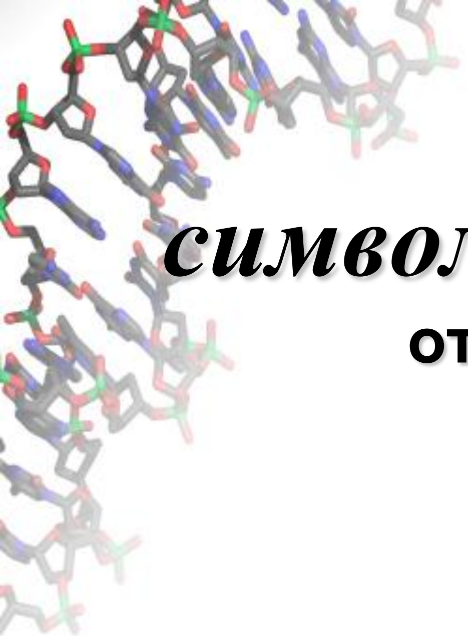


Анализ символьных последовательностей

от биоинформатики до лингвистики

М.А. Ройтберг

Занятие 3

ГИПЕРГРАФЫ

(часть 2)

Яндекс

25 февраля 2014

ПОВТОРЕНИЕ.
ЧТО БЫЛО НА 2-М ЗАНЯТИИ

ГИПЕРГРАФЫ: ФОРМАЛЬНЫЕ ОПРЕДЕЛЕНИЯ

1. Гиперграф, гиперребро

Ориентированный гиперграф (ОГ-граф) над полукольцом *A* – это

пятерка $G = (V, q_0, E, A, f)$, где

V – конечное множество вершин;

$q_0 \in V$ – *стартовая* вершина;

E – конечное множество *гиперребер,*

т.е. пар вида (v, W), где

$v \in V$,

W – упорядоченный список вершин из V;

A – полукольцо;

$f: E \to A$ – функция пометок на ребрах.

2. Гиперребро. Начальная вершина. Конечные вершины.

Вершина *v* называется *начальной* вершиной гиперребра *e=(v, W)*, вершины из *W* называются конечными вершинами этого гиперребра; значение *f(e)* $\in$ *A* называется *весом* гиперребра *e.*

Вершина v ОГ-графа G называется терминальной, если в G нет гиперребра, в котором v является начальной вершиной.

Аналог пути в графе - гиперпуть в гиперграфе

3. Гиперпуть.

Гиперпуть в ОГ-графе G = (V, q0, E, f) –это такое конечное упорядоченное помеченное дерево T, что

1) каждый узел дерева T помечен вершиной ОГ-графа G; одна вершина v из V может соответствовать нескольким узлам дерева T.

2) каждому внутреннему узлу r дерева T соответствует гиперребро e = (v, W), причем узел r помечен вершиной v, а список пометок в сыновьях узла r, взятых в порядке, предписанном деревом T, совпадает со списком W.

Корневое гиперребро гиперпути *T* – гиперребро, соответствующее корню *T.*
Начальная (корневая) вершина – вершина, соответствующее корню *T.*

4. Ацикличность

- Определение. ОГ-граф $G = (V, q_0, E)$ называется *ациклическим,* если в *G* не существует потока, в котором корень помечен той же вершиной, что и еще какой-либо узел этого потока.
- Очевидно, в *ациклическом* ОГ-графе существует лишь конечное число различных потоков.
- **Вес R*(T) гиперпути T* (аналог веса пути на графах) определяется рекурсивно.**

5. Вес гиперпути: рекурсивное определение

- Пусть $G = (V, q_0, E, f)$ – ОГ-граф; T – гиперпуть в G. Определим *вес R(T) гиперпути T* следующим образом.
- Если гиперпуть T состоит из единственного узла, то $R(T) = i$, где i – нейтральный элемент относительно операции * (умножения).
- Пусть корень T помечен гиперребром $e = (v, W)$; $x_1, \ldots, x_N$ – упорядоченный список сыновей корня дерева T; T_k – поддерево дерева T с корнем в узле x_k ($k = 1, \ldots, N = |W|$). Тогда

$$R(T) = f(e) * R(T_1) * \ldots * R(T_N)$$

5. Вес гиперпути: явное определение

- Пусть $G = (V, q_0, E, f)$ – ОГ-граф; T – гиперпуть в G. Пусть корень T помечен гиперребром $e = (v, W)$; $x_1, \ldots, x_N$ – упорядоченный список сыновей корня дерева T; T_k – поддерево дерева T с корнем в узле x_k ($k = 1, \ldots, N = |W|$).Тогда

$$R(T) = f(e) * R(T_1) * \ldots * R(T_N)$$

Вес гиперпути T это произведение весов гиперребер (в смысле операции *), соответствующих узлам T, причем порядок перемножения соответствует *левому обходу* дерева T в глубину.

6. Терминальные и полные пути

- Гиперпуть *T* в ОГ-графе *G* называется *терминальным,* если все его листья соответствуют терминальным вершинам ОГ-графа *G*.

- Терминальный гиперпуть называется *полным,* если его корню соответствует стартовая вершина ОГ-графа *G*.

7. Основная задача на гиперграфах

- Гиперпуть *T* в ОГ-графе *G* называется *терминальным,* если все его листья соответствуют терминальным вершинам ОГ-графа *G*.
- Терминальный гиперпуть называется *полным,* если его корню соответствует стартовая вершина ОГ-графа *G*.

- Определение 6.2. *Обобщенной статистической суммой* ОГ-графа *G* называется сумма (в смысле операции +) *S(G)* весов всех его полных гиперпутей.

- ***Задача.*** **Найти обобщенную статистическую сумму для заданного ациклического ОГ-графа G**

РЕШЕНИЕ ОСНОВНОЙ ЗАДАЧИ

Найти обобщенную статистическую сумму для заданного ациклического ОГ-графа G

- *Обозначения. Пусть* $v \in V$.
- G_v – это подграф ОГ-графа G, порожденный всеми вершинами G, достижимыми из v, т.е. вершинами, которые встречаются в гиперпутях, корень которых соответствует вершине v.
- *$S(V)$ – значение обобщенной статистической суммы для гиперграфа G_v – т.е. сумма весов всех полных путей в G_v*

Вычисление $S(V_0)$ ОГ-графа G (V_0, q_0, E, f)

Общее описание алгоритма

- Перебираем вершины *ОГ-графа G* в обратном топологическом порядке и для каждой вершины $v \in V$ вычисляем S(V).
- Для любой терминальной вершины *v* положим $S(v)=i$, где *i* – правый нейтральный элемент относительно «умножения».
- ***Как вычислять S(v), если v - не терминальная вершина?***

Вычисление S(v) для внутренней вершины v.

- $e_1 = \langle v; x_1, \ldots, x_k \rangle$, $e_2 = \langle v; y_1, \ldots, y_m \rangle$,
- $e_3 = \langle v; z_1, \ldots, z_n \rangle$ - все гиперребра с начальной вершиной *v*
- ***H(w)*** – множество всех гиперпутей с начальной вершиной *w*
- $\boldsymbol{H_e(g)}$ – множество всех гиперпутей с корневым гиперребром *g*
  - $\boldsymbol{H(v) = H_e(e_1) + H_e(e_2) + H_e(e_3)}$
- *Пусть* $\boldsymbol{S_e(e_r)}$ *обозначает сумму весов всех гиперпутей из* $H_e(e_r)$, $r = 1, 2, 3.$
  - $$\boldsymbol{S(v) = S_e(e_1) + S_e(e_2) + S_e(e_3)}$$

-

Вычисление $S_e(e_1)$ - 1

- $e_1 = <v; x_1, \ldots, x_k>$; $H_e(e_1)$ – множество всех гиперпутей с корневым гиперребром g
- $S_e(e_1)$ - *сумма весов всех гиперпутей из* $H_e(e_1)$.
  - $S_e(e_1) = SUM(R(T) \mid T \in H_e(e_1))$
- *Пусть* T *– гиперпуть из* $H_e(e_1)$; Пусть корень T помечен гиперребром $e = (v, W)$; T_j – поддерево дерева T с корнем j-м сыне корня T. *Тогда* T_j лежит в $H(x_j)$ и
  - $R(T) = f(e_1)*R(T_1)*\ldots*R(T_k)$
- Отсюда
- $S_e(e_1) = SUM(f(e_1)*R(T_1)*\ldots*R(T_k) \mid T \in H_e(e_1)) =$
- $= f(e_1)* SUM(R(T_1)*\ldots*R(T_k) \mid T \in H_e(e_1))$

Вычисление $S_e(e_1)$ - 2

- $S_e(e_1) = SUM(f(e_1)*R(T_1)*\ldots*R(T_k) \mid T \in H_e(e_1)) =$
- $= f(e_1)*SUM(R(T_1)*\ldots*R(T_k) \mid T \in H_e(e_1))$

$$SUM(R(T_1)*\ldots*R(T_k) \mid T \in H_e(e_1)) =$$
$$= SUM(R(X_1)*\ldots*R(X_k) \mid X_1 \in H(x_1); \ldots; X_k \in H(x_k)) =$$
$$= SUM(R(X_1) \mid X_1 \in H(x_1))* \ldots * SUM(R(X_k) \mid X_1 \in H(x_k)) =$$

- $= S(x_1)*\ldots*S(x_k)$

- $S_e(e_1) = f(e_1)*S(x_1)*\ldots*S(x_k)$

-

Вычисление S(*v*) для внутренней вершины v. Окончание

- $S(v) = S_e(e_1) + S_e(e_2) + S_e(e_3)$

- $S_e(e_1) = f(e_1) * S(x_1) * \ldots * S(x_k)$

$$S(v) = f(e_1) * S(x_1) * \ldots * S(x_k) + f(e_2) * S(y_1) * \ldots * S(y_m) + f(e_3) * S(z_1) * \ldots * S(z_n)$$

Время работы алгоритма

Предполагаем: время перехода к обработке очередной вершины и время доступа к ранее вычисленным суммам не зависят от размера графа.

$$S(v) = f(e_1) * S(x_1) * \ldots * S(x_k) +$$
$$+ f(e_2) * S(y_1) * \ldots * S(y_m) +$$
$$+ f(e_3) * S(z_1) * \ldots * S(z_n)$$

T – сумма времен обработки гиперребер. Время обработки гиперрребра ~ количества его концевых вершин => ***T*** ~ суммарное количество концевых вершин всех гиперребер.

- Если степени всех гиперребер ограничены, то

T ~ количество гиперребер.

ОСТАЛОСЬ ПОСЛЕ 2-ГО ЗАНЯТИЯ:

- Что делать, если обратный топологический порядок на множестве вершин неизвестен?

- Как вычислить сумму весов всех гиперпутей, содержащих данное гиперребро или данную вершину?

- Как перенести алгоритм Дейкстры на гиперграфы?

ОСТАЛОСЬ
ПОСЛЕ 2-го ЗАНЯТИЯ:

- Что делать, если обратный топологический порядок на множестве вершин неизвестен?

  === Рассмотрено на семинаре.

- Как вычислить сумму весов всех гиперпутей, содержащих данное гиперребро или данную вершину?

  === **БУДЕТ СЕГОДНЯ**

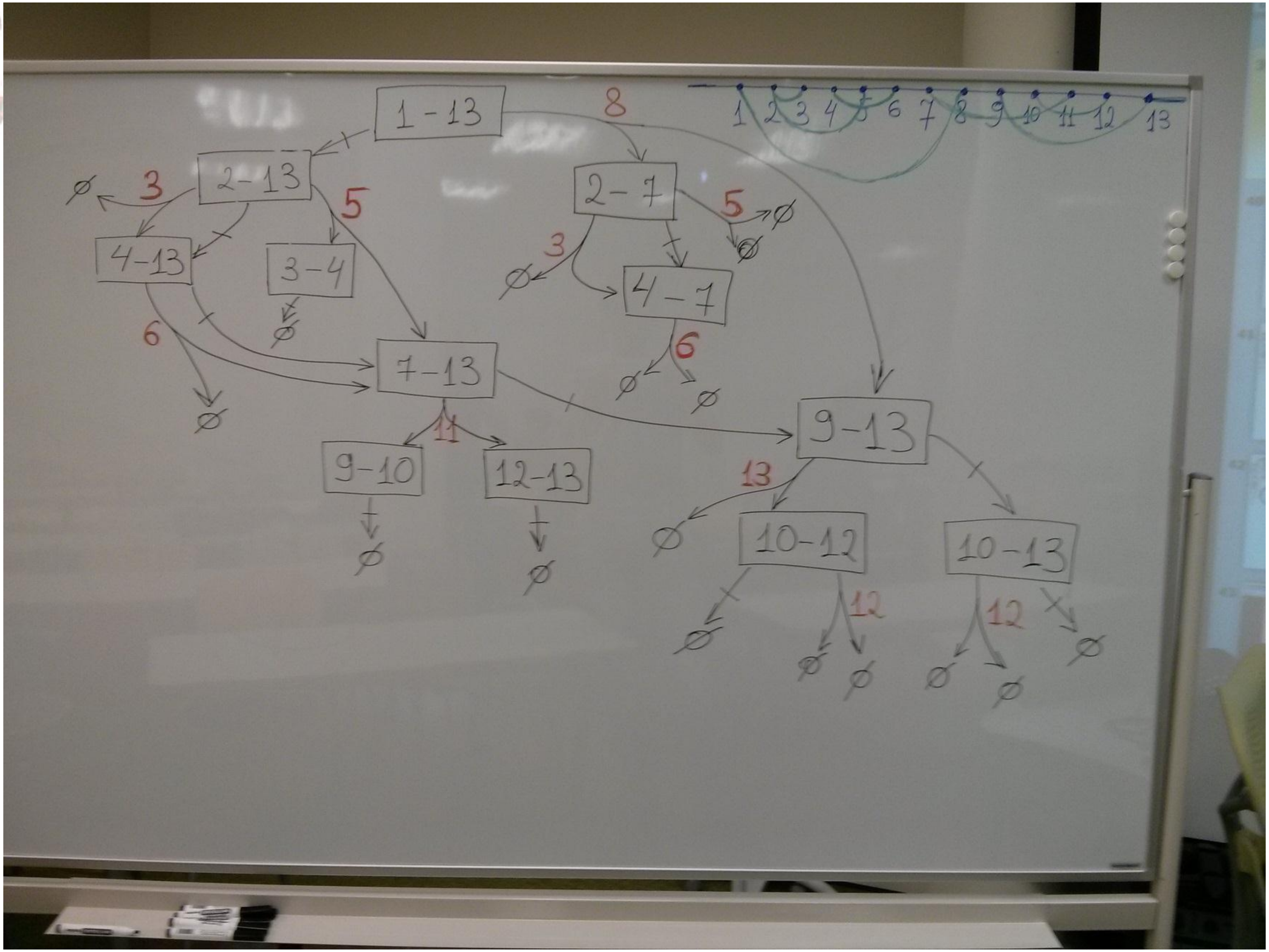
- Как перенести алгоритм Дейкстры на гиперграфы?

  === Если успеем…

Специальные суммы для гиперграфов

- 1. Постановка задачи. Воспоминания о графах.

Пример: гиперграф для задачи о дугах



- **Объяснить обозначения (красная цифра – конец дуги, начало дуги – начало отрезка)**
- **!!! Граф – сильно ациклический**
- **Кольцо – коммутативное.**

- **ДОДЕЛАТЬ!**
-

- **ПРИМЕР: Все гиперпути, проходящие, через [7, 13].**
- **!!! Нужен более сложный пример – чтобы слева от [7, 13] что-то было**
    - **ДОДЕЛАТЬ!**

Специальные суммы для гиперграфов

- 1. Постановка задачи. Воспоминания о графах.

Специальные суммы для гиперграфов

- Пусть дан ОГ-граф $G = (V, v_0, E, K, f)$

  V – конечное множество вершин;

  $v_0 \in V$ – *стартовая* вершина;

  E – конечное множество *гиперребер,*

  т.е. пар вида (v, W), где $v \in V$,

  W – упорядоченный список вершин из V;

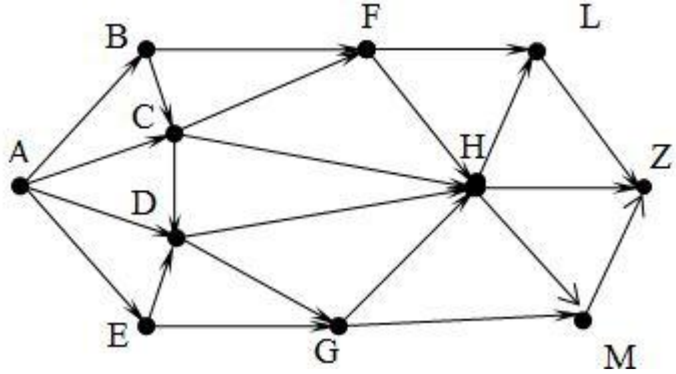
  K – полукольцо;

  $f: E \to A$ – функция пометок на ребрах.

(Полная) Специальная (обобщенная статистическая) сумма S(v) для вершины *v* – это сумма весов всех полных гиперпутей, содержащих вершину *v*

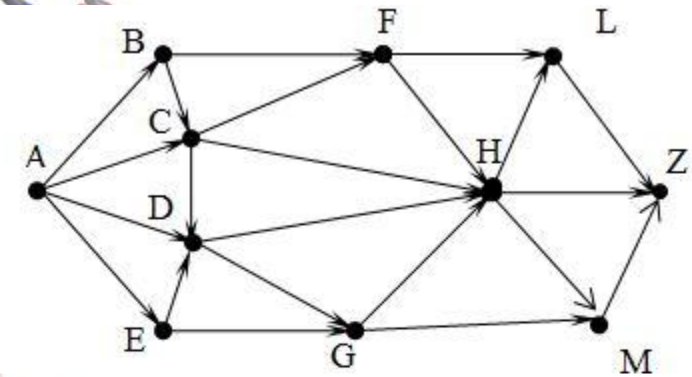
Задача. Найти значение *S(v)* для всех вершин *ОГ-графа G*

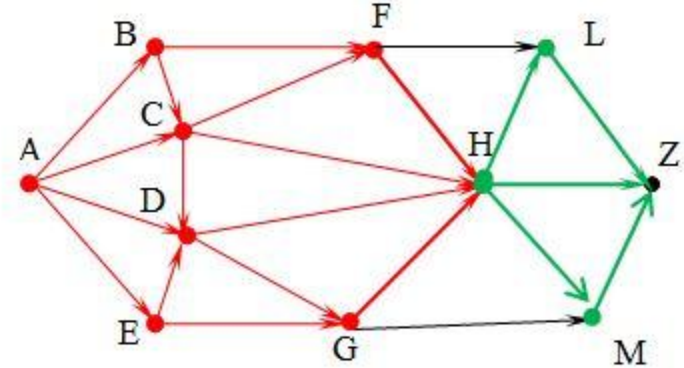
Как это было у графов



(Полная) Специальная (обобщенная статистическая) сумма S(v) для *в*ершины *v* – это сумма весов всех полных путей, содержащих вершину *v*

Как это было у графов





(Полная) Специальная (обобщенная статистическая) сумма S(v) для вершины *v* – это сумма весов всех полных путей, содержащих вершину *v.*

v-инициальный путь – инициальный путь,
который заканчивается в *v*

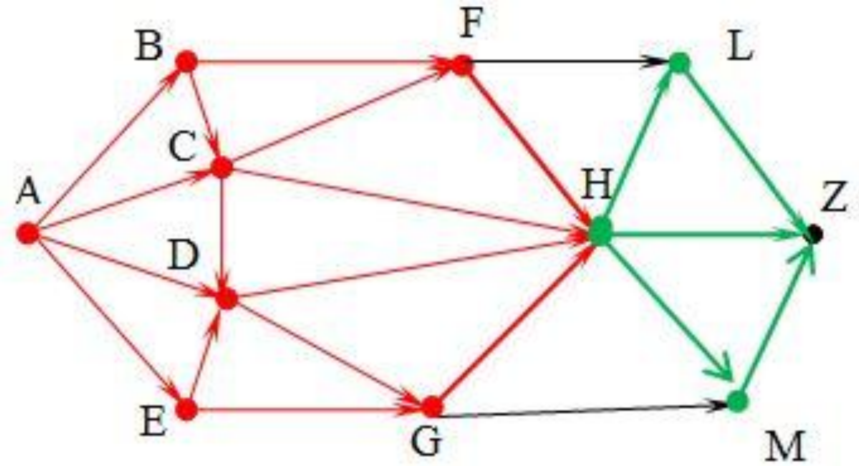
$S_I(v)$ – это сумма весов всех *v*-инициальных путей

v-терминальный путь **-** терминальный *путь,*
который начинается в v

$S_T(v)$ – это сумма весов всех *v*-терминальных путей.

$$\mathbf{S(v) = S_I(v) * S_T(v)}$$

Как это было у графов



$$S(v) = S_I(v) * S_T(v)$$

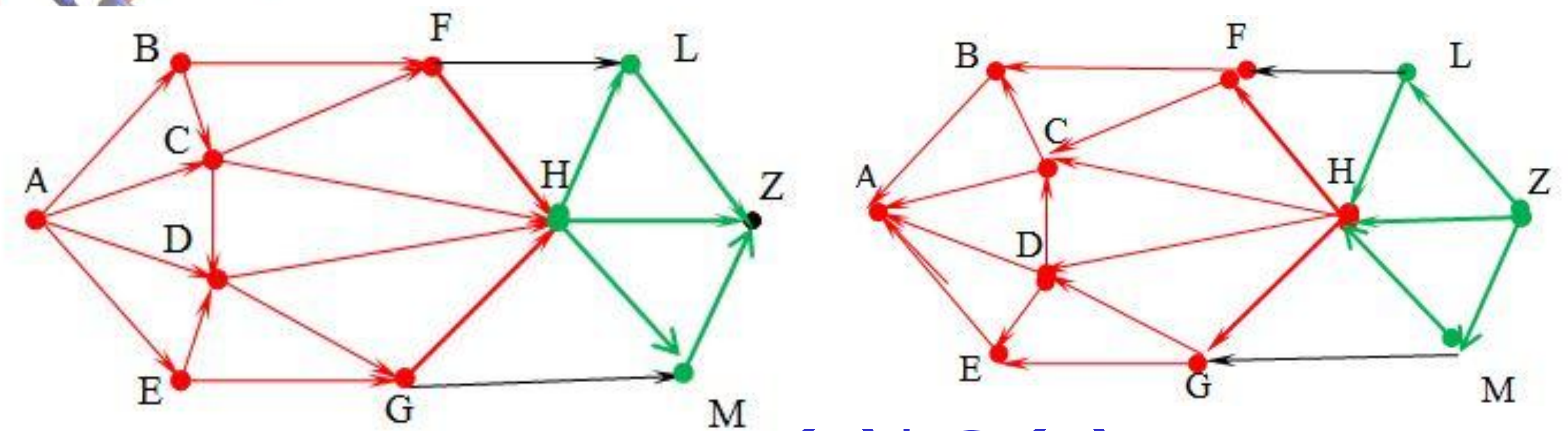
$$S(H) = \sum_{i,\,j}\{P(St_i(H)\cdot Term_j(H)\} =$$

$$= \sum_{i,\,j}\{P(St_i(H))*P(Term_j(H))\} =$$

$$= \sum_i\{P(St_i(H))\} * \sum j\{P(Term_j(H))\} =$$

$$= S_I(H) * S_T(H)$$

Как это было у графов



$$S(v) = S_I(v) * S_T(v)$$

Значение $S_T(v)$ **для всех** вершин *v* находим с помощью алгоритма ДП.
Значения $S_I(v)$ **для всех** вершин *v* находим с помощью алгоритма ДП для «обращенного» графа.

Для гиперграфов стрелки обращать нельзя!

Значение $S_T(v)$ **для всех** вершин *v* *гиперграфа* находим с помощью алгоритма ДП.

НО:

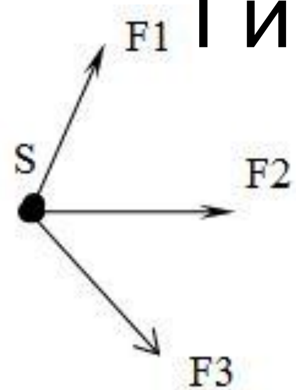
Значения $S_I(v)$ **для всех** вершин *v* *гиперграфа* *НЕ МОЖЕМ* найти с помощью алгоритма ДП для «обращенного» гиперграфа.

Граф

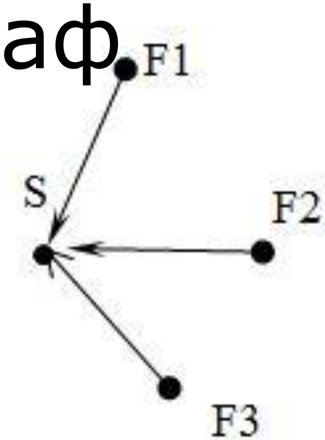
Ребро

Ребро

Гиперграф



Гиперребро



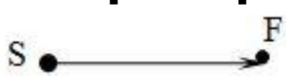
НЕ гиперребро

Для гиперграфов стрелки обращать нельзя!

Значение $S_T(v)$ для всех вершин *v* <u>*гиперграфа*</u> находим с помощью алгоритма ДП.

ЧТО БУДЕТ АНАЛОГОМ $S_I(v)$ ДЛЯ ГИПЕРГРАФОВ?

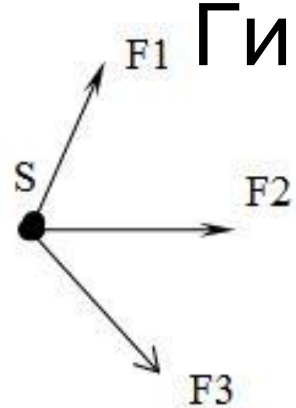
Граф



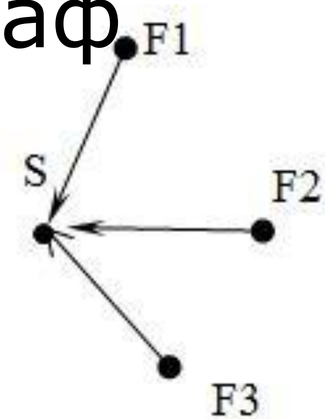
Ребро

Ребро

Гиперграф



Гиперребро

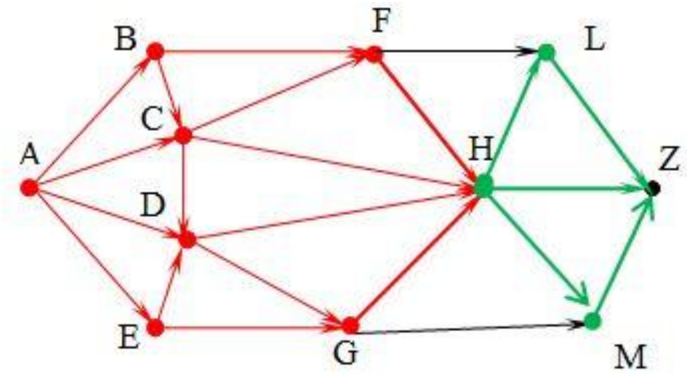


НЕ гиперребро

Специальные суммы для гиперграфов

- 2. Обобщение понятия *v-и*нициального пути и алгоритм

Как это было у графов: переформулировка опредедения



Опр. 1. v-инициальный путь – это
инициальный путь, который заканчивается в *v*

Опр.2. v-инициальный путь – это
полный путь, из которого удалена часть пути, которая начинается в *v* (если в пути нет узла, помеченного *v*, то для этого полного пути *v*-инициальный путь не существует).

v-инициальный путь для гиперграфов

v-инициальный гиперпуть – это полный гиперпуть, из которого удалено поддерево с корнем в узле, помеченном вершиной *v.*

Для полного гиперпути,
в котором нет узлов с пометкой v,
v-инициального пути не существует.

v-инициальный путь для гиперграфов

***v*-инициальный** гиперпуть – это полный гиперпуть, из которого удалено поддерево с корнем в узле, помеченном вершин**ой** *v*.

Предположение: АОГ-граф ***сильно*** *ациклический.*

ПРИМЕР.

- 6 полных деревьев – 2 v-инициальных и 3 v-терминальных.
- Рисунок.
- **ДОДЕЛАТЬ!!**

v-инициальный путь для гиперграфов

***v*-инициальный** гиперпуть – это полный гиперпуть, из которого удалено поддерево с корнем в узле, помеченном вершиной *v*.

$S_I(v)$ – это сумма весов всех *v*-инициальных путей.

***v*-терминальный** путь **-** терминальный *путь,* который начинается в v

$S_T(v)$ – это сумма весов всех *v*-терминальных путей.

$$\mathbf{S(v) = S_I(v) * S_T(v)}$$

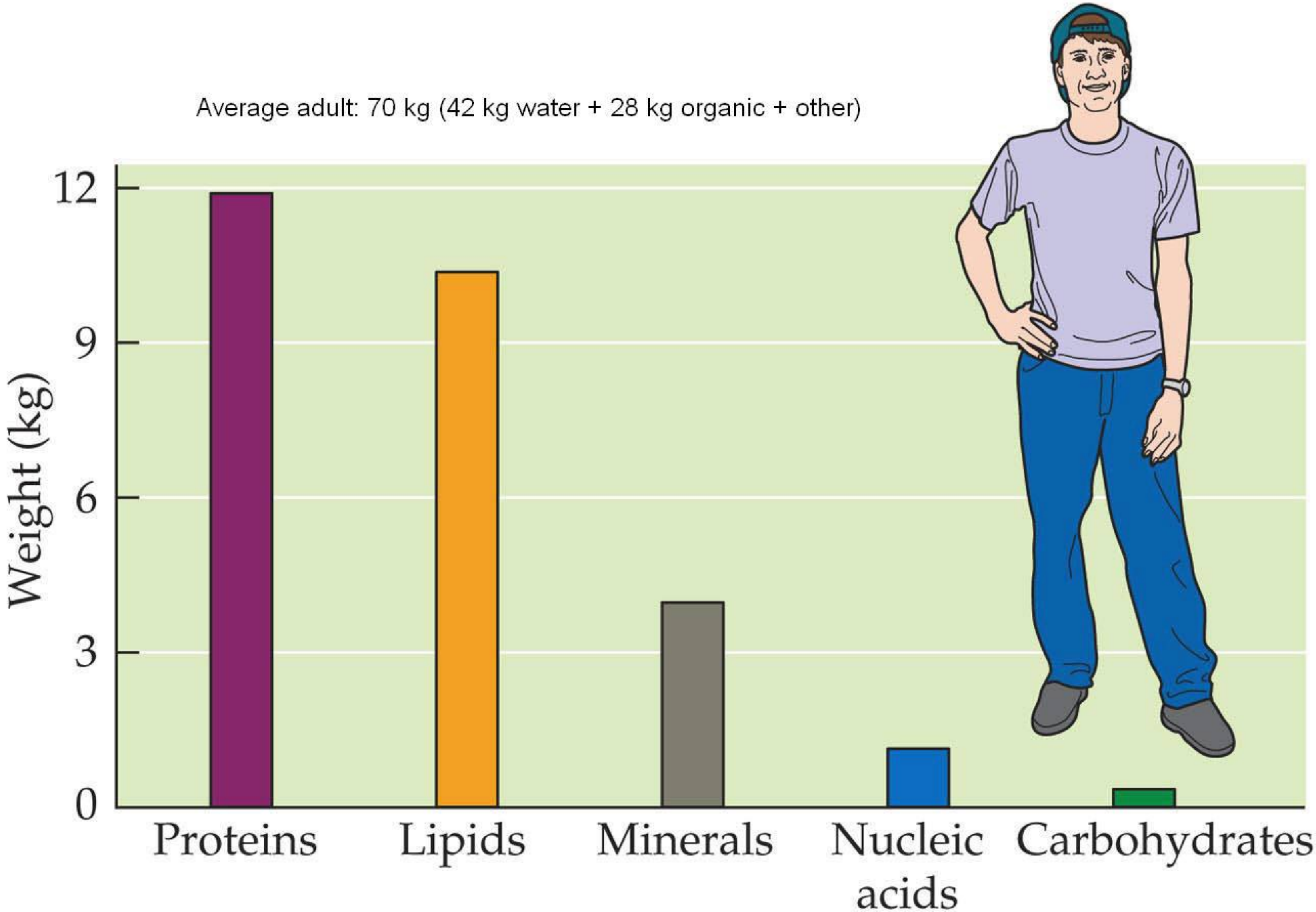
v-сумма для гиперграфов

$$S(v) = S_I(v) * S_T(v)$$

В случае гиперграфов требуется
коммутативность умножения

Отступление

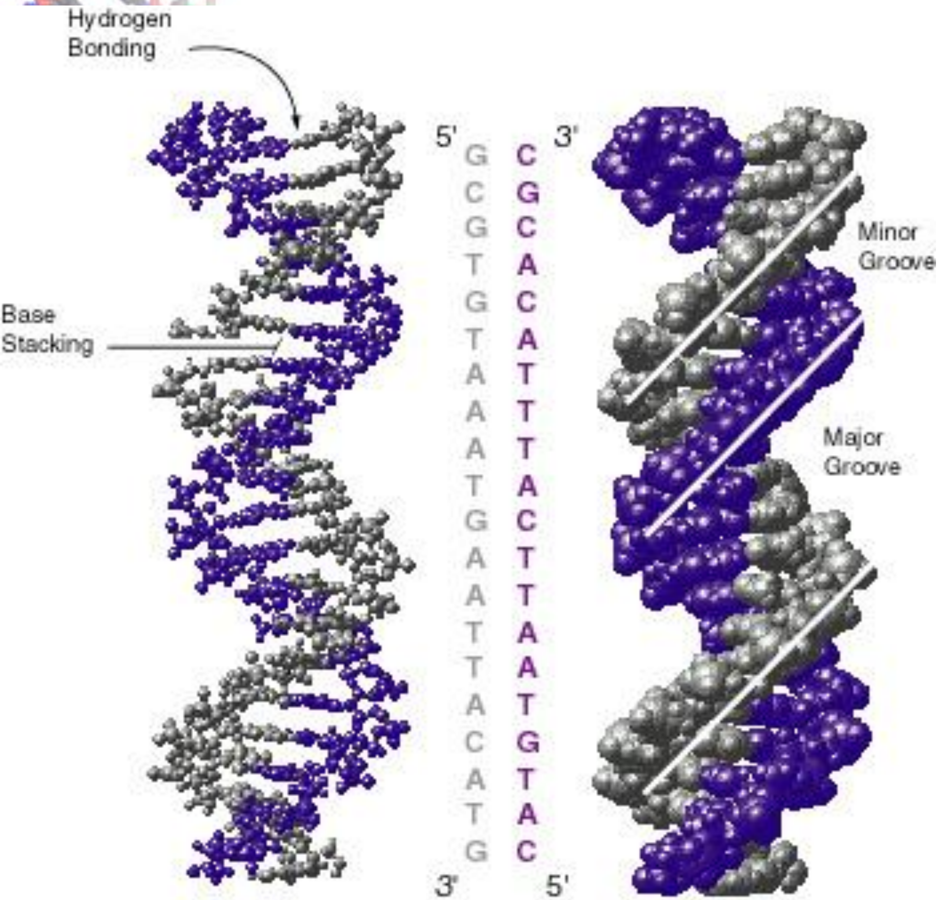
Немного биологии



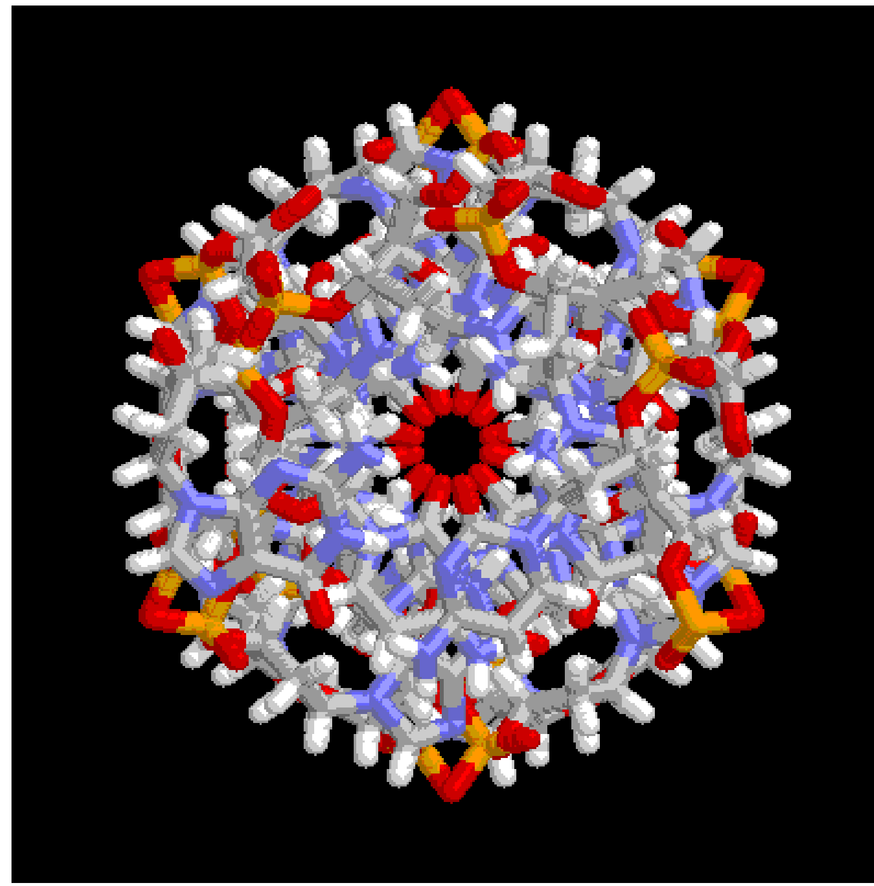
ANIMAL PHYSIOLOGY, Figure 4.1

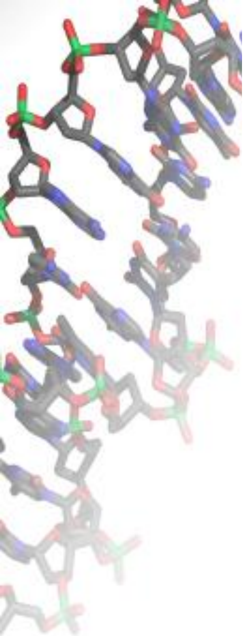
ДНК: 2 нити; L ~ $10^5 - 10^9$

нуклеотиды (4)

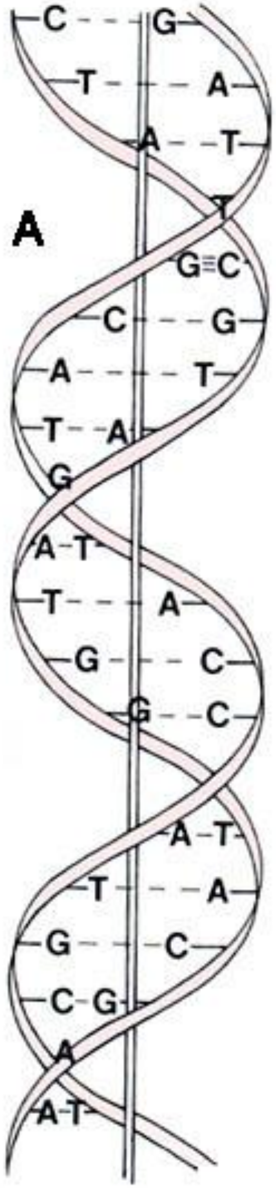


Miesfeld / Applied Molecular Genetics
Fig. 01.03. DNA Structure.



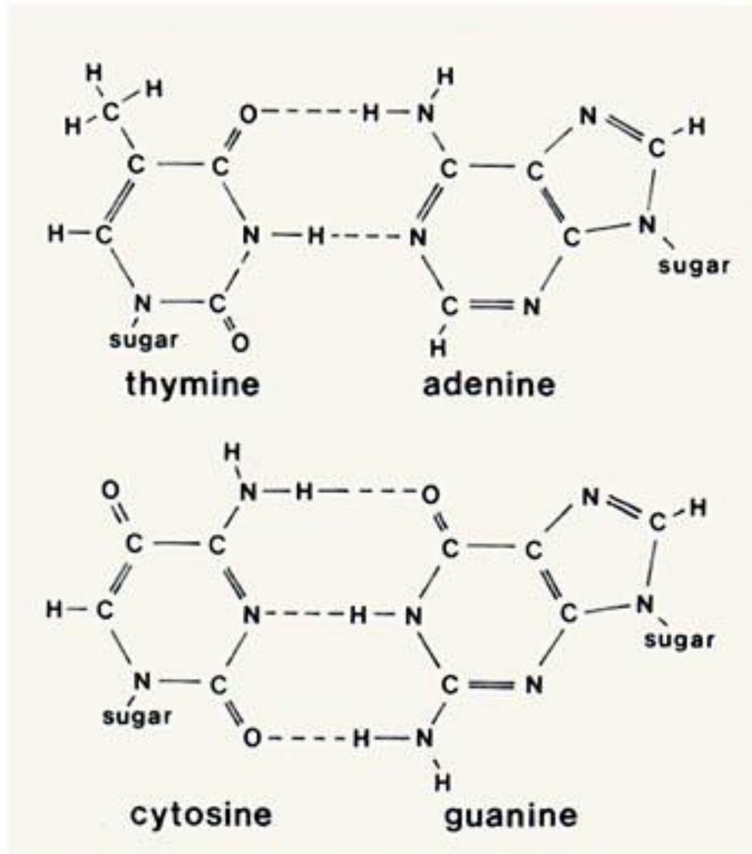


A



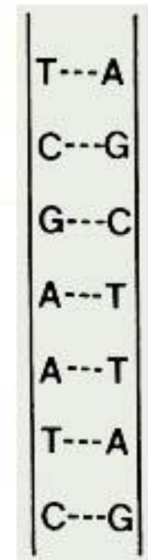
C G
T A
A T
T
G≡C
C G
A T
T A
G
A T
T A
G C
G C
A T
T A
G C
C G
A
A T

B



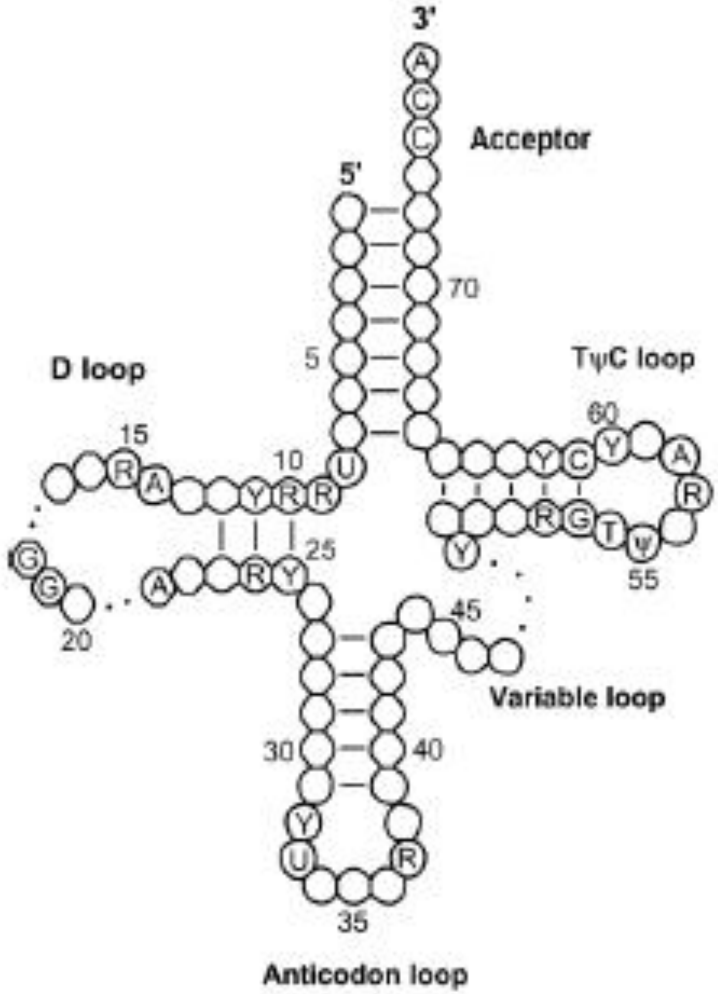
sugar
sugar
thymine
adenine
sugar
sugar
cytosine
guanine

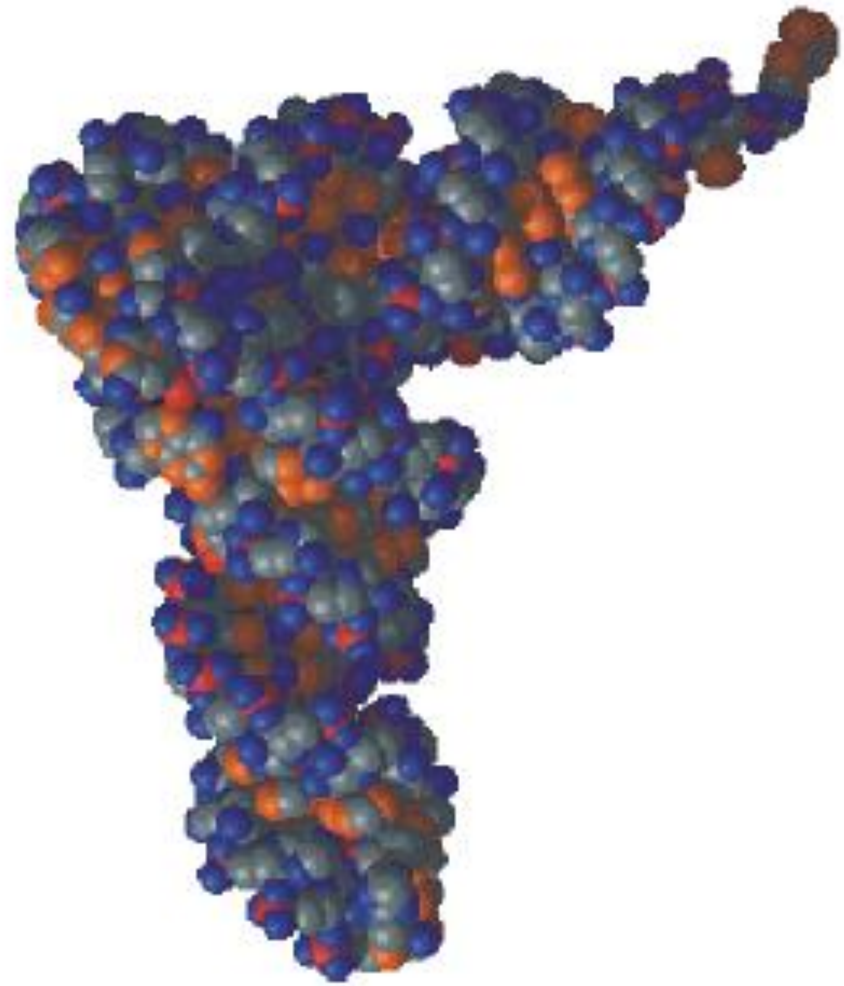
C



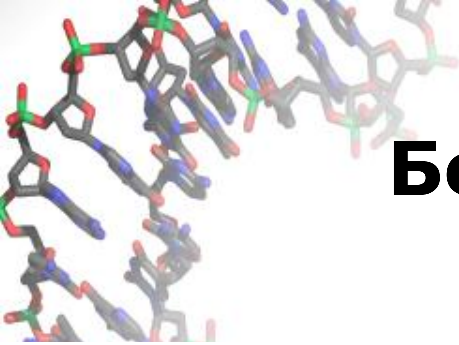
T---A
C---G
G---C
A---T
A---T
T---A
C---G

РНК: 1 нить; L ~ $10^2 – 10^3$
нуклеотиды (4)

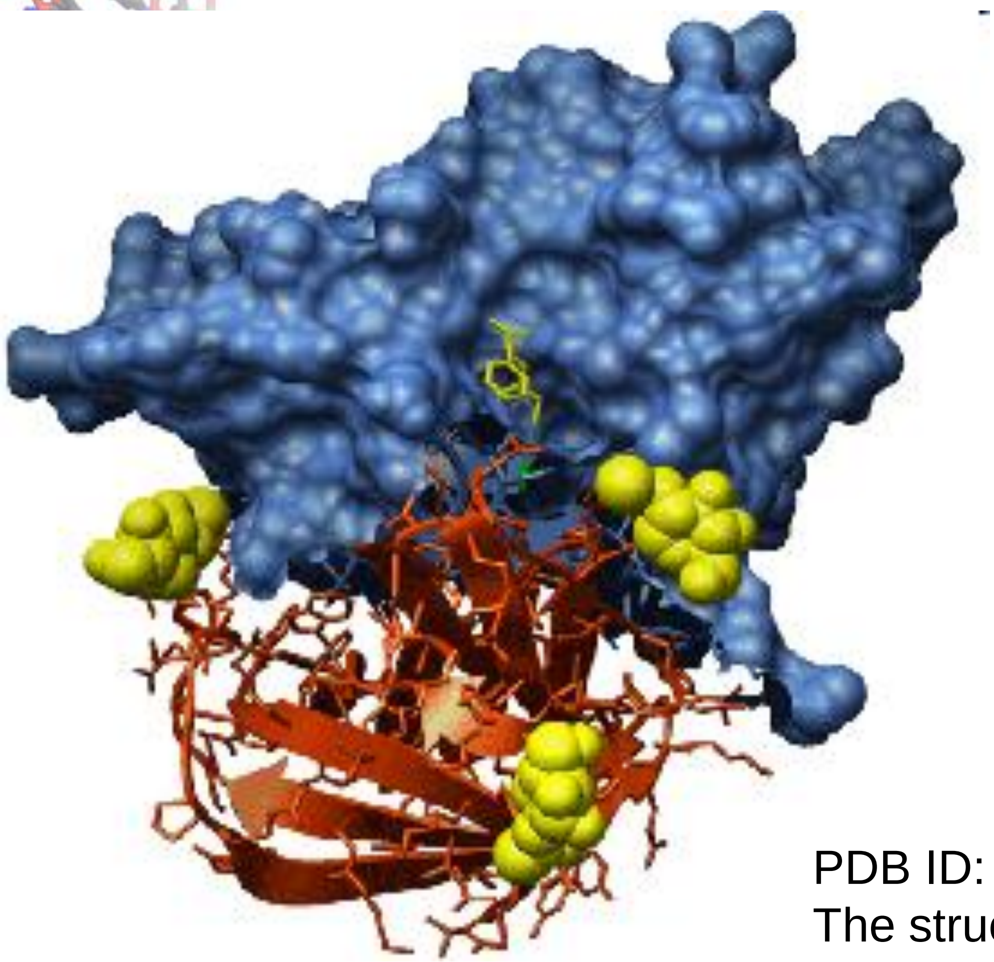




From Paul Higgs

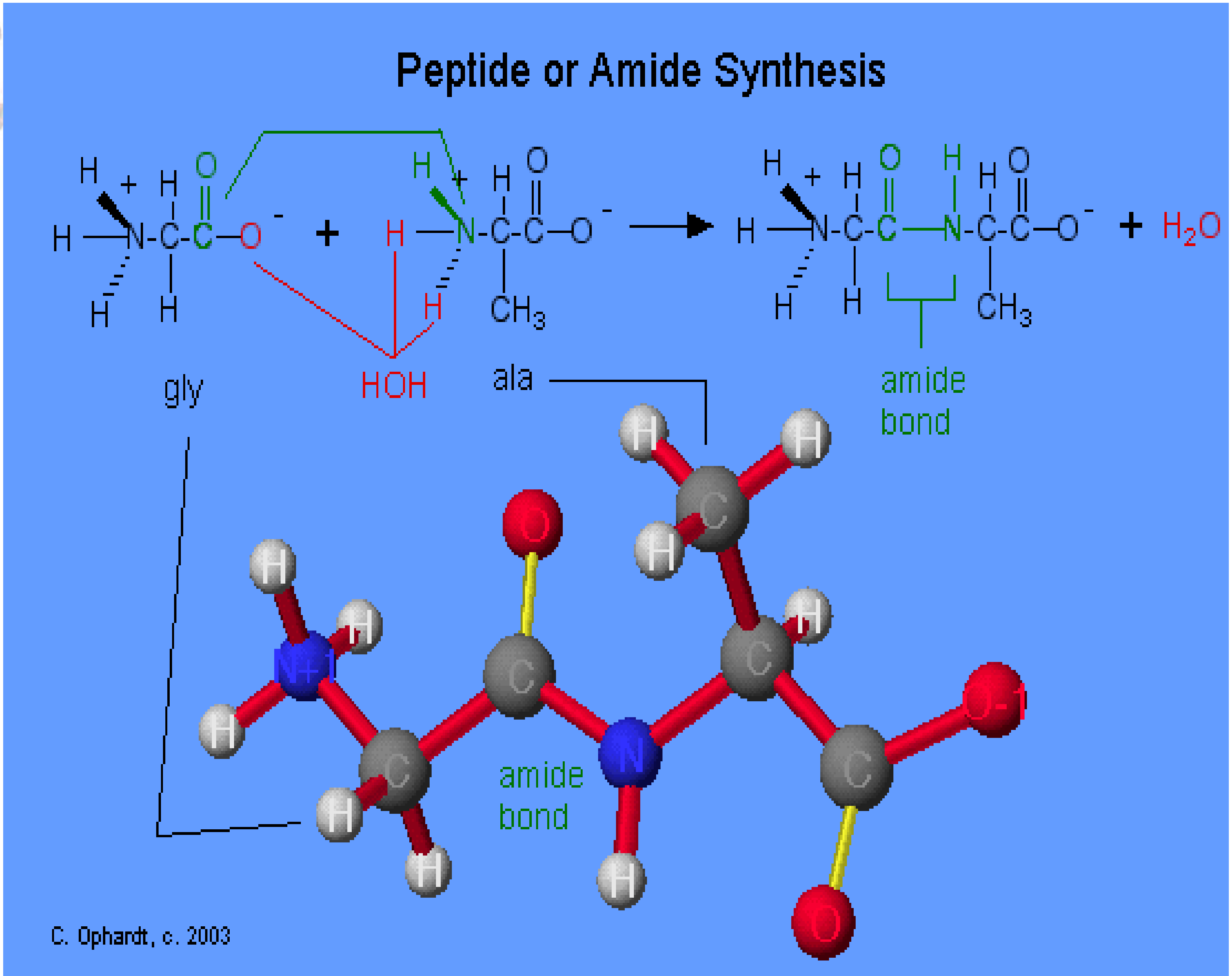


Белки: 1 нить; L ~ $10^2 - 10^3$
аминокислоты (20)



PDB ID: **2act** E.N. Baker, E.J. Dodson (1980):
The structure of actinidin at 1.7 Ångstroms

...Gly + Ala... = ...GA...

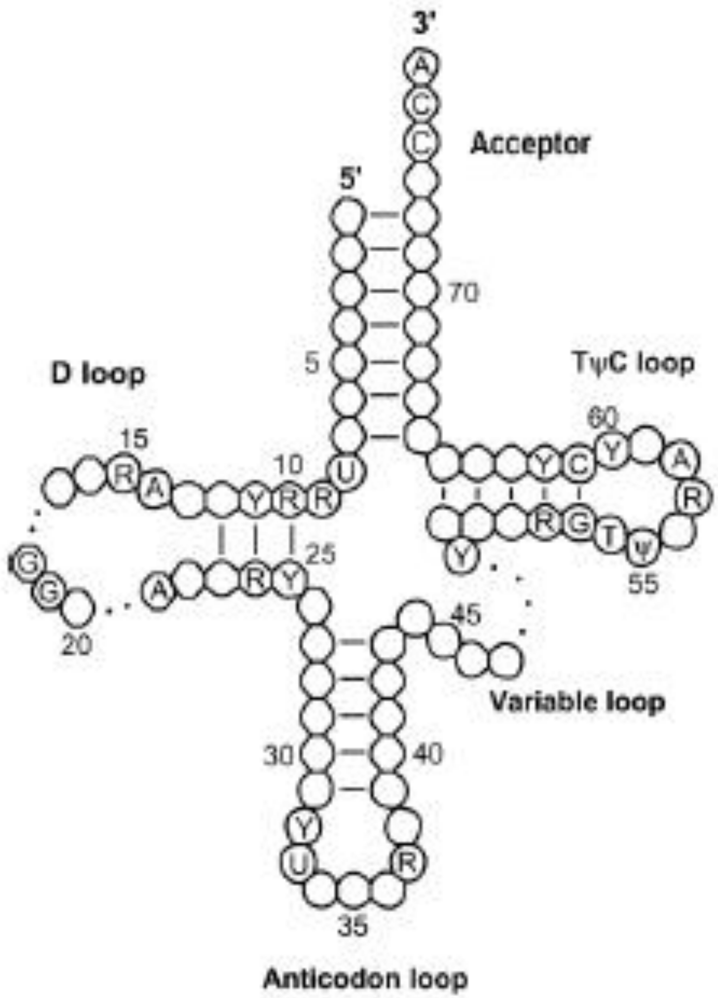


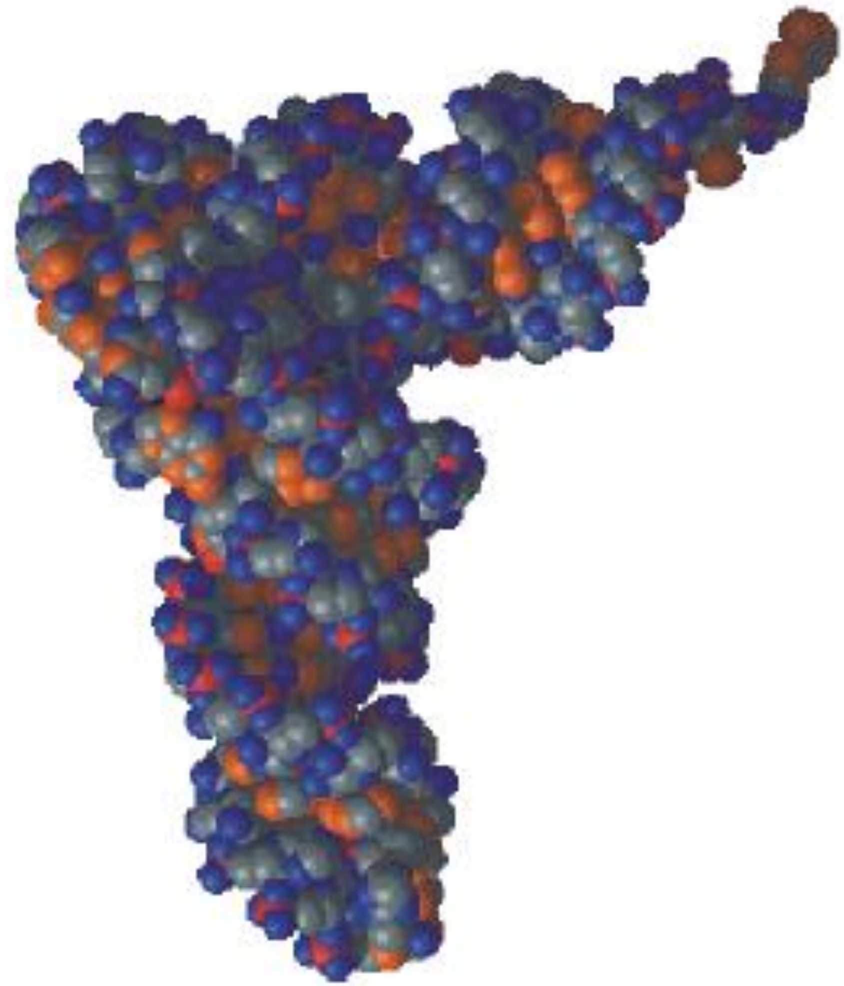
Peptide or Amide Synthesis
gly
HOH
ala
amide bond
amide bond
C. Ophardt, c. 2003

ИСТОРИЯ и ДЛИНЫ

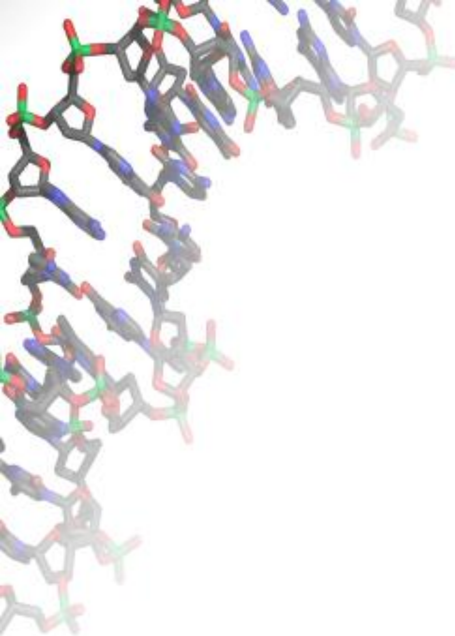
- tRNA - (**1964**) - **75** bases (old, slow, complicated method)
- First complete DNA genome: X174 DNA (**1977**) - **5386** bases
- human mitochondrial DNA (**1981**) - **16,569** bases
- tobacco chloroplast DNA (**1986**) - **155,844** bases
- First complete bacterial genome (*H. Influenzae*)(**1995**) - **1.9 x 10^6** bases
- Yeast genome (eukaryote at ~ **1.5 x 10^7**) completed in **1996**
- Several archaebacteria
- *E. coli* -- **4 x 10^6** bases [**1998**]
- Several pathogenic bacterial genomes sequenced
  - Helicobacter pyloris, Treponema pallidium, Borrelia burgdorferi, Chlamydia trachomatis, Rickettsia prowazekii, Mycobacterium tuberculosis
- Nematode C. elegans (~ **4 x 10^8**) - December **1998**
- Human genome (rough draft completed **2000**) - **3 x 10^9** base
- **2010** – rat, mouse, pig, fugu, etc, full genomes **50 x 10^9**
- ***~2015*** – individual human genomes (“$1000 per genome”)

Раздел 3.
Предсказание вторичной структуры РНК.
3.1. История

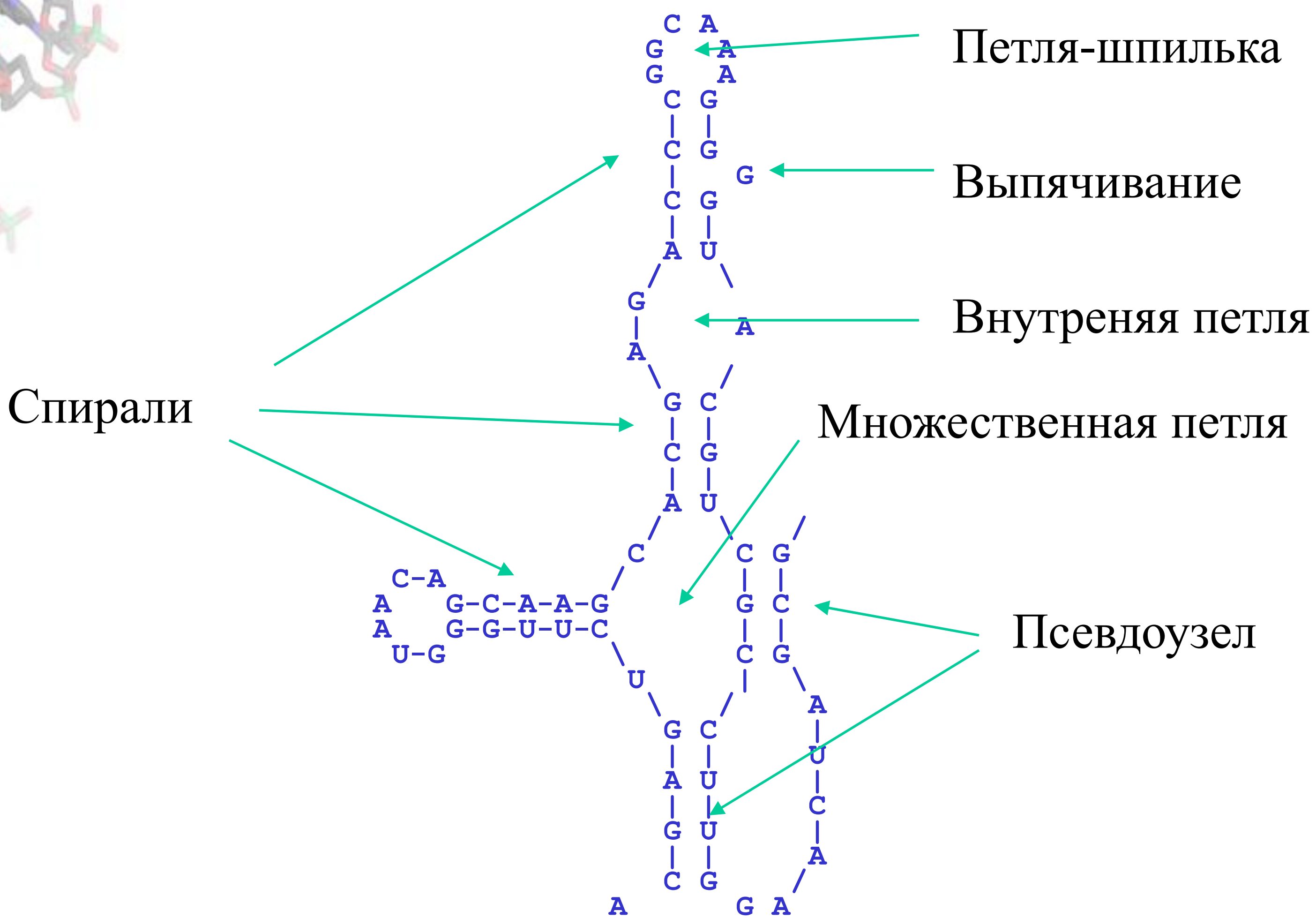




From Paul Higgs

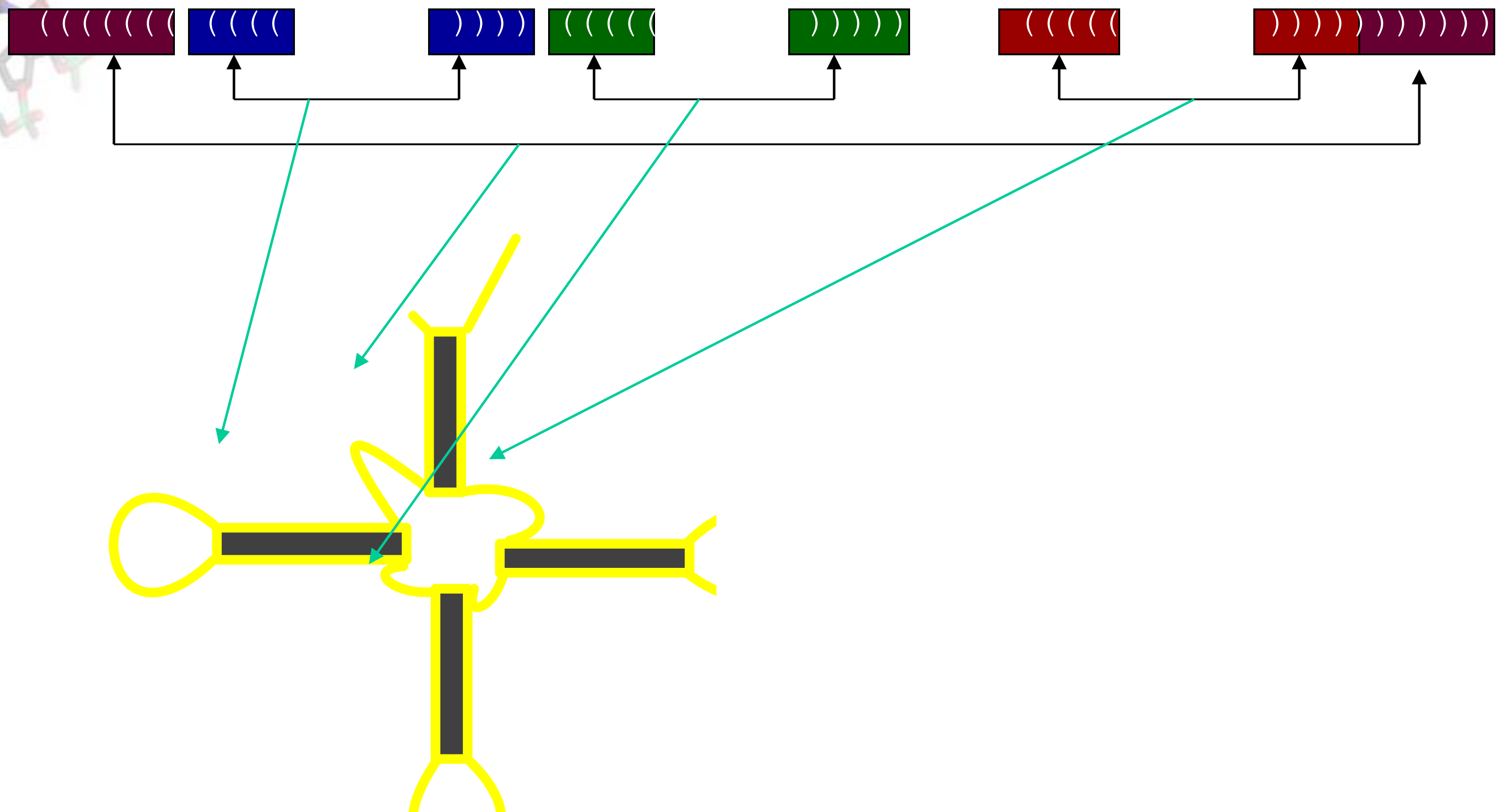


Элементы вторичной структуры



Представление вторичной структуры без псевдоузлов. "Скобочная структура"

gggctaTAGCTCAGcTGGGAGAGCgcctgcTTtgcACgcaggagGTCtgcGGTTCGAtCCCgcatagctccaCCA



Предсказание вторичной структуры РНК, свободной от псевдоузлов

- Задача 1. Найти скобочную структуру, содержащую максимально возможное количество скобок.
  - $T = \boldsymbol{O(L^3)}$

Уточнение постановки задачи – 1 (согласование с экспериментом)

- 1а. Разрешать связи, отличные от A-T, G-C.
- 1б. Считать не количество связей, а их суммарную энергию, энергия каждой возможной пары считается известной.

С алгоритмической точки зрения задача практически не меняется.

Уточнение постановки задачи – 2 (согласование с экспериментом)

- Учитывать вклад нуклеотидов, не участвующих в образовании водородных связей (“Nearest Neighbor model”).

  Приводит к существенному усложне-нию алгоритма, однако порядок сложности не меняется $T = \boldsymbol{O(L^3)}$

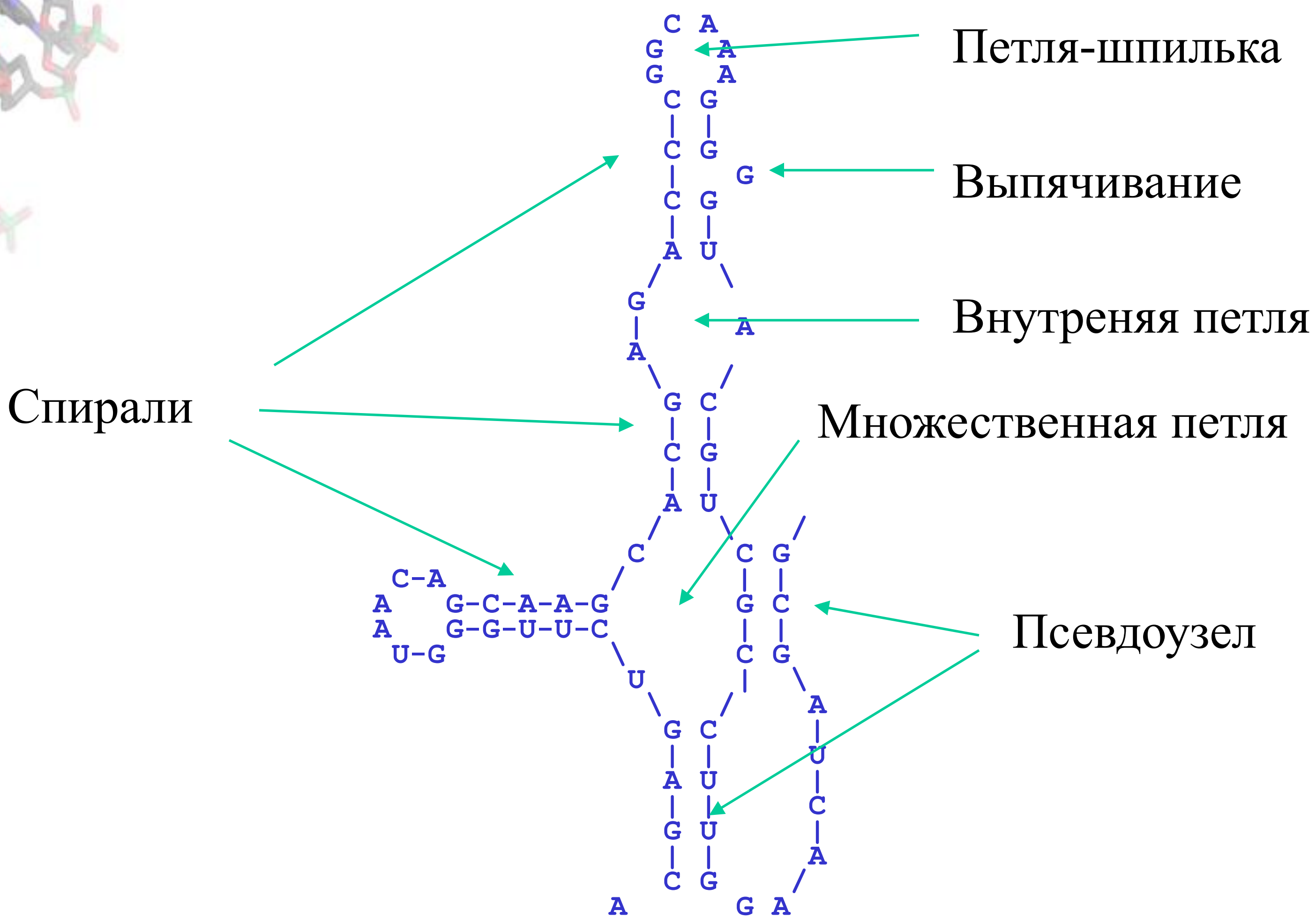
Уточнение постановки задачи – 3 (согласование с экспериментом)

- Молекула РНК может принимать не ту структуру, которой мы приписали оптимальную энергию, а несколько иную, например, из-за того, что мы не знаем точных значений энергетических параметров. Поэтому полезно не искать одну «оптимальную» структуру, а проанализировать все возможные структуры и оценить вероятность образования каждой отдельной связи («статистический вес» связи). Это также можно решить методом динамического программирования.

Уточнение постановки задачи – 4 (согласование с экспериментом)

- многие авторы пытаются выяснить вторичную структуру РНК, не сводя ее к какой-либо алгоритмической оптимизационной задаче, а путем моделирования реального процесса «сворачивания» молекулы РНК (т. е. установления и исчезновения водородных связей).

Элементы вторичной структуры



3а. Предсказание вторичной структуры РНК (внутренние петли, неветвящиеся структуры)

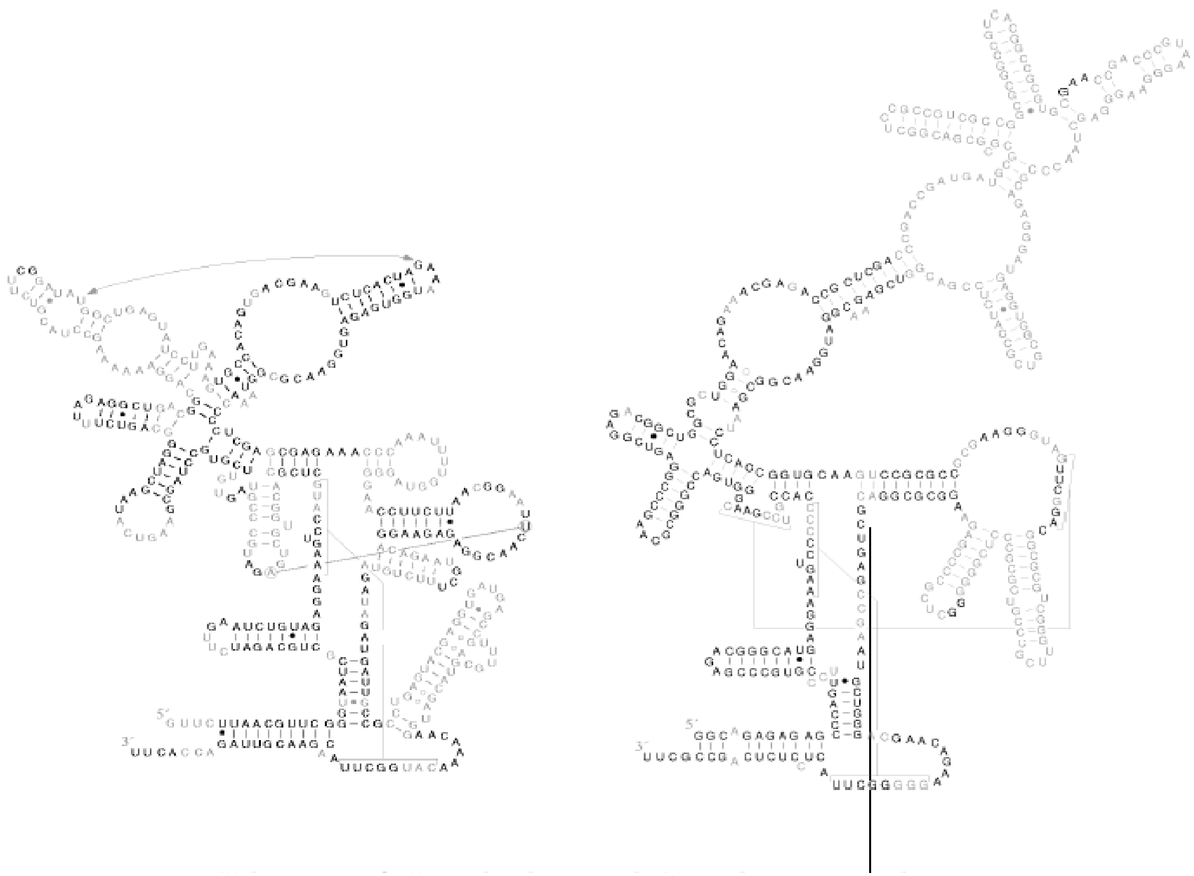
Время работы алгоритма:

Время: O(M•log(L)).

L – длина РНК

M – число возможных спариваний (M < L^2)

3б. Выравнивание последовательностей РНК с заданной вторичной структурой.



3б. Выравнивание последовательностей РНК с заданной вторичной структурой.

Весовая система– **это пятерка <*M, g, d, b, c*>, где**

M **– весовая матрица замен;**

g **и** ***d*** **– коэффициенты аффинной весовой функции удалений фрагментов;**

b **– бонус за одновременное сопоставление двух спаренных нуклеотидов,**

c **– штраф за «потерю» спаривания нуклеотидов**

$$W(G) = \sum_{k=1..n} M(S_1[p_k], S_2[q_k]) - g\cdot m - d\cdot l + b\cdot t - c\cdot k$$

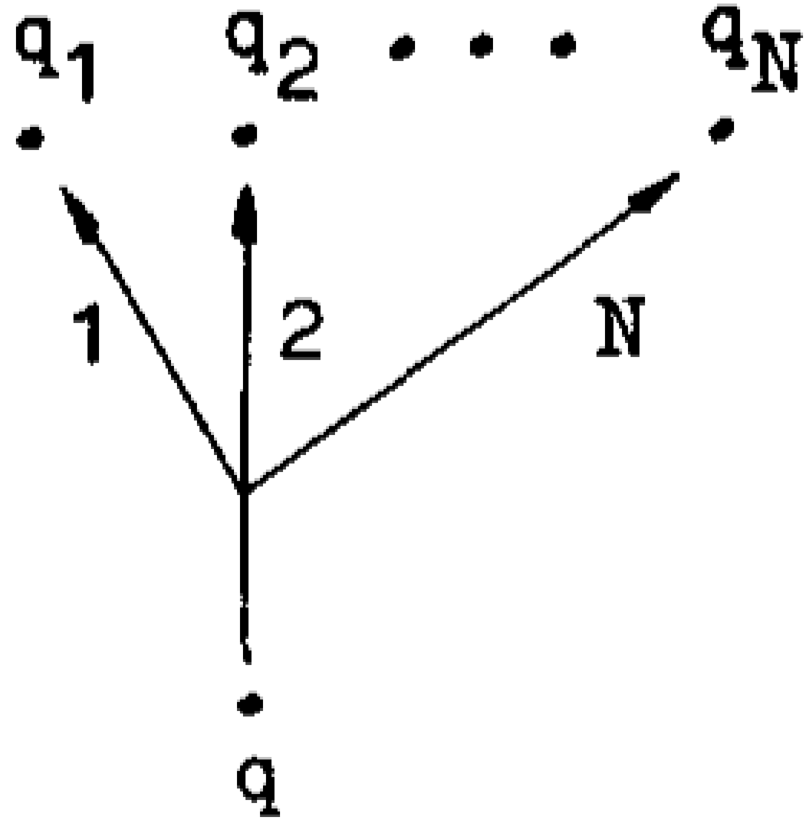
Время: $O(m^2\, n \log^2(n))$

Память: $O(m^2 n \log(n))$

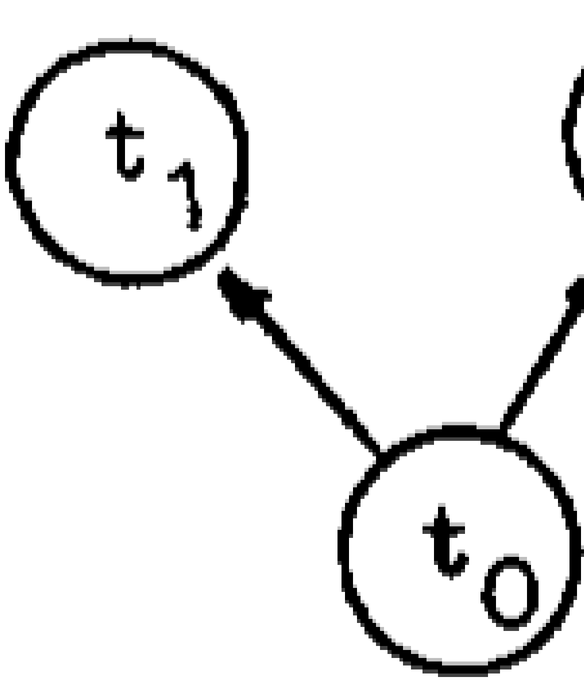
Раздел 3.
Предсказание вторичной структуры РНК.

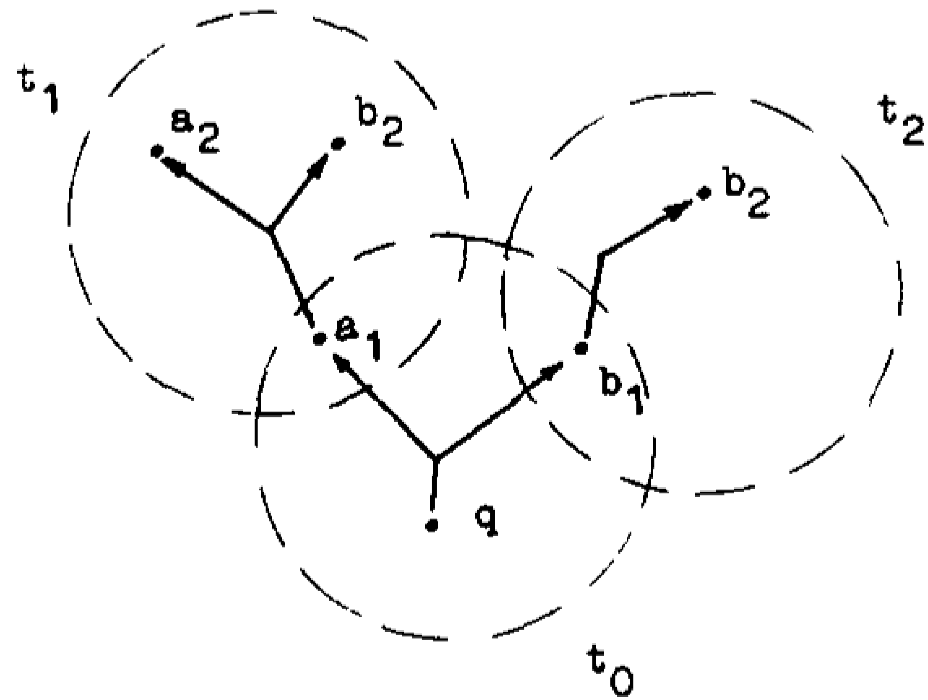
3.2. Оптимальная структура РНК и гиперграфы

(Ориентированный) Гиперграф: множество вершин и множество гиперребер

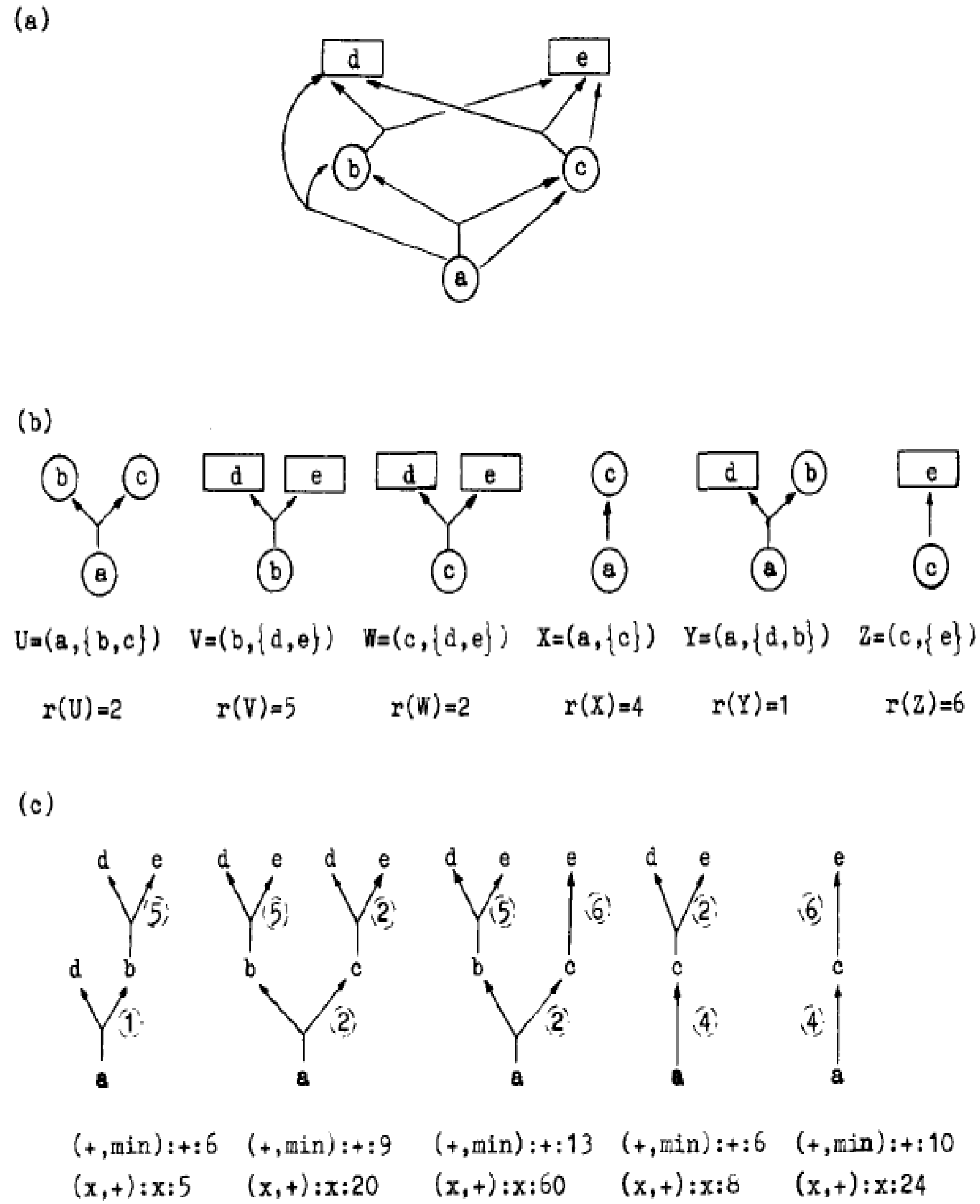


Гиперпуть





Вес гиперпути – ПРОИЗВЕДЕНИЕ (*) весов гиперребер


(a)
(b)
U=(a,{b,c}) V=(b,{d,e}) W=(c,{d,e}) X=(a,{c}) Y=(a,{d,b}) Z=(c,{e})
r(U)=2 r(V)=5 r(W)=2 r(X)=4 r(Y)=1 r(Z)=6
(c)
(+,min):+:6 (+,min):+:9 (+,min):+:13 (+,min):+:6 (+,min):+:10
(x,+):x:5 (x,+):x:20 (x,+):x:60 (x,+):x:8 (x,+):x:24

Графы и гиперграфы
Основные понятия

Вершина

Ребро ***Гиперребро***

Вершина-источник

Тупиковая вершина (сток)

Путь ***Гиперпуть***

Инициальный (гипер)путь

Терминальный (гипер)путь

Полный (гипер) путь:

Начальная вершина – источник;

Конечные вершины - тупиковые

Графы и гиперграфы

Основные понятия

Вес гипер(ребра)

«Умножение»: как вычислять вес (гипер)пути

«Сложение»: целевая функция [коммутат.]

Дистрибутивность:

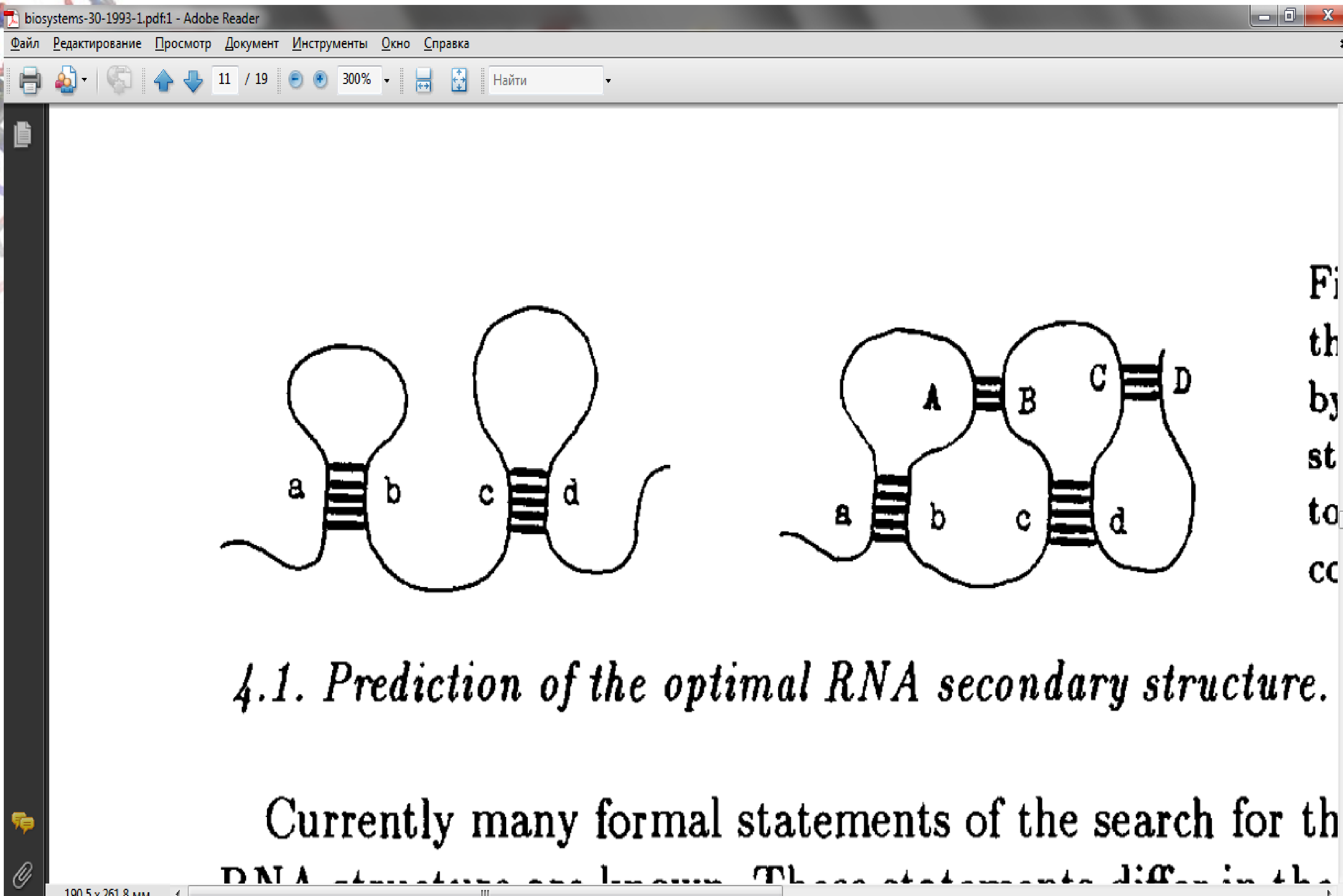
a*(b+c) = a*b+a*c; (b+c)*a = b*a+c*a

Вес пути ***Вес гиперпути***

ПРОБЛЕМА:

НАЙТИ «СУММУ» ВЕСОВ
ВСЕХ ПОЛНЫХ (ГИПЕР)ПУТЕЙ

Вторичная структура РНК (оптимальная скобочная структура)



4.1. Prediction of the optimal RNA secondary structure.

Currently many formal statements of the search for th

ВТОРИЧНАЯ СТРУКТУРА
строки *P[1..N]*

Алфавит: {A, T; G, C}

Комплементарные буквы:

A <-> T;

G <-> C

Пара (=дуга) *(i, j)* разрешена, если буквы *P[i] и P[j] комплементарны*

ВТОРИЧНАЯ СТРУКТУРА строки *P[1..N]*

- Множество пар $S = \{(i\_1, j\_1), \ldots, (i\_K, j\_K)\}$ *такое, что*

  (ia) $1 <= i\_r < j\_r <= N$ $(r = l, \ldots, K)$;

  (ib) все пары в *S* разрешены

(ii) если $(i,j) \in S$ и $(i', j') \in S$, то отрезки

[i, j] и [i′, j′] либо вложены один в другой,

либо не пересекаются

Вес структуры S – количество пар в ней.

$$W(S) = K$$

ОПТИМАЛЬНАЯ ВТОРИЧНАЯ СТРУКТУРА строки *P[1..N]*

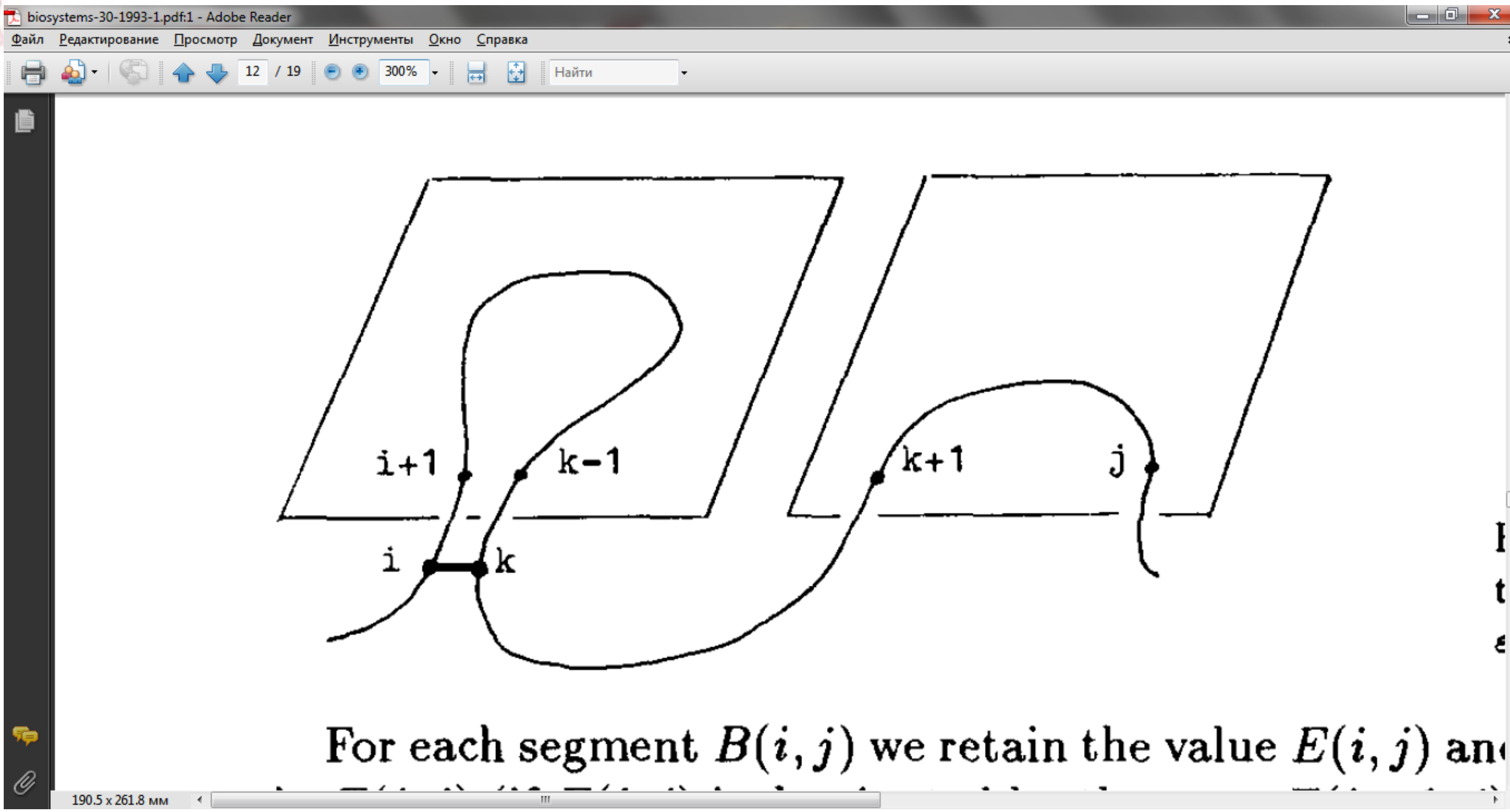
Оптимальная вторичная структура для строки *P – вторичная структура для P, имеющая наибольший возможный вес.*

$$W(P) = \max\{W(S) \mid S – \text{структура для } P\}$$

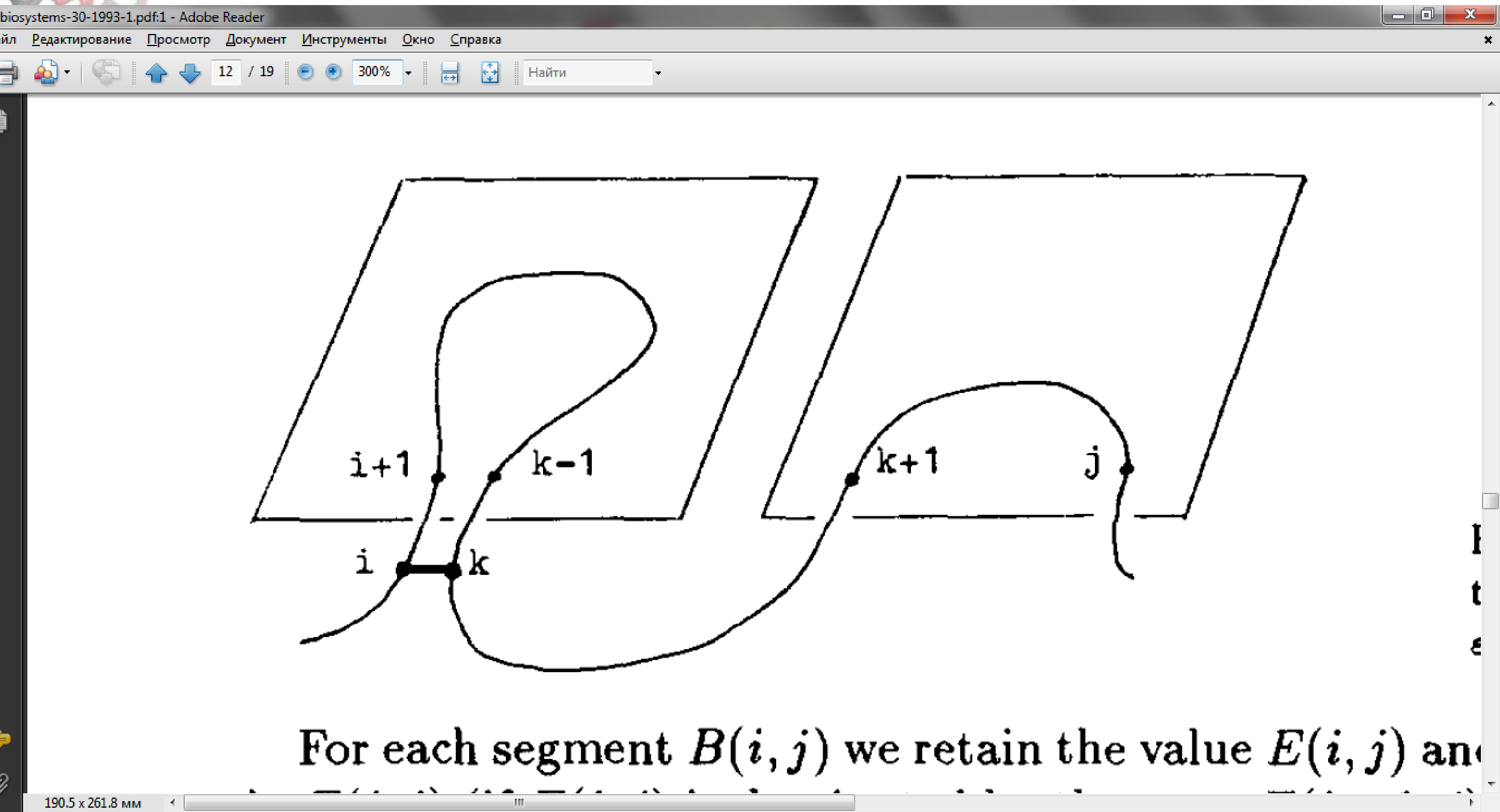
W(P) = 0, если длина P равна 0 или 1

ЗАДАЧА: Найти оптимальную структуру для данной строки

$W(i, j) = \max\{$«склейка» (i, k) допустима $|$ $1 + W(i+1, k-1) + W(k+1, j)\}$



$$W(i, j) = \max\{W(i+1, j),$$
$$\max\{k: «склейка» (i, k) \text{ допустима} \mid 1 + W(i+1, k-1) + W(k+1, j)\}\}$$



Пример: РНК и гиперпуть