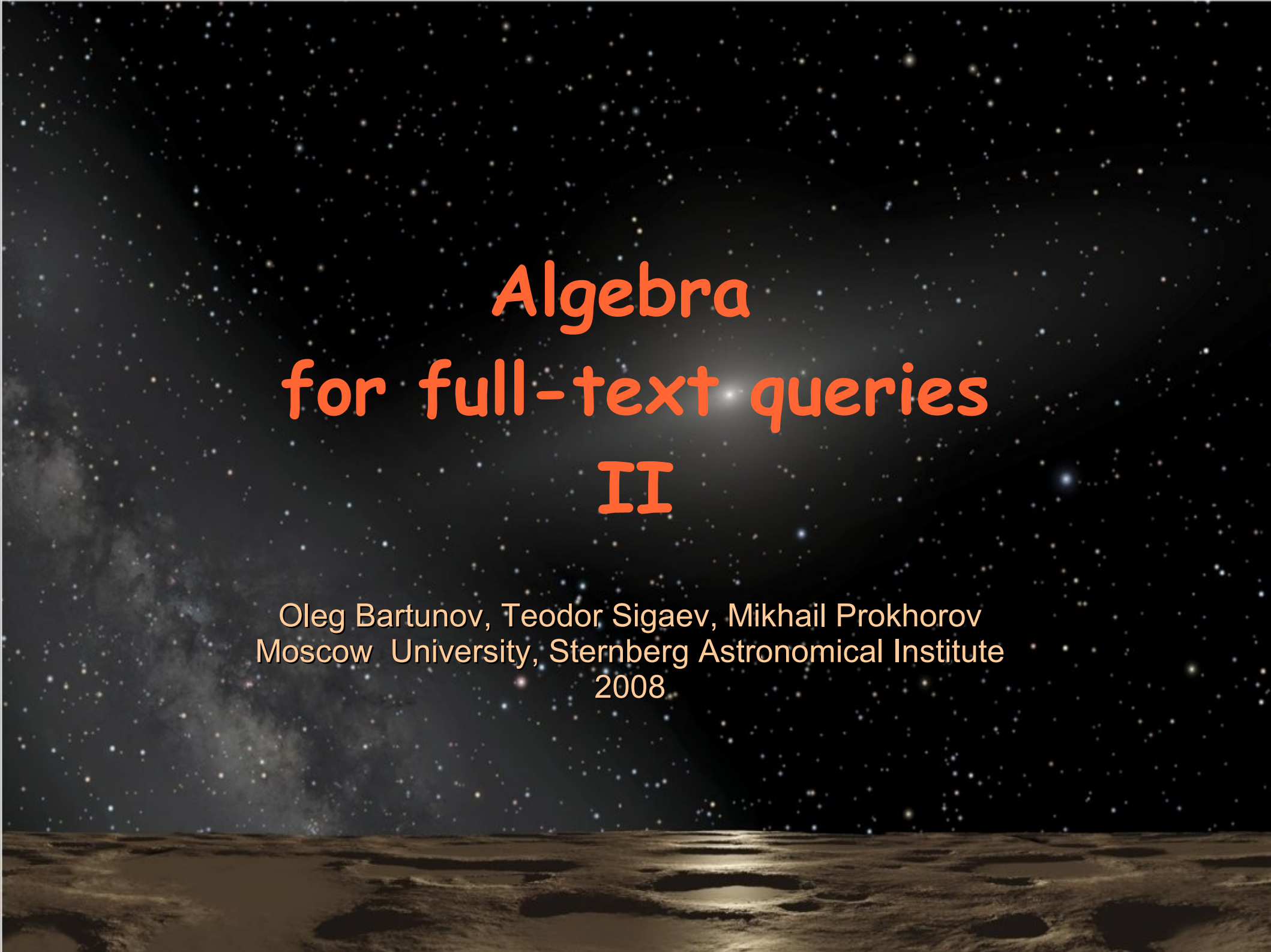

Algebra
for full-text queries
II
Oleg Bartunov, Teodor Sigaev, Mikhail Prokhorov
Moscow University, Sternberg Astronomical Institute
2008

Операция $: *BEFORE* (*AFTER*)

- Поиск фразы из 2 слов

  a $[n] b == ∃ i,j : $pos(b)_i - pos(a)_j = n$

- Операторr BEFORE ($n) ищет фразу
  - *с заданным порядком слов* (*операндов*) — a BEFORE b (или b AFTER a)
  - *с заданным расстоянием* n (default is 1)

Обобщения: операции $[m,n] и т.п.

- a $[m,n] b == ∃ i,j: $m \le pos(b)_i - pos(a)_j \le n$
- a $[<n] b == ∃ i,j: $0 < pos(b)_i - pos(a)_j \le n$
- a $[>n] b == ∃ i,j: $pos(b)_i - pos(a)_j > n$
- и т.п.

Действия с $

- a $[n] b = b $[-n] a
- !(a $[n] b) = $\forall$ i,j : $pos(b)_i - pos(a)_j$!= n
- !!(a $[n] b) = a $[n] b
- a $!b = a & ($\exists$ i,j : $pos(!b)_i - pos(a)_j = 1$)
- !a $ b = b & ($\exists$ i,j : $pos(b)_i - pos(!a)_j = 1$)
- !a $!b = ($\exists$ i,j : $pos(!b)_i - pos(!a)_j = 1$)
- a $ (b | c) = a $ b | a $ c
- (b | c) $ a = b $ a | c $ a
- a $ (b & c) = b & c & (a $ b | a $ c);
- (b & c) $ a = b & c & (b $ a | c $ a)

Действия общего вида

- a $[n] b – в тексте присутствует b через n позиций после a (или a за n позиций перед b)
- a $[n] b = b $[-n] a – n позиций перед то же самое, что на -n позиций за (после)
- !(a $[n] b) – в тексте нет b через n позиций после a
- !!(a $[n] b) = a $[n] b – двойное отрицание $ дает исходный результат

Операции с отрицанием

- a $!b – в тексте есть a за которым идет не b
- !a $ b – в тексте есть b перед которым стоит не a
- !a $!b – в тексте есть последовательная пара слов первое из которых не a, а второе не b

Сочетание $ с AND и OR

- a $ (b | c) – в тексте есть a за которым идет b или c
- (b | c) $ a – в тексте есть b или c за которым идет a
- a $ (b & c) – в тексте есть a за которым идет b И c (см. пояснение ниже)
- (b & c) $ a – в тексте есть b И c за которым идет a (см. пояснение ниже)

Ускорение поиска

- Рассмотрим определение основной операции

  a $[n] b == ∃ i,j : $pos(b)_i - pos(a)_j = n$

- Для ускорения выполнения запроса усложним выражение

  a $[n] b == a & b & (∃ i,j : $pos(b)_i - pos(a)_j = n$)

- Смысл усложнения: если в тексте нет одного из аргументов (a или b), то фразы быть не может и проверка позиций не нужна

Отрицание $

- Рассмотрим отрицание основной операции

  $!(a \$[n] b) == \forall\ i,j : pos(b)_i - pos(a)_j != n$

- Для ускорения запроса усложним выражение

  $a \$[n] b == !a \mid !b \mid (\forall\ i,j : pos(b)_i - pos(a)_j != n)$

- Смысл усложнения: если в тексте нет одного из аргументов (a или b), то запрос верен и провека списков позиций не требуется

Сочетание $ и &

- Рассмотрим сочетание операций $ и &: **a $ (b & c)**
- Возможны две интерпретации:
  - a $ (b & c) = (a $ b) & (a $ c) – в тексте должны присутствовать обе пары a $ b и a $ c
  - a $ (b & c) = b & c & (a $ b | a $ c) – в тексте должны присутствовать оба аргумента b и c и один из них должен следовать за a: (a $ b | a $ c)

- Пример с заменой синонимов

'some' $ 'telefonsvarer' => 'some' $ ('telefon' & 'svar')

служит аргументом в пользу 2-го варианта

Запросы с отрицаниями

Вырождение запросов с отрицаниями

- a $!b – . . . ab . . . ab . . . aab . . . ab . . .
- a $[2] !b – ... abb ... acb ... aabb ... azb ...
- a $[<3] !b – ... abb ... acb ... aabb ... azb ...
- !a $ b – . . . ab . . . ab . . . aab . . . ab . . .
- !a $!b – . . . ab . . . ab . . . ax . . . xb . . . xy . . .

НО данному выражению не удовлетворяет только текст

n ≥0 раз -> aaa...aaabbb...bbb <- m≥0 раз

Регулярные выражения и $

- a [xyz] b <=> a $ (x | y | z) $ b
- a . b <=> a $[2] b
- a . . . b <=> a $[4] b
- a .* b <=> a $[1,∞] b
- a [xy]{1,2} b <=> a $ (x | y | x$x | x$y | y$x | y$y) $ b

Фразы (>2 слов)

Фраза задается:

1) a ... b ... c ... b ... d ... . x - порядком слов

2) $ $[4] $[<3] $[3,7] $[n] ... $[m] - интервалами

В тексте положение фразы задается позициями:

posL – самое левое слово

posR – самое правое слово

*) Задание posL и posR может определить несколько фраз

**) для $ и $[n] posR(A)-posL(A) = Σn_i

***) Для фразы из одного слова “a”: posL(a)=posR(a)=pos(a)

Обощение $ на фразы

Пусть даны фразы

A = (a $ b $ c ... y $ z),

B = (a1 $ ... $ z1),

тогда

A $[n] B == ∃ i,j : $\text{posL}(B)_i - \text{posR}(A)_j = n$

Поиск фраз в тексте

Определение через поиск слов

$$a\ \$[n_a]\ b\ \$[n_b]\ c\ \ldots\ x\ \$[n_x]\ y\ \$[n_y]\ z ==$$

$$\exists\ i_a, i_b, \ldots, i_y, i_z: \quad pos(b)i_b - pos(a)i_a = n_a\ \&$$
$$pos(c)i_c - pos(b)i_b = n_b\ \&$$
$$\ldots$$
$$pos(y)i_y - pos(x)i_x = n_x\ \&$$
$$pos(z)i_z - pos(y)i_y = n_y$$

Поиск фраз в тексте

Определение через поиск фраз

$A == (a1\ \$[n_a]\ b1 \ldots y1\ \$[n_y]\ z1) ==$

$\exists i_a,i_b,\ldots,i_y,i_z: pos(b1)i_b - pos(a1)i_a = n_a\ \&\ \ldots\ \&\ pos(z1)i_z - pos(y1)i_y = n_y$

$B == (a2\ \$[m_a]\ b2 \ldots y2\ \$[m_y]\ z2) ==$

$\exists j_a,j_b,\ldots,j_y,j_z: pos(b2)j_b - pos(a2)j_a = m_a\ \&\ \ldots\ \&\ pos(z2)j_z - pos(y2)j_y = m_y$

$A\ \$[k]\ B == \exists\ i,j : posL(B)_i - posR(A)_j = k \Rightarrow$

$posR(A) = pos(z1)\ \&\ posL(B) = pos(a2) \Rightarrow$

$\exists\ i_a,\ldots,i_z,j_a,\ldots,j_z:$

$pos(b1)i_b - pos(a1)i_a = n_a\ \&\ \ldots\ \&\ pos(z1)i_z - pos(y1)i_y = n_y$

$\&\ pos(a2)j_a - pos(z1)i_z = k\ \&$

Поиск фраз в тексте

Вывод:

Разбиение фразы на произвольные части

не меняет результата.

Это позволяет организовывать поиск рекурсивно,

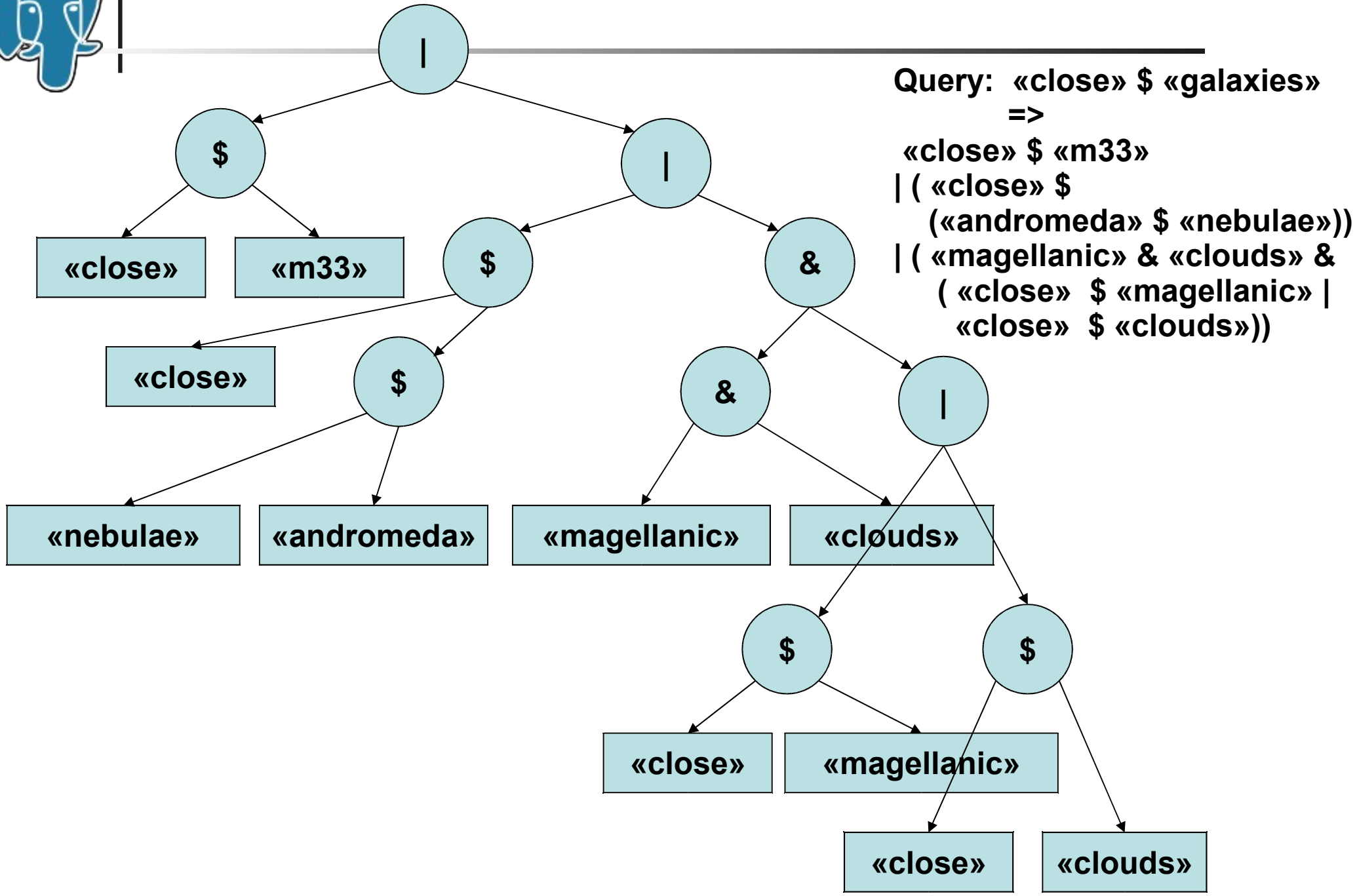
наиболее удобным образом.

Частный случай (2 слова+1)

$$(a \$[n]\ b)\ \$[m]\ c = (a \$[n]\ b)\ \&\ c\ \&$$
$$(\exists i,j: posL(c)_j - posR(\text{“}ab\text{”})_i = m) \Rightarrow$$
$$(a \$[n]\ b)\ \&\ c\ \&\ (\exists i,j: pos(c)_j - posR(\text{“}ab\text{”})_i = m) \Rightarrow$$
$$(a\ \&\ b\ \&\ (\exists k,l: pos(b)_k - pos(a)_l = n))\ \&$$
$$c\ \&\ (\exists i,j: pos(c)_j - posR(\text{“}ab\text{”})_i = m) =$$
$$= a\ \&\ b\ \&\ c\ \&$$
$$(\exists k,l: pos(b)_k - pos(a)_l = n)\ \&\ (\exists j: pos(c)_j - pos(b)_k = m) =$$
$$= a\ \&\ b\ \&\ c\ \&\ (\exists j,k,l: pos(b)_k - pos(a)_l = n\ \&$$
$$pos(c)_j - pos(b)_k = m) =$$
$$= a \$[n]\ b \$[m]\ c$$

Аналогично: $a \$[n]\ (b \$[m]\ c) = a \$[n]\ b \$[m]\ c$

Оптимизация поиска



Query: «close» $ «galaxies»
=>
«close» $ «m33»
| («close» $
(«andromeda» $ «nebulae»))
| («magellanic» & «clouds» &
(«close» $ «magellanic» |
«close» $ «clouds»))

Поглощение выражений

- a $ b ⊂ a $[n] b (n≥1)
- a $[m] b ⊂ a $[n] b (n≥m)
- a $[m1,n1] b ⊂ a $[m,n] b (m≤m1 & n1≤n)
- b $ c ⊂ a $ b $ c $ d

Если правое выражение TRUE => левое тоже TRUE

Если левое выражение известно – ускоряется вычисление правого

Оптимальный порядок поиска фраз

Вариант 1: по структуре фразы

```
  4?        1          2          1        4?
a $ b $ c $ d $ e $ f $ g $ c $ d $ h $ i $ ... $ x $ y $ z
          |<---------- 3 ---------->|
```

Вариант 2: по частоте слов ([1] – редкое, [n] – частое)

```
          [1]  [m]                                  [2] [n]
a $ b $ c $ d $ e $ f $ g $ h $ ... $ x $ y $ x
      5?   1   4?                              3?  2
```