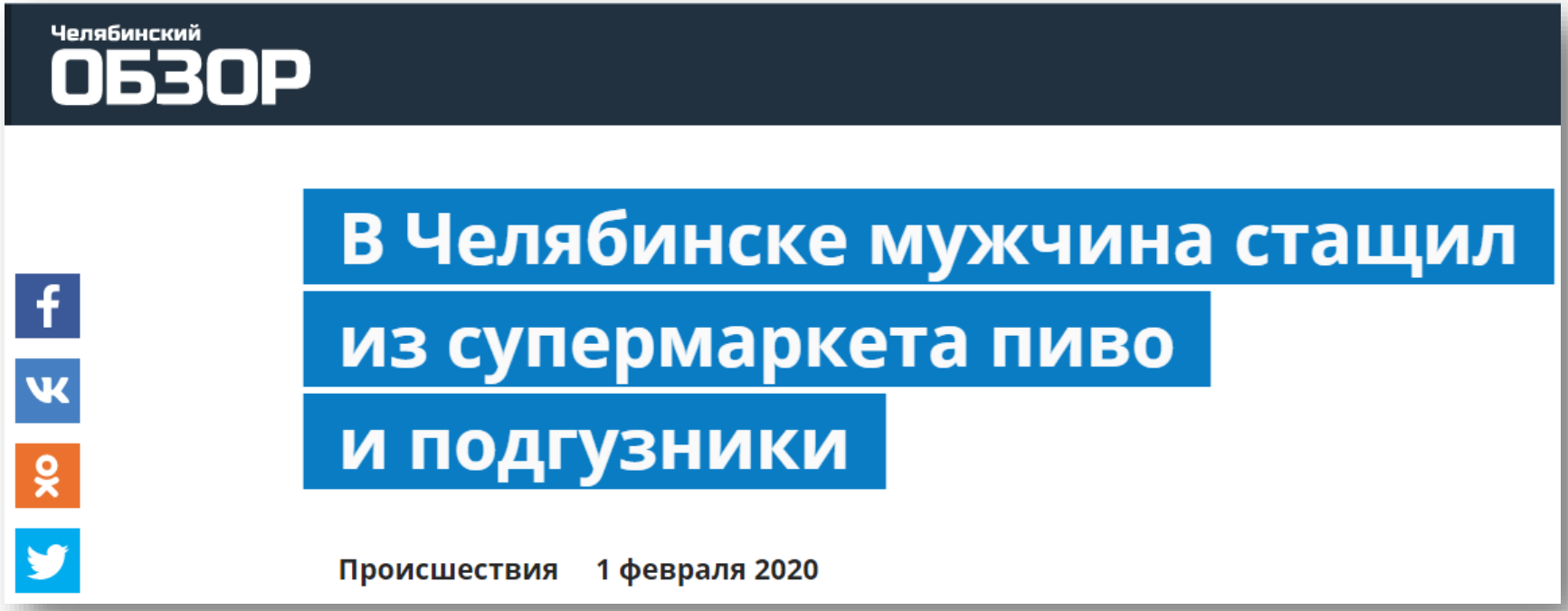
Задача поиска шаблонов в данных

Мыслить по шаблону –
вернейший способ завалить дело.

Дж.Ф. Энрайт


Челябинский ОБЗОР

В Челябинске мужчина стащил из супермаркета пиво и подгузники

Происшествия 1 февраля 2020

https://obzor174.ru/v-chelyabinske-muzhchina-stashchil-iz-supermarketa-pivo-i-podguzniki

Содержание

- **Постановка задачи**
  - **Частые наборы**
  - **Ассоциативные правила**
- **Основные алгоритмы поиска частых наборов**
  - **Apriori**
  - **Eclat**
  - **FP-Growth**
- Меры полезности шаблонов:
  - support
  - confidence
  - lift и др.
- Поиск шаблонов в Больших данных
  - Компактное представление частых наборов
  - Иерархии наборов
  - Фрагментация и сэмплинг для поиска частых наборов

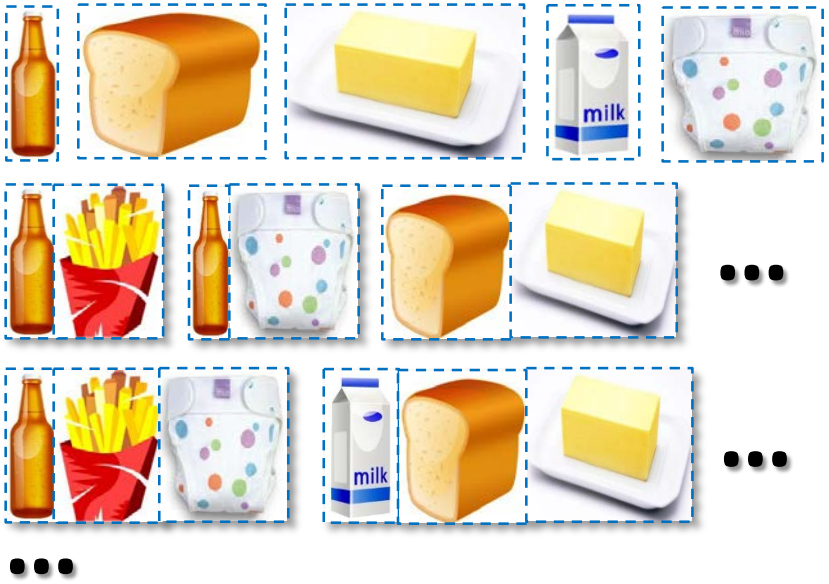
Задача анализа рыночной корзины

- Какие наборы товаров в супермаркете часто покупают совместно?

Частый набор *vs.* шаблон

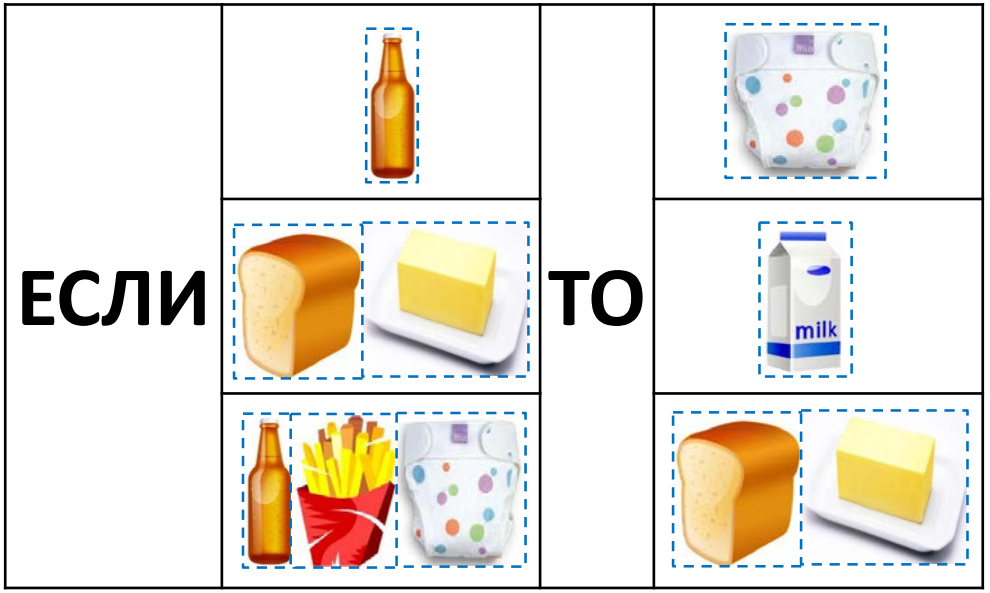
Частые k-наборы

$1 \le k \le k_{max}$



Шаблоны (ассоциативные правила)

антецедент $\rightarrow$ *консеквент*



А при чем здесь  и  ?

- В 1992 г. в США был проведен анализ 1.2 млн. рыночных корзин в 25 магазинах формата «у дома» компании Osco, который выявил частый набор  для покупок в рабочие дни с 5 до 7 час. вечера
- Руководство Osco не стало ставить эти товары рядом на полках, поскольку им были неясны причины такого частого набора
- *Объяснение*: молодая семья приходит с работы домой, жена отправляет мужа в ближайший магазин купить для ребенка  , а муж дополнительно покупает себе 

Объекты, наборы, транзакции

- Объекты (items):

$$\mathcal{I} = \{i_1, \dots, i_m\}$$

- Набор (itemset):

$$I \subseteq \mathcal{I},\ I \neq \emptyset;\ k\text{-набор: } I \subseteq \mathcal{I}, |I| = k$$

- Транзакции (transaction database):

$$\mathcal{D} = \{(tid; I) \mid I \subseteq \mathcal{I}\}$$

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Cola, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Cola, Diaper, Milk

$\mathcal{I} = \{\text{Beer, Bread, Cola, Diaper, Milk}\}$

Шаблон, поддержка, достоверность

- Шаблон: $A \to B, A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset$
- Поддержка

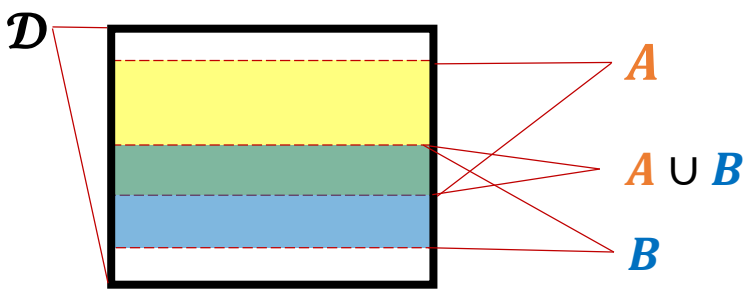
Более точно: $P(E_A \cap E_B)$

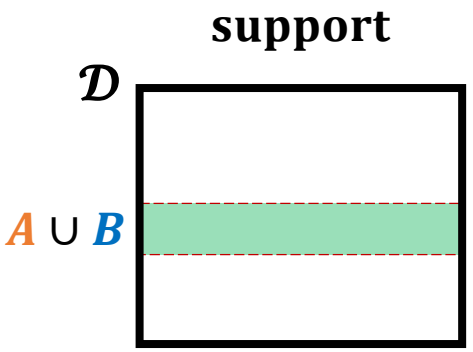
$$\sup(A \to B) = P(A \cup B) = \frac{|\{t \in \mathcal{D} \mid (\mathrm{A} \cup \mathrm{B}) \subseteq t.I\}|}{|\mathcal{D}|}$$

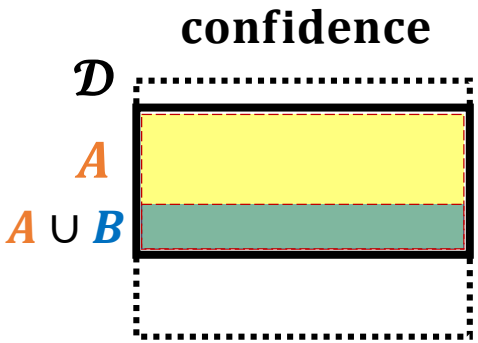
- Достоверность

Более точно: $P(E_B \mid E_A)$

$$\mathrm{conf}(A \to B) = P(B|A) = \frac{\sup(A \cup B)}{\sup(A)}$$







Поддержка и достоверность шаблона

- Поддержка:

$$\sup(A \to B) = P(A \cup B)$$

- Достоверность:

$$\text{conf}(A \to B) = P(B|A)$$

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	**Beer**, Cola, **Diaper, Milk**
4	**Beer**, Bread, **Diaper, Milk**
5	Bread, Cola, **Diaper, Milk**

$$\{\text{Diaper}, \text{Milk}\} \to \text{Beer}$$

$$sup = \frac{2}{5} = 0.4, \qquad \text{conf} = \frac{2}{3} = 0.67$$

Частый набор

- $minsup$ – порог поддержки (параметр)
- $I \subseteq \mathcal{I}$ – частый $\Leftrightarrow \sup(I) \geq minsup$
- Множество всех частых наборов: $\mathcal{L} = \bigcup_{k=1}^{k_{max}} \mathcal{L}_k$,
  $\mathcal{L}_k$ – множество частых k-наборов

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Cola, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Cola, Diaper, Milk

$$\boldsymbol{minsup = 0.6}$$

$$\mathcal{L}_1 = \{\text{Beer, Bread, Diaper, Milk}\}$$

$$\mathcal{L}_2 = \left\{\begin{matrix}\{\text{Beer, Diaper}\}, \{\text{Bread, Diaper}\}, \\ \{\text{Diaper, Milk}\}\end{matrix}\right\}$$

$$\boldsymbol{minsup = 0.1}$$

$$\mathcal{L}_3 = \left\{\begin{matrix}\{\text{Beer, Bread, Diaper}\}, \\ \{\text{Beer, Diaper, Milk}\}, \\ \{\text{Bread, Diaper, Milk}\}\end{matrix}\right\}$$

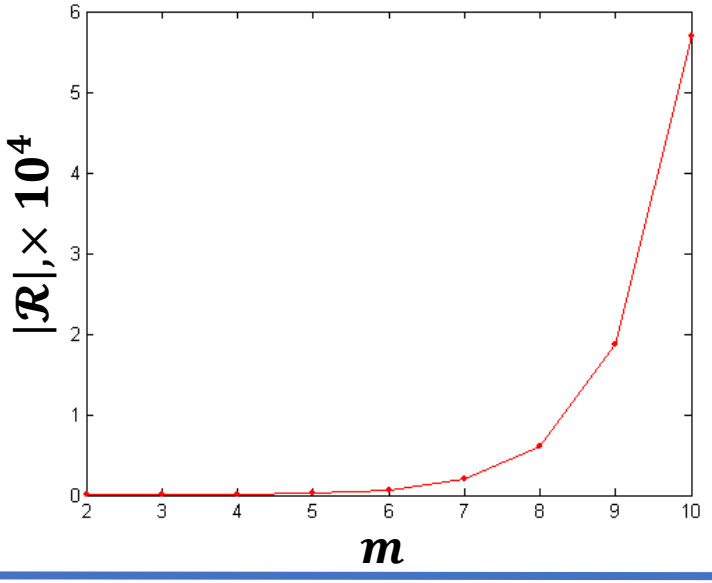
Устойчивый шаблон

- $minconf$ – порог достоверности (параметр)
- Шаблон $A \to B$ устойчив, если
  $\sup(A \to B) \geq minsup \wedge \text{conf}(A \to B) \geq minconf$
- **Постановка задачи поиска шаблонов**
  - Дано:
    $\mathcal{I} = \{i_1, \dots, i_m\}, \mathcal{D} = \{t_1, \dots, t_n\}, minsup, minconf$
  - Найти:
    $$\mathcal{R} = \{A \to B \mid A, B \subseteq \mathcal{I}, A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset, \sup(A \to B) \geq minsup, \text{conf}(A \to B) \geq minconf\}$$

Поиск шаблонов: полный перебор

1. Сгенерируем $\mathcal{R} = \{A \to B \mid A, B \subseteq \mathcal{I}, A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset\}$
2. $\forall r \in \mathcal{R}$ вычислим $\sup(r)$, $\text{conf}(r)$
3. Отбросим $\forall r \in \mathcal{R}$: $\sup(r) < minsup$, $\text{conf}(r) < minconf$

- $|\mathcal{R}| = \sum_{k=1}^{m-1}\left[C_k^m \cdot \sum_{i=1}^{m-k} C_i^{m-k}\right] = 3^m - 2^{m+1} + 1$



m	$\lvert\mathcal{R}\rvert$
6	602
10	57 002

Поиск устойчивых шаблонов полным перебором вычислительно невозможен

Поиск шаблонов: идеи

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Cola, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Cola, Diaper, Milk

$\{Milk,Diaper\} \rightarrow \{Beer\}$ (s=0.4, c=0.67)
$\{Milk,Beer\} \rightarrow \{Diaper\}$ (s=0.4, c=1.0)
$\{Diaper,Beer\} \rightarrow \{Milk\}$ (s=0.4, c=0.67)
$\{Beer\} \rightarrow \{Milk,Diaper\}$ (s=0.4, c=0.67)
$\{Diaper\} \rightarrow \{Milk,Beer\}$ (s=0.4, c=0.5)
$\{Milk\} \rightarrow \{Diaper,Beer\}$ (s=0.4, c=0.5)

- Шаблоны примера – разбиения одного набора
- Разбиения одного набора имеют равную поддержку, но могут иметь разную достоверность
- Для поиска шаблонов обработку поддержки и достоверности можно отделить друг от друга

Поиск шаблонов: алгоритм

1. Найдем все частые наборы (с поддержкой не ниже $minsup$)
2. Сгенерируем шаблоны, выполняя разбиение каждого частого набора; устойчивыми будут шаблоны с достоверностью не ниже $minconf$

Поиск частых наборов: полный перебор

1. Сгенерируем кандидатов в частые наборы: $\mathcal{C} = \mathcal{P}(\mathcal{I}) \setminus \emptyset$
2. $\forall c \in \mathcal{C}$ вычислим $\sup(c)$
3. Отбросим $\forall c \in \mathcal{C}$: $\sup(c) < minsup$

- Сложность: $O(n \cdot 2^m \cdot w)$,
  где w – средняя длина транзакции

Поиск частых наборов полным перебором вычислительно невозможен

Отбрасывание заведомо редких наборов

- Антимонотонность поддержки

$$\forall X, Y\colon X \subseteq Y \Leftrightarrow \sup(X) \geq \sup(Y)$$

  – поддержка набора всегда не больше поддержки любого своего подмножества

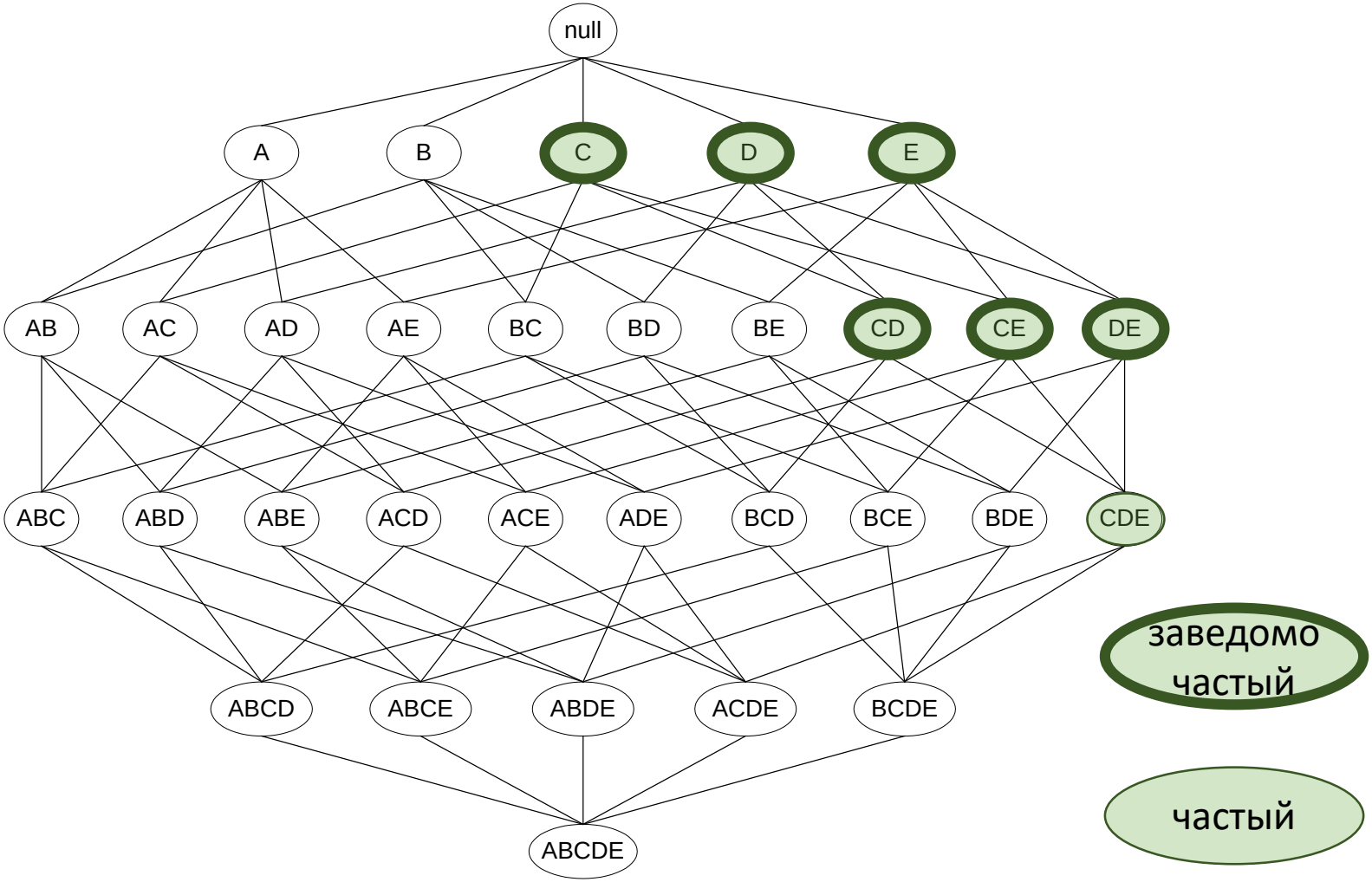
- Принцип Априори
  – любое подмножество частого набора является частым набором

$$\sup(Y) \geq minsup \Leftrightarrow \forall X \subseteq Y \sup(X) \geq minsup$$

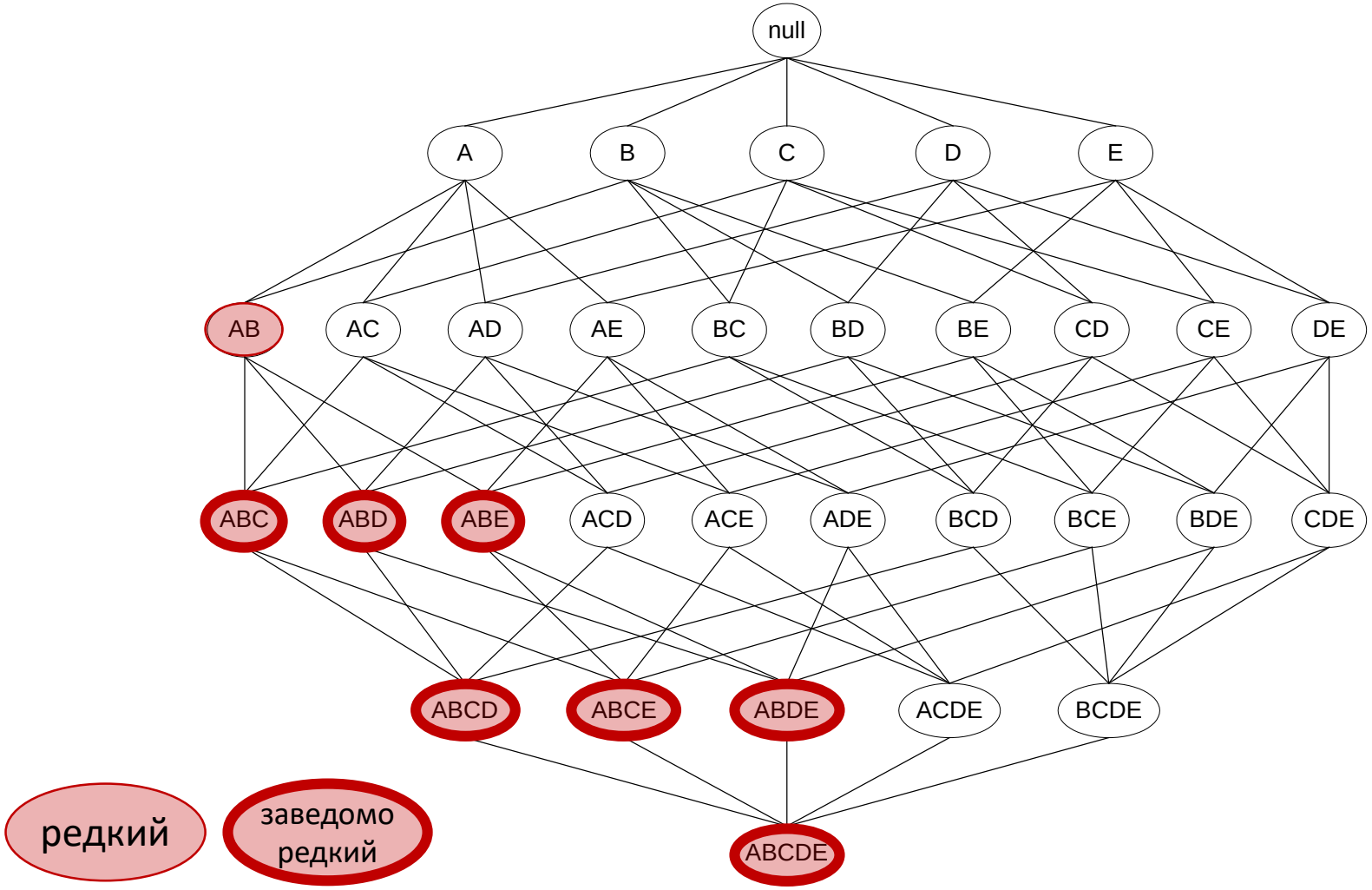
  – если некое подмножество набора является редким набором, то набор является редким

$$\sup(Y) < minsup \Leftrightarrow \exists X \subseteq Y \sup(X) < minsup$$

Принцип Априори



Принцип Априори



Генерация наборов-кандидатов

- $k = 1$: $C_1 := \mathcal{I}$ (множество 1-наборов)
- $k = 2$: $C_2 := \mathcal{L}_1 \times \mathcal{L}_1$
  - Декартово произведение множества частых 1-наборов на себя
- $k \geq 3$:
  - Соединение множества частых $(k-1)$-наборов с самим собой
    - $C_k := \mathcal{L}_{k-1} \bowtie_\Theta \mathcal{L}_{k-1}$
      $X = (x_1, \ldots, x_{k-1}),\ \ Y = (y_1, \ldots, y_{k-1}),\ X, Y \in \mathcal{L}_{k-1}$,
      элементы в наборах лексикографически упорядочены
      $$\Theta = (\wedge_{i=1}^{k-2} x_i = y_i) \wedge (x_{k-1} < y_{k-1})$$
    - $X \bowtie_\Theta Y = (x_1, \ldots, x_{k-2}, x_{k-1}, y_{k-1})$
  - Отбрасывание заведомо редких наборов из C_k без вычисления поддержки по принципу Априори
    - если $I \subseteq C_k$, но $I \notin \mathcal{L}_{k-1}$, то отбросить I

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $$C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$

  Вычисление поддержки

  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$

  Вычисление поддержки

  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$

- $\mathcal{L}_2 = \{\mathbf{A}\mathrm{B}, \mathbf{A}\mathrm{C}, \mathbf{A}\mathrm{D}, \mathbf{A}\mathrm{E}, \mathbf{B}\mathrm{C}, \mathbf{B}\mathrm{D}, \mathbf{B}\mathrm{E}, \mathbf{C}\mathrm{D}, \mathbf{C}\mathrm{E}\}$

  Самосоединение

  $C_3 = \{\mathbf{A}\mathrm{BC}, \mathbf{A}\mathrm{BD}, \mathbf{A}\mathrm{BE}, \mathbf{A}\mathrm{CD}, \mathbf{A}\mathrm{CE}, \mathbf{A}\mathrm{DE}, \mathbf{B}\mathrm{CD}, \mathbf{B}\mathrm{CE}, \mathbf{B}\mathrm{DE}, \mathbf{C}\mathrm{DE}\}$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$
  Декартово произведение
  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$
  Вычисление поддержки
  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$
- $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}\}$
  Самосоединение
  $C_3 = \{\mathrm{ABC}, \mathrm{ABD}, \mathrm{ABE}, \mathrm{ACD}, \mathrm{ACE}, \mathrm{ADE}, \mathrm{BCD}, \mathrm{BCE}, \mathrm{BDE}, \mathrm{CDE}\}$
  Вычисление поддержки
  $\mathcal{L}_3 = \{\mathrm{ABC}, \mathrm{ABD}, \mathrm{ABE}, \mathrm{ACD}, \sout{\mathrm{ACE}}, \sout{\mathrm{ADE}}, \mathrm{BCD}, \sout{\mathrm{BCE}}, \mathrm{BDE}, \mathrm{CDE}\}$
- $\mathcal{L}_3 = \{\mathbf{AB}\mathrm{C}, \mathbf{AB}\mathrm{D}, \mathbf{AB}\mathrm{E}, \mathrm{ACD}, \mathrm{BCD}, \mathrm{BDE}, \mathrm{CDE}\}$
  Самосоединение
  $C_4 = \{\mathbf{AB}\mathrm{CD}, \mathbf{AB}\mathrm{CE}, \mathbf{AB}\mathrm{DE}\}$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$

  Вычисление поддержки

  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$
- $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}\}$

  Самосоединение

  $C_3 = \{\mathrm{ABC}, \mathrm{ABD}, \mathrm{ABE}, \mathrm{ACD}, \mathrm{ACE}, \mathrm{ADE}, \mathrm{BCD}, \mathrm{BCE}, \mathrm{BDE}, \mathrm{CDE}\}$

  Вычисление поддержки

  $\mathcal{L}_3 = \{\mathrm{ABC}, \mathrm{ABD}, \mathrm{ABE}, \mathrm{ACD}, \sout{\mathrm{ACE}}, \sout{\mathrm{ADE}}, \mathrm{BCD}, \sout{\mathrm{BCE}}, \mathrm{BDE}, \mathrm{CDE}\}$
- $\mathcal{L}_3 = \{\mathbf{ABC}, \mathbf{ABD}, \mathbf{ABE}, \mathrm{ACD}, \mathrm{BCD}, \mathrm{BDE}, \mathrm{CDE}\}$

  Самосоединение

  $C_4 = \{\mathbf{ABCD}, \mathbf{ABCE}, \mathbf{ABDE}\}$

  Отбрасывание

  $C_4 = \{\mathbf{ABCD}, \mathrm{A}\mathbf{BCE}, \mathbf{A}\mathrm{B}\mathbf{DE}\}$ и $\mathbf{BCE} \notin \mathcal{L}_3$, $\mathbf{ADE} \notin \mathcal{L}_3$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$
  Декартово произведение
  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$
  Вычисление поддержки
  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$
- $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}\}$
  Самосоединение
  $C_3 = \{\mathrm{ABC}, \mathrm{ABD}, \mathrm{ABE}, \mathrm{ACD}, \mathrm{ACE}, \mathrm{ADE}, \mathrm{BCD}, \mathrm{BCE}, \mathrm{BDE}, \mathrm{CDE}\}$
- $\mathcal{L}_3 = \{\mathbf{AB}\mathrm{C}, \mathbf{AB}\mathrm{D}, \mathbf{AB}\mathrm{E}, \mathrm{ACD}, \mathrm{BCD}, \mathrm{BDE}, \mathrm{CDE}\}$
  Самосоединение
  $C_4 = \{\mathbf{AB}\mathrm{CD}, \mathbf{AB}\mathrm{CE}, \mathbf{AB}\mathrm{DE}\}$
  Отбрасывание
  $C_4 = \{\mathbf{AB}\mathrm{CD}, \mathrm{A}\mathbf{BCE}, \mathrm{A}\mathbf{BDE}\} \Rightarrow C_4 = \{\mathrm{ABCD}\}$

Алгоритм Априори

Input:

- D, a database of transactions;
- min_sup, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)    L1 = find_frequent_1-itemsets(D);
(2)    for (k = 2; Lk−1 ≠ φ; k++) {
(3)        Ck = apriori_gen(Lk−1);
(4)        for each transaction t ∈ D { // scan D for counts
(5)            Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)            for each candidate c ∈ Ct
(7)                c.count++;
(8)        }
(9)        Lk = {c ∈ Ck | c.count ≥ min_sup}
(10)   }
(11)   return L = ∪k Lk;
```

Алгоритм Априори

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)     for each itemset l_1 ∈ L_{k-1}
(2)         for each itemset l_2 ∈ L_{k-1}
(3)             if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2])
                    ∧... ∧ (l_1[k-2] = l_2[k-2]) ∧ (l_1[k-1] < l_2[k-1]) then {
(4)                 c = l_1 ⋈ l_2; // join step: generate candidates
(5)                 if has_infrequent_subset(c, L_{k-1}) then
(6)                     delete c; // prune step: remove unfruitful candidate
(7)                 else add c to C_k;
(8)             }
(9)     return C_k;

procedure has_infrequent_subset(c: candidate k-itemset;
          L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
(1)     for each (k-1)-subset s of c
(2)         if s ∉ L_{k-1} then
(3)             return TRUE;
(4)     return FALSE;
```

```
(1)   L_1 = find_frequent_1-itemsets(D);
(2)   for (k = 2; L_{k-1} ≠ φ; k++) {
(3)       C_k = apriori_gen(L_{k-1});
(4)       for each transaction t ∈ D { // scan D for counts
(5)           C_t = subset(C_k, t); // get the subsets of t that are candidates
(6)           for each candidate c ∈ C_t
(7)               c.count++;
(8)       }
(9)       L_k = {c ∈ C_k | c.count ≥ min_sup}
(10)  }
(11)  return L = ∪_k L_k;
```

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$C_1 = \mathcal{I}$

Items	SUP
I1	
I2	
I3	
I4	
I5	
I6	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$C_1 = \mathcal{I}$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2
I6	1

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

C_1

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2
I6	**1**

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$

Items	SUP
I1,I2	
I1,I3	
I1,I4	
I1,I5	
I2,I3	
I2,I4	
I2,I5	
I3,I4	
I3,I5	
I4,I5	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$

Items	SUP
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$

Items	SUP
I1,I2	4
I1,I3	4
I1,I4	**1**
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	**0**
I3,I5	**1**
I4,I5	**0**

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

Items	SUP
I1,I2,I3	
I1,I2,I5	
I1,I3,I5	
I2,I3,I4	
I2,I3,I5	
I2,I4,I5	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

Items	SUP
I1,I2,I3	
I1,I2,I5	
I1,**I3,I5**	
I2,**I3,I4**	
I2,**I3,I5**	
I2,**I4,I5**	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

$C_4 = \mathcal{L}_3 \bowtie \mathcal{L}_3$

Items	SUP
I1,**I2,I3,I5**	

$\mathcal{L}_4 = \emptyset$

Items	SUP

Алгоритм Apriori: пример

$D, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}$

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$\cup$

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$\cup$

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

Поиск шаблонов: алгоритм

$\mathcal{L} := \cup_{k=1}^{k_{max}} \{I \subseteq \mathcal{I} \mid |I| = k, \sup(I) \geq minsup\}$

$\mathcal{R} := \emptyset$

for all $I \in \mathcal{L}$

  for all $S \in \mathcal{P}(I) \setminus \emptyset$ **do**

    if $\frac{\sup(I)}{\sup(S)} \geq minconf$ **then**

      $pattern := "S \rightarrow I \setminus S"$

      $\mathcal{R} := \mathcal{R} \cup pattern$

$\mathcal{P}(I)$ – множество всех подмножеств I

Генерация шаблонов: пример

D,
$minsup = 0.2$
$minconf = 0.7$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	**2**

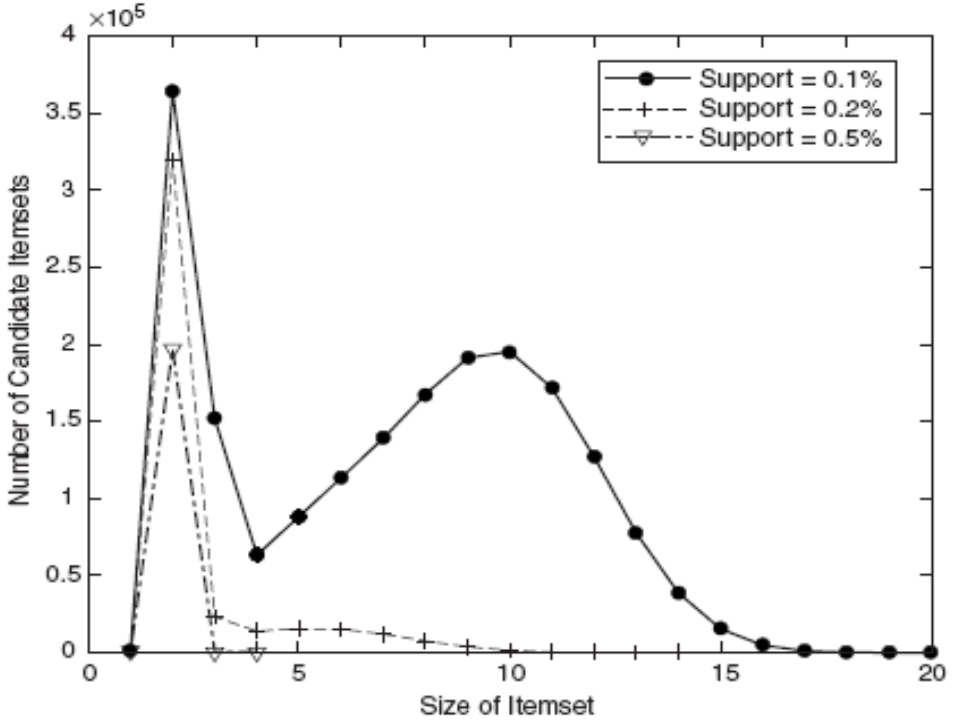
Шаблоны

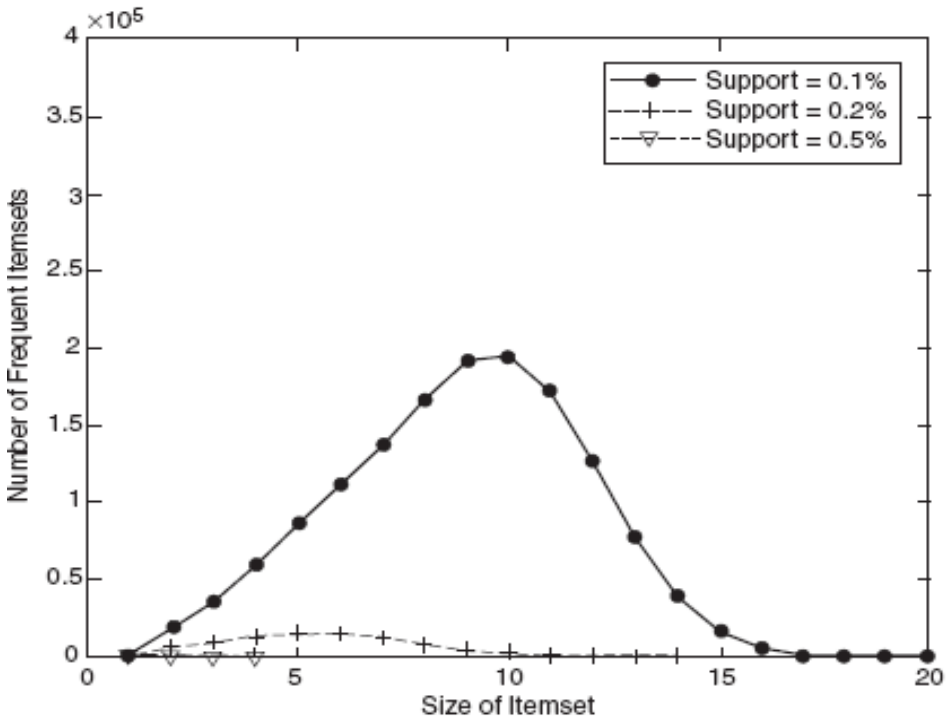
для {I1,I2,I5}

Rule	CONF
{I1,I2}→I5	2/4
{I1,I5}→I2	**2/2**
{I2,I5}→I1	**2/2**
I1→{I2,I5}	2/6
I2→{I1,I5}	2/7
I5→{I1,I2}	**2/2**

Факторы, влияющие на сложность Apriori

minsup (порог поддержки)



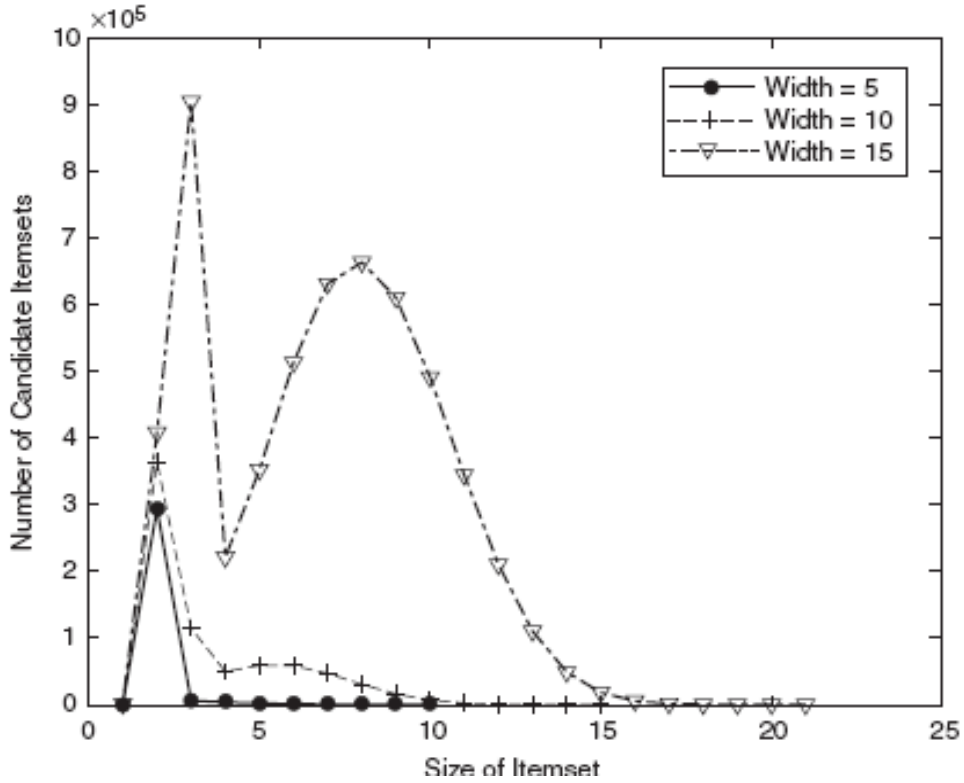


$\bigcup_k |C_k|$

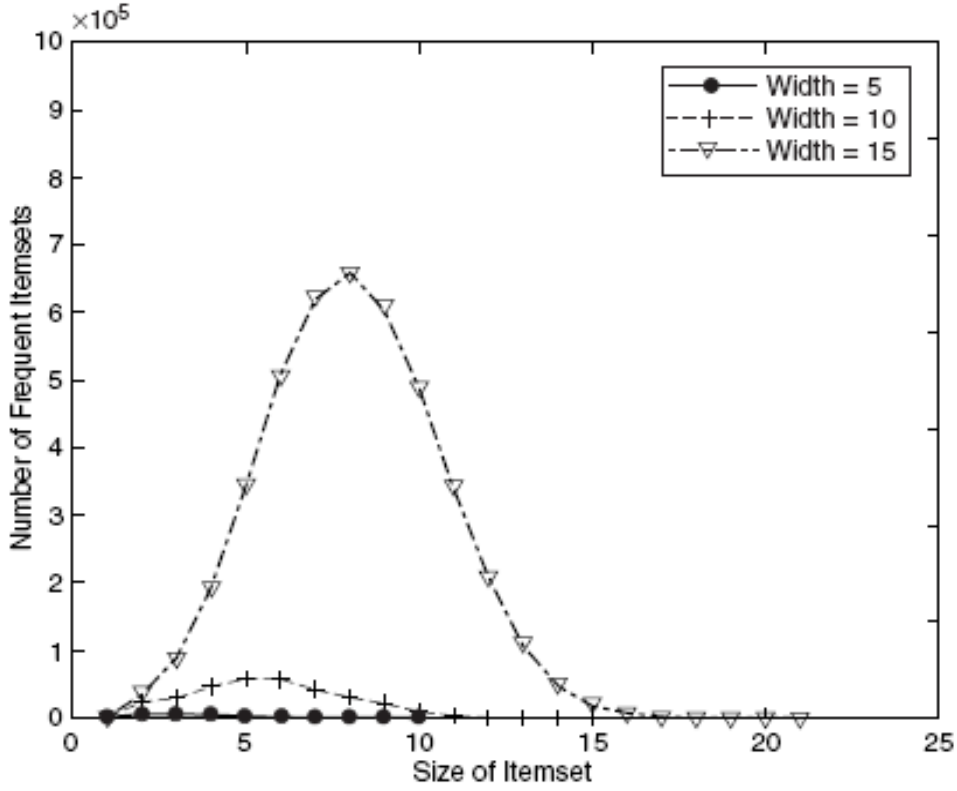
$\bigcup_k |\mathcal{L}_k|$

Факторы, влияющие на сложность Apriori

$|\bar{T}|$ (средняя длина транзакции)



$\bigcup_k |C_k|$



$\bigcup_k |\mathcal{L}_k|$

Недостатки Apriori

- Потенциально большое число генерируемых кандидатов
  - $C_1 = \mathcal{I}$
  - $C_2 = \mathcal{L}_1 \times \mathcal{L}_1 \Rightarrow |C_2| = |\mathcal{L}_1|^2$
  - $k \geq 3$: $C_k = \mathcal{L}_{k-1} \bowtie_{\Theta} \mathcal{L}_{k-1}$
- Многократное сканирование D и проверка вхождения каждого кандидата в каждую транзакцию
  - $O(|D| \cdot |C_k| \cdot |\bar{T}|)$

Алгоритм ECLAT

- Вертикальный формат данных
  - $TIDlist(A) ::= \{T \in D \mid A \subseteq T\}$
  - $\sup(A) = |TIDlist(A)|$
  - $\sup(A, B) = |TIDlist(A) \cap TIDlist(B)|$

Алгоритм ECLAT

$D, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

C_1

Item	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2
I6	**100**	**1**

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

Алгоритм ECLAT

$\mathcal{L}_1$

Item	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2

$$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$$

$$TIDlist(A, B) = TIDlist(A) \cap TIDlist(B)$$

C_2

Items	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	**40**	**1**
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	**80**	**1**

Алгоритм ECLAT

$\mathcal{L}_2$

Items	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2

$\mathcal{L}_1$

Item	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

C_3

Items
I1,I2,I3
I1,I2,I5
I1,I3,I5
I2,I3,I4
I2,I3,I5
I2,I4,I5

Apriori

$TIDlist(A, B) = TIDlist(A) \cap TIDlist(B)$

$\mathcal{L}_3$

Items	TID_list	SUP
I1,I2,I3	80, 90	2
I1,I2,I5	10, 80	2

Алгоритм FP-Growth

- Поиск частых наборов без генерации кандидатов
- Длинный частый набор строится из коротких, используя только локально частые наборы
  - Пусть abc частый набор
  - Выполнить проектирование $\mathcal{D}$ на abc: $\mathcal{D}|abc$ (получить транзакции, содержащие abc)
  - Если d – частый набор в $\mathcal{D}|abc$, то $abcd$ – частый набор

Алгоритм FP-Growth: предобработка

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

Items	SUP
I2	7
I1	6
I3	6
I4	2
I5	2

$\boldsymbol{D}$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

Редкие наборы
не рассматриваются

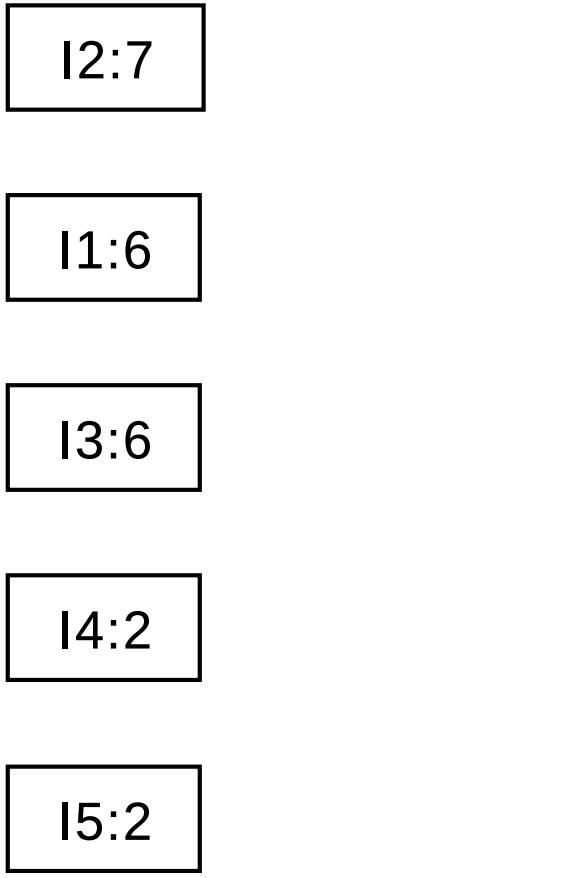
Построение FP-дерева

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

sorted $\mathcal{L}_1$

Items	SUP
I2	7
I1	6
I3	6
I4	2
I5	2



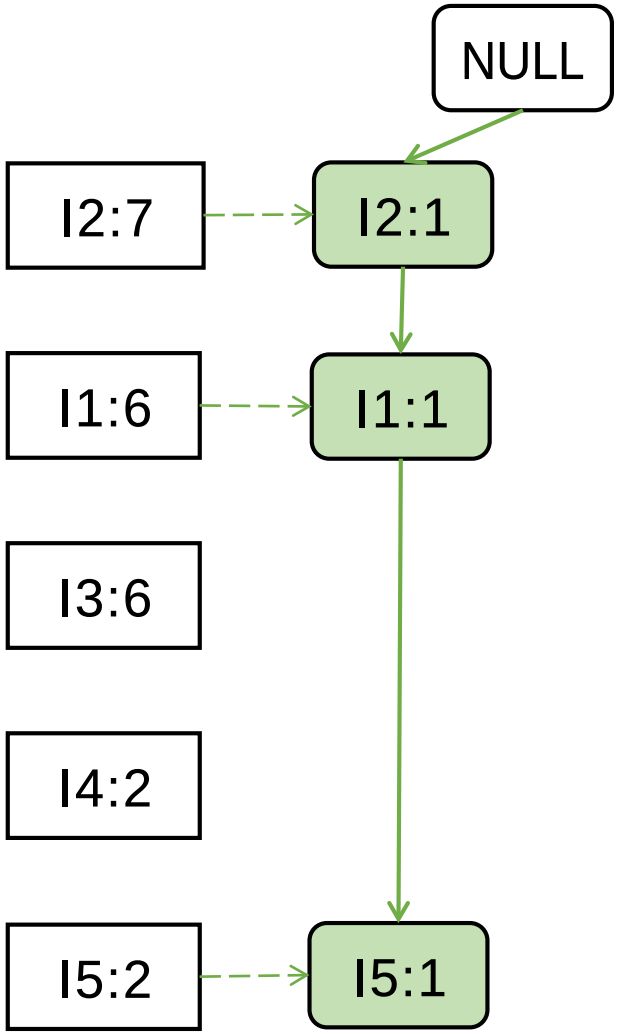
Построение FP-дерева

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

sorted $\mathcal{L}_1$

Items	SUP
I2	7
I1	6
I3	6
I4	2
I5	2

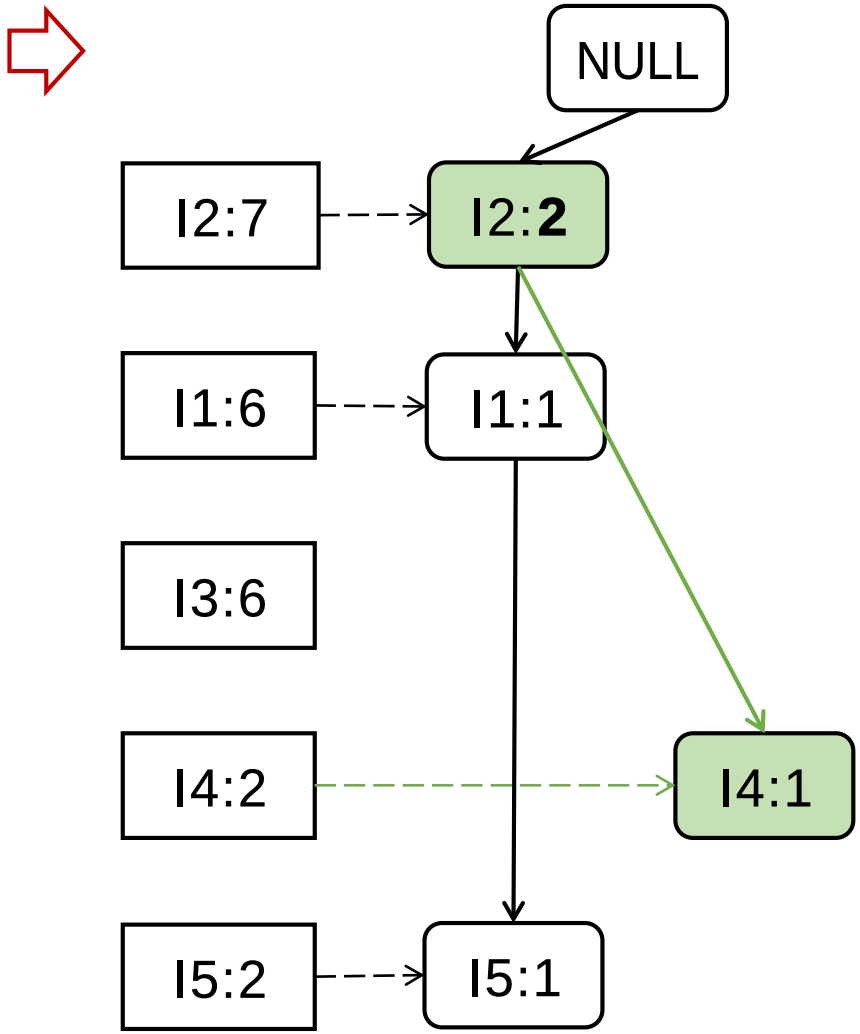


Построение FP-дерева

$\boldsymbol{D}, minsup = 0.2$ sorted $\mathcal{L}_1$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

Items	SUP
I2	7
I1	6
I3	6
I4	2
I5	2



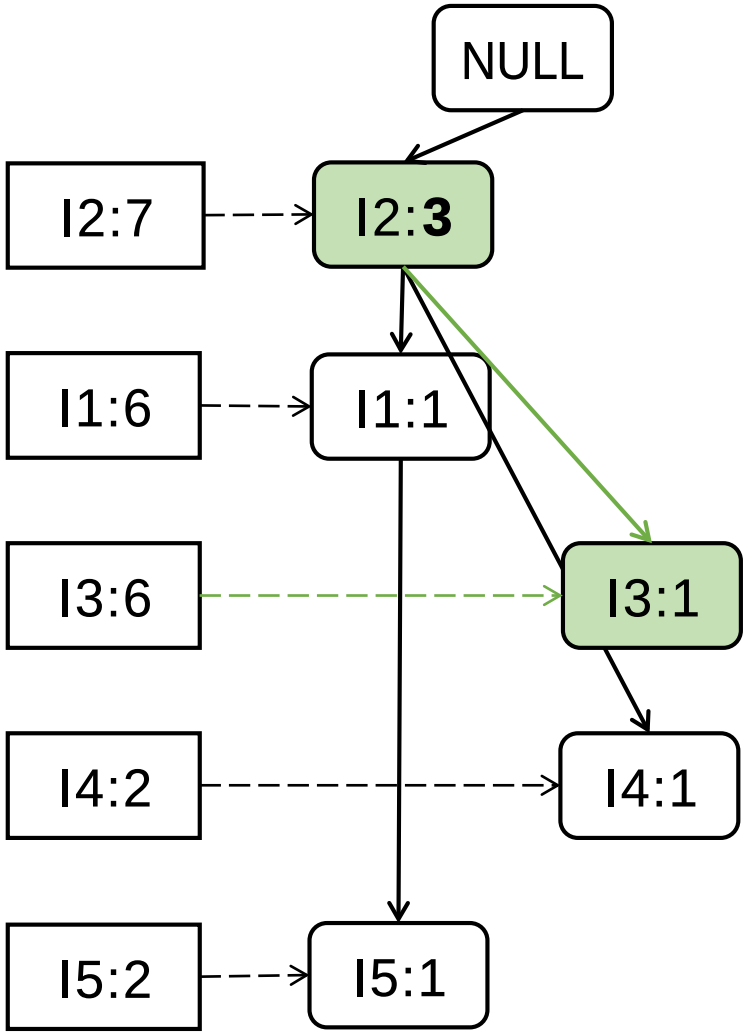
Построение FP-дерева

$D, minsup = 0.2$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

sorted $\mathcal{L}_1$

Items	SUP
I2	7
I1	6
I3	6
I4	2
I5	2



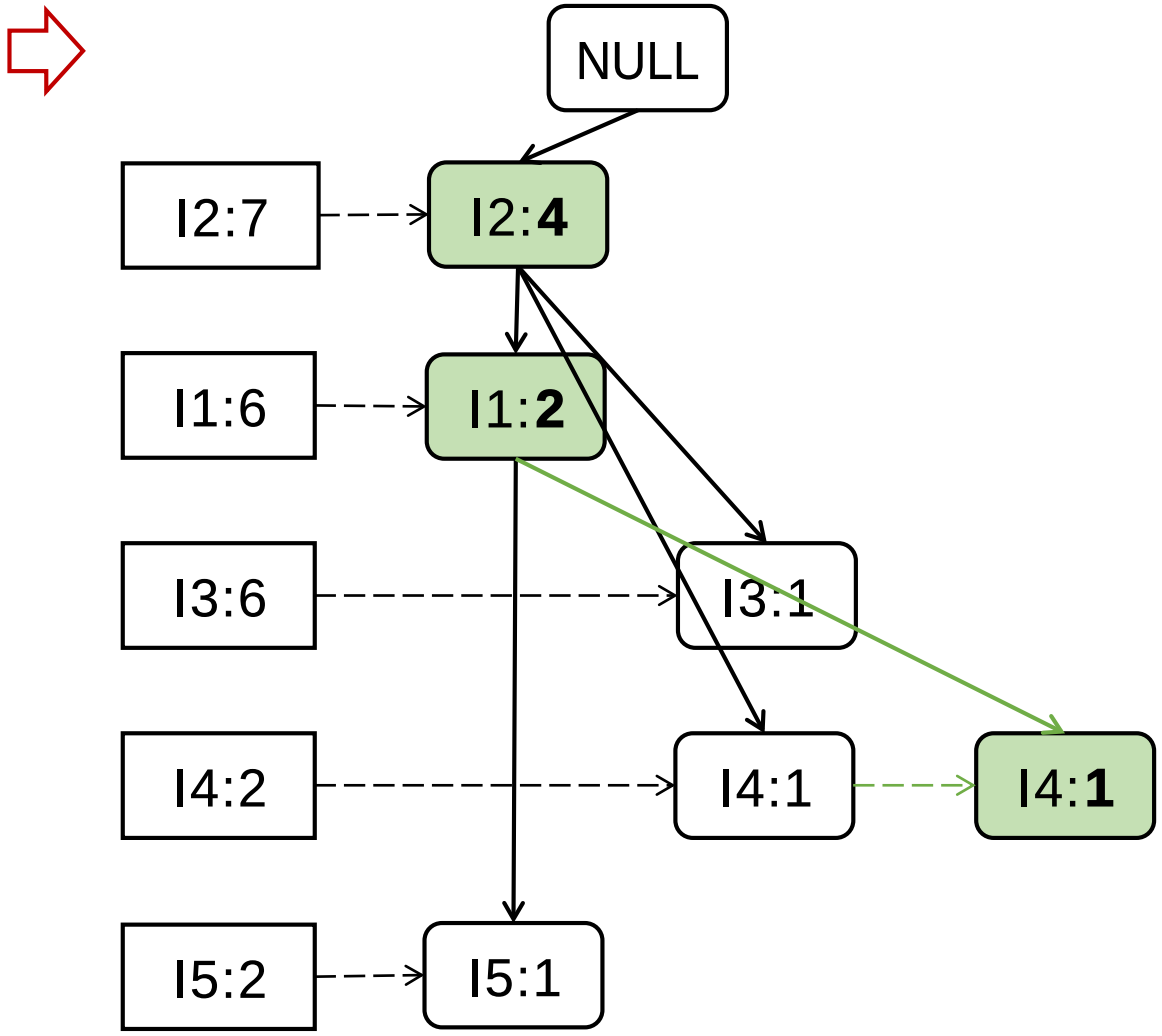
Построение FP-дерева

$D, minsup = 0.2$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

sorted $\mathcal{L}_1$

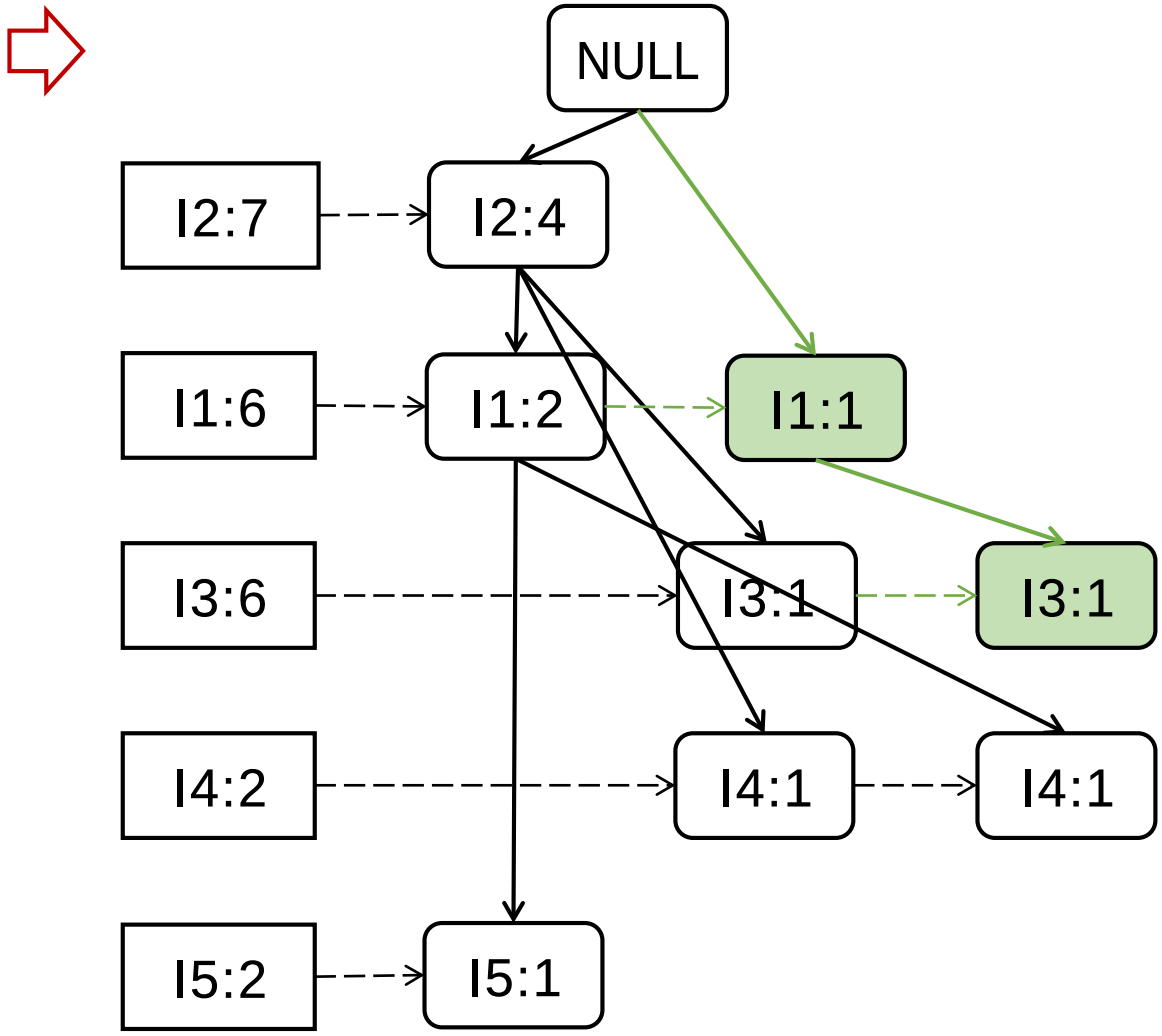
Items	SUP
I2	7
I1	6
I3	6
I4	2
I5	2



Построение FP-дерева

$\boldsymbol{D}, minsup = 0.2$

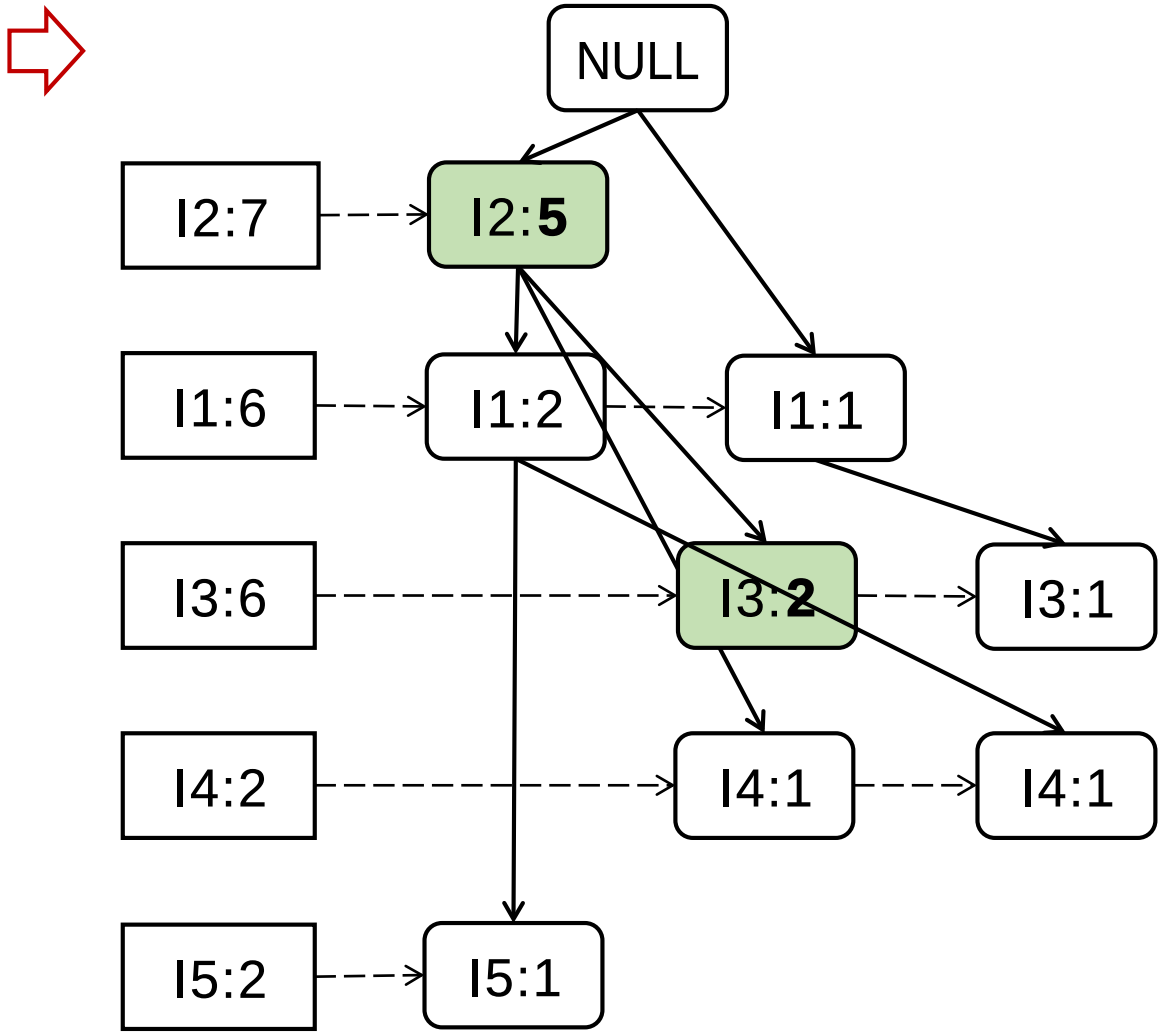
TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6



Построение FP-дерева

$\boldsymbol{D}, minsup = 0.2$

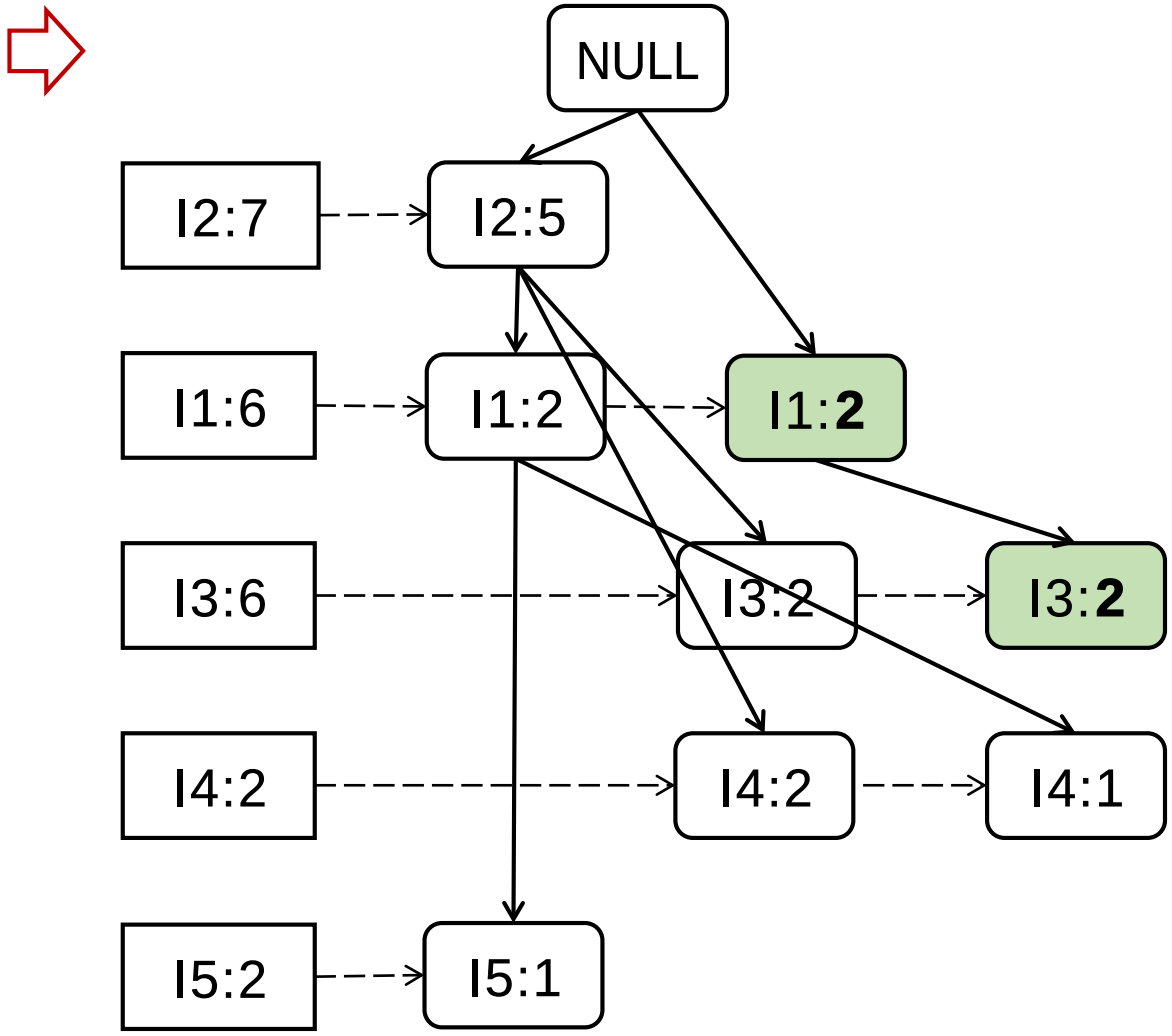
TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6



Построение FP-дерева

$\boldsymbol{D}, minsup = 0.2$

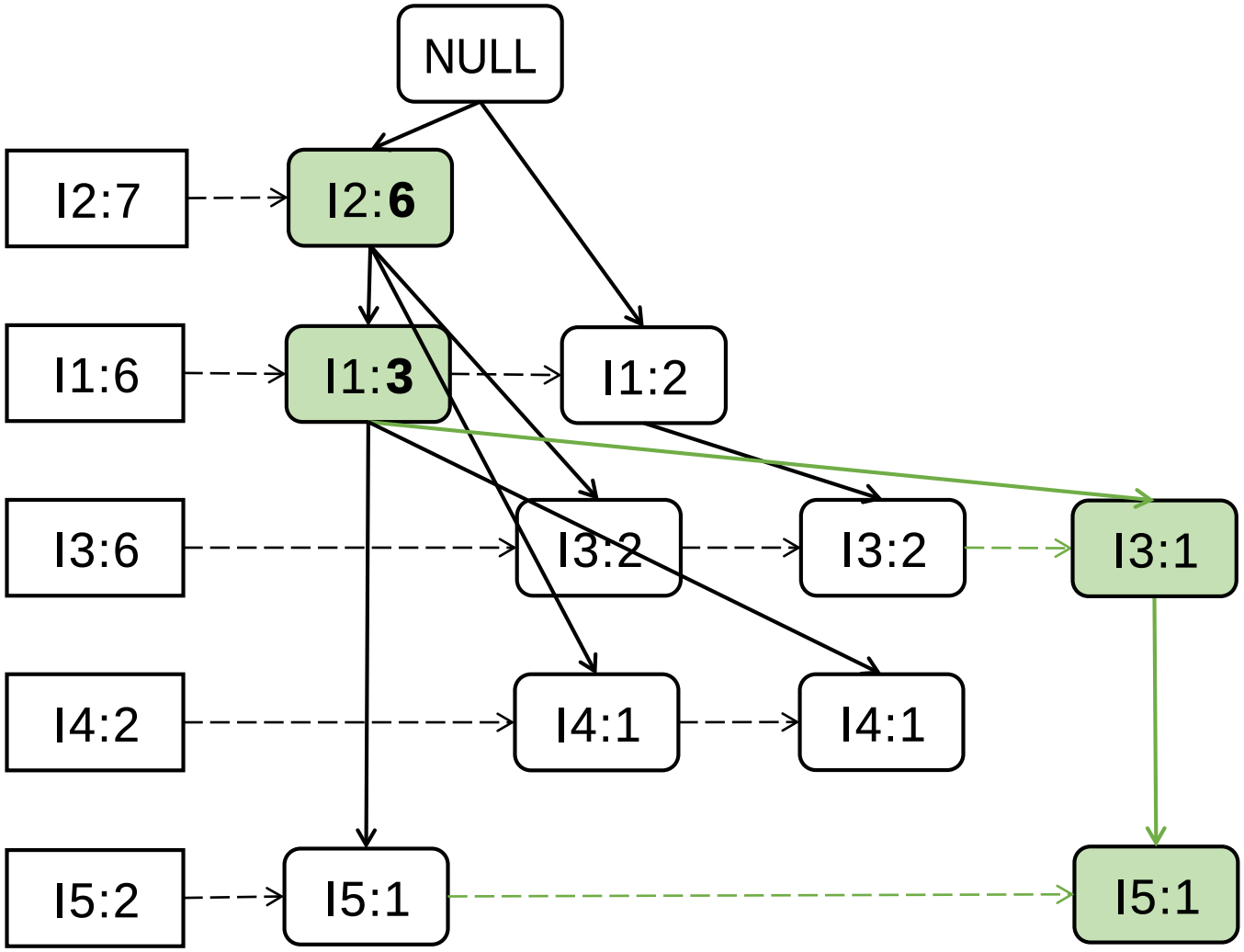
TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6



Построение FP-дерева

$D, minsup = 0.2$

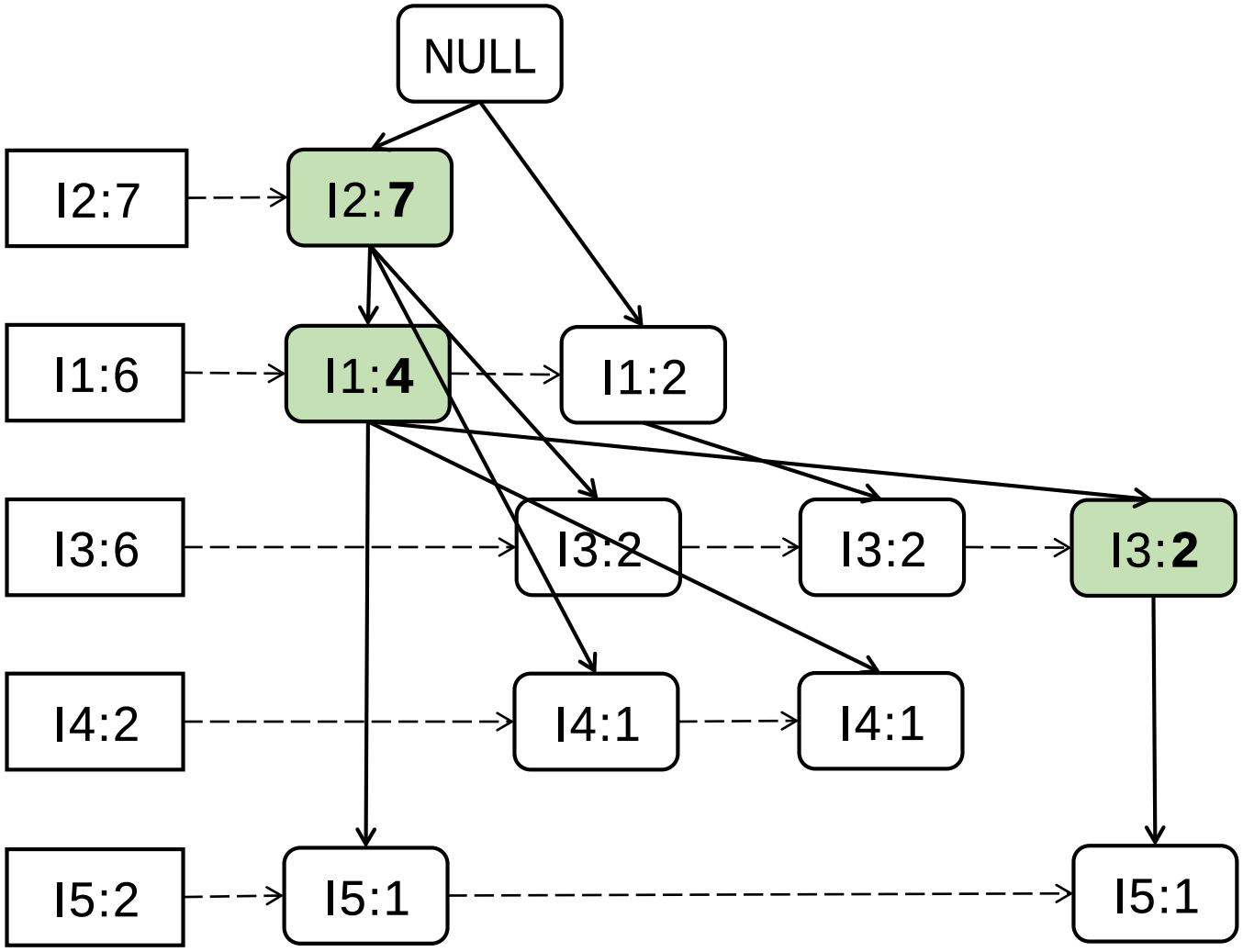
TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6



Построение FP-дерева

$D, minsup = 0.2$

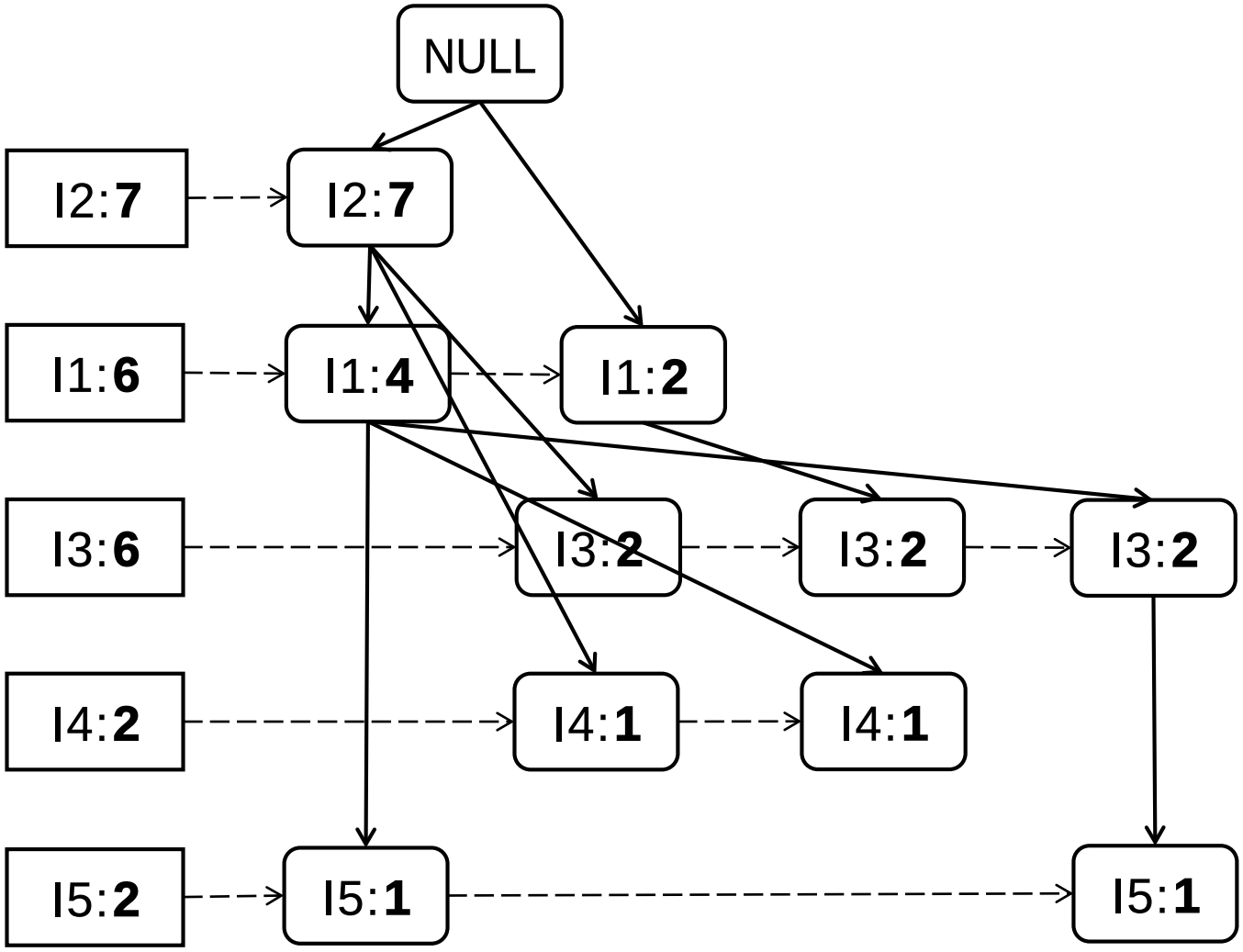
TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6



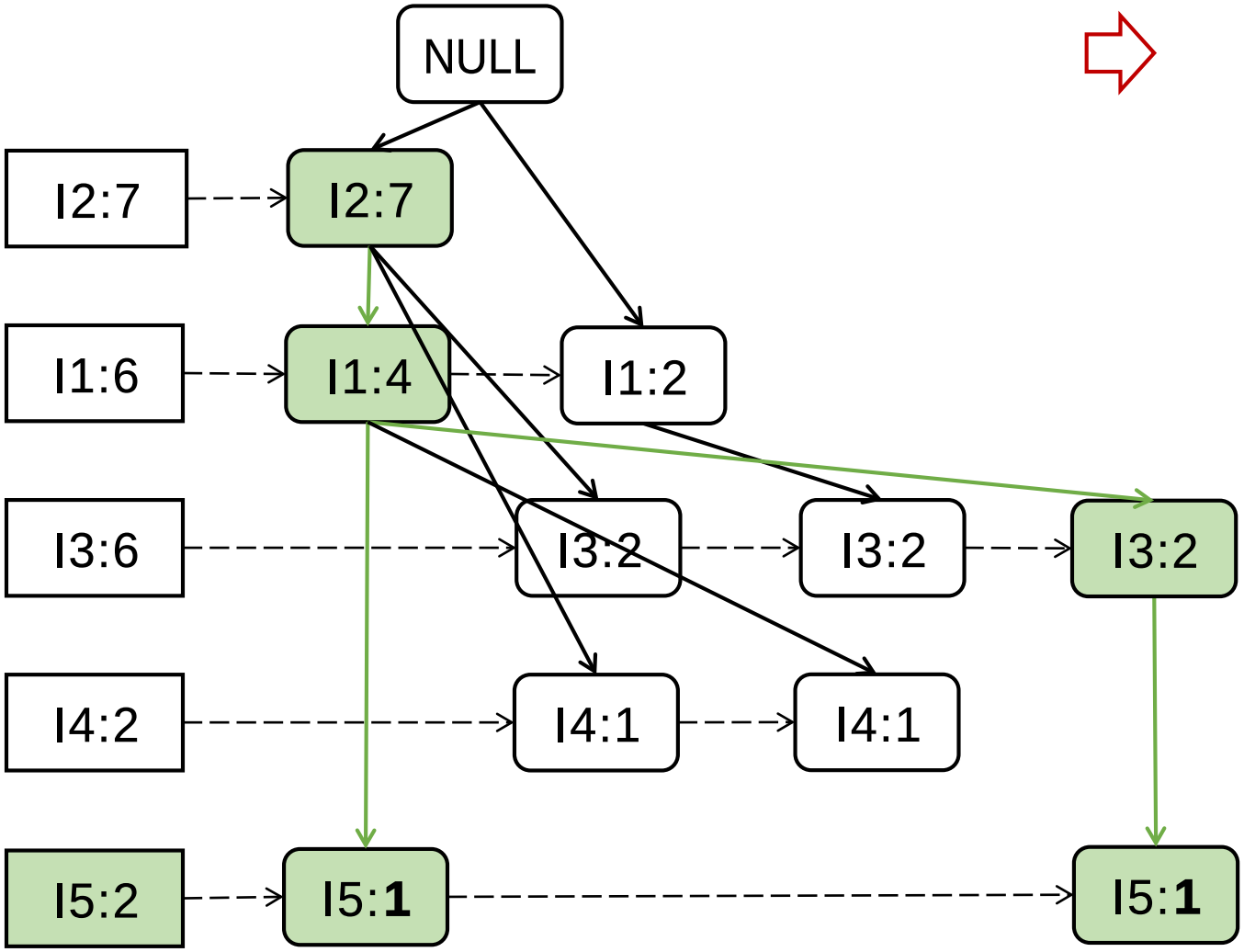
Построение FP-дерева

$D, minsup = 0.2$

TID	Items
10	I2, I1, I5
20	I2, I4
30	I2, I3
40	I2, I1, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I2, I1, I3, I5
90	I2, I1, I3
100	I6

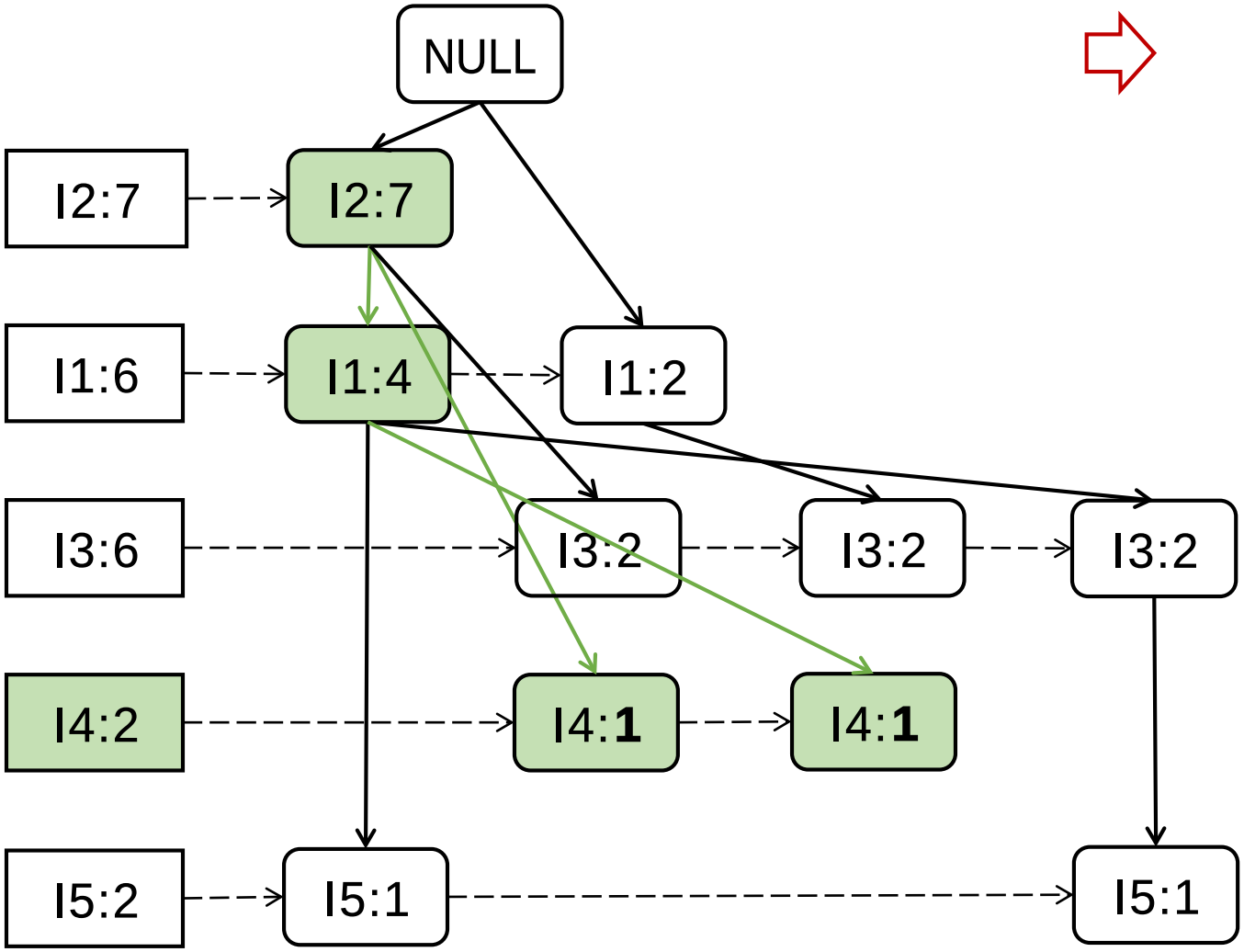


Обход FP-дерева: условные базисы



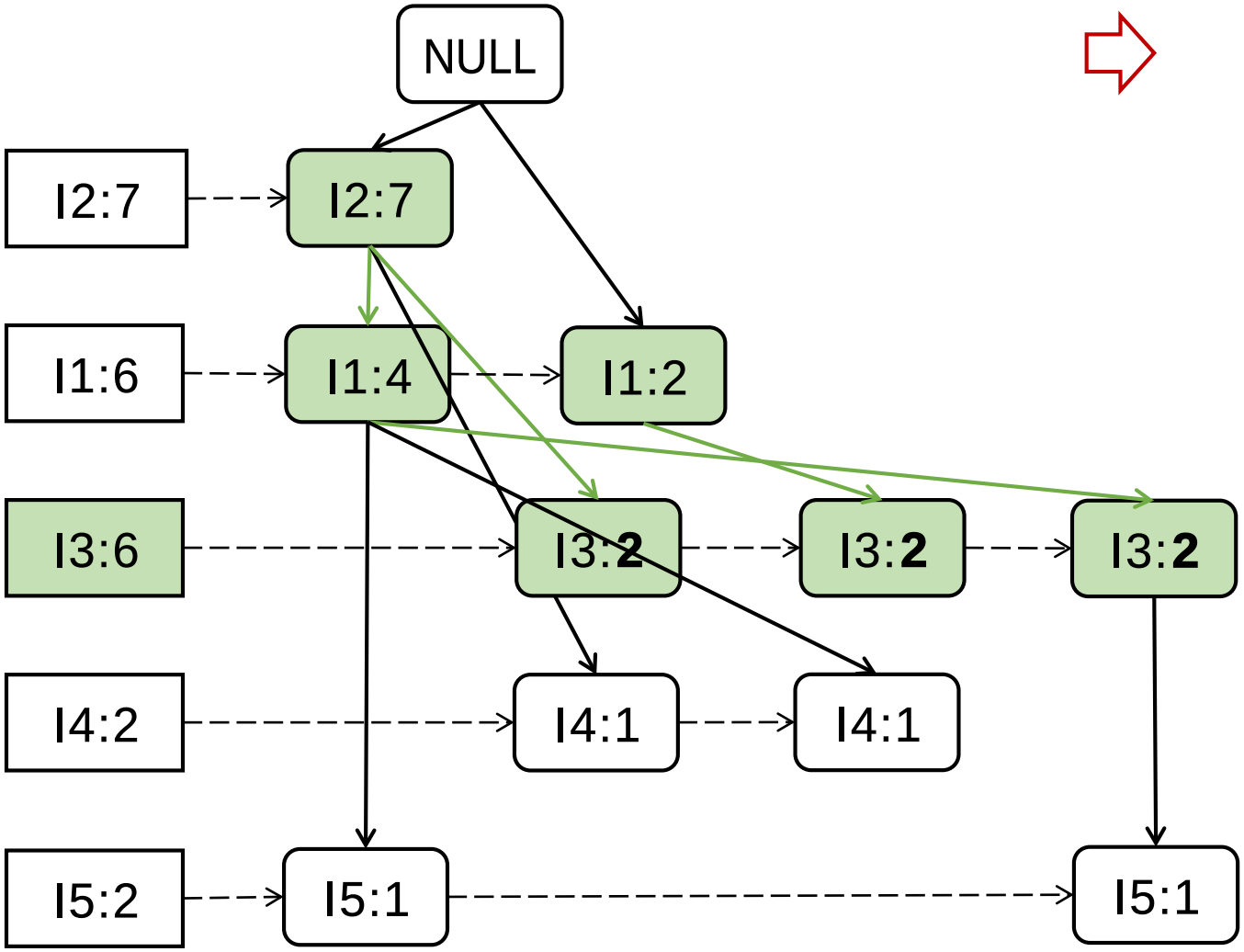
Item	Conditional pattern base
I2	∅
I1	
I3	
I4	
I5	{I2,I1:**1**}, {I2,I1,I3:**1**}

Обход FP-дерева: условные базисы



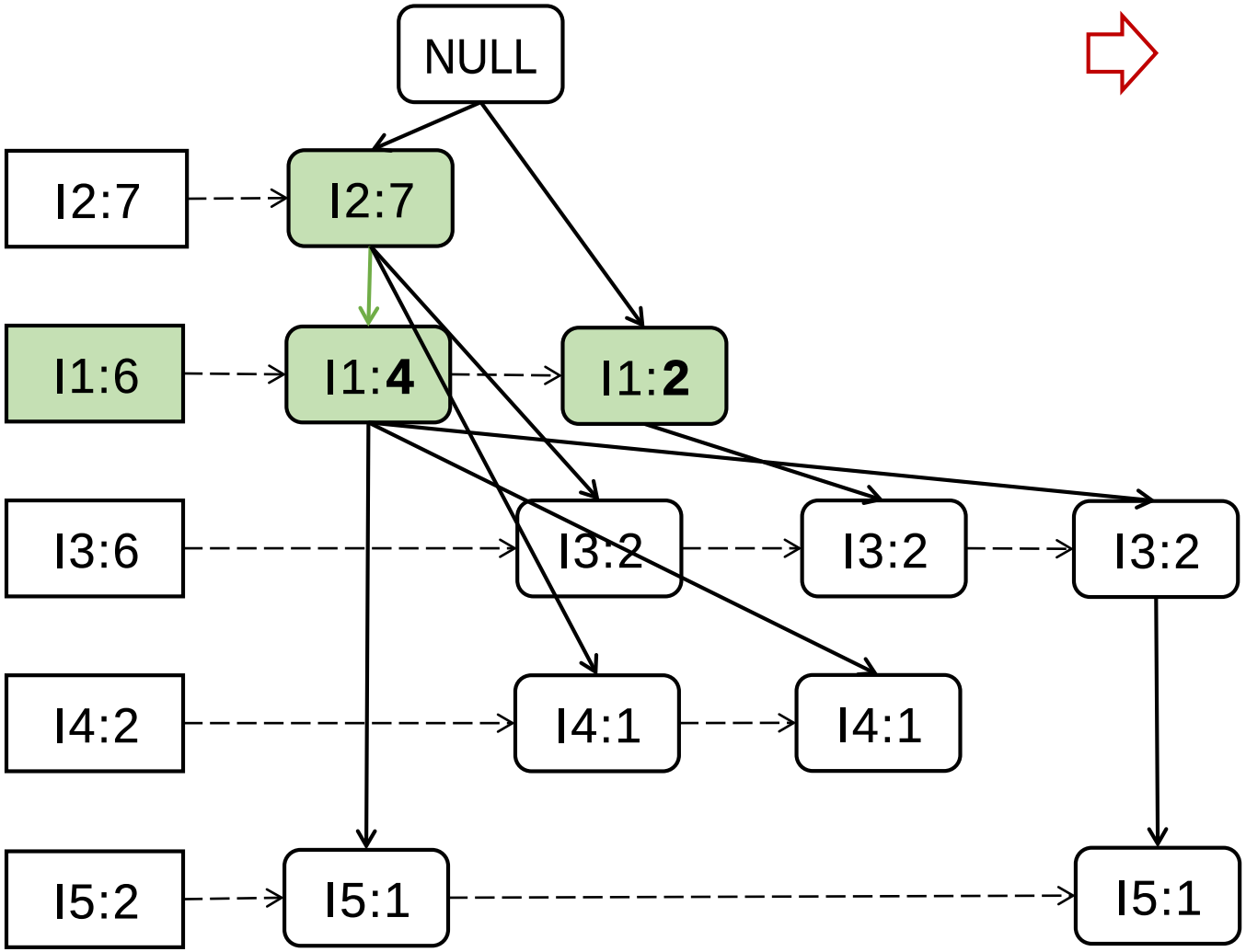
Item	Conditional pattern base
I2	∅
I1	
I3	
I4	{I2,I1:**1**}, {I2:**1**}
I5	{I2,I1:1}, {I2,I1,I3:1}

Обход FP-дерева: условные базисы



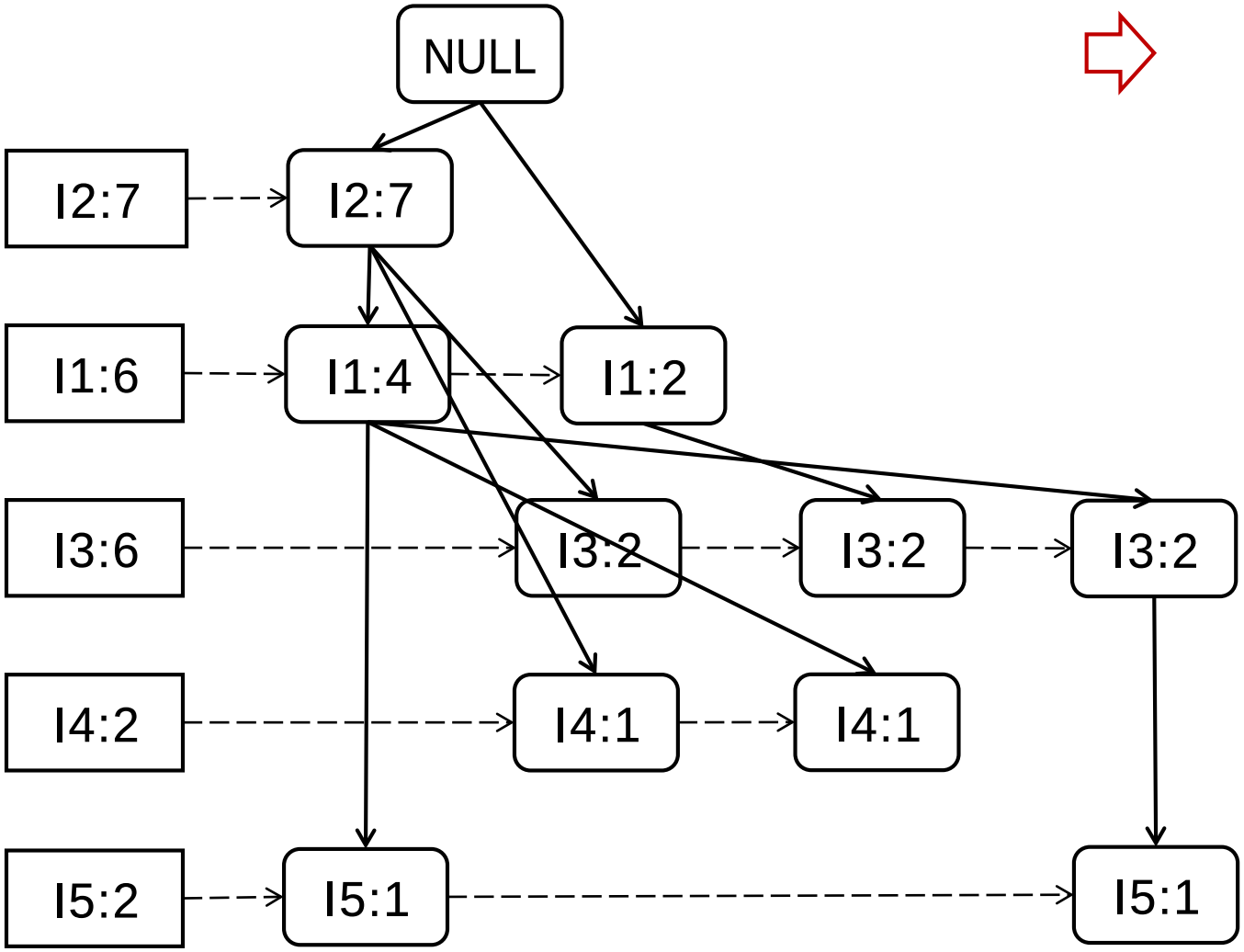
Item	Conditional pattern base
I2	∅
I1	
I3	{I2,I1:**2**}, {I2:**2**}, {I1:**2**}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

Обход FP-дерева: условные базисы



Item	Conditional pattern base
I2	∅
I1	{I2:**4**}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

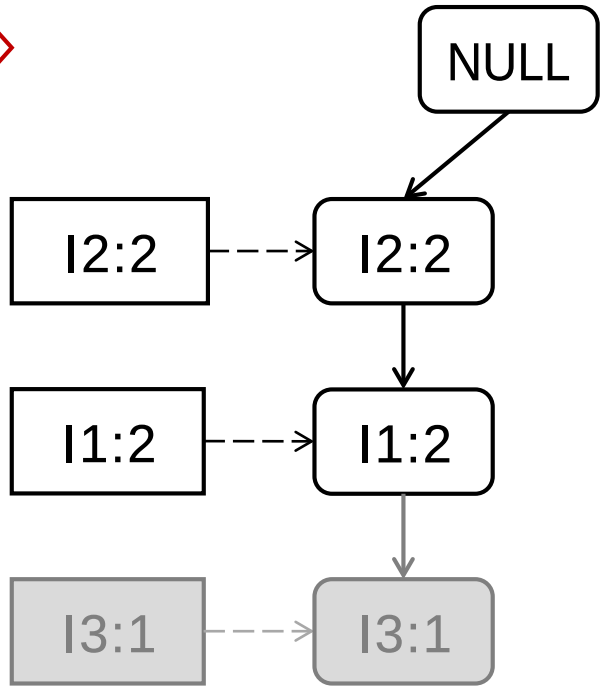
Обход FP-дерева: условные базисы



Item	Conditional pattern base
I2	∅
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

Обход FP-дерева: условные FP-деревья

Item	Conditional pattern base
I2	∅
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I2:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

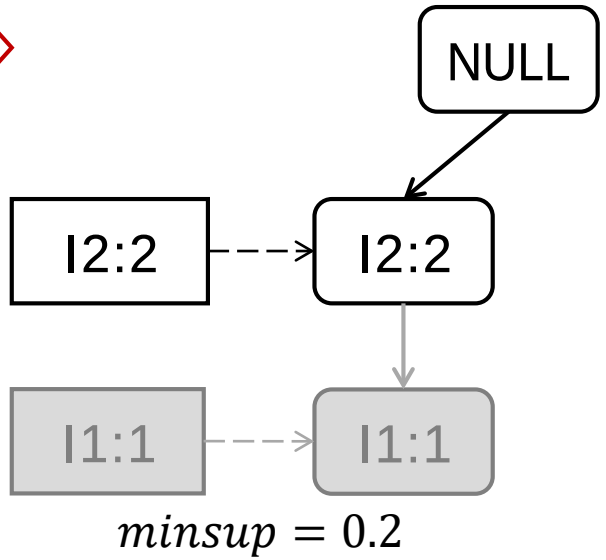


$minsup = 0.2$

Item	Conditional FP-tree
I2	∅
I1	
I3	
I4	
I5	{I2:2, I1:2}

Обход FP-дерева: условные FP-деревья

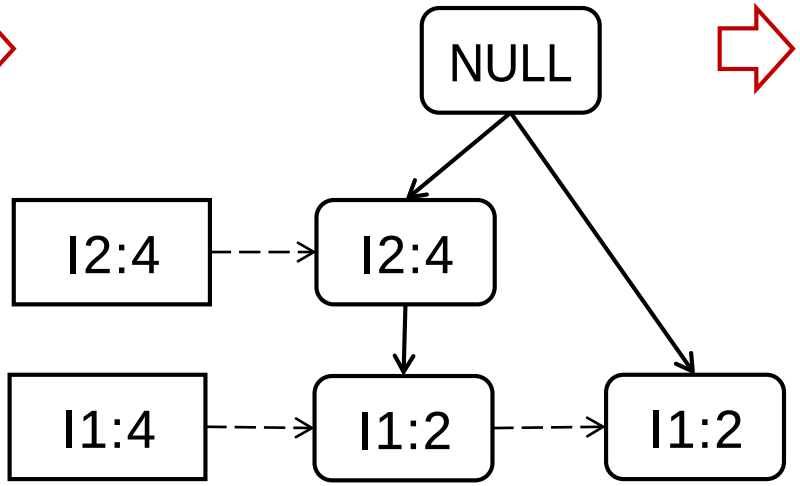
Item	Conditional pattern base
I2	∅
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I2:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}



Item	Conditional FP-tree
I2	∅
I1	
I3	
I4	{I2:2}
I5	{I2:2}, {I1:2}

Обход FP-дерева: условные FP-деревья

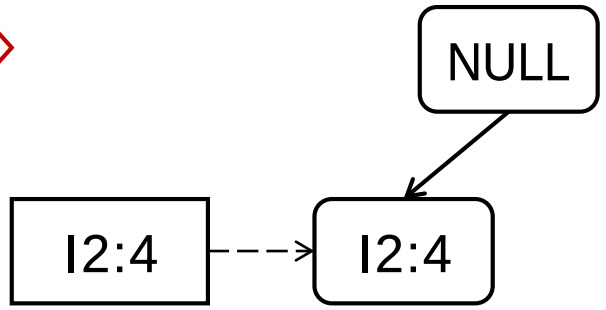
Item	Conditional pattern base
I2	∅
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}



Item	Conditional FP-tree
I2	∅
I1	
I3	{I2:4,I1:2} {I1:2}
I4	{I2:2}
I5	{I2:2, I1:2}

Обход FP-дерева: условные FP-деревья

Item	Conditional pattern base
I2	∅
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}



Item	Conditional FP-tree
I2	∅
I1	{I2:4}
I3	{I2:4,I1:2} {I1:2}
I4	{I2:2}
I5	{I2:2, I1:2}

Обход FP-дерева: условные FP-деревья

Item	Conditional pattern base
I2	∅
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

Item	Conditional FP-tree
I2	∅
I1	{I2:4}
I3	{I2:4,I1:2} {I1:2}
I4	{I2:2}
I5	{I2:2, I1:2}

Обход FP-дерева: частые наборы

Item	Conditional pattern base
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

Item	Conditional FP-tree
I1	{I2:4}
I3	{I2:4,I1:2} {I1:2}
I4	{I2:2}
I5	{I2:2, I1:2}

Prefix	Frequent Itemsets
I1	{I2,I1:4}
I3	{I2,I3:4}, {I1,I3:4}, {I2,I1,I3:2}
I4	{I2,I4:2}
I5	{I2,I5:2}, {I1,I5:2}, {I2,I1,I5:2}

Обход FP-дерева: частые наборы

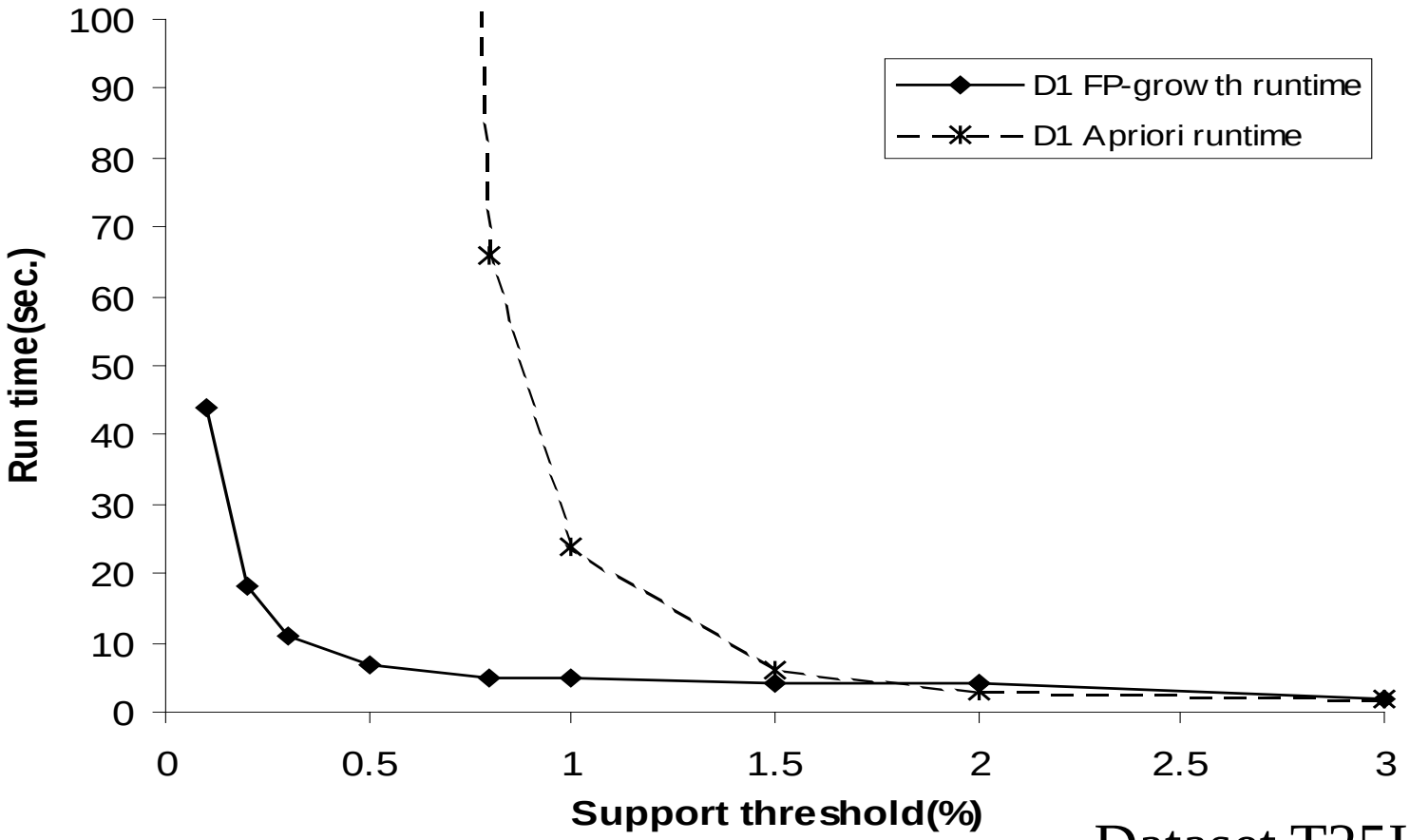
Item	Conditional pattern base
I1	{I2:4}
I3	{I2,I1:2}, {I2:2}, {I1:2}
I4	{I2,I1:1}, {I2:1}
I5	{I2,I1:1}, {I2,I1,I3:1}

Item	Conditional FP-tree
I1	{I2:4}
I3	{I2:4,I1:2} {I1:2}
I4	{I2:2}
I5	{I2:2, I1:2}

Prefix	Frequent Itemsets
I1	{I2,I1:4}
I3	{I2,I3:4}, {I1,I3:4}, {I2,I1,I3:2}
I4	{I2,I4:2}
I5	{I2,I5:2}, {I1,I5:2}, {I2,I1,I5:2}

$\cup\ \mathcal{L}_1$

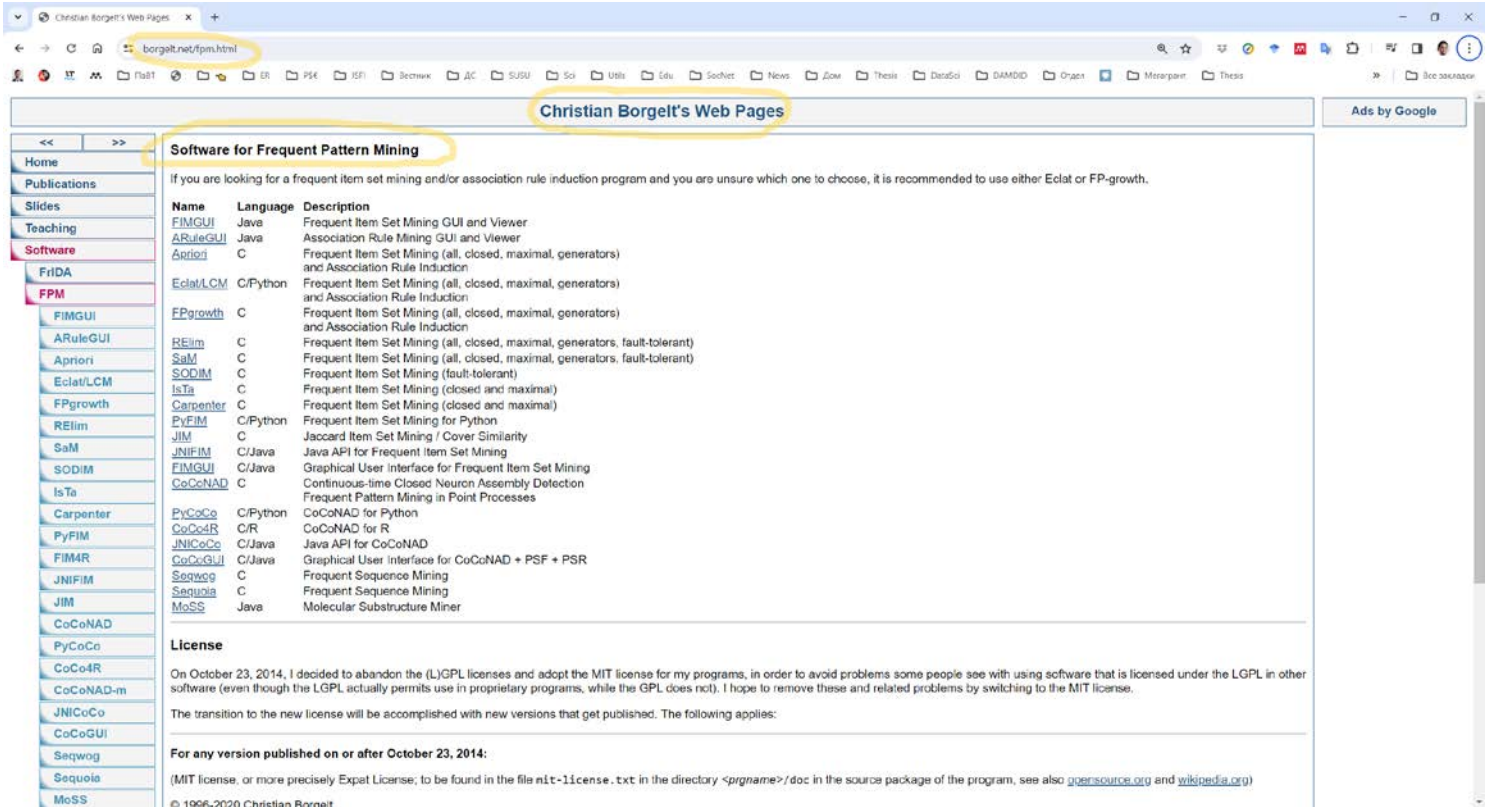
FP-Growth *vs.* Apriori



Dataset T25I20D10K

Реализация алгоритмов поиска шаблонов

- Christian Borgelt, Prof. for Data Science at the Paris Lodron University of Salzburg



- IBM Synthetic Data Generator for Itemsets and Sequences
  https://github.com/zakimjz/IBMGenerator

Литература

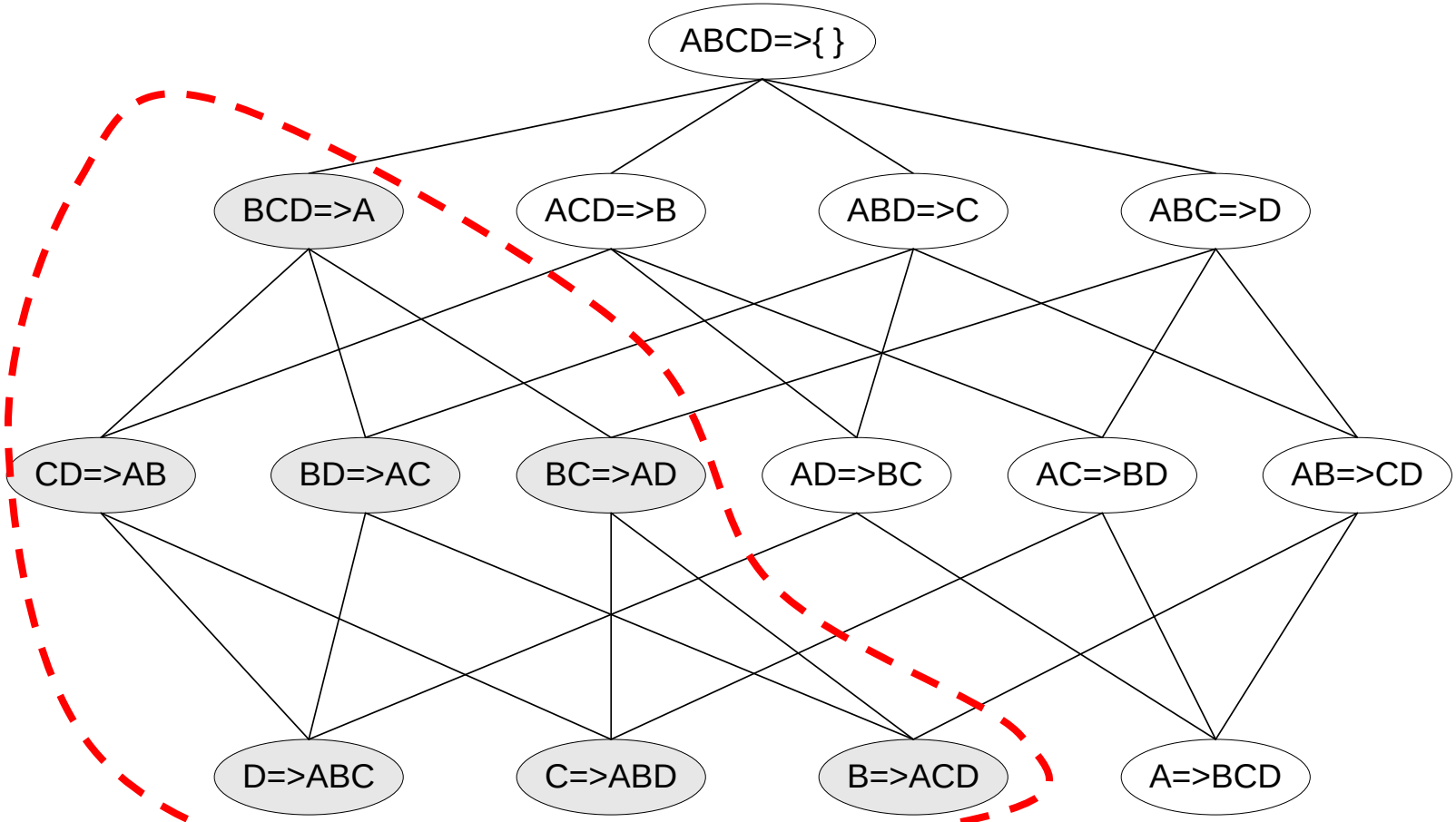
- Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd Edition. Morgan Kaufmann, 2011. 740 p. ISBN 978-0123814791
  - Chapter 6. Mining Frequent Patterns, Associations, and Correlations: Basic Concepts and Methods, pp. 243-278
- Tan P.-N., Steinbach M., Karpatne A., Kumar V. Introduction to Data Mining. 2nd Edition. Pearson, 2019. 839 p. ISBN-13: 978-0-13-312890-1
  - 5. Association Analysis: Basic Concepts and Algorithms, pp. 357-450

Отбрасывание заведомо неустойчивых шаблонов

- В общем случае достоверность не обладает свойством антимонотонности
- **Теорема**.
  Пусть A, B – непустые наборы и $A \subseteq B$.
  Если $\text{conf}(A \to B \setminus A) < minconf$, то
  $\forall \tilde{A} \subseteq A \ \text{conf}\left(\tilde{A} \to B \setminus \tilde{A}\right) < minconf$

  Доказательство.
  $\text{conf}(A \to B \setminus A) = \frac{\sup(B)}{\sup(A)}$, $\text{conf}\left(\tilde{A} \to B \setminus \tilde{A}\right) = \frac{\sup(B)}{\sup(\tilde{A})}$.
  $\tilde{A} \subseteq A \Rightarrow \sup\left(\tilde{A}\right) \geq \sup(A)$.

Отбрасывание заведомо неустойчивых шаблонов



Заведомо
неустойчивые шаблоны