**Методы поиска ассоциативных правил**

**Алгоритм AIS**. Первый *алгоритм* поиска *ассоциативных правил*, называвшийся *AIS* [62], (предложенный Agrawal, Imielinski *and* Swami) был разработан сотрудниками исследовательского центра *IBM* Almaden в 1993 году. С этой работы начался интерес к *ассоциативным правилам* ; на середину 90-х годов прошлого века пришелся пик исследовательских *работ* в этой области, и с тех пор каждый год появляется несколько новых алгоритмов.

В алгоритме *AIS* кандидаты *множества* наборов генерируются и подсчитываются "на лету", во время сканирования *базы данных*.

**Алгоритм SETM**. Создание этого алгоритма было мотивировано желанием использовать язык *SQL* для вычисления часто встречающихся наборов товаров. Как и *алгоритм* *AIS*, SETM также формирует кандидатов "на лету", основываясь на преобразованиях *базы данных*. Чтобы использовать стандартную операцию объединения языка *SQL* для формирования кандидата, SETM отделяет формирование кандидата от их подсчета.

Неудобство алгоритмов *AIS* и SETM - излишнее генерирование и подсчет слишком многих кандидатов, которые в результате не оказываются часто встречающимися. Для улучшения их работы был предложен *алгоритм* Apriori [63].

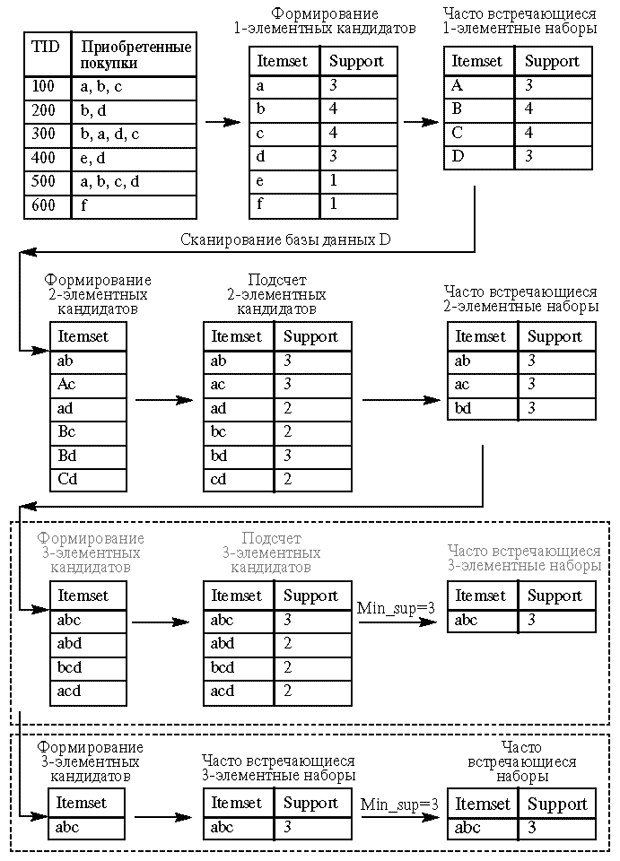
Работа данного алгоритма состоит из нескольких этапов, каждый из этапов состоит из следующих шагов:

* формирование кандидатов;
* подсчет кандидатов.

**Формирование кандидатов** (candidate *generation*) - этап, на котором *алгоритм*, сканируя базу данных, создает множество i-элементных кандидатов (i - номер этапа). На этом этапе *поддержка* кандидатов не рассчитывается.

**Подсчет кандидатов** (candidate counting) - этап, на котором вычисляется *поддержка* каждого i-элементного кандидата. Здесь же осуществляется *отсечение* кандидатов, *поддержка* которых меньше минимума, установленного пользователем (min\_sup). Оставшиеся i-элементные наборы называем часто встречающимися.

Рассмотрим работу алгоритма Apriori на примере *базы данных* D. Иллюстрация работы алгоритма приведена на [рис. 15.1](https://www.intuit.ru/studies/professional_skill_improvements/1210/courses/6/lecture/186?page=3#image.15.1). Минимальный уровень *поддержки* равен 3.



**Рис. 15.1.**Алгоритм Apriori

На первом этапе происходит формирование одноэлементных кандидатов. Далее *алгоритм* подсчитывает *поддержку* одноэлементных наборов. Наборы с уровнем *поддержки* меньше установленного, то есть 3, отсекаются. В нашем примере это наборы e и f, которые имеют *поддержку*, равную 1. Оставшиеся наборы товаров считаются часто встречающимися одноэлементными наборами товаров: это наборы a, b, c, d.

Далее происходит формирование двухэлементных кандидатов, подсчет их *поддержки* и *отсечение* наборов с уровнем *поддержки*, меньшим 3. Оставшиеся двухэлементные наборы товаров, считающиеся часто встречающимися двухэлементными наборами ab, ac, bd, принимают участие в дальнейшей работе алгоритма.

Если смотреть на работу алгоритма прямолинейно, на последнем этапе *алгоритм* формирует трехэлементные наборы товаров: abc, abd, *bcd*, *acd*, подсчитывает их *поддержку* и отсекает наборы с уровнем *поддержки*, меньшим 3. Набор товаров abc может быть назван часто встречающимся.

Однако *алгоритм* Apriori уменьшает количество кандидатов, отсекая - априори - тех, которые заведомо не могут стать часто встречающимися, на основе информации об отсеченных кандидатах на предыдущих этапах работы алгоритма.

*Отсечение* кандидатов происходит на основе предположения о том, что у часто встречающегося набора товаров все подмножества должны быть часто встречающимися. Если в наборе находится *подмножество*, которое на предыдущем этапе было определено как нечасто встречающееся, этот кандидат уже не включается в формирование и подсчет кандидатов.

Так наборы товаров ad, bc, *cd* были отброшены как нечасто встречающиеся, *алгоритм* не рассматривал набор товаров abd, *bcd*, *acd*.

При рассмотрении этих наборов формирование трехэлементных кандидатов происходило бы по схеме, приведенной в верхнем пунктирном прямоугольнике. Поскольку *алгоритм* априори отбросил заведомо нечасто встречающиеся наборы, последний этап алгоритма сразу определил набор abc как единственный трехэлементный часто встречающийся набор (этап приведен в нижнем пунктирном прямоугольнике).

*Алгоритм* Apriori рассчитывает также *поддержку* наборов, которые не могут быть отсечены априори. Это так называемая негативная область (*negative* *border*), к ней принадлежат наборы-кандидаты, которые встречаются редко, их самих нельзя отнести к часто встречающимся, но все подмножества данных наборов являются часто встречающимися.