Регулярные множества и регулярные выражения

Элементами регулярного множества являются цепочки символов, образованные из символов алфавита X, т.е. регулярное множество представляет собой некоторый формальный язык.

Регулярными множествами являются:

1. ∅ — пустое множество;
2. {ε} — множество, состоящее из одной пустой цепочки;
3. {x} — множество, состоящее из одной цепочки, представляющей собой один символ алфавита X;
4. если A и B — произвольные регулярные множества, то множества A∪B, AB, A*, B* также являются регулярными множествами;
5. ничто другое не является регулярным множеством.

Одним из способов задания регулярного множества является регулярное выражение.

Регулярными выражениями являются:

1. 0 — регулярное выражение, обозначающее ∅;
2. ε — регулярное выражение, обозначающее {ε};
3. x — регулярное выражение, обозначающее {x};
4. если a и b — регулярные выражения, обозначающие регулярные множества A и B, то:

   a+в — регулярное выражение, обозначающее множество A∪B;

   ав — регулярное выражение, обозначающее множество AB;

   a* — регулярное выражение, обозначающее множество A*;

   в* — регулярное выражение, обозначающее множество B*.

При записи регулярных выражений можно использовать круглые скобки, как и в обычных арифметических выражениях, для указания порядка выполнения операций. Для уменьшения числа скобок используются приоритеты операций: итерация самая приоритетная; менее приоритетна конкатенация; самый низкий приоритет у объединения. При отсутствии скобок операции выполняются слева направо с учётом приоритета.

Для сокращения записи регулярного выражения можно использовать систему выражений вида:

$d_1=r_1$

$d_2=r_2$

...

$d_n=r_n$

где d_i — различные имена, а каждое r_i — регулярное выражение над символами $X \cup \{d_1,d_2,...,d_{i-1}\}$, т.е. символами основного алфавита и ранее определенными символами. Таким образом, для любого r_i можно построить регулярное выражение над X, повторно заменяя имена регулярных выражений на обозначаемые ими регулярные выражения.

Регулярные множества и регулярные выражения представляют собой разные сущности: регулярное множество — это множество цепочек (в общем случае бесконечное), а регулярное выражение — это формула (составленная из конечного числа символов), схематично показывающая, как может быть построено соответствующее её регулярное множество.

Пример.

Регулярные выражения r_1 и r_2 в алфавите X={ц, ., ±, -}:

r_1=((ц + (± + -)ц)ц*. + (. + (± + -).)ц)ц*

r_2=цц*.ц* + (± + -)цц*.ц* + (± + -).цц* + .цц*

Одно и то же регулярное множество может быть представлено различными регулярными выражениями, например выражениями r_1 и r_2 (см. пример). Два регулярных выражения называются *эквивалентными*, если они определяют одно и то же регулярное множество

.

Для любых регулярных выражений a, b и c справедливы равенства:

1. a+b = b+a	6. a+a* = a*	11. a+a = a
2. a+(b+c) = (a+b)+c	7. aa* = a*a	12. 0a = a0 = 0
3. a(b+c) = ab+ac	8. a(ba)* = (ab)*a	13. 0+a = a+0 = a
4. (a+b)c = ac+bc	9. (a*)* = a*	14. εa = aε = a
5. a(bc) = (ab)c	10. 0* = ε	15. ε+a* = a*+ε = a*

Эти равенства можно доказать, проверяя равенство соответствующих множеств цепочек. Их можно использовать для преобразования, в том числе и для упрощения регулярных выражений.

Регулярные выражения и конечные распознаватели

Любой регулярный язык может быть задан регулярным выражением и может быть распознан конечным распознавателем, следовательно, для каждого конечного распознавателя существует регулярное выражение, определяющее тот язык, который допускается распознавателем, и, обратно, для каждого регулярного выражения существует конечный распознаватель, который допускает только то множество цепочек, которое задаёт регулярное выражение. Далее рассмотрим алгоритм получения регулярного выражения, определяющего множество цепочек, допускаемых заданным конечным распознавателем и алгоритм построения конечного распознавателя, допускающего множество цепочек, заданных регулярным выражением.

Получение регулярного выражения из конечного распознавателя

Введём в рассмотрение модель *обобщённого графа переходов* как расширение графа конечного распознавателя.

В обобщённом графе переходов одна начальная и произвольное количество допускающих вершин, а дуги, в отличие от графа конечного распознавателя, помечены не символами алфавита, а регулярными выражениями.

Цепрочка допускается обобщённым графом переходов, если эта цепочка принадлежит множеству, описываемому конкатенацией регулярных выражений, которые помечают путь из начальной вершины в допускающую.

Язык, допускаемый обобщённым графом переходов, представляет собой множество всех допускаемых им цепочек.

Граф конечного распознавателя является частным случаем обобщённого графа переходов, поэтому все языки, допускаемые конечными распознавателями, допускаются и обобщёнными графами переходов.

Обобщённый граф переходов можно представить в *нормализованной форме*, в которой только одна допускающая вершина, из которой не исходит ни одна дуга, а в единственную начальную вершину не входит ни одна дуга.

Для этого в обобщённый граф переходов нужно ввести новую допускающую вершину и провести в неё дуги — ε-переходы — из каждой допускающей вершины (за искючением введённой вершины), а исходные допускающие вершины не считать допускающими.

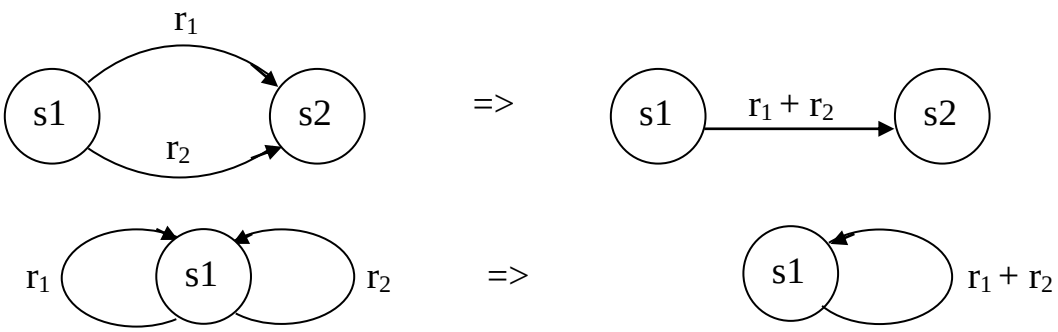
Если в начальную вершину входит хотя бы одна дуга, то ввести новую вершину, провести из неё дугу — ε-переход — в начальную вершину и начальной вершиной считать только введённую вершину.

Нормализованный обобщённый граф переходов можно преобразовать в граф, содержащий только две вершины — одну начальную и одну допускающую, которые соединены одной дугой, отмеченной регулярным выражением, описывающим множество, допускаемое исходным графом.

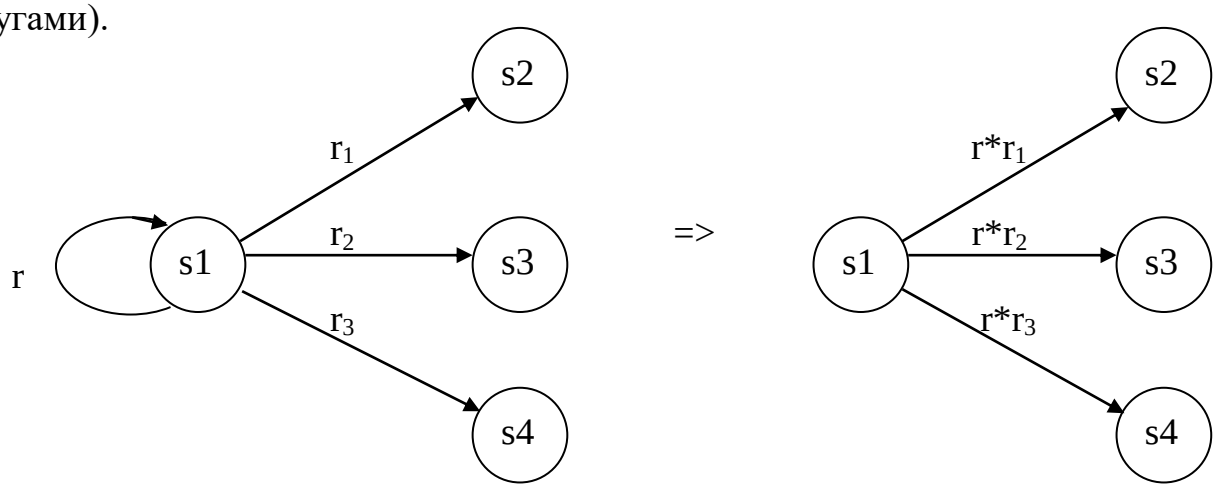
Такое преобразование выполняется путём исключения дуг и вершин.

При исключении дуг и вершин применяются следующие три правила:

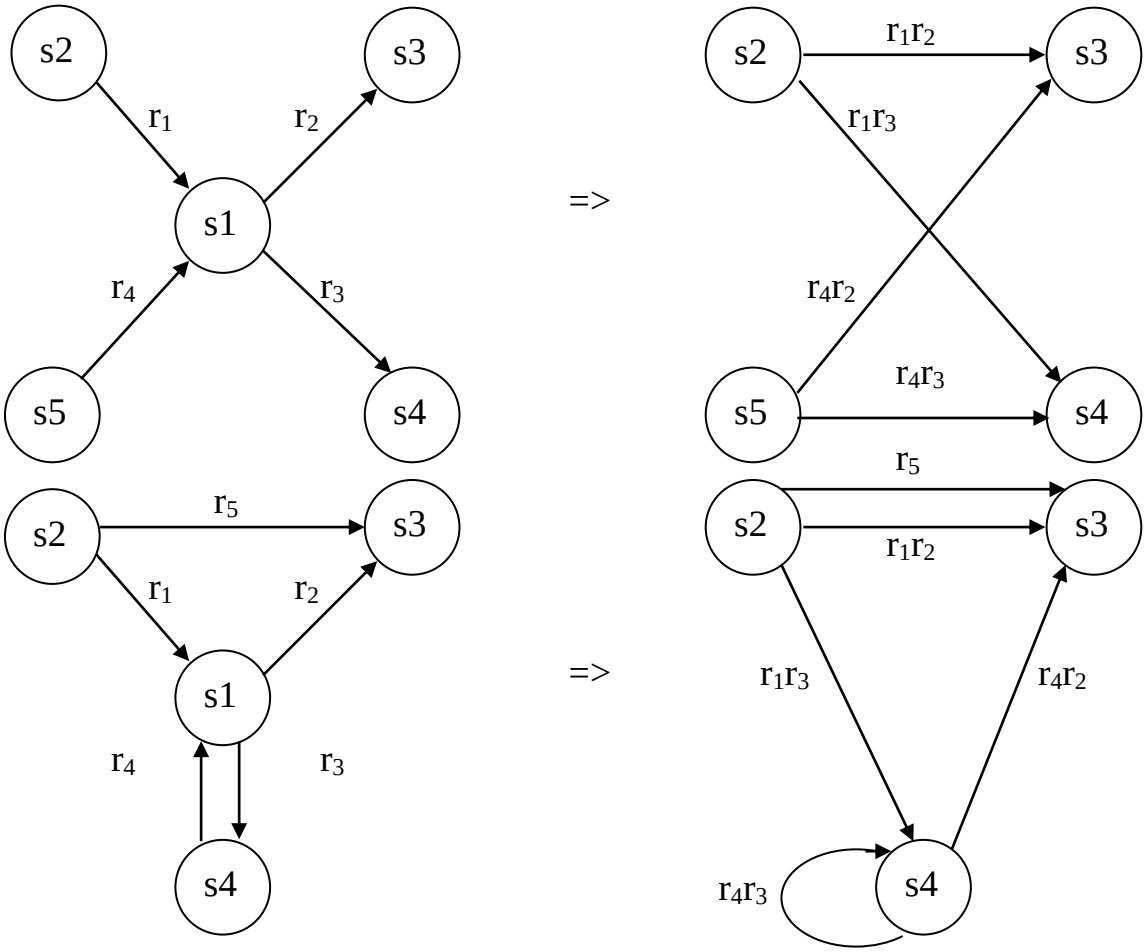
1) правило исключения “параллельных” дуг. Две дуги с одинаковым началом, одинаковым концом и с метками r_1 и r_2 соответственно, заменяются одной дугой с теми же началом и концом, отмеченной выражением $r_1 + r_2$.



2) правило исключения “петель”. Если существует вершина, отличная от начальной и допускающей, являющаяся началом и концом дуги, отмеченной r , то такая дуга (“петля”) исключается, а метки всех выходящих из вершины дуг конкатенируются слева с r* . Если же ни одна дуга (после исключения “петли”) не выходит из вершины, то вершина исключается вместе со всеми входящими в неё дугами. Вершина исключается и в том случае, если в неё не входит ни одна дуга (исключается вместе со всеми выходящими из неё дугами).



3) правило исключения вершин. Вершина без петли, отличная от начальной и допускающей, исключается. Каждый путь от вершины, из которой выходит дуга, ведущая в исключаемую вершину, до вершины, в которую входит дуга, исходящая из исключаемой вершины, заменяется дугой, отмеченной конкатенацией меток на дугах рассматриваемого пути. Допускаются циклические пути и дуги, соединяющие начальную и конечную вершину пути.



Алгоритм получения регулярного выражения, определяющего множество цепочек, допускаемых заданным конечным распознавателем.

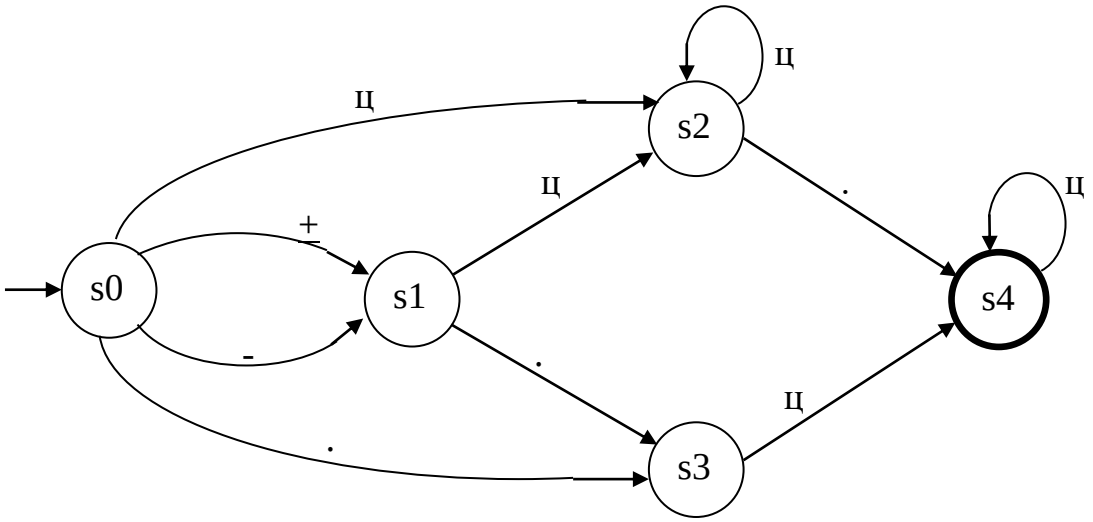
1. Преобразовать граф конечного распознавателя в нормализованный обобщённый граф переходов.

2. Во всех возможных случаях применить правило исключения “параллельных” дуг. Во всех возможных случаях применить правило исключения “петель”. Если возможно, применить правило исключения вершины и выполнить п.2, иначе перейти к п.3.

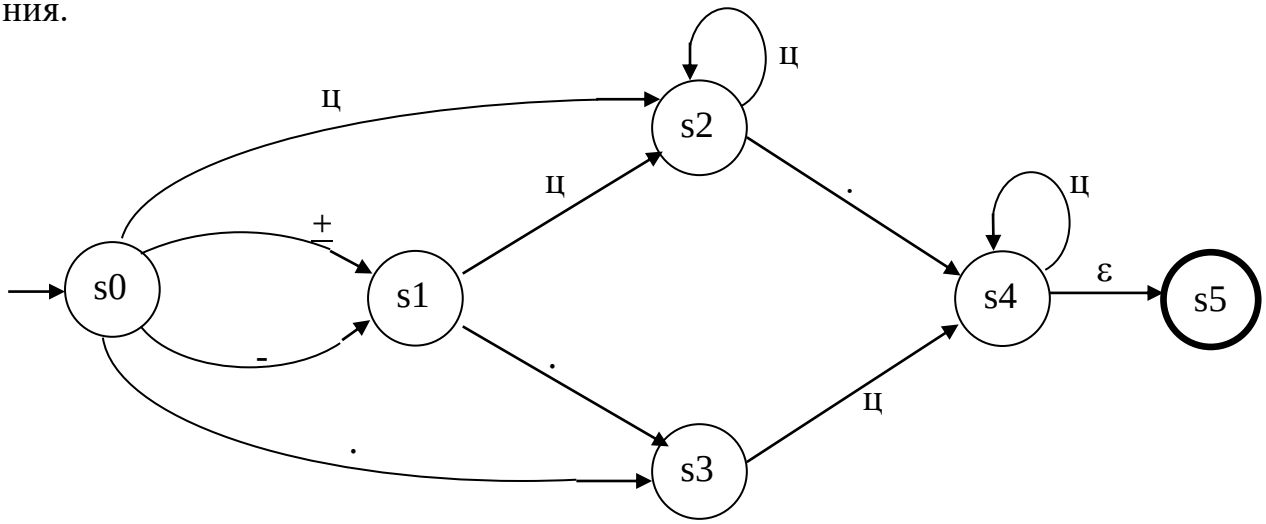
3. Если в полученном графе не оказалось дуг, провести дугу от начальной вершины к допускающей и отметить её символом ∅.

4. Дуга, ведущая от начальной вершины к допускающей, отмечена регулярным выражением, определяющим множество цепочек, допускаемых заданным конечным распознавателем.

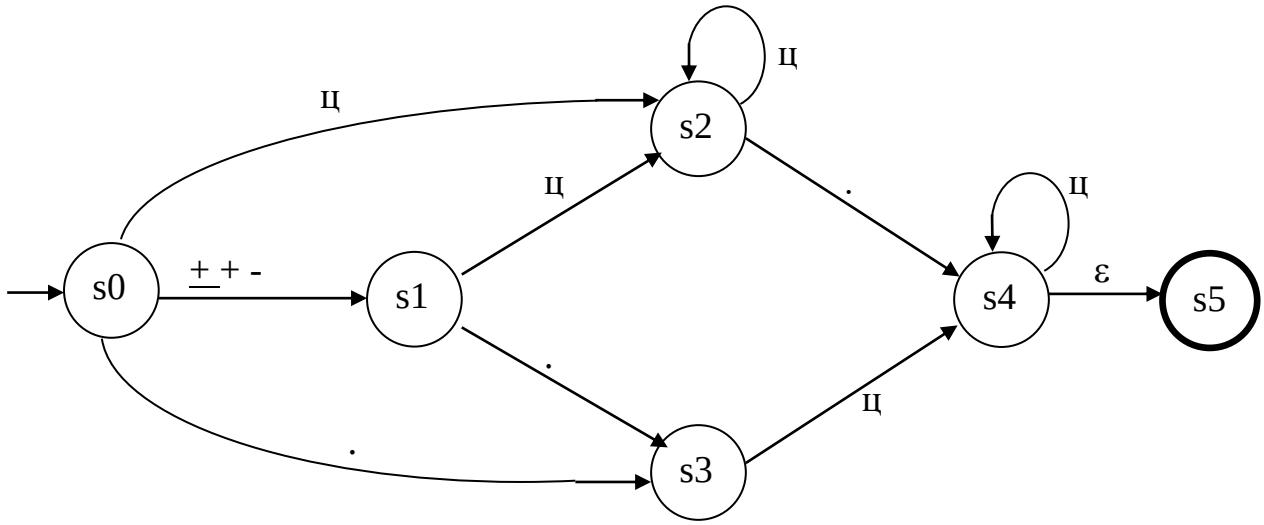
Пример.
Граф конечного распознавателя.



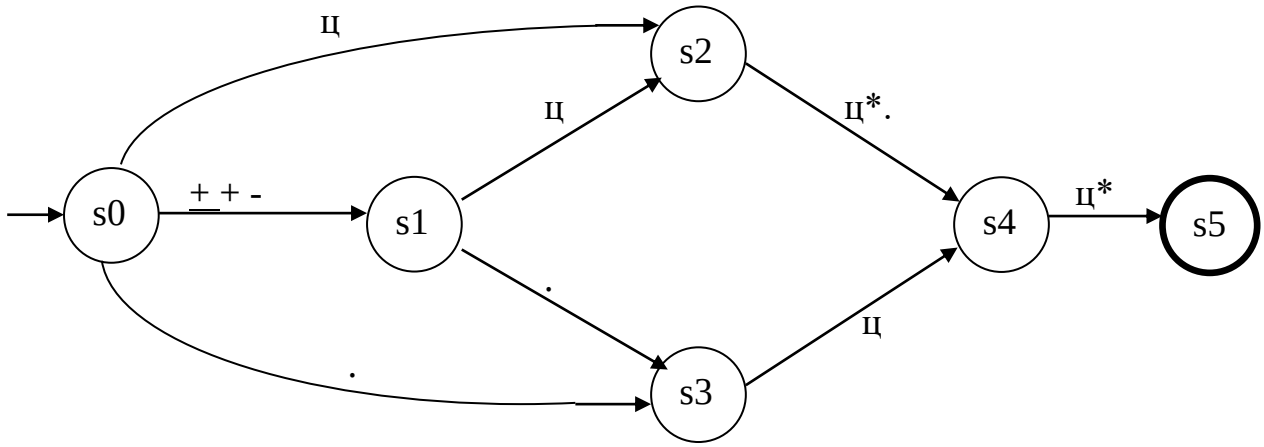
1. Преобразование графа конечного распознавателя в нормализованный обобщённый граф переходов добавлением нового допускающего состояния.



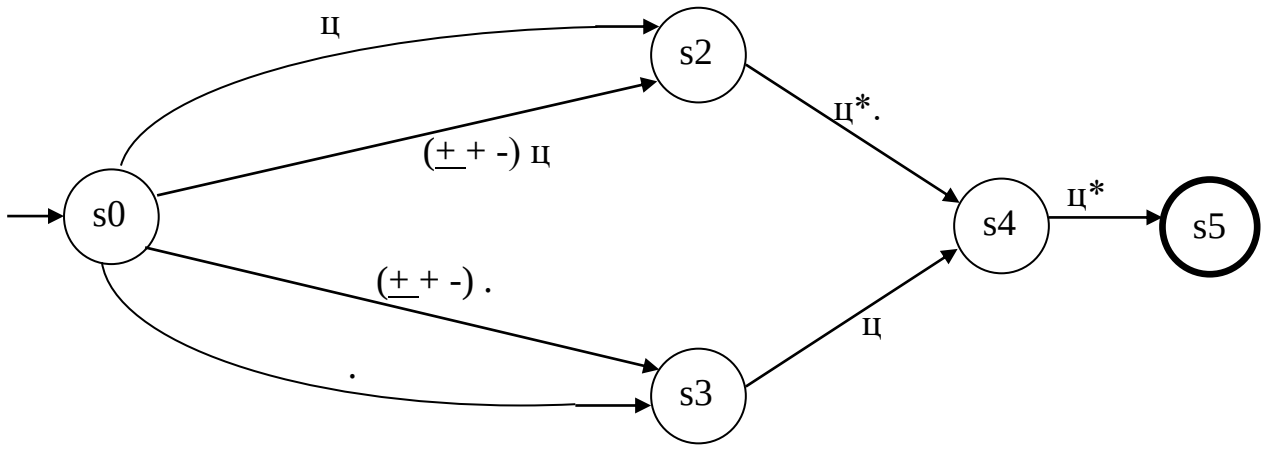
2. Исключение “параллельных” дуг.



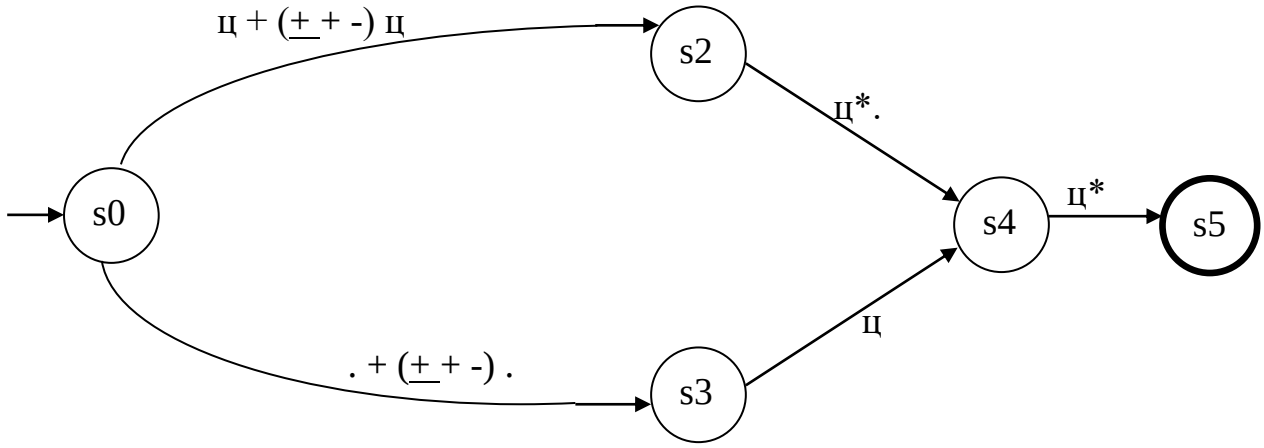
3. Исключение “петель”.



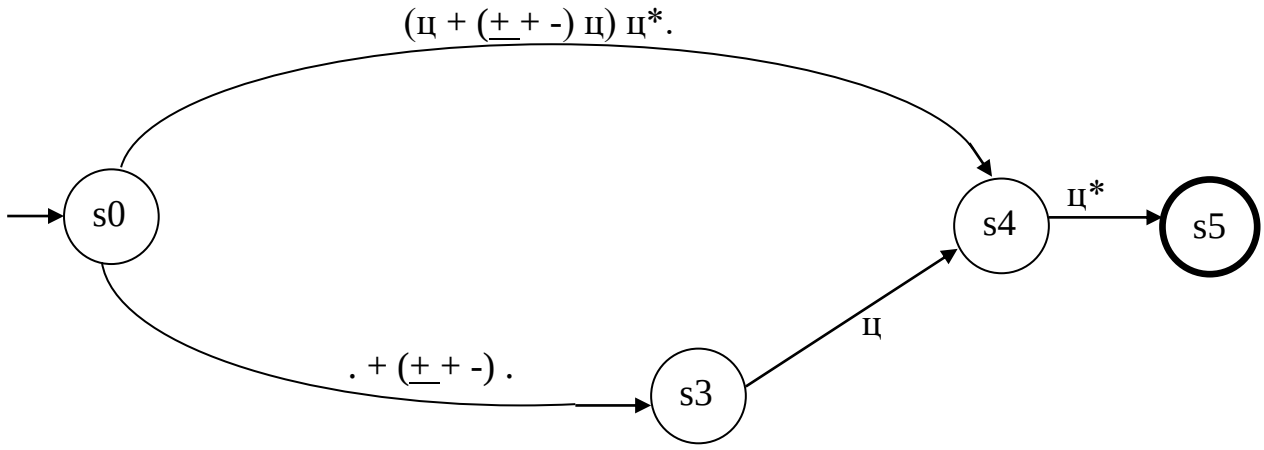
4. Исключение вершины s1 .



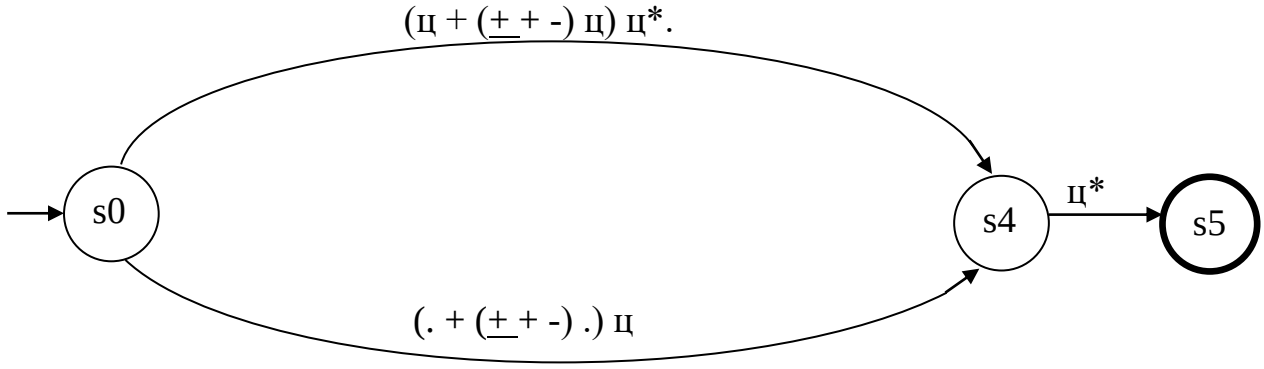
5. Исключение “параллельных” дуг.



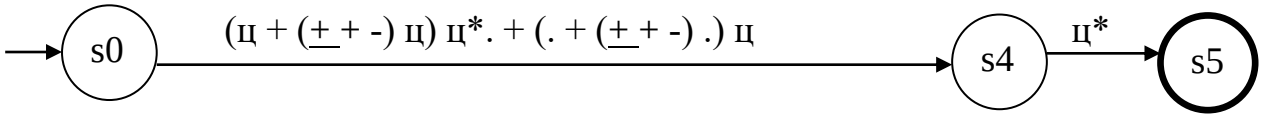
6. Исключение вершины s2 .



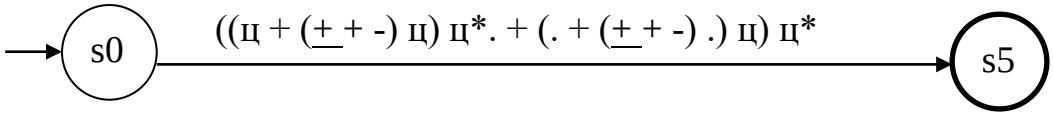
7. Исключение вершины s3 .



8. Исключение “параллельных” дуг.



9. Исключение вершины s4 .



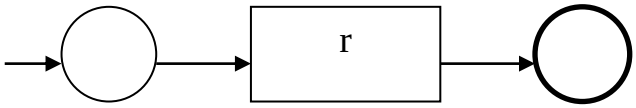
10. Искомое регулярное выражение

r = ((ц + (+ + -) ц) ц*. + (. + (+ + -) .) ц) ц*

Построение конечного распознавателя по регулярному выражению

Пусть задано регулярное выражение r .

Конечный распознаватель (возможно, недетерминированный), допускающий множество цепочек, определяемых регулярным выражением r, содержащий одну начальную вершину, в которую не входит ни одна дуга, и одну допускающую вершину, из которой не исходит ни одна дуга, представим следующей моделью:



Начальная вершина

В прямоугольнике сосредоточны внутренние вершины

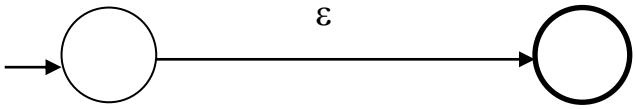
Допускающая вершина

Дуга, ведущая из начальной вершины в прямоугольник, соответствует всем дугам, идущим из начальной вершины, а дуга, ведущая из прямоугольника в допускающую вершину, соответствует всем дугам, идущим в допускающую вершину при графовом представлении конечного распознавателя.

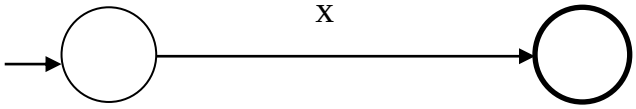
В зависимости от вида регулярного выражения r , прямоугольник в модели можно детализировать по следующим правилам:

1) если $r=\varnothing$, то прямоугольник исключается

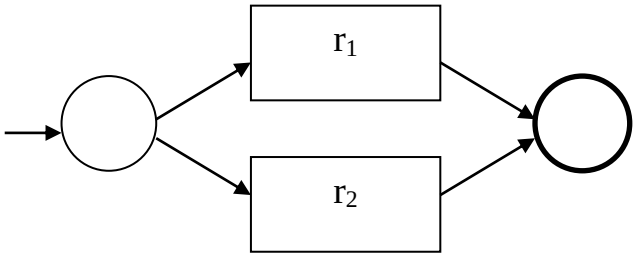
2) если $r=\varepsilon$, то прямоугольник заменяется одной дугой, отмеченной символом ε.



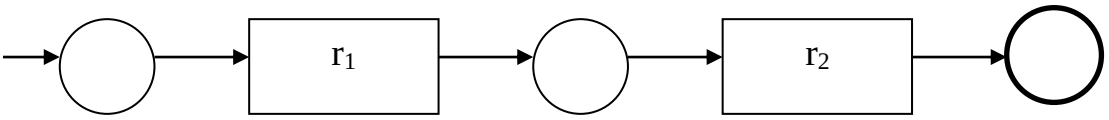
3) если $r=x$, $x \in X$, то прямоугольник заменяется одной дугой, отмеченной символом x.



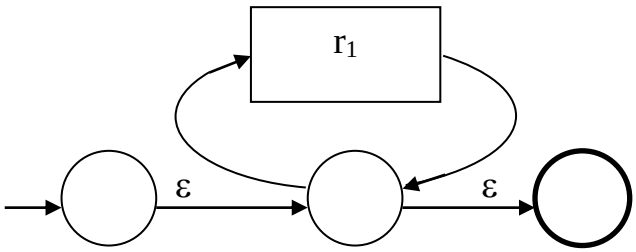
4) если $r=r_1+r_2$, то прямоугольник с меткой r заменяется двумя “параллельно” соединёнными прямоугольниками с метками r_1 и r_2 . Начальные и допускающие вершины прямоугольников с метками r_1 и r_2 совмещаются.



5) если $r=r_1r_2$, то прямоугольник с меткой r заменяется двумя “последовательно” соединёнными через дополнительную вершину прямоугольниками с метками r_1 и r_2. Новая вершина представляет собой совмещение допускающей вершины прямоугольника с меткой r_1 с начальной вершиной прямоугольника с меткой r_1.



6) если $r=r_1*$, то прямоугольник с меткой r заменяется новой вершиной с “петлёй”, на которой находится прямоугольник с меткой r_1. Дуги, соединяющие начальную и допускающую вершину с новой вершиной, отмечаются символом ε. Новая вершина представляет собой совмещение начальной и допускающей вершин для прямоугольника с меткой r_1.



Алгоритм построения детерминированного конечного распознавателя, допускающего множество цепочек, определяемых регулярным выражением r.

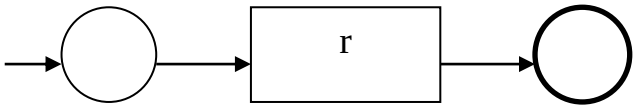
1. Конечный распознаватель, допускающий множество цепочек, определяемых регулярным выражением r , представить моделью, содержащей начальную, допускающую вершину и прямоугольник, в который вписано регулярное выражение r .

2. Пока в модели есть прямоугольники, детализировать их по правилам 1-6 .

Пример.

Регулярное выражение r = ((ц + (+ + -) ц) ц*. + (. + (+ + -) .) ц) ц*

1. Конечный распознаватель, допускающий множество цепочек, определяемых регулярным выражением r , представим моделью:

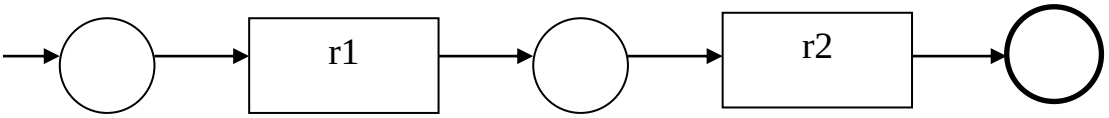


2. Регулярное выражение r = ((ц + (+ + -) ц) ц*. + (. + (+ + -) .) ц) ц* представим как r=r1r2, где

r1=((ц + (+ + -) ц) ц*. + (. + (+ + -) .) ц)

r2=ц*

По правилу 5 получаем:

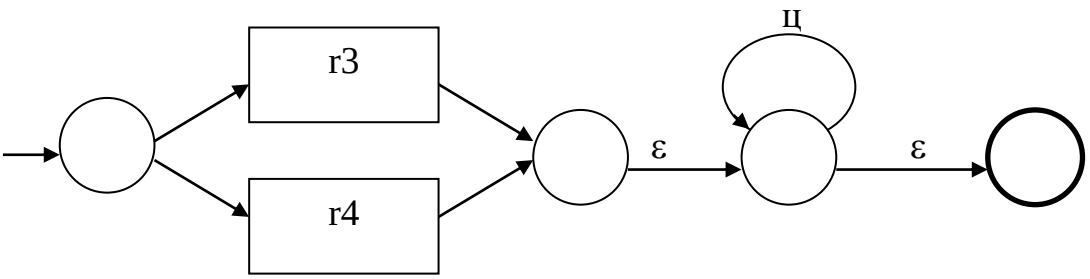


3. Регулярное выражение r1=((ц + (+ + -) ц) ц*. + (. + (+ + -) .) ц) представим как r1=r3 + r4, где

r3=(ц + (+ + -) ц) ц*.

r4=(. + (+ + -) .) ц

Применяя правило 4 к прямоугольнику с меткой r1 и правила 6 и 3 к прямоугольнику с меткой r2=ц*, получаем:



4. Регулярное выражение r3=(ц + (+ + -) ц) ц*. представим как r3=r5r6, где

r5=(ц + (+ + -) ц) ц*

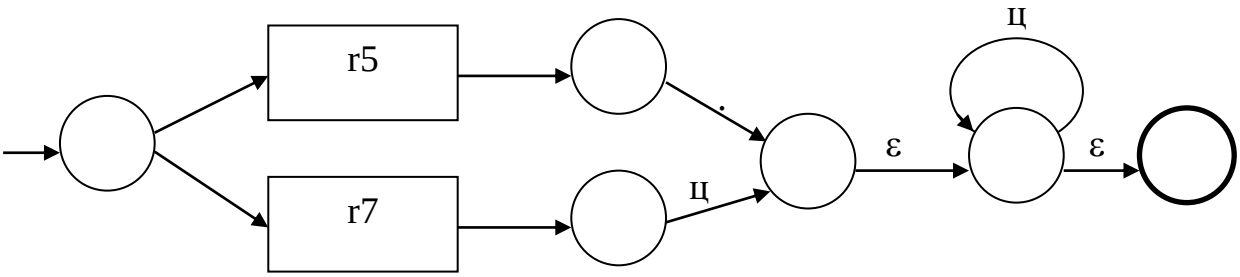
r6=.

Регулярное выражение r4=(. + (+ + -) .) ц представим как r4=r7r8, где

r7=. + (+ + -) .

r8=ц

Применяя правила 5 и 3 к прямоугольникам с метками r3 и r4, получаем:



5. Регулярное выражение r5=(ц + (+ + -) ц) ц* представим как r5=r9r10, где

r9=ц + (+ + -) ц

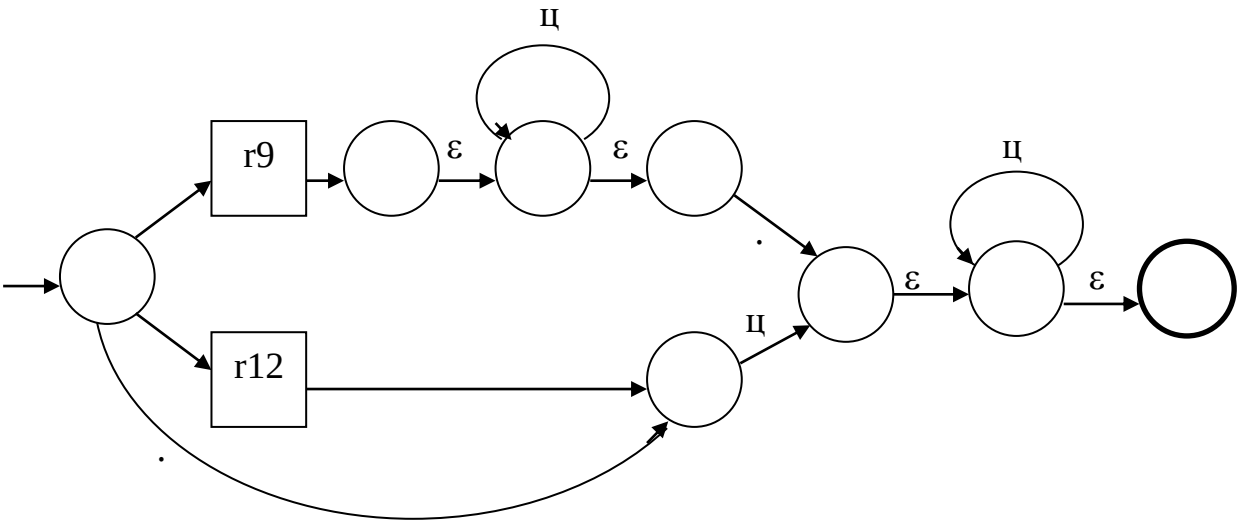
r10=ц*

Регулярное выражение r7=. + (+ + -) . представим как r7=r11 + r12, где

r11=.

r12= (+ + -) .

Применяя правила 5 и 6 к прямоугольнику с меткой r5 и правила 4 и 3 к прямоугольнику с меткой r7, получаем:

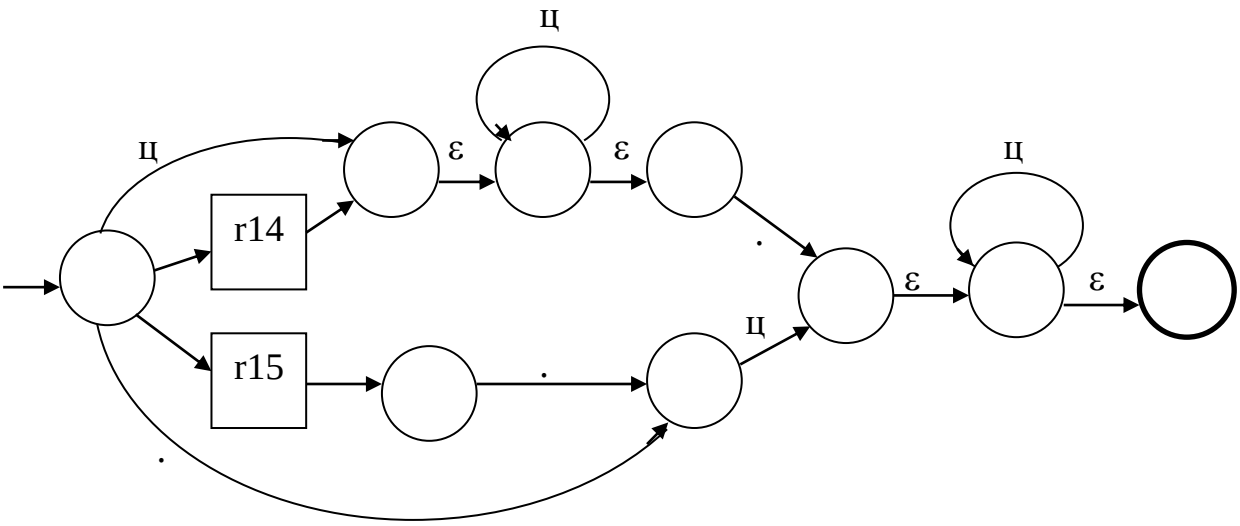


6. Регулярное выражение r9=ц + (+ + -) ц представим как r9=r13 + r14, где

r13=ц
r14=(+ + -) ц

Регулярное выражение r12= (+ + -) . представим как r12=r15r16, где
r15=+ + -
r16=.

Применяя правила 4 и 3 к прямоугольнику с меткой r9 и правила 5 и 3 к прямоугольнику с меткой r12, получаем:



7. Регулярное выражение r14=(+ + -) ц представим как r14=r17r18, где
r17=+ + -
r18=ц

Применяя правила 5, 4 и 3 к прямоугольнику с меткой r14 и правила 4 и 3 к прямоугольнику с меткой r12, получаем недетерминированный конечный распознаватель: