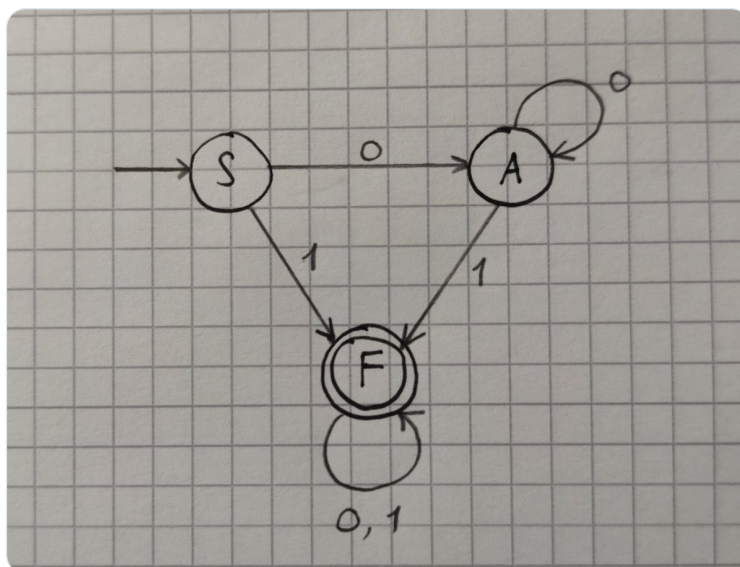


Валерий Жила @ValeriiZhyla

Oct 14, 2021 · 71 tweets · [ValeriiZhyla/status/1448572183354224641](https://twitter.com/ValeriiZhyla/status/1448572183354224641)



Я постараюсь рассказать о конечных автоматах так, чтоб и ребёнку стало понятно!



Конечные автоматы – они же Finite-state machines – одна из самых базовых и простых концепций Computer Science. Чтоб разобраться в ней, не нужно вообще никаких специальных знаний – достаточно знать буквы латинского алфавита (хотя бы парочку) и несколько цифр

Мотивацию, лирику и несколько уточнений оставим на конец треда. А сейчас – ближе к телу!

Автомат состоит из состояний и переходов между ними. Количество состояний ограничено, поэтому автомат и называется "конечным"

Состояние это именно то, чем кажется. Выключатель света (если он не застревает) может находиться только в двух состояниях – свет включен и свет выключен

Светофор может находиться (если он работает) в трех состояниях: зелёный, жёлтый, красный

Состояние автомата нужно понимать именно так, не строя дополнительных догадок и умозаключений. Это максимально базовая концепция, без оговорок.

Переходы между состояниями – реакция на какое-то событие, которая переводит автомат из состояния А в состояние Б. Внешнее событие (нажатие кнопки) переводит выключатель из состояния ВЫКЛ в состояние ВКЛ.

Внешнее событие (сигнал таймера в электросхеме) переводит светофор из состояния ЗЕЛЁНЫЙ в состояние ЖЁЛТЫЙ.

Состояния часто связаны между собой. Например, из одного состояния автомат может перейти не в любое состояние по желанию, а только по одному из предварительно заданных переходов

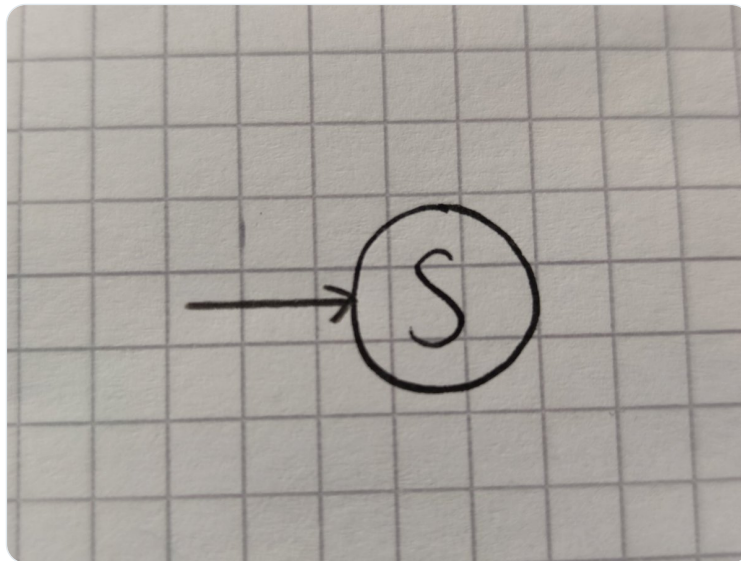
Светофор не может стать из зелёного красным, минуя жёлтый цвет (я абсолютно не уверен в этом факте, но допустим)

Так много текста, а пока ничего не понятно. Время картинок!

Итак, каждый автомат должен с чего-то начинаться. Это называется его начальным состоянием, и обозначается состоянием со стрелочкой, ведущей к нему извне (а не от другого состояния).

Такой состояние часто называют S (от слова Start)

Вот так вот:



У автомата может быть только одно начальное состояние (и, соответственно, только одна стрелочка "извне")

Когда мы начинаем работать с автоматом, он находится в этом, и только в этом состоянии, до получения первого сигнала для перехода в другое состояние

Когда электрики подключают светофор, первый цвет, которым он загорится может быть только зелёный (очередной выдуманный факт про светофоры)

Хорошо, стартовое состояние S у нас есть, но толку нам от автомата, который ничего больше не делает?

Согласен, толку немного. Предлагаю это исправить!

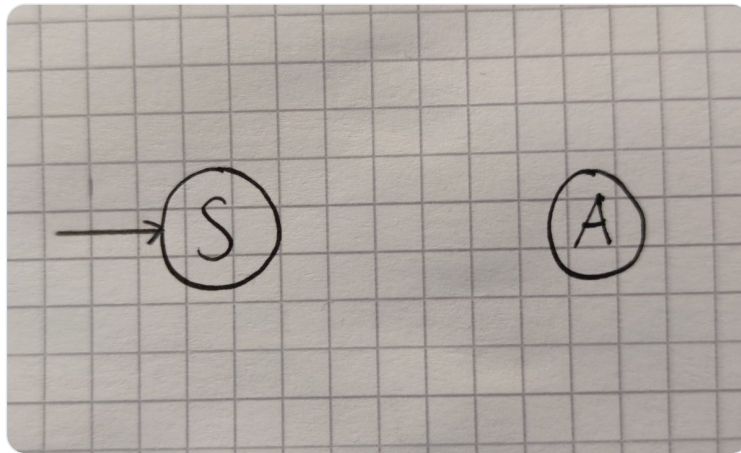
Добавим к нашему зародышу автомата ещё одно состояние – A.

В данный момент из S никак нельзя попасть в A.

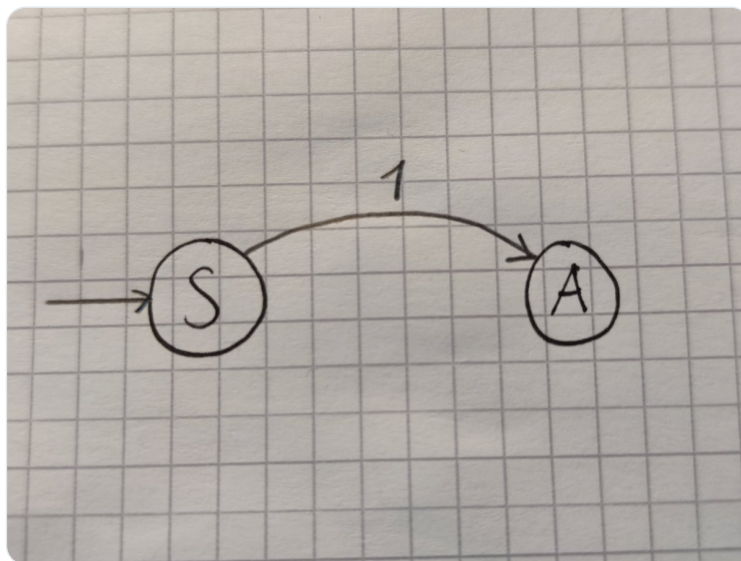
Как сказали бы серьезные бородачи и бородачихи – граф несвязный!

Как скажем мы – из S нельзя по стрелочка попасть в A, грусть печаль.

Исправим и это!



Итак, сейчас из S в A ведет стрелочка, но над ней появилась какая-то единица. Что это за зверь – пока не понятно, но понятно, что теперь из S в A можно как-то попасть!



Эта единичка – условие для перехода. Помните я говорил про внешние сигналы? Это они и есть!

Наш автомат перейдет в состояние A только в том случае, если получит единицу.

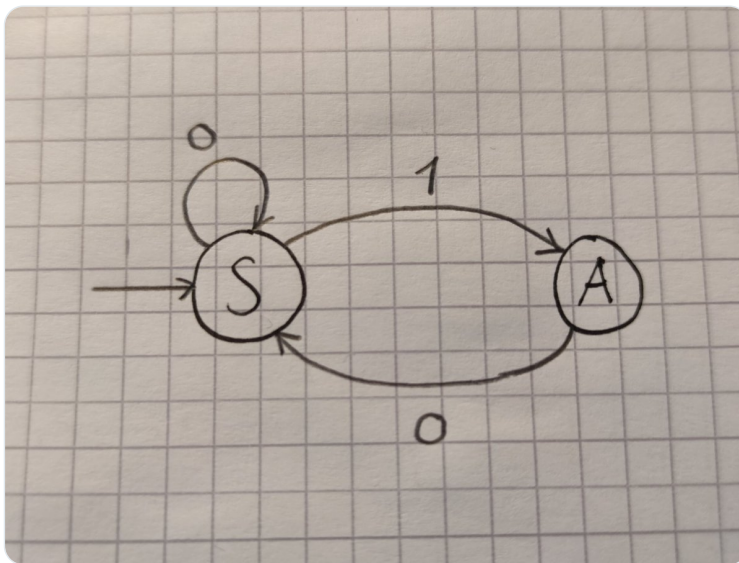
В зависимости от применения в реальной жизни, единица может быть нажатием определенной кнопки на кофемашине, или сигналом "следующий цвет, пожалуйста" от таймера в управляющий блок светофора

Очень часто в теории автоматов рассматривают только два сигнала – 0 и 1. В общем случае – их будет ровно столько, сколько создатель автомата захочет

Добавим в наш маленький автомат ещё и сигнал 0, а то с одним видом сигнала очень скучно работать (но возможно!)

"Ничего себе ты разогнался, товарищ, что стало с нашим прекрасным автоматом?!"

Спокойствие! Сейчас всё станет понятно)



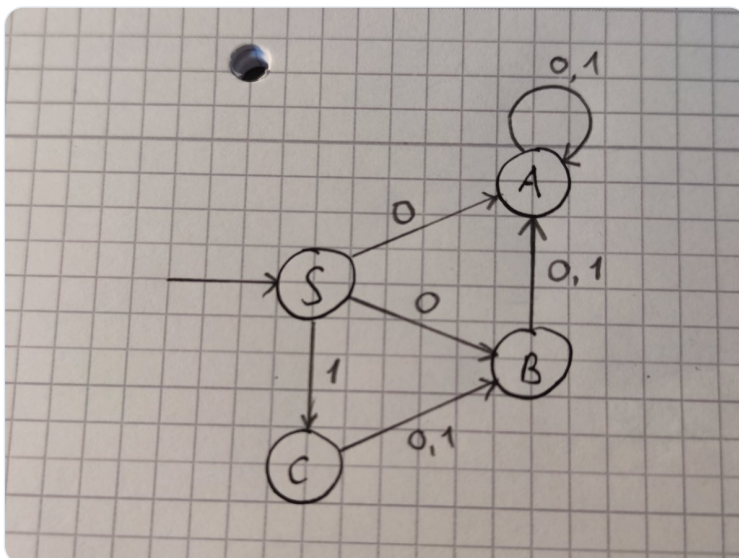
Новый переход из A в S по сигналу 0 вопросов вызвать не должен – он аналогичен переходу из S в A по сигналу 1. А вот новый переход из S в S по сигналу 0 может показаться странным, но это очень удобная штука.

Можно создать переход из любого состояния в любое состояние, так почему бы не перейти в себя?

Одно маленькое, но очень-очень важное условие – из каждого состояния автомата может быть ровно по одному переходу на каждый вид сигнала.

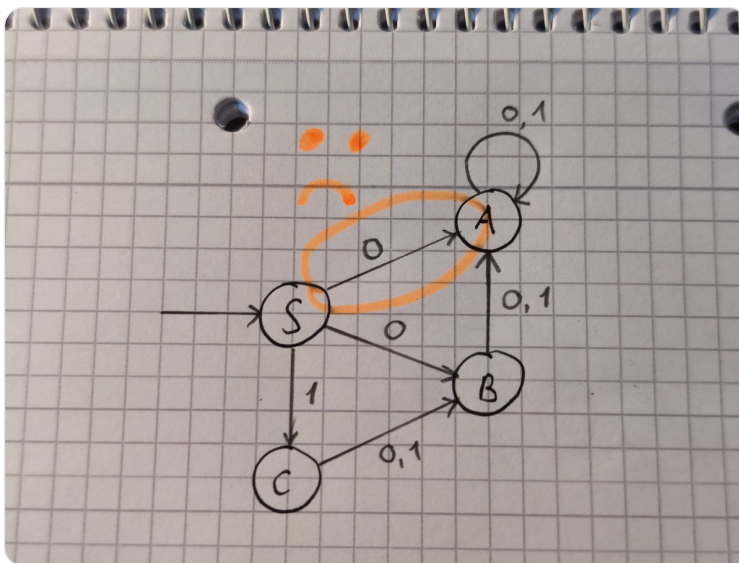
О том, что будет, если отбросить это условие, я расскажу в скором времени, в следующем треде по CS.

Это было небольшое отступление, покажу на примере:



...

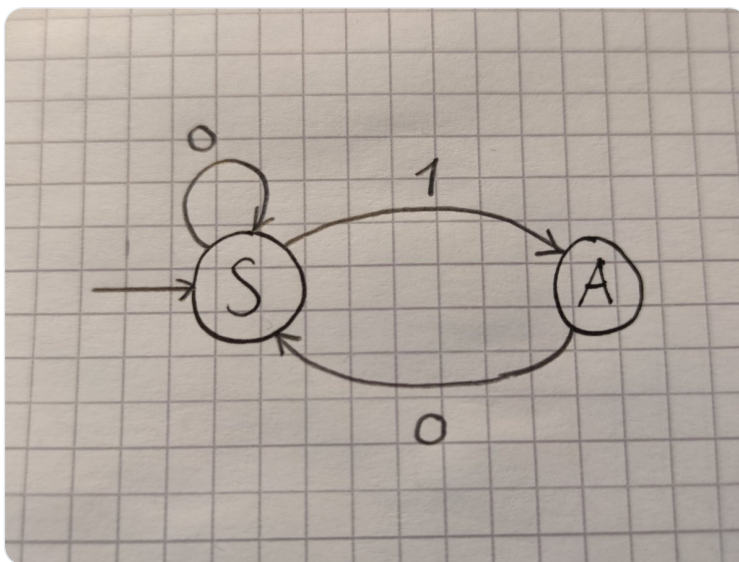
Этот автомат нарушает условие того, что из одного состояния по одному сигналу может быть только один переход. Из S ведут аж две стрелочки с нулём, одна в A, другая в B.



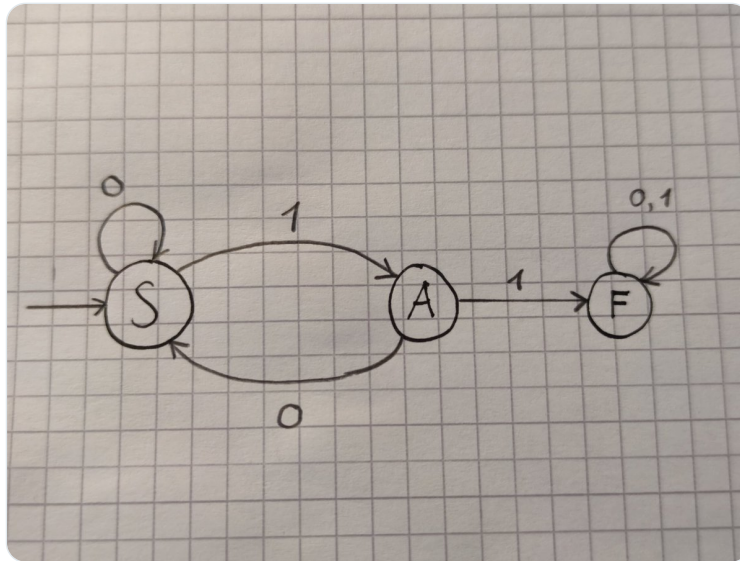
Так же тут можно увидеть, что из A в A ведет стрелочка с двумя сигналами. Так делать можно, для экономия места – это значит, переходим по сигналу 0 или по сигналу 1

Если сигнала всего 2, такой переход будет вызывать бесконечный цикл – автомат застрянет в состоянии A, и больше никак из него не выйдет

Вернемся к нашему простому автомату:



У него тоже нарушено одно правило – из A по сигналу 1 нет перехода. Исправим и это!



Теперь наш автомат соответствует всем критериям настоящего автомата! Такой автомат вполне подойдёт в планировании состояний какого-то объекта в программировании, или для того, чтоб сделать светофор, или описать какую-то схему, описать цикл деления животную клетки и так далее

Вот только для нужд науки, и особенно Computer Science – этого маловато.

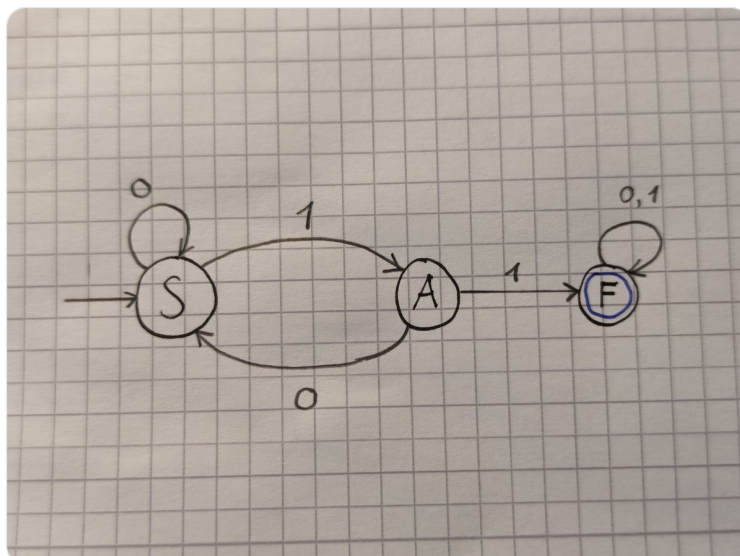
Не хватает одной маленькой детали – конечного состояния.

Остановка автомата в конечном состоянии - это момент, когда автомат говорит: "я переварил по очереди все входящие сигналы, пощёлкал состояниями, начиная с S и далее по списку, и выдаю свой вердикт – ДА!"

Остановка автомата в обычном состоянии (а не конечном) – "Вердикт: НЕТ!"

Возможно, это немножко сложно, но я покажу на примере, и все станет ясно. Это уже почти финиш, обещаю)

Конечное состояние обозначается двумя кружочками, и часто носит название F (от слова Finish). Вот как выглядит наш знакомый автомат, которому дали конечное состояние (у F теперь кружок в кружке, я выделил его синим цветом):



Что будет делать наш автомат теперь? Он скажет ДА последовательности сигналов (так называемому "слову") которое содержит два сигнала 1 в самом конце. И НЕТ в любом другом случае. Проверьте сами!

Программисты скажут – "это очень напоминает регулярные выражения..." И попадут в точку!

Конечные автоматы как раз и описывают регулярные выражения. Есть доказанная теорема: для каждого конечного автомата можно создать регулярное выражение, которое полностью его описывает, и наоборот. Они "эквивалентны"

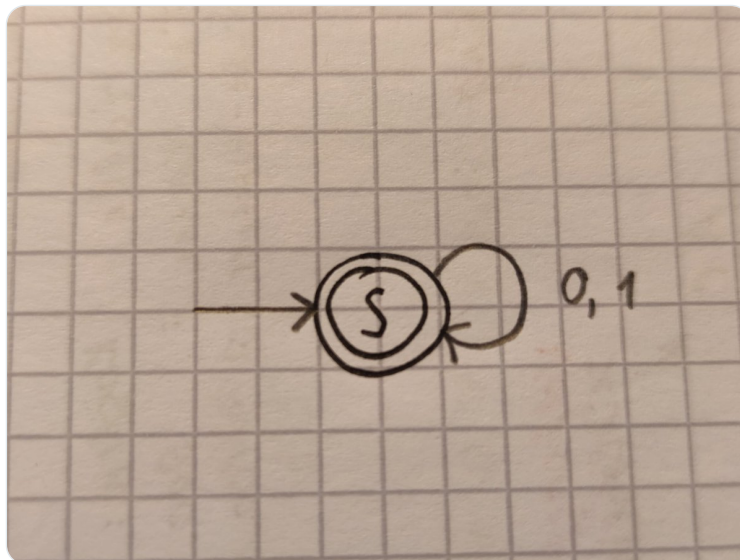
Но это отступление для тех, кто знает, что такое регулярные выражения, и к теме не относится)

Конечных состояний (кружок в кружке, остановка в нем - значит ответ ДА), в отличии от стартового состояния может быть несколько. Все состояния автомата могут быть конечными – и тогда этот автомат будет говорить ДА на любую последовательность сигналов

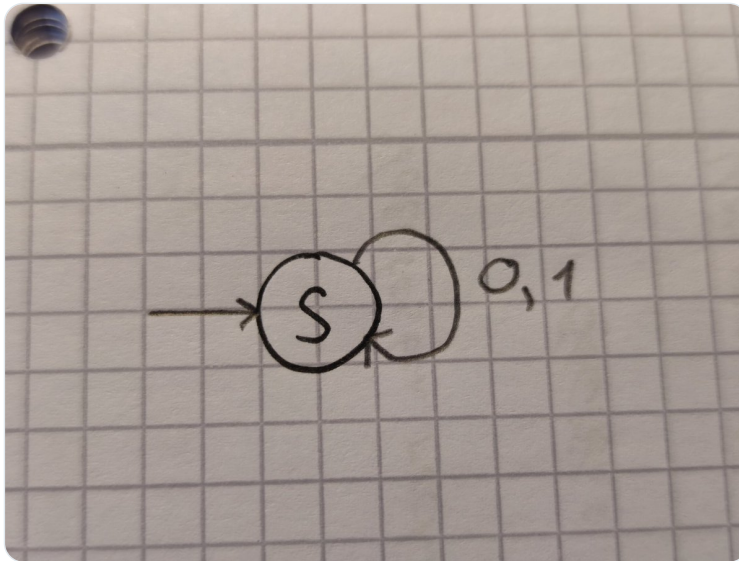
Рассмотрим маленький пример на эту тема. Даже два примера!

Начальное состояние имеет такие же права и обязанности, как остальные – и может быть конечным!

Этот автомат говорит ДА после обработки любой последовательности сигналов, и даже если сигналов вообще нет

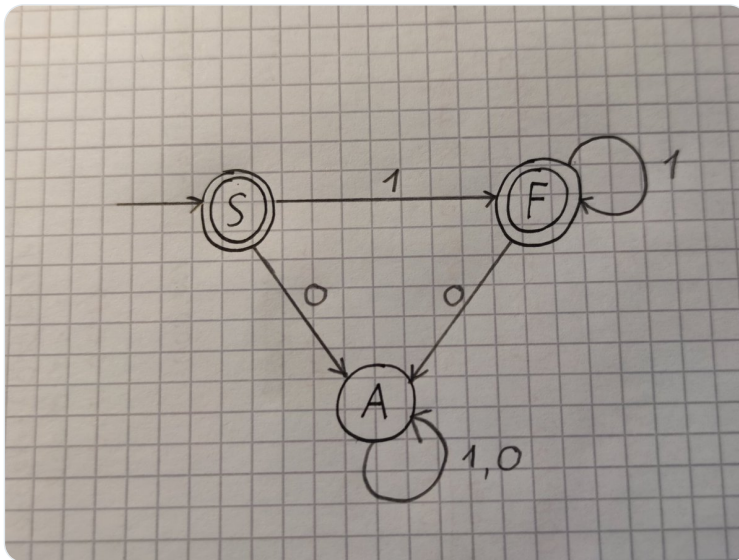


А вот его злой брат-близнец: он говорит НЕТ абсолютно любой последовательности сигналов – какой бы она не была, автомат остановится в обычном, а не в конечном состоянии



На закуску, давайте напишем чуть более осмысленный автомат

Вот какой красавец! Что он делает? Он принимает любую последовательность сигналов (и даже пустую последовательность) и говорит ДА, если она состоит из любого количества единиц



Но стоит затесаться там хотя бы одному нулю – автомат улетит в "мусорное состояние" А, из которого выхода нет.

нужно →

0						✓
1						✓
1	1	1	1			✓
1	1	1	0	1		✗
0	0	0	1	1		✗

Любой автомат можно задать как в графической форме (как мы делали), так и в форме таблицы переходов

Вот пример таблицы переходов для нашего последнего автомата

переходы

Состояние	Конечн?	0	1
S	✓	A	F
A	✗	A	A
F	✓	A	F

Start: S

Поздравляю! Теперь вы имеете довольно хорошее представление о конечных автоматах. Теперь пойдут научные оговорки и мотивация ко всему этому

Оговорка раз:

Мы рассмотрели конечные автоматы, у которых количество состояний конечно. Существуют так же бесконечные автоматы – но это намного сложнее и имеет околонулевую практическую значимость – наша Вселенная конечна, и количество частиц в ней тоже

Количество конфигураций этих частиц тоже (скорее всего, я не физик ни на секунду) конечно.

Оговорка два: мы рассмотрели один из двух основных видов конечных автоматов – автомат Мура. Существуют ещё автоматы Мили. Автоматы Мура и Мили эквивалентны – и свободно переводимы из одного вида в другой – но автоматы Мили используются намного реже.

Оговорка три: помните условия, что автомат однозначно реагирует на сигналы, и не имеет нескольких переходов из одного состояния в другое для одного сигнала?

Это условие называется детерминизмом, то есть сегодня мы рассмотрели детерминистичные конечные автоматы.

Если это условие откинуть, то автоматы будут, соответственно, недетерминистичными. Как раз о них пойдет речь в следующем треде про Computer Science.

Теперь про мотивацию, и где автоматы вообще используются. Просто ответ: да везде, всё, что имеет какие-либо функции, является автоматом в широком смысле этого понятия. Даже человеческий организм – это огромный набор простых и сложных детерминистичных конечных автоматов

Не буду ничего говорить тут про нервную деятельность, там всё немного сложнее

Мотивация для программистов: как я уже сказал, автоматы – это регулярные выражения, они же регексы. Автоматы широко применяются в архитектуре ПО.

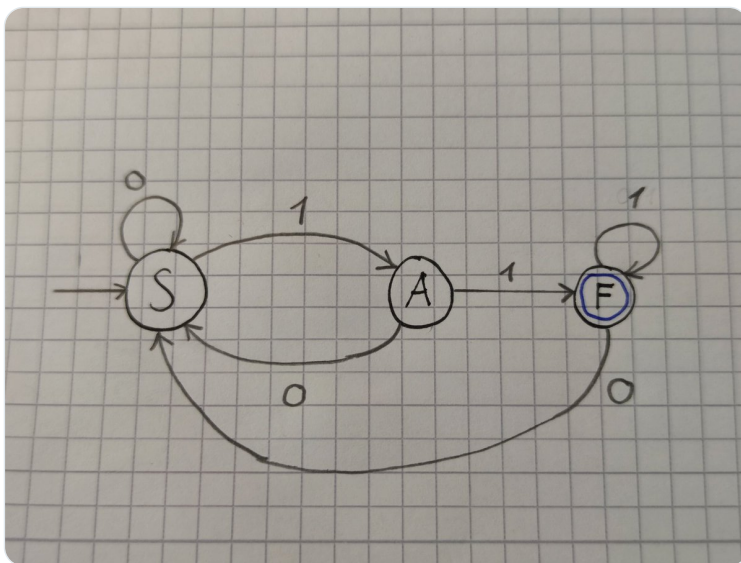
Они используются в самих языках программирования – синтаксические и лексические анализаторы, которые выдают вам Syntax Error или Compilation Time Error – реализованы именно как конечный автомат

На этом я заканчиваю эту гигантскую историю. Надеюсь, большая часть была всем понятна. Если остались вопросы – задавайте, постараюсь на всё ответить)

[@threadreaderapp](#) unroll

Как мне тут уже правильно заметили, я писал одно, а думал про другое. Этот автомат говорит ДА не в случае с двумя сигналами 1 в конце, а в случае с двумя 1 подряд в любом месте

Вот пример правильного автомата, говорящего ДА только на последовательности сигналов с минимум 2 единицами в конце



Спасибо [@lexx_it](#) за внимательность!