***Задача 1.***Описать функционирование автомата, который запоминает один символ (или осуществляет задержку сигнала).

***Таблица***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Состояние автомата до входного сигнала  (то, что он «помнил») | Входной сигнал | Состояние автомата  после входного сигнала (запоминает поступивший сигнал) | Выходной сигнал (автомат выдает  то, что он «помнил» до входного сигнала) | ***Граф*** |
| 0 | 0 | 0 | 0 | Граф элементарного запоминающего автомата |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 |

***Задача 2.***Опишите конечный автомат проверки на четность числа подающихся на вход единиц, если, как и в предыдущем примере, входной и выходной алфавиты состоят из 0 и 1, причем выходной сигнал 0 означает, что число всех поступивших на вход 1 было четным, а 1 - нечетным. Сколько состояний имеет придуманный конечный автомат. Является ли число состояний минимальным? Представьте придуманный автомат в виде таблицы и в виде графа.

### Работа конечного автомата

Рассмотрим протокол работы запоминающего автомата. Начальным его состоянием естественно считать 0. Пусть автомат работает на следующей последовательности входных сигналов: 11001.

Заполним таблицу протокола работы автомата:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Время | 0 | 1 | 2 | 3 | 4 |
| Входной сигнал | 1 | 1 | 0 | 0 | 1 |
| Состояние |  |  |  |  |  |
| Выходной сигнал |  |  |  |  |  |

В начале выходной последовательности появился 0, который хранился в памяти автомата перед началом его работы.

Обратите внимание на задержку сигнала автоматом и на потерю на выходе последней 1 (она сохраняется в памяти автомата).

***Задача 3.*** В условиях предыдущего упражнения напишите протокол работы построенного конечного автомата на последовательности 0101100111.

***Задача 4.* Самообучающийся автомат Клода Шеннона для игры в чет-нечет**

Однажды кибернетики решили посоревноваться в искусстве программирования и договорились сделать автоматы для игры в чет-нечет и устроить между ними турнир. Чей автомат выиграет, тот и будет самым искусным кибернетиком.

В день турнира все соревнующиеся принесли большие автоматы, а Клод Шеннон принес маленький. В современных терминах это означает, что его программа была небольшой и изящной. Началось соревнование. Каждая партия протекала так: противники называли 0 или 1, если у обоих автоматов цифры оказывались равными, выигрывал один из них (по предварительной договоренности их создателей), если разные - другой. После этого ходы противников и результат партии (1- выиграл, 0 - проиграл) вводились в автоматы. Это делалось для того, чтобы у автоматов была возможность анализа сыгранных партий и учета полученных результатов в следующих партиях. Когда турнир закончился, оказалось, что с завидным преимуществом выиграл автомат Клода Шеннона.

Кибернетикам не терпелось узнать, что же придумал Клод Шеннон. Оказалось, что его автомат имел две стратегии: одна заключалась в том, что называлось число, совпадающее с ходом противника в предыдущей партии, вторая - в том, что называлось противоположное число. Все обучение заключалось в правиле переключения с одной стратегии на другую. Если в очередных трех партиях автомат одерживал две или три победы, он сохранял стратегию, если нет - менял.

Проанализируем автомат Клода Шеннона. Если считать, что переключение стратегий делает не сам автомат, а, например, его создатель, то автомат Клода Шеннона можно представить в виде двух простых конечных автоматов. Один - просто осуществляет задержку сигнала, то есть передает входной сигнал на выход в следующем такте работы, второй - инвертирует сигнал.

*Электронную гадалку, которую мы сейчас опишем, придумал создатель теории информации К. Шеннон. Работает она так. Человек пишет на бумаге число 0 или 1. Машина этого числа не знает, но печатает 0, 1 или 2. Двойка означает, что машина не берется угадать написанное число, а 0 или 1 — ее предположение о написанном числе. После этого человеку сообщают предположение машины, а в машину вводят число, написанное человеком.*

*Вначале машина играет неважно, но после двух-трех десятков проб начинает угадывать в 90% случаев, сколько бы человек ни пытался ее запутать. Это производит впечатление.*

*Устроена программа так. В ней имеется 5-индексный массив A[0:1, 0:2, 1:1, 0:2, 0:1]  из 72 элементов. Вначале массив очищен нулями, и машина первые три раза печатает двойки. В дальнейшем машина помнит несколько последних ходов своих и человека. Если человек последними написал числа a1, a2, a3  и машина на это отвечала b1, b2, b3, то в ячейку A[a1, b1, a2, b2, a3] добавляется единица, то есть машина запоминает, что после комбинации a1, b1, a2, b2 человек выбрал число a3. Чтобы предсказать, что теперь напишет человек, машина сравнивает числа A[a2, b2, a3, b3, 0] и A[a2, b2, a3, b3, 1]. Если первое сильно превосходит второе, то предсказывается число 0, если наоборот, то число 1, а если они отличаются мало, то печатает число 2, то есть отказывается угадывать. Можно усовершенствовать программу, добавляя на ходе i в нужную ячейку не единицу, а число (1.1)i, и тем самым уменьшая вес старых событий, которые человек успевает забыть.*

*Запрограммируйте гадалку так, чтобы цифры, написанные человеком, и цифры, предсказанные машиной, располагались на экране парами и чтобы человек видел последние 10-20 пар. Показывайте все время на экране текущий процент верных угадываний. Испытайте вариант гадалки, не учитывающий своих предсказаний, но зато руководствующийся более длинными сериями чисел человека.*

*Если бы человек определял свои числа бросанием монеты или с помощью случайных чисел, то программа не смогла бы угадать заметно более 50% чисел. Но человек не умеет задавать числа случайно, и электронная гадалка расшифровывает его тактику или психологию.*

***Задача 5.***

Напишите программу для игры в чет-нечет, моделирующую автомат Клода Шеннона. Попробуйте модернизировать его, меняя число стратегий (например, возможна стратегия "называть всегда 1") и число партий, после которых осуществляется переключение стратегий. Устройте соревнование между различными версиями программ и определите лучшую из них.