Для работы необходим текстовый файл с прошлого шага, в нём содержится информация о кол-ве состояний в исходном и новом КА, кол-ве различных переходов. В текущей реализации мы представляем, что сигналы для переходов – возрастающая числовая последовательность двоичных чисел. Второй строчкой получаем выход из SAT-решателя. В третьей передаётся информация о выходах с новых состояний.

Пока программа умеет генерировать синтезируемый КА Мура с использованием 2х @always блоков в языке SystemVerilog.

Код генератора, используя информацию об исходных данных и выходах SAT-решателя, преобразует функцию двоичной логики в читаемую информацию(**словарь**: ключ - новое состояние, value – массив со связями между другими состояниями). По новому представлению информации генерируется шапка и инструкция переходов исследуемого КА на языке SystemVerilog.

def traceToState(state) – связывает каждое новое состояние с к его выходами и условиями переходов для удобного предствления информации о генерировании КА.

def decTobin(dec) - конвертер двоичных чисел, понятных языку verilog необходим, потому что формат двоичного представления чисел в языке python3 отличается от необходимого.

def create\_state(state): - модуль генерирующий код описывающий каждое новое состояние, проходя последовательно по **словарю**.