

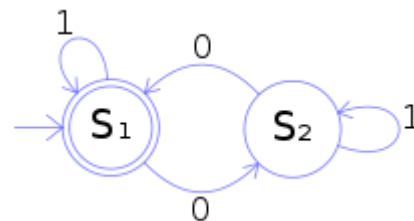
Theorievragen:

1. Dit tabel heb ik van de gekregen Wikipediabron gehaald.

Automaton hebben verschillenden talen, een taal is de lijst van woorden die de automata accepteert, niet alle automata zullen dezelfde lijst van woorden hebben daarom hebben ze een andere taal.

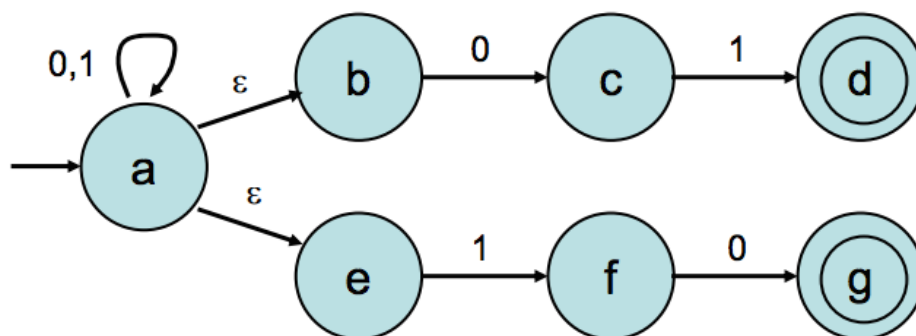
Automaton	Recognizable language
Nondeterministic/Deterministic Finite state machine (FSM)	regular languages
Deterministic pushdown automaton (DPDA)	deterministic context-free languages
Pushdown automaton (PDA)	context-free languages
Linear bounded automaton (LBA)	context-sensitive languages
Turing machine	recursively enumerable languages
Deterministic Büchi automaton	ω-limit languages
Nondeterministic Büchi automaton	ω-regular languages
Rabin automaton, Streett automaton, Parity automaton, Muller automaton	

2. Een deterministisch automata moet voor elke input een transition hebben. 'There must be exactly one transition function for every input symbol in Σ from each state.'¹ Bij een niet deterministisch automata moet dat niet. 'Unlike DFAs, NDFAs are *not* required to have transition functions for every symbol in Σ , and there can be multiple transition functions in the same state for the same symbol.'²



Een deterministisch automata heeft deze form:

Een niet deterministisch automata heeft deze form:



¹ Finite State Machines. *Brilliant.org*. Retrieved 14:01, November 11, 2020, from <https://brilliant.org/wiki/finite-state-machines/>

² Finite State Machines. *Brilliant.org*. Retrieved 14:05, November 11, 2020, from <https://brilliant.org/wiki/finite-state-machines/>

DFA vs NDFA

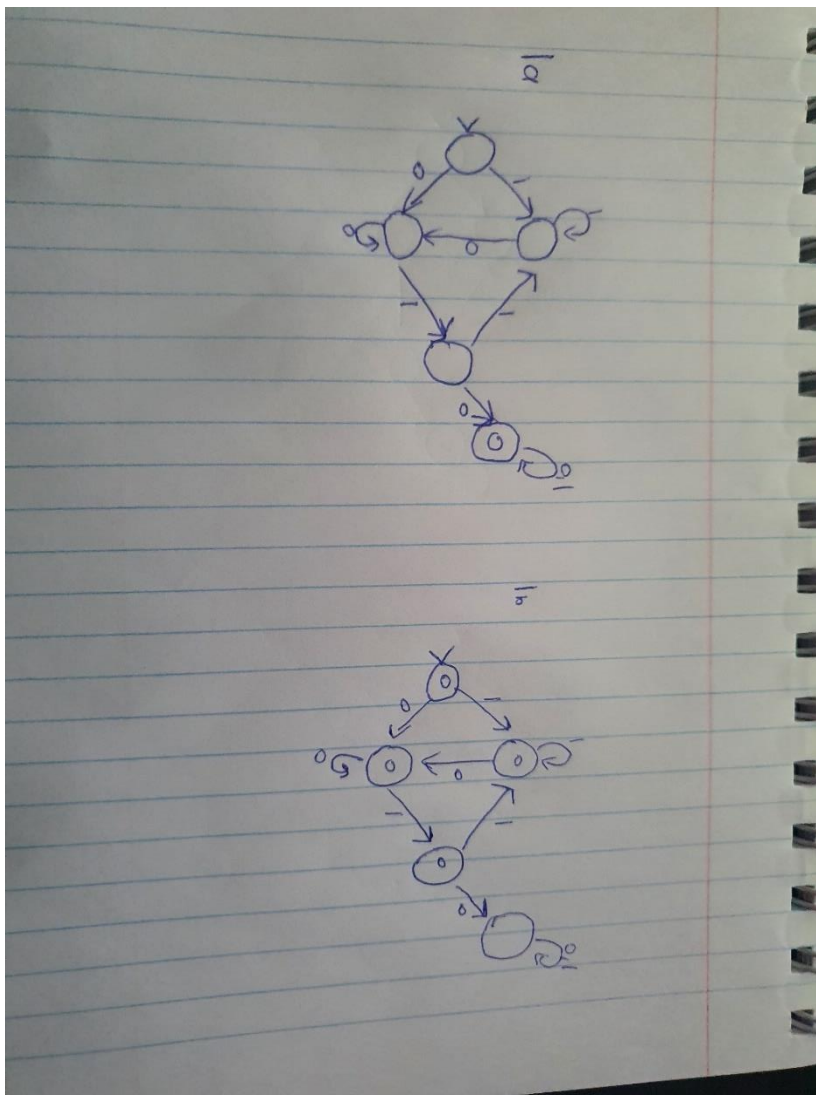
The following table lists the differences between DFA and NDFA.

DFA	NDFA
The transition from a state is to a single particular next state for each input symbol. Hence it is called <i>deterministic</i> .	The transition from a state can be to multiple next states for each input symbol. Hence it is called <i>non-deterministic</i> .
Empty string transitions are not seen in DFA.	NDFA permits empty string transitions.
Backtracking is allowed in DFA	In NDFA, backtracking is not always possible.
Requires more space.	Requires less space.
A string is accepted by a DFA, if it transits to a final state.	A string is accepted by a NDFA, if at least one of all possible transitions ends in a final state.

3

Automata:

1. 1a is de automata die de substring 010 bevatten, 1b is het tegenovergestelde.

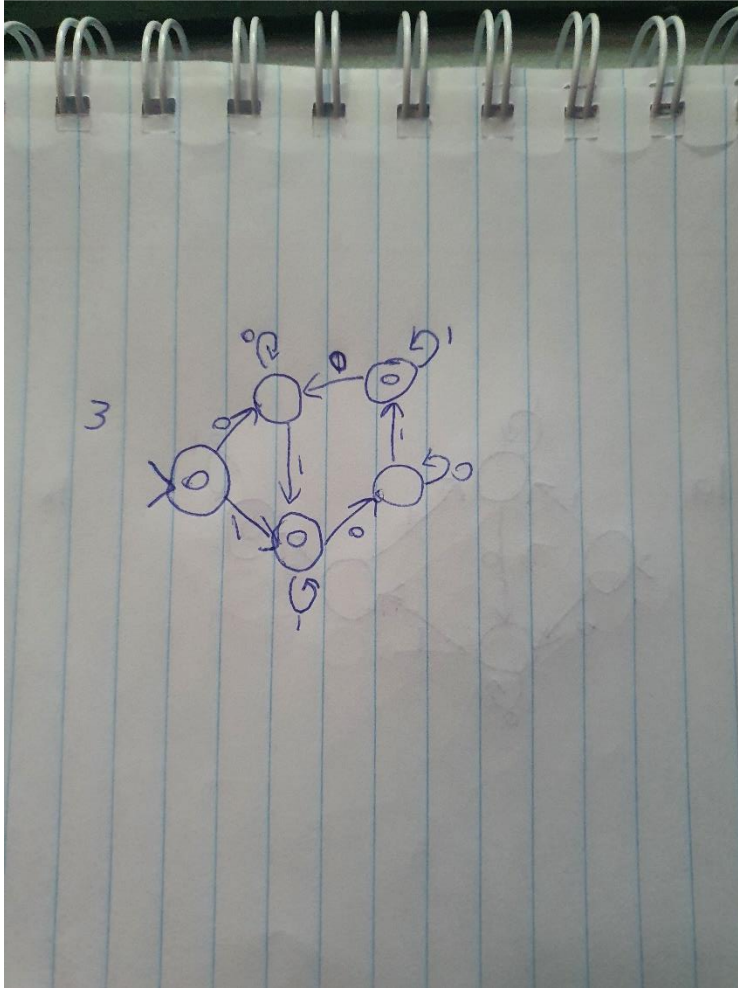


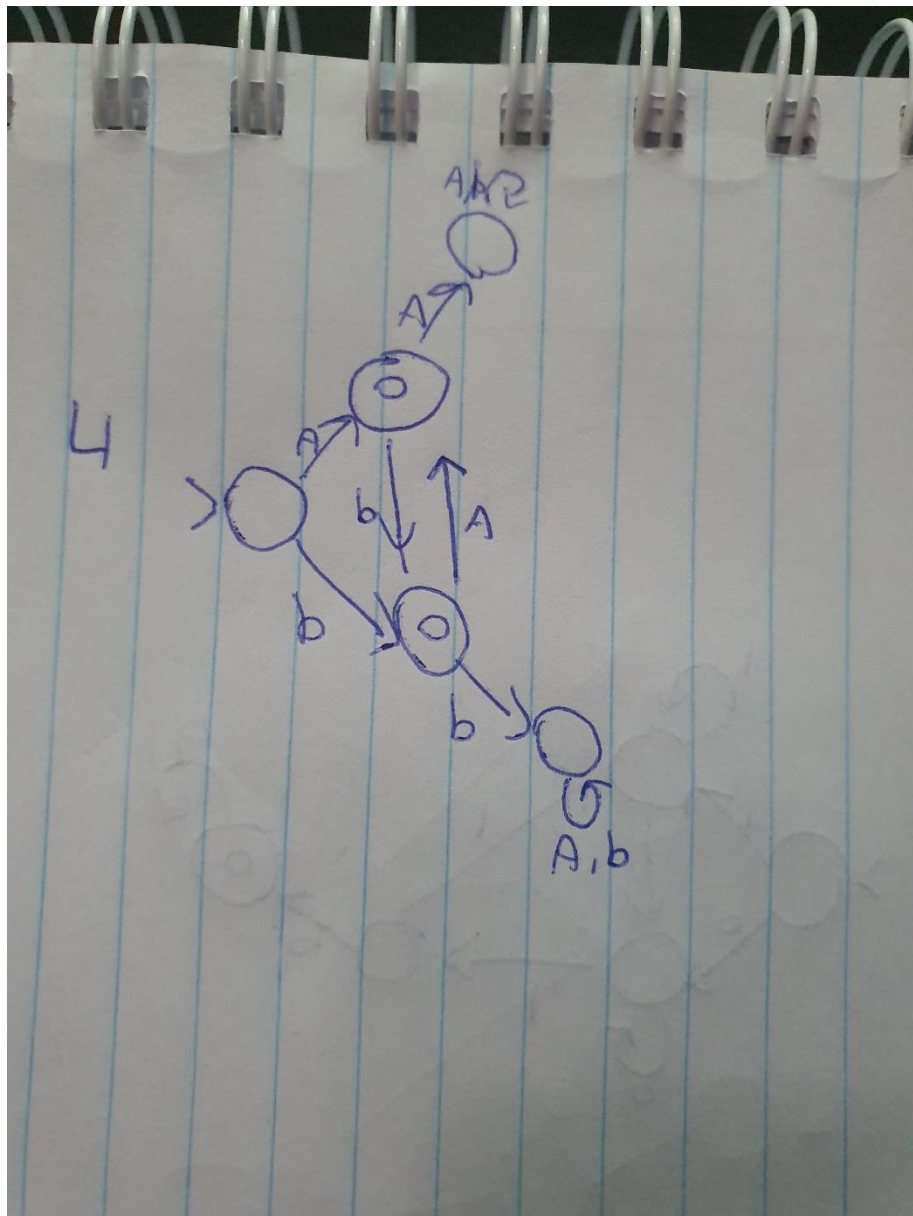
³ https://www.tutorialspoint.com/automata_theory/non_deterministic_finite_automaton.htm

2. Snap niet wat bedoeld wordt met 'beschrijf welke talen deze beschrijven', willen jullie weten wat ze doen? Of bij welke van de talen uit de eerste theorievraag ze horen? Beide dan maar.

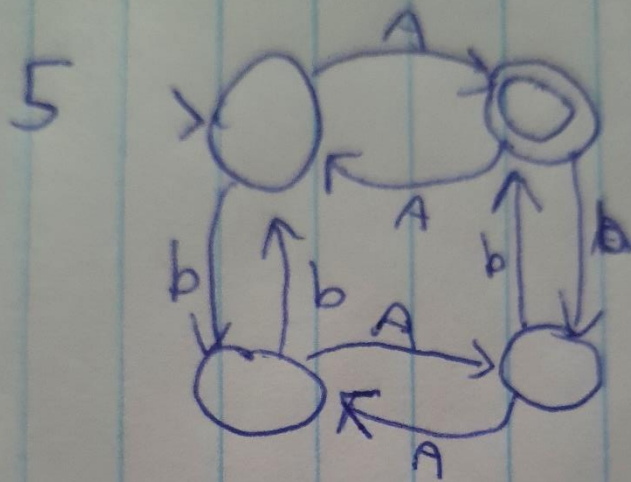
De linker van de twee die accepteert alleen sub strings die beginnen en eindigen op een A en geen twee A's en B's achter elkaar. Bijvoorbeeld ababababa. In reguliere expressie omdat oneven letters zijn A even zijn B

De rechter accepteert alleen als de sub string alleen een B is of als de sub string eindigt na A's op één B. Voorbeelden: B, AAAAB, AAAAAAAAAAAB. Ook in reguliere expressie.





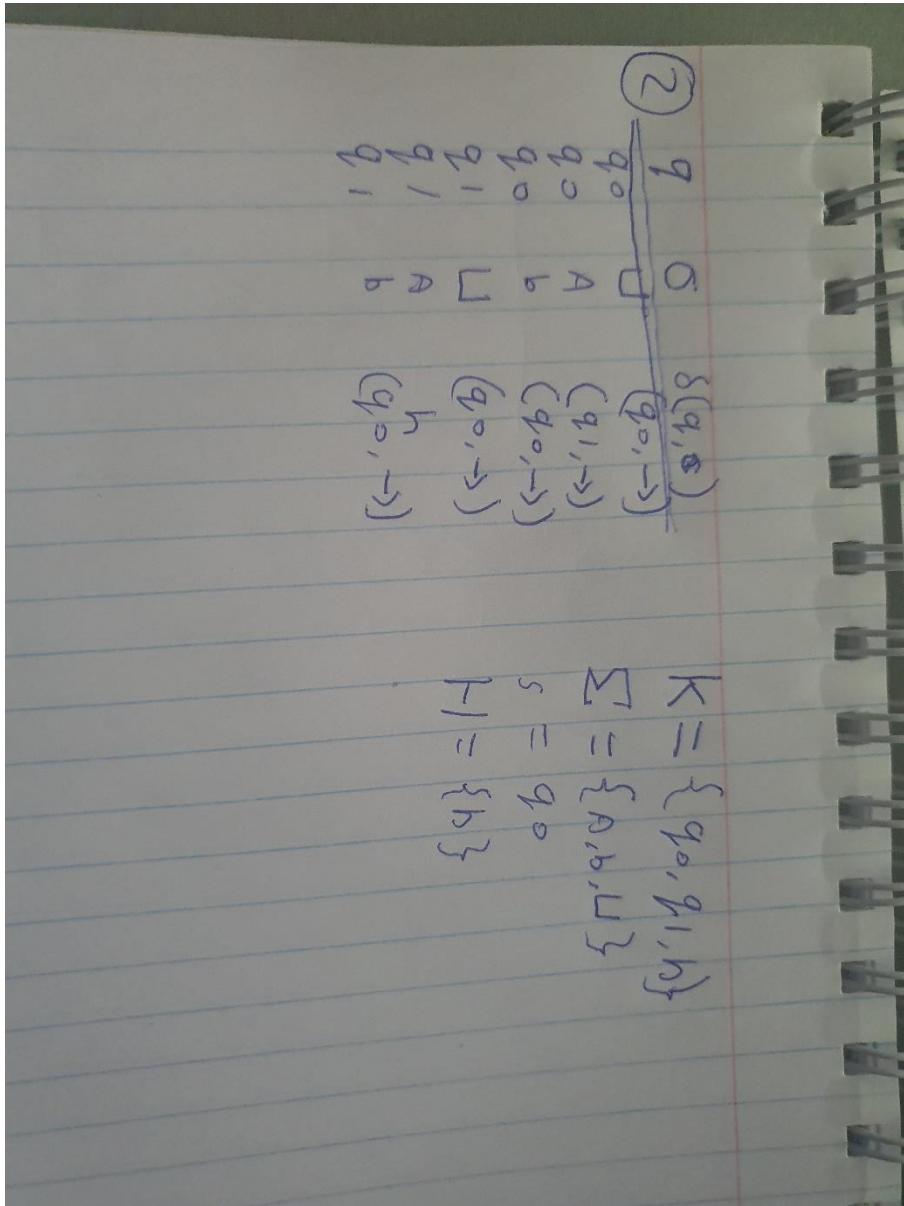
4.



5.

Turing Machines:

1. De machine kijkt waar de eerste A is en de dichtstbijzijnde B links van die A. dan Hold



- 2.
3. Deze begrijp ik (nog) niet, er wordt in de opdracht een voorbeeld gegeven met AABBB. Als we nu even dat voorbeeld pakken, wat willen jullie als uitkomst hebben? ABBu (u = blank) of ABBA (dat de eerste input nu achter aan staat). Het uitwerken lukt voor beide niet dus maakt niet uit.
4. Deze machine gaat naar rechts totdat het een blank tegenkomt, dan gaat hij terug totdat hij een 1, 0 of een blank tegenkomt. Maar aangezien dat het hele alfabet is gaan hij maar ééntje naar links. Als dat een 1 is wordt het een 0, als het een 0 is wordt het een 1 en als het een blank is wordt het een 1. Als het een 0 wordt dan gaat hij weer terug naar het links gaan gedeelte. Dus dan gaat hij weer een naar links. Binaire rekenen is de functie.

Aanpassing: De aanpassing doet precies hetzelfde maar dan naar links en dan rechts. Foto: volgende bladzijde.

