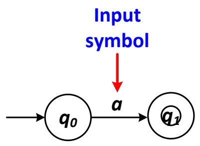
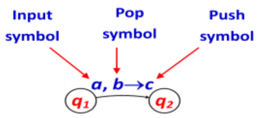
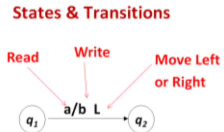
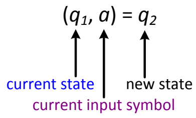
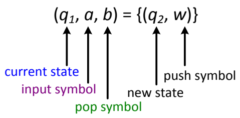
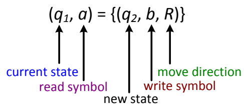
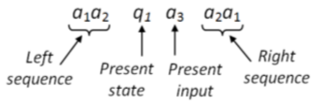
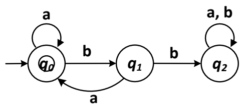
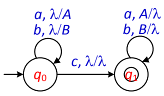
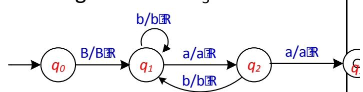
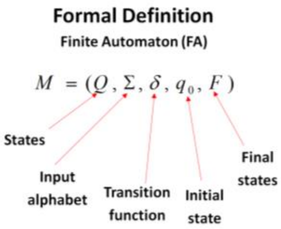
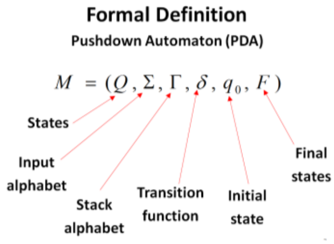
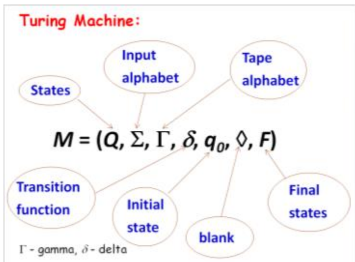
Table 6.5 Comparison of the differences between FA, PDA and TM

Issues	**FA**	**PDA**	**TM**
Label of the transitions in the state diagram	One symbol - input symbol 	Three symbols - input, pop and push symbols 	Three symbols - read, write and move direction symbols 
Transition functions	$\delta(q_1, a) = q_2$ • It means that with input a, move from state q_0 to state q_2 • The first two (q_1, a) are input. • The last one (q_2) is output. 	$\delta(q_1, a, b) = \{(q_2, w)\}$ or $\delta(q_1, a, b) = \{(q_2, w), (q_3, w)\}$ • It means that – From state q_1. – Read *a* from the tape, – Pop the string *b* from the stack, – To state q_2, – Push string *w* onto the stack. • The first three (q_1, a, b) are input. • The last two (q_2, w) are output. 	$\delta(q_1, a) = \{(q_2, b, R)\}$ • It means that – From state q_1. – Read *a* from the tape, – Write *b* to the tape, – To state q_2, – Move to right of the tape. • The first two (q_1, a) are input. • The last three (q_2, b, R) are output. 
Configurations / Instantaneous Description	Represented with the ordered pair $[q_i, s]$ where • $q_i \in Q$ is the machine's current state; • $s \subseteq w$ and $w \in \Sigma^*$ is the remaining unprocessed input.	• Represented with a triple $(p, w, \alpha) \in (K, \Sigma^*, \Gamma^*)$ where • *p* is the current state • *w* is the remaining input • α is the current stack contents	Denoted uq_ivB where *B* is blank symbol, all tape positions, to the right of the B are blanks and *uv* is the string spelled by the symbols on the tape from the left-hand boundary to the *B*. 
Computations	The FA M_1:  The computations of M with input strings *abba* is $[q_0, abba] \vdash [q_0, bba]$ $\vdash [q_1, ba]$ $\vdash [q_2, a]$ $\vdash [q_2, \lambda]$. accepts	The PDA M_2:  The computation of M_2 with input *abcba* is $[q_0, abcba, \lambda] \vdash [q_0, bcba, A]$ $\vdash [q_0, cba, BA]$ $\vdash [q_1, ba, BA]$ $\vdash [q_1, a, A]$ $\vdash [q_1, \lambda, \lambda]$. accepts	The Turing machine M_3:  accepts the language $(a \cup b)^*aa(a \cup b)^*$. The computation of M_3 with input *aabb* is $q_0BaabbB \vdash Bq_1aabbB$ $\vdash Baq_2abbB$ $\vdash Baaq_3bbB$. accepts
Determinism	Both deterministic (DFA) and non-deterministic (NFA) • Every state of DFA always has exactly one exiting transition arrow for each symbol in the alphabet while the NFA may has more. • In a DFA, labels on the transition arrows are from the	Non-deterministic only • PDAs are non-deterministic. Allowed non-deterministic transitions - Multiple transitions on same pop/input, transitions may but do not have to push or pop.	Deterministic only • Turing machine are deterministic. No lambda transitions allowed.

	alphabet while NFA can have an arrow with the label ε.		
Formal definitions (how many member of the tuple? What they are?)	A FA is a 5-tuple (Q, Σ, δ, q_0, F). 	A PDA is a 6-tuple (Q, Σ, Γ, δ, q_0, F). 	A TM is a 7-tuple (Q, Σ, Γ, B, δ, q_0, F). 
Acceptance criteria	FA accepts *w* if the machine end up in a final state.	PDA accepts *w* if the machine end up in a final state with an empty stack.	A string is accepted by final state if the computation halts in a final state, but the TM need not read the entire input string to accept the string.
Example of state diagram for the language *aa**