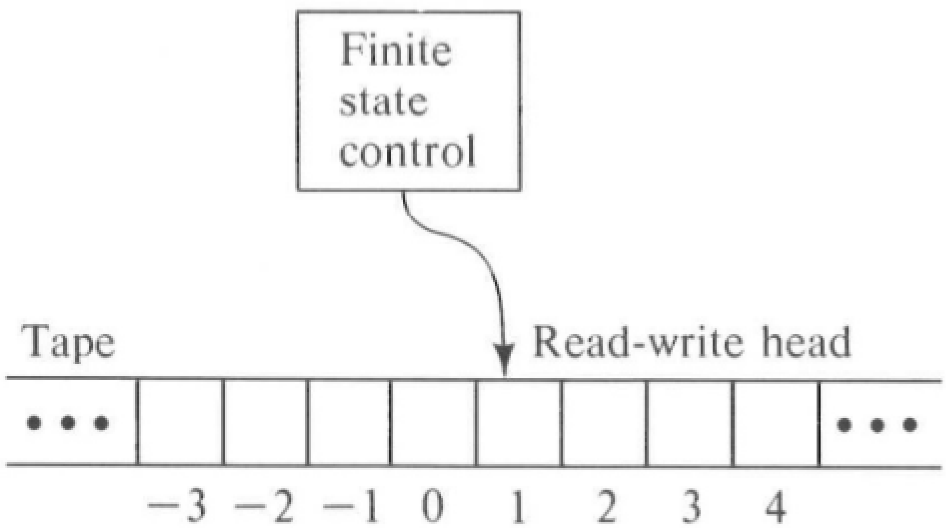
DTM and NDTM

Deterministic Turing Machine (DTM)

- In order to formalize the notion of an algorithm, we will need to fix a particular model for computation.
- We will consider the deterministic one-tape Turing machine (DTM), which consists of finite state control, a read-write head and a tape made up of a infinite sequence of squares (labeled with integers for convenience).



- A program for DTM consists of:
  - A finite set Γ of tape symbols
  - A subset of tape symbols, called input symbols $\Sigma \subset \Gamma$
  - A blank symbol $b \in \Gamma \land b \notin \Sigma$
  - A finite set of states Q including a start-state q_0 and two halt states (finish states) q_Y and q_N.
  - A transition function $\delta : (Q - \{q_Y, q_N\}) \times \Gamma \to Q \times \Gamma \times \{-1, 1\}$ (a function taking all states except two halts states and any tape symbol and

returning a state, a tape symbol and a plus or minus one)

The DTM procedure:

- The input to DTM is a string of input symbols $x \in \Sigma^*$ placed on tape squares 1 to $|x|$. All other tape squares contain the blank symbol b.
- The program starts in state q_0 and ends when the state is either q_Y (answer yes), or q_N (answer no)
- The read-write head starts scanning at square 1.
- Then the procedure is:
  1. Take the symbol $s \in \Gamma$ which is currently under the read-write head
  2. Take the current state of the program $q \in Q$
  3. Then the transition function gives us $\delta(q, s) = (q', s', \Delta)$
  4. Erase the current symbol s under the read-write head and write the new symbol s'
  5. Move one square to the left if $\Delta = -1$, or one square to the right if $\Delta = 1$.
  6. Change the current state from q to q'
  7. Repeat until the current state is either q_Y or q_N.

- Example of a program for DTM:

$$\Gamma = \{0, 1, b\},\ \Sigma = \{0, 1\}$$

$$Q = \{q_0, q_1, q_2, q_3, q_Y, q_N\}$$

q	0	1	b
q_0	$(q_0, 0, +1)$	$(q_0, 1, +1)$	$(q_1, b, -1)$
q_1	$(q_2, b, -1)$	$(q_3, b, -1)$	$(q_N, b, -1)$
q_2	$(q_Y, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$
q_3	$(q_N, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$

$\delta(q, s)$

Nondeterministic Turing Machine (NDTM)

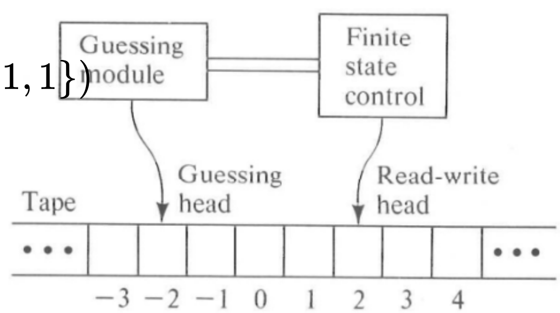
- Nondeterministic Turing Machine is generally used to define what a class of computational problems such that it is possible to "verify" that a solution to a problem indeed solves this particular instance in polynomial time (NP).
- Below are two definitions of NDTM, since Pawlak was not specific about which one we should use. Most likely we only need to know the first one, but I added the second one just in case.

- The NDTM operates exactly like a DTM with one exception:
- The transition function δ from the DTM definition is replaced by a relation called the transition relation:

$$\delta \subseteq ((Q - \{q_Y, q_N\}) \times \Gamma) \times (Q \times \Gamma \times \{-1, 1\})$$

- The transition relation defines all possible actions that could happen, which differs from the transition function that outputs

- The NDTM has exactly the same structure as a DTM, except that it is augmented with a *guessing module* having its own write-only head as illustrated below:



exactly one possible action that could happen.

- In NDTM there are many paths of computation, instead of just one path of computation in DTM. Those paths are defined by all possible actions that could take place given a state and a symbol from the read-write head.
- A NDTM return yes if and only if at least one of those possible computational paths leas into an accepting state q_Y.
- The NDTM program is specified in the same way as the DTM program.
- The NDTM procedure differs from that of DTM in that it takes place in two distinct stages:

1. Stage One (guessing stage)
    - Initially the input string is written in tape squares 1 through $|x|$, other squares are blank, the read-write head is at square 1 and the guessing head is at square -1.
    - The guessing module then either writes something to the square or moves the guessing head, but it can only move to the left.
    - The guessing module chooses randomly which symbols from Γ to write on the tape and how long will it operate.
    - Once the guessing module has finished the NDTM moves to the stage two.
2. Stage Two (normal operation)
    - After the guessing module wrote some new symbols onto the tape the NDTM proceeds just like a DTM.

- The NDTM then starts in state q_0 and the read-write head is at square 1 just as usual.
- The program executed by NDTM usually involves checking those squares that were written to by the guessing module.

- Motivation:
  - Let's consider the decision version of the traveling salesman problem. Suppose someone claimed, for a particular instance of this problem, that the answer for that instance is "yes". If we were sceptical, we could demand that they prove their claim by providing us with a tour having the required properties (distance less than the given bound). It would then be a simple matter for us to verify the truth or falsity of their claim merely by checking that that they provided us with is actually a tour and, if so, computing its length and comparing that quantity to the given bound. Furthermore, we could specify our "verification procedure" as a general algorithm that has time complexity polynomial in length of the instance.
  - In general those type of decision problems where given a "guess" solution we can verify it in polynomial time are defined as problems in the NP class.
  - The NDTM is used as an abstract machine that could take a problem in NP and produce an answer in polynomial time.