Function: parse_automaton_data

Description:

This function reads an automaton file to extract information like states, alphabet, initial state, final states, and transitions. It then ensures the consistency and validity of this information before generating an instance of the FiniteAutomata class.

Input:

- **file_name** (optional): The name of the file containing automaton information. If not provided, the default filename is "FA.in".

Output:

- An instance of the **FiniteAutomata** class representing the parsed automaton.

Class: FiniteAutomata

Description:

This class represents a finite automaton, encapsulating its states, alphabet, transitions, and functionalities to ascertain determinism and validate input sequences against the automaton's logic.

Attributes:

- **states**: List of states comprising the automaton.
- **alphabet**: List of symbols constituting the automaton's alphabet.
- **initial_state**: The designated initial state of the automaton.
- **final_states**: List of final states indicating acceptance criteria.
- **transitions**: Dictionary mapping state transitions based on input symbols.

Method: **is_deterministic()**

Description:

Determines if the automaton adheres to deterministic behavior, ensuring each state and input symbol leads to a single subsequent state.

Input:

- None

Output:

- **True** if the automaton is deterministic; otherwise, returns **False**.

Method: **is_sequence_accepted(sequence)**

Description:

Checks whether a given sequence of symbols is accepted by the automaton following its defined transition rules.

Input:

- **sequence**: The sequence of symbols to be evaluated against the automaton.

Output:

- **True** if the sequence is accepted; otherwise, returns **False**.

Function: validate_states

Description:

This function verifies the integrity of the automaton's structure by confirming the validity of states, initial state, final states, transitions, and their alignment with the defined alphabet.

Input:

- **states**: List of states in the automaton.
- **initial_state**: The initial state of the automaton.
- **final_states**: List of final states in the automaton.
- **transitions**: Dictionary representing state transitions based on input symbols.
- **alphabet**: List of symbols present in the alphabet.

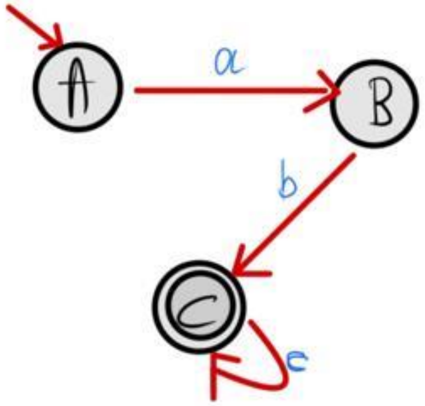
Output:

- Raises specific exceptions with detailed error messages if any inconsistencies or invalidities are found during validation.

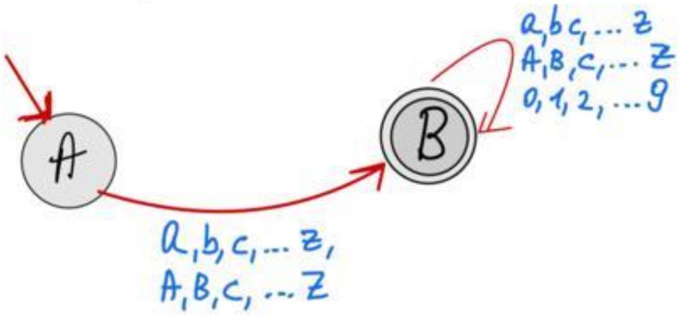
BNF-> FINITE AUTOMATA

```
<newline> ::= `\n`
<FA> ::=
<set_of_states><newline><alphabet><newline><initial_state><newline><final_states><newline><transitions>

<state> ::= `A`|`B`|...|`Z`|`a`|`b`|...|`z`
<states> ::= <state>|<state> ` ` <states>
<set_of_states> ::= `states = `<states>

<alphanumeric> ::= `A`|`B`|...|`Z`|`a`|`b`|...|`z`|`0`|`1`|...|`9`
<set_of_alphabet> ::= <alphanumeric>|<alphanumeric><set_of_alphabet>
<alphabet> ::= `alphabet = ` <alphabet_set>

<initial_state> ::= `initialState = ` <state>

<final_states> ::= `finalStates = ` <states>

<transition> ::= `(` <state> `,` <alphanumeric> `=>` <state> `)`
<set_of_transitions> ::= <transition><newline>|<transition><newline><set_of_transitions>
<transitions> ::= `transitions=` <newline> <set_of_transitions>
```

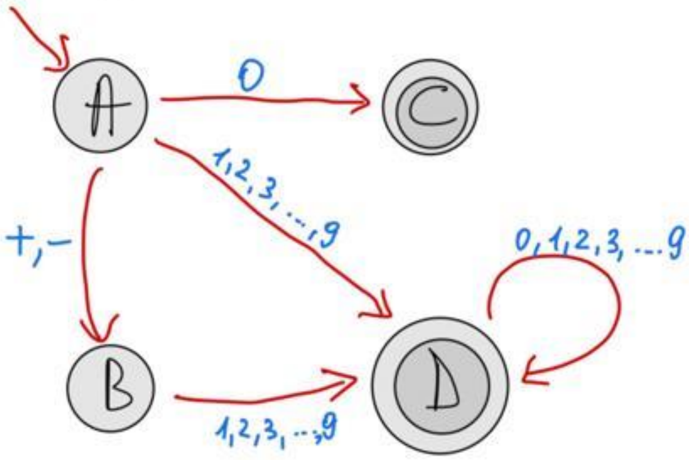
Example FA.in



identifiers FA.in



integer constants FA.IN


A
0
C
1,2,3,...,9
+,-
0,1,2,3,...9
B
D
1,2,3,...,9