Index: 4235230034

Name: Obeng Hamza

To classify the languages \(L(w)\) given a DFA that accepts a certain language \(L\), we need to check whether the language defined by the tail \(w\) (i.e., \((0+1)^**w\)) falls into one of three categories:

1. **Contained in \(L\)**: All strings ending with \(w\) are in \(L\).

2. **Disjoint from \(L\)**: No string ending with \(w\) is in \(L\).

3. **Neither contained nor disjoint**: Some strings ending with \(w\) are in \(L\) and some are not.

To make this determination, we follow these steps:

1. **Simulate the DFA**: Check how the DFA processes the string \(w\).

2. **Check the state after processing \(w\)**: Determine if the DFA reaches an accepting state after processing \(w\).

3. **Test with arbitrary prefixes**: Check how the DFA reacts to arbitrary prefixes followed by \(w\).

Classification of the Given Languages

1. **\(L(1111001)\)**:

- Check if the DFA reaches an accepting state after processing the string "1111001".

- If the DFA ends in an accepting state after "1111001", then \(L(1111001)\) is contained in \(L\).

- If it does not reach an accepting state, further testing with different prefixes is required.

2. **\(L(11011)\)**:

- Simulate the DFA on "11011".

- If the DFA ends in an accepting state, then \(L(11011)\) is contained in \(L\).

- Otherwise, check with different prefixes to see if any combination of prefixes followed by "11011" is accepted.

3. **\(L(110101)\)**:

- Simulate the DFA on "110101".

- If the DFA ends in an accepting state, then \(L(110101)\) is contained in \(L\).

- If the DFA does not accept "110101", check with various prefixes to determine the relationship with \(L\).

4. **\(L(00011101)\)**:

- Simulate the DFA on "00011101".

- If the DFA ends in an accepting state, then \(L(00011101)\) is contained in \(L\).

- If not, test with different prefixes to see if there are any strings in \(L\) that end with "00011101".

General Approach

1. **Simulate the DFA**: Process the string \(w\) through the DFA.

2. **Analyze the Result**:

- If the DFA reaches an accepting state, \(L(w)\) is contained in \(L\).

- If the DFA does not reach an accepting state, further check is required with various prefixes.

3. **Test Prefixes**:

- If some prefixes followed by \(w\) are accepted and some are not, then \(L(w)\) is neither contained nor disjoint.

- If no prefixes followed by \(w\) are accepted, then \(L(w)\) is disjoint from \(L\).

Example

Assuming a DFA that reaches an accepting state on processing "1111001":

- \(L(1111001)\) is contained in \(L\).

If it does not reach an accepting state:

- Further prefix testing is necessary to determine if \(L(1111001)\) is disjoint or neither.

Given the provided context, the exact transitions and states of the DFA are not detailed, but the approach remains consistent: simulate, analyze, and test prefixes to classify each \(L(w)\).