Developing Regular Expressions: Some Examples

Example 1: Develop a regular expression for the following (regular) set:

\[ \{w \in \{a, b\}^* : \#_a(w) \text{ and } \#_b(w) \text{ are both even}\}. \]

- First note that the given language is accepted by the following DFA:

- Introduce a new start state \( p \) and a unique final state \( f \), and introduce appropriate \( \varepsilon \)-transitions.
- Eliminate state $B$, leading to the following (equivalent) transition graph:

```
• Eliminate state $D$, leading to the following (equivalent) transition graph:

```

- Eliminate state $C$, leading to the following (equivalent) transition graph:

```
Finally, eliminate state $A$, leading to the following transition graph:

```
\[ p \xrightarrow{(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*} f \]
```

Regular expression is now immediate: \((aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*\).

It is instructive to build a transition graph (with $\epsilon$-transitions) corresponding to the regular expression thus obtained.

Example 2: Regular expression for \(\{w \in \{0, 1\}^*: w\) does not contain 000 as a substring\).

First note that the given language is accepted by the following DFA:

```
\[ A \xrightarrow{1} B \xrightarrow{1} C \xrightarrow{0, 1} D \]
```

Discard the trap state $D$, and remove all transitions associated with that state.
• Introduce a new start state \( p \) and a unique final state \( f \), and introduce appropriate \( \varepsilon \)-transitions, leading to the following (equivalent) transition graph.

• Eliminate state \( C \), leading to the following (equivalent) transition graph:

• Eliminate state \( B \), leading to the following (equivalent) transition graph:
• Eliminate state $A$, leading to the following (equivalent) transition graph:

- Regular expression is now immediate: $(1 + 01 + 001)^* (\varepsilon + 0 + 00)$. 

— o —