Half

- Half
  - $\frac{1}{2} L = \{ x \in \Sigma^* : \text{there exists } y \in \Sigma^* \text{ with } |y| = |x| \text{ such that } xy \in L \}$

- Theorem 3.4.1. If $L$ is regular, then so is $\frac{1}{2} L$.
  - Proof idea: Guess the appropriate middle state $q$ and move forward from $q_0$ and $q$ simultaneously.
  - Each state except the initial state is a triple: [the guessed middle state, the state we are in after processing a prefix of string $x$ starting from state $q_0$, the state we are in after processing a prefix of the guessed string $y$ starting from state $q$].
  - Guessing and the “there exists” in the definition suggest that nondeterminism will be useful.
Morphism (Homomorphism)

- **Morphism on strings**: Let $\Sigma$ and $\Delta$ be alphabets. A morphism is a map $h$ from $\Sigma^*$ to $\Delta^*$ that satisfies $h(xy) = h(x)h(y)$ for all strings $x, y \in \Sigma^*$.

- **Morphism on languages**: $h(L) = \bigcup_{x \in L} \{h(x)\}$.

- **Theorem 3.3.1**: For languages $L, L_1, L_2 \subseteq \Sigma^*$ and morphism $h: \Sigma^* \rightarrow \Delta^*$, we have
  
  a) $h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$
  
  b) $h(L_1 L_2) = h(L_1)h(L_2)$
  
  c) $h(L^*) = h(L)^*$
Substitution

- **Substitution on strings**: Let $\Sigma$ and $\Delta$ be alphabets. A substitution is a map $s$ from $\Sigma^*$ to $2^\Delta^*$ that maps each letter $a \in \Sigma$ to a language $L_a$ and satisfies:
  - a) $s(\varepsilon) = \{\varepsilon\}$,
  - b) $s(xy) = s(x)s(y)$ for all strings $x, y \in \Sigma^*$

- **Substitution on languages**: $s(L) = \bigcup_{x \in L} \{s(x)\}$.

- **Generalization of Theorem 3.3.1**. For languages $L, L_1, L_2 \subseteq \Sigma^*$ and substitution $h: \Sigma^* \rightarrow 2^\Delta^*$, we have
  - a) $s(L_1 \cup L_2) = s(L_1) \cup s(L_2)$
  - b) $s(L_1 L_2) = s(L_1)s(L_2)$
  - c) $s(L^*) = s(L)^*$

- **Theorem 3.3.5**. The class of regular languages is closed under substitution by regular languages.
Applications of Morphism and Substitution

- If $L \subseteq \{a, b, c\}^*$ is a regular language, is the language formed by removing the letter $c$ from all the strings in $L$ regular?

- If $L \subseteq \{a, b\}^*$ is a regular language, is the language formed by inserting the letter $c$ in all possible ways into strings in $L$ regular?
Inverse Morphism

- **Definition:** If $h: \Sigma^* \rightarrow \Delta^*$ is a morphism, and $L \subseteq \Delta^*$, then we define $h^{-1}(L) = \{x \in \Sigma^*: h(x) \in L\}$.

- **Examples**
  - Given a language $L \subseteq \{a, b\}^*$, consider the language formed by inserting the letter $c$ in all possible ways into strings in $L$.
  - **Shuffle**
    - The shuffle of two strings (not necessarily of the same length) consists of all strings obtained by interleaving the letters as in shuffling a deck of cards.
    - The shuffle of two languages: $\text{shuff}(L_1, L_2) = \bigcup_{x \in L_1, y \in L_2}\{\text{shuff}(x, y)\}$.
    - Different ways of defining shuffle using morphisms and inverse morphisms.

- **Theorem 3.3.9.** If $L \subseteq \Delta^*$ is regular, and $h: \Sigma^* \rightarrow \Delta^*$ is a morphism, then $h^{-1}(L)$ is regular.