homomorphism of languages

Canonical name	HomomorphismOfLanguages
Date of creation	2013-03-22 18:55:06
Last modified on	2013-03-22 18:55:06
Owner	CWoo (3771)
Last modified by	CWoo (3771)
Numerical id	12
Author	CWoo (3771)
Entry type	Definition
Classification	msc 68Q45
Synonym	ϵ-free
Synonym	non-erasing
Related topic	StringSubstitution
Related topic	Substitution
Defines	homomorphism
Defines	antihomomorphism
Defines	λ-free

Let Σ_1 and Σ_2 be two alphabets. A function $h : \Sigma_1^* \to \Sigma_2^*$ is called a *homomorphism* if it is a semigroup homomorphism from semigroups Σ_1^* to Σ_2^*. This means that

- h preserves the empty word: $h(\lambda) = \lambda$, and
- h preserves concatenation: $h(\alpha\beta) = h(\alpha)h(\beta)$ for any words $\alpha, \beta \in \Sigma_1^*$.

Since the alphabet Σ_1 freely generates Σ_1^*, h is uniquely determined by its restriction to Σ_1. Conversely, any function from Σ_1 extends to a unique homomorphism from Σ_1^* to Σ_2^*. In other words, it is enough to know what $h(a)$ is for each symbol a in Σ_1. Since every word w over Σ is just a concatenation of symbols in Σ, $h(w)$ can be computed using the second condition above. The first condition takes care of the case when w is the empty word.

Suppose $h : \Sigma_1^* \to \Sigma_2^*$ is a homomorphism, $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$. Define

$$h(L_1) := \{h(w) \mid w \in L\} \qquad \text{and} \qquad h^{-1}(L_2) := \{v \mid h(v) \in L_2\}.$$

If L_1, L_2 belong to a certain family of languages, one is often interested to know if $h(L_1)$ or $h^{-1}(L_2)$ belongs to that same family. We have the following result:

1. If L_1 and L_2 are regular, so are $h(L_1)$ and $h^{-1}(L_2)$.
2. If L_1 and L_2 are context-free, so are $h(L_1)$ and $h^{-1}(L_2)$.
3. If L_1 and L_2 are type-0, so are $h(L_1)$ and $h^{-1}(L_2)$.

However, the family $\mathscr{F}$ of context-sensitive languages is not closed under homomorphisms, nor inverse homomorphisms. Nevertheless, it can be shown that $\mathscr{F}$ is closed under a restricted class of homomorphisms, namely, λ-free homomorphisms. A homomorphism is said to be *λ-free* or *non-erasing* if $h(a) \neq \lambda$ for any $a \in \Sigma_1$.

Remarks.

- Every homomorphism induces a substitution in a trivial way: if $h : \Sigma_1^* \to \Sigma_2^*$ is a homomorphism, then $h_s : \Sigma_1 \to P(\Sigma_2^*)$ defined by $h_s(a) = \{h(a)\}$ is a substitution.
- One can likewise introduce the notion of antihomomorphism of languages. A map $g : \Sigma_1^* \to \Sigma_2^*$ is an antihomomorphism if $g(\alpha\beta) =$

$g(\beta)g(\alpha)$, for any words α, β over Σ_1. It is easy to see that g is an antihomomorphism iff $g \circ \mathrm{rev}$ is a homomorphism, where rev is the reversal operator. Closure under antihomomorphisms for a family of languages follows the closure under homomorphisms, provided that the family is closed under reversal.

References

[1] S. Ginsburg, *The Mathematical Theory of Context-Free Languages*, McGraw-Hill, New York (1966).

[2] H.R. Lewis, C.H. Papadimitriou *Elements of the Theory of Computation.* Prentice-Hall, Englewood Cliffs, New Jersey (1981).