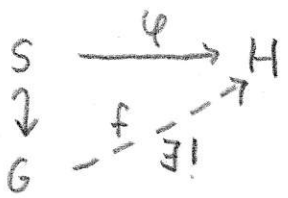
1. Free groups

Definitions and simple facts

1.1 **Definition.** Let G be a group. A basis of G is a subset $S \subset G$ with the following property.

(*) For any group H and any map $\varphi : S \longrightarrow H$ there exists a unique homomorphism $f : G \longrightarrow H$ such that $f(s) = \varphi(s)$ for all $s \in S$.



φ map of sets

f homomorphism of groups

1.2 **Definition.** A group is called free if it possesses a basis. [If G has a basis S we say that G is free with basis S or that G is free over S]

1.3 **Proposition.** For any set T there is a group G and an injective map $\tau : T \longrightarrow G$ such that G is free with basis $\tau(T)$.

Proof. Consider the set $T \times \{-1, 1\}$. For $t \in T$ we write for $(t, 1)$ t or t^1 (i.e. $t = t^1 = (t, 1)$) and for $(t, -1)$ we write t^{-1}; i.e. we write $T \sqcup T^{-1}$ for $T \times \{-1, 1\}$

A word in $T \cup T^{-1}$ is a finite sequence

$$t_1^{\varepsilon_1} \cdot \dots \cdot t_n^{\varepsilon_n}, \quad \varepsilon_i \in \{\pm 1\}, \ t_i \in T,$$

of elements of $T \cup T^{-1}$; the empty word is also a word which we denote by 1. The set of words in $T \cup T^{-1}$ is denoted by $W(T)$. By concatenating words we define a product „$\cdot$" on $W(T)$. The product is associative with 1 as a 2-sided unit.

To introduce inverses we introduce an equivalence relation „$\sim$" on $W(T)$ as follows.

an elementary expansion of a word w is the insertion of $t t^{-1}$ or $t^{-1} t$ somewhere in w, $t \in T$, i.e. $w = w_1 \cdot w_2 \dashrightarrow w' = w_1 t t^{-1} w_2$ or

$$w_1 t^{-1} t \, w_2$$

an elementary reduction of a word is the opposite process, i.e. the removal of a $t^{-1}t$ or $t t^{-1}$ from the word.

Two words are called equivalent if one can pass from one to the other by finitely many expansions and reductions

1.4 Remark: $w_i \sim w_i'$, $i = 1, 2$; then $w_1 w_2 \sim w_1' w_2'$

Proof. by induction on the number of expansions and reductions to pass from w_i to w_i' (Exercise)

consequently, the product on $W(T)$ passes down to a product of $F(T) := W(T)/\sim$ given by

$[w_1][w_2] := [w_1 w_2]$, where $[x]$ means equiv. class of the word x. It is obviously associative with $[1]$ as unit and the inverse of

$$[t_1^{\varepsilon_1} \cdots t_n^{\varepsilon_n}] \text{ is } [t_n^{-\varepsilon_n} \cdots t_1^{-\varepsilon_1}]$$

where we set $t^{-(-1)} = t^1 = t$. Thus

$F(T)$ is a group.

Before we show that $F(T)$ is in fact free with basis $\{[t] : t \in T\}$ we solve the so called word problem for the group $F(T)$. We are looking for some algorithm that allows us to answer the following

1.5 Question: Given words $w, w' \in W(T)$. When is $[w] = [w']$ in $F(T)$?

1.6 Definition: A word w is called reduced if it does not contain a subword of the form $t^{\varepsilon} t^{-\varepsilon}$ with $\varepsilon \in \{\pm 1\}$, i.e. if w is not of the form $w_1 t^{\varepsilon} t^{-\varepsilon} w_2$ for some $t \in T$, $w_1, w_2 \in W(T)$.

1.7 Proposition: For every element $[w] \in F(T)$ there is a unique reduced word $\bar{w}$ with $[w] = [\bar{w}]$

Furthermore, if the length $\ell(w)$ of $w = t_1^{\varepsilon_1} \cdots t_n^{\varepsilon_n}$ is defined to be n then there is an algorithm which produces in (at most) $\lfloor \frac{n}{2} \rfloor$ steps the unique reduced word equivalent to w.

Proof. Here is an algorithm. Read the word w starting from the left until you hit for the first time a pair of consecutive letters of the form $t^{\varepsilon} t^{-\varepsilon}$, $\varepsilon = \pm 1$. If there is no such pair, we are done. Otherwise, remove this pair and start anew. Since at each step the word length is reduced by 2 we end after $\lfloor \frac{n}{2} \rfloor$ steps either by a 1-letter word or by 1 unless the algorithm stopped before.

To prove 1.7, it remains to show that our algorithm applied to two equivalent words w and w' produces the same reduced word

(This implies in particular: if w and w' are reduced and equivalent then they are identical. For our algorithm leaves a reduced word unchanged)

We do this by induction on the number of expansions and reductions we need to go from w to w'. I.e. it suffices (to show:) if w' is obtained from w by an expansion then our algorithm produces the same reduced word for w' as for w.

1.8: This is an (easy) exercise. □