RENAMING AND ERASING IN SZILARD LANGUAGES

Matthias Höpner
Manfred Opp

Institut f. Informatik, Univ. Hamburg
Schlüterstr. 70 , D-2000 Hamburg 13

Abstract

We characterize the images of Szilard languages under alphabetical homomorphisms using so called label grammars and show: If L is a label language (i.e. a language generated by such grammar) then $L - \{\lambda\}$ is a coding (renaming) of some Szilard language. This result shows that arbitrary homomorphisms do not have more generating power than nonerasing ones except that they generate the empty word. Combining this result with other properties of label languages, established elsewhere, one obtains characterizations of label languages and of codings of Szilard languages including those by finite shuffle expressions and equations.

Since one might interprete label grammars as a special kind of labelled Petri nets, where each transition has exactly one input arc, we solved the elemination problem of λ-transitions for this restricted class of Petri nets, even if there occur infinite firing-sequences using only λ-transitions within a net.

Introduction

Informally Szilard languages describe the derivation process of context-free grammars. This is done by writing down the rules exactly in the order they have been used to yield a terminating derivation. Very often one denotes labels instead of the rules itself where each rule has exactly one label. This one to one correspondence supplies the class of Szilard languages with strictly deterministic properties, which force them to form an anti AFL and not to contain all the regular sets.

From a pure theoretical point of view this is a very unlucky situation. Moreover one might be interested only in the important part of a derivation not denoting for instance labels of rules like $A \longrightarrow B$. This is only possible within the theory of Szilard languages if one allows λ-labels for the rules. Arbitrary labelling is necessary if one likes to supply rules which are structurally similar with the same label.

This suggestions give the motivation to introduce label grammars, which exactly generate the images of Szilard languages under alphabetical homomorphisms. The class $\mathcal{H}_{\alpha}(\mathcal{S}_{\mathcal{Z}})$ of label languages now also has nice formal properties. (See Höpner 74b and Höpner/Opp 77). Regarding the similarities between context-free languages and label languages as for instance finite expression representations, the connection with recognizable sets of trees, and the characterization using equations one might ask, whether there exist more similarities. One of the most interesting questions might be the problem, whether the λ-labels do or do not have a strong effect on the generation of label languages.

Results

Passing some lemmas we show that there exists a result similar to the Greibach normalform theorem, which unfortunately cannot be proved with the same methods.

Definition 'label grammar'

A label grammar G is a tupel $G = (N,R,A_o,X)$, where $N = \{A_o,A_1,\ldots,A_n\}$ is a finite set of nonterminals, $A_o \in N$ is the start symbol, $X = \{x_1,x_2,\ldots,x_m\}$ is a finite set of labels, and $R \subset N \times (X \cup \{\lambda\}) \times N^*$ is a finite set of context-free, labelled rules.

Notation

A rule $(A,x,w) \in R$ may also be denoted by $A \xrightarrow[x]{} w$, and it is called an 'x-rule' for short. Recall that in a label grammar there might exist more than one x-rule for some $x \in X \cup \{\lambda\}$. In fact the rules of a label grammar are context-free rules, to each of which a symbol from X or the empty word is assigned.

The following convention will be used: Nonterminals will be denoted by capital letters $A,B,C,\ldots$ or $A_i,B_i,\ldots$, $i \in \mathbb{N}_o$. Strings of nonterminals will be denoted by small $u,v,w,u',v',u_i,v_i,\ldots$, labels will be denoted by $x,y,z,x_i,y_i,z_i,\ldots$ and strings of labels will be written as $\alpha, \beta, \gamma, \alpha_i, \beta_i, \gamma_i,\ldots$

The derivation process for label grammars will be defind according to the ordinary notation of derivations in context-free grammars but disregarding the order of the symbols that form a sentential form. In the sequel Ψ denotes the Parikh mapping $\Psi : N^* \longrightarrow \mathbb{N}_o^{card(N)}$. For vectors $a,b \in \mathbb{N}_o^{card(N)}$, $a = b$, $a \geqslant b$, $a < b$, $a + b$, $a - b$ will be understood componentwise. The vector $(0,\ldots,0)$ will be denoted by $\underline{0}$.

Definition 'one step derivation'

Let $w_1 \in N^+$, $w_2 \in N^*$, $(A,x,v) \in R$. Then $w_1 \Longrightarrow_{(A,x,v)} w_2$ holds true, if and only if:

$$\text{(i)} \quad \Psi(w_1) - \Psi(A) \geqslant \underline{0}$$
$$\text{(ii)} \quad \Psi(w_2) = \Psi(w_1) - \Psi(A) + \Psi(v) \ .$$

If we are only interested in the label of the underlying rule, we write $w_1 \Longrightarrow_x w_2$ instead of $w_1 \Longrightarrow_{(A,x,v)} w_2$.

We extend this notion for arbitrary derivations by:

Definition 'many step derivation'

for each $\alpha \in X^*$ we define the relation '$\overset{*}{\Longrightarrow}_\alpha$' by:

(i) $w \overset{*}{\Longrightarrow}_\lambda w$ for each $w \in N^+$

(ii) if $w_1 \overset{*}{\Longrightarrow}_\alpha w_2$ and $w_2 \overset{*}{\Longrightarrow}_x w_3$, then $w_1 \overset{*}{\Longrightarrow}_{\alpha x} w_3$.

A derivation $w_1 \overset{*}{\Longrightarrow}_\alpha w_2$ is called 'α-derivation'.

Definition 'label language'

A language $L \subset X^*$ is called label language iff there exists a label grammar $G = (N,R,A_o,X)$ such that:

$$L = LL(G) := \{ \alpha \in X^* \mid A_o \overset{*}{\underset{\lambda}{\Longrightarrow}} \alpha \text{ in } G \} .$$

It is obvious that each label language, as it is defined here, is the image of a Szilard language of a context-free grammar under alphabetical homomorphism. Therefore we denote the family of all label languages by $\mathcal{H}_\alpha(\mathcal{L}_2)$.

We recall a normalform theorem as it can be found in Höpner 74α .

Theorem 'first normalform'

For each label grammar $G = (N,R,A_o,X)$ we can find a label grammar $\tilde{G} = (\tilde{N},\tilde{R},A_o,X)$ such that the following holds:

(i) $\tilde{G}$ is reduced, that means: for each $A \in N$ there exists a derivation $A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} wA \overset{*}{\underset{\beta}{\Longrightarrow}} \lambda$.

(ii) $\tilde{G}$ doesn't contain a rule of the form (A, λ, λ) .

(iii) $LL(\tilde{G}) = LL(G) - \{\lambda\}$

The construction of the grammar $\tilde{G}$ is similar to that one used to eleminate the empty word within context-free grammars, so we need no detailed proof.

In the next part we define a restricted class of derivations, called 'normal derivations', and show that the set of words generated by normal derivations only is exactly the whole label language of the underlying grammar.

Definition 'simple derivation', 'x-sequence', 'path of a sequence'

A derivation of the form

$$w_o \overset{}{\underset{(B_o,x_o,v_o)}{\Longrightarrow}} w_1 \overset{}{\underset{(B_1,x_1,v_1)}{\Longrightarrow}} w_2 \quad \cdots \quad w_n \overset{}{\underset{(B_n,x_n,v_n)}{\Longrightarrow}} w_{n+1} \quad , n \geqslant 0$$

within a given label grammar is called 'simple derivation' if and only if:

$$\Psi(v_i) - \Psi(B_{i+1}) \geqslant \underline{0} \quad \text{for } 0 \leqslant i \leqslant n-1 .$$

In the case that $x_i = \lambda$ for $0 \leqslant i \leqslant n-1$ and $x_n \neq \lambda$ holds, this simple x_n-derivation is called 'x_n-sequence'. The word $B_o B_1 \cdots B_n \in N^+$ is called the 'path' of this x_n-sequence.

Definition 'sequenced derivation'

A derivation of the form $w_o \overset{*}{\underset{y_1}{\Longrightarrow}} w_1 \overset{*}{\underset{y_2}{\Longrightarrow}} w_2 \cdots w_{n-1} \overset{*}{\underset{y_n}{\Longrightarrow}} w_n$ within a given label grammar is called a 'sequenced derivation' iff each subderivation $w_i \overset{*}{\underset{y_{i+1}}{\Longrightarrow}} w_{i+1}$, $0 \leqslant i \leqslant n-1$, is an y_{i+1}-sequence.

Not every derivation within a label grammar is a sequenced derivation, ofcourse, but we can show:

Theorem 'sequenced derivations are sufficient'

Let G be a label grammar in first normalform, then

$LL(G) = \{ \alpha \in X^+ \mid$ there exists a sequenced derivation $A_o \overset{*}{\Longrightarrow}_{\alpha} \lambda$ in $G \}$.

Proof:

We define a transformation T_1 on arbitrary α-derivations d, such that $T_1(d)$ is a sequenced α-derivation. The transformation successively rearranges the λ-rules of the given derivation.

Definition 'the transformation T_1'

step 0 : Let d be the given derivation.

step 1 : Mark the rightmost unmarked λ-rule in the derivation d. If there is no such rule, then output d. Stop.

step 2 : Shift the λ-rule which has been marked in step 1 to the right as far as possible. This means: until the next rule needs a nonterminal, which only occurs in the substring generated by this λ-rule. This yields the new derivation d. Return to step 1.

Of course the transformation T_1 terminates for every input derivation d, since each time we reach step 2 one additional λ-rule is marked and there is only a finite number of λ-rules within d. On the othrr hand T_1 terminates only if all the λ-rules have been moved to the rightmost position which is characterized by the fact that the lefthand side of the next rule can be found only within the righthand side of the former one. Moreover there is no λ-rule at the very end of the derivation d since G has been assumed to be in first normalform, i.e. each terminating rule has a label $x \neq \lambda$. Therefore the output of the transformation T_1 is a sequenced derivation generating the same word α . This proves our theorem.

Now we are going to define a certain 'loop structure' for x-sequences using another transformation T_2. To do this we define 'A-loops' for each $A \in N$.

Definition 'A-loops within a sequence'

Let $w_o \overset{(B_1,x_1,v_1)}{\Longrightarrow} w_1 \cdots \Longrightarrow w_{i-1} \overset{(B_i,x_i,v_i)}{\Longrightarrow} w_i \cdots \quad w_{j-1} \overset{(B_j,x_j,v_j)}{\Longrightarrow} w_j \cdots \Longrightarrow w_{n-1} \overset{(B_n,x_n,v_n)}{\Longrightarrow} w_n$ be an x_n-sequence ($x_l = \lambda$ for $1 \leq l \leq n-1$, $x_n \in X$).

If $B_i = B_j \in N$ and the path $B_i \cdots B_{j-1} \in N^+$ doesn't contain a symbol twice (i.e. $B_l \neq B_k$ for $i \leq l \neq k \leq j-1$) then the subderivation

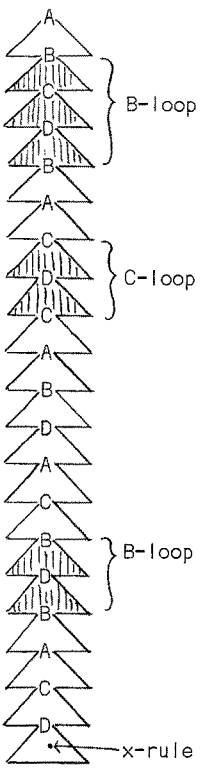
$B_i \overset{(B_i,\lambda,v_i)}{\Longrightarrow} u_i \overset{(B_{i+1},\lambda,v_{i+1})}{\Longrightarrow} u_{i+1} \cdots \overset{(B_{j-1},\lambda,v_{j-1})}{\Longrightarrow} u_{j-1}$

is called a 'B_i-loop' of this sequence.

We want to point out, that A-loops of a sequence use only λ-rules which are pairwise different and at most card(N) of them. Thus there is only a finite number of different A-loops for each $A \in N$ that can be constructed within a label grammar.

Example

A more pictoral representation of A-loops may be necessary. Suppose the following derivation tree of some x-sequence (we choose a special example for demonstration) :



A triangle like  denotes the application of a rule $A \longrightarrow uBv$. Thus the base of a triangle represents the sentential form generated by this rule.

To each A-loop of a sequence corresponds a subword w of the path of the whole sequence, which is the path of the loop and has the following properties: (i) no symbol in w occurs twice, and (ii) the symbol appearing immediately next to the subword w equals the first symbol of w . In the example we may mark some of the loops within the path as follows:
A[BCD]BA[CD]CABDAC[BD]BACD. One can see that we havn't marked all the loops. A complete marking without partial overlapping would be the following one:
A[BCD]BA[CD]C[ABD]AC[BD]BACD. The transformation T_2 as defined next will use a unique marking of loops within a sequence.

Definition 'the transformation T_2 for sequences'

Let s be a given x-sequence of a label grammar.

step 0 : Start at the very left of s , no loops within s are marked.

step 1 : Find the leftmost unmarked loop within s and mark it. A loop within a sequence is called leftmost, if there exist no other loop containing a rule in front of this loop. Thus the leftmost loop is uniquely defined by the longest prefix of the path which doesn't contain a symbol twice.

step 2 : If in step 1 no further loop has been found, then do step 3, otherwise repeat step 1.

step 3 : If for each $A \in N$ all the marked A-loops are directly following each other (no rule is separating two A-loops) then stop. Otherwise rearrange the marked loops within the sequence in such a way, that for each $A \in N$ the A-loops are directly following the first one within the relative order they had before. Repeat step 1.

Let us apply the transformation T_2 to the sequence of the previous example.

Example 'application of T_2'

(step 0) Let s be the sequence with the path ABCDBACDCABDACBDBACD .

(step 1, step 2) We mark the leftmost loop : A[BCD]BACDCABDACBDBACD .

Applying step 1 and step 2 three times more yields: A[BCD]BA[CD]C[ABD]AC[BD]BACD .

No further loops can be found, so we apply step 3. In this step we have to erase the B-loop with path BD at the end of the sequence and insert this loop next to the first B-loop. This yields a sequence with the following path : A[BCDBD]BA[CD]C[ABD]ACBACD. Since we have rearranged the loops we repeat step 1. We find another A-loop and mark it: A[BCDBD]BA[CD]C[ABDACB]ACD . We pass step 2 and reach step 3 where we find that no rearranging is necessary, the stop condition is met and therefore we have finished. The sequence with the above path is the unique output of the transformation T_2 .

Without proof we state:

Fact 'properties of the transformation T_2'

T_2 terminates for every input sequence s and the path w of the output sequence has the following form:

$$w = u_1 v_1^1 \cdots v_{k(1)}^1 u_2 v_1^2 \cdots v_{k(2)}^2 u_3 \cdots u_n v_1^n \cdots v_{k(n)}^n u_{n+1}$$

where:

- $0 \leq n \leq \mathrm{card}(N)$
- $k(i) \in \mathbb{N}_0$ for $1 \leq i \leq n$
- $u_i \in N^+$ for $2 \leq i \leq n+1$, $u_1 \in N^*$
- $v_j^i \in N^*$ for $1 \leq i \leq n$ and $1 \leq j \leq k(i)$
- $\mathrm{lg}(u_i) \leq \mathrm{card}(N)$ for $1 \leq i \leq n+1$
- $\mathrm{lg}(v_j^i) \leq \mathrm{card}(N)$ for $1 \leq i \leq n$, $1 \leq j \leq k(i)$

Each v_j^i and v_m^l is a path of B_i-loop respectively B_l-loop, and if $i \neq l$ then $B_i \neq B_l$. If we define $\underline{max} := \mathrm{card}(N) \cdot (\mathrm{card}(N) + 1)$ for a given label grammar, then we have $1 \leq \mathrm{lg}(u_1 u_2 \cdots u_{n+1}) \leq \underline{max}$ for the above path of the output x-sequence $T_2(s)$. Note that rearranging loops within a sequence doesn't change the generated label.

We now define 'normal sequences' as the invariants under this transformation.

Definition 'normal x-sequence', 'normal derivation'

Each x-sequence s for which $T_2(s)$ is equal to s itself is called a 'normal x-sequence'. A derivation is called 'normal derivation' if and only if it is a sequenced derivation, each sequence of which is a normal sequence.

We obtain:

Theorem 'normal derivations are sufficient'

For each label grammar in first normalform the following holds:

$$LL(G) = \{ \alpha \in X^* \mid \text{there exists a normal } \alpha\text{-derivation } A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} \lambda \}.$$

Proof:

We know that it is sufficient to consider only sequenced derivations. Now suppose s is a sequenced α-derivation built up by the sequences $s_1, s_2, \ldots, s_m$. We apply the transformation T_2 to each sequence s_i. This gives a new collection of normal sequences $T_2(s_1), T_2(s_2), \ldots, T_2(s_m)$. These normal sequences can be composed to the desired normal α-derivation.

Having defined normal derivations, we can construct a label grammar without λ-rules equivalent to a given arbitrary label grammar in first normalform.

Construction ' λ-free label grammar'

Let $G = (N,R,A_o,X)$ be a label grammar in first normalform.

Define a new alphabet: $\overline{N} := \{ \overline{A} \mid A \in N \}$.

Define the operation ' $\overline{\ }$ ' for each $w = B_1B_2\cdots B_n \in N^+$ by $\overline{w} := \overline{B}_1\overline{B}_2\cdots\overline{B}_n \in \overline{N}^+$ and let $\overline{\lambda} := \lambda$. Let $\underline{max} := card(N)\cdot(card(N)+1)$ as before.

Define sets of rules by:

$$R_1 := \left\{ (A,x,w\bar{u}) \;\middle|\; A \overset{*}{\underset{x}{\Longrightarrow}} w \text{ is a x-sequence in } G \text{ with path } v \in N^+, \text{ such that } lg(v) \leqslant \underline{max} \text{ and } u \text{ is obtained from } v \text{ by deletion of some (all, no) symbols} \right\}$$

$$R_2 := \left\{ (\overline{A},x,w_1w_2\bar{u}) \;\middle|\; \text{there exists a rule } (B,x,w_2\bar{u}) \in R_1 \text{ and there exists an A-loop } A \overset{*}{\underset{\lambda}{\Longrightarrow}} w_1AB \text{ in } G \right\}$$

$$R_3 := \left\{ (\overline{A},x,w_1w_2\bar{u}\overline{A}) \;\middle|\; (\overline{A},x,w_1w_2\bar{u}) \in R_2 \right\}$$

Note: The sets R_1, R_2, and R_3 are finite, since each pathlength of the underlying derivations is bounded by some constant.

The new label grammar $\tilde{G}$ is finally defined by:

$G = (N \cup \overline{N}, R_1 \cup R_2 \cup R_3, A_o, X)$. By definition $\tilde{G}$ doesn't contain a λ-rule.

We are now in the position to formulate and prove our main result. Because of the lack of space we cannot prove it completely, but we are able to present the difficult part of the proof.

Theorem 'we can do without λ-rules'

Let G be an arbitrary label grammar, then there exists a label grammar $\tilde{G}$ such that $LL(\tilde{G}) = LL(G) - \{\lambda\}$ and $\tilde{G}$ doesn't contain a rule of the form (A,λ,w) .

Proof:

For the given label grammar G we construct a label grammar in first normalform which generates $LL(G) - \{\lambda\}$. Thus without restriction we may assume G to be in first normalform. For this label grammar G we construct $\tilde{G}$ as it is defined in the above construction. $\tilde{G}$ has no λ-rules and it is left to show $LL(\tilde{G}) = LL(G)$.

The proof of $LL(\tilde{G}) \subset LL(G)$ follows the idea that each rule of $\tilde{G}$ has been constructed by derivations within G and is not presented here.

For the proof of the statement $LL(G) \subset LL(\tilde{G})$ we need some auxiliary notations and definitions.

Notation 'lists of words'

Let $w_1, \ldots, w_n \in N^*$ be words, then $[w_1, \ldots, w_n]$ is called a 'list'. The empty list is denoted by $[]$. Let $l_1 = [w_1, \ldots, w_n]$ and $l_2 = [v_1, \ldots, v_m]$ be lists, then: $cons(l_1,l_2) := [w_1, \ldots, w_n, v_1, \ldots, v_n]$.

Let l be a list and w be a word, then $delete(w,l)$ denotes the list which is obtained from the list l by deletion of the first occurence of w in this list. If l doesn't contain w, then $delete(w,l) := l$.

Definition 'collection of free loops within normal sequences'

Let s be a normal sequence of the label grammar $G = (N,R,A_o,X)$. (Recall that $N = \{A_o, \ldots, A_n\}$).

$F(s)$ denotes the 'collection of free loops' within s and is defined as follows:

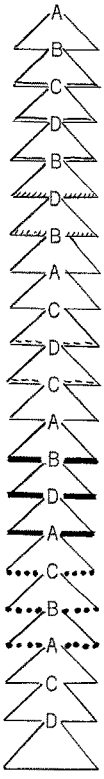
$F(s) := (F_o(s), F_1(s), \ldots, F_n(s))$, where for each $0 \leq i \leq n$, $F_i(s)$ is a list of words from N defined by:

$F_i(s)$ is the list of all the words $v \in N^+$ for which $A_i \overset{*}{\underset{\lambda}{\Longrightarrow}} vA_i$ is a marked loop within s after applying the transformation T_2. Note $s = T_2(s)$. Moreover $F_i(s)$ contains v as often as the loop $A_i \overset{*}{\underset{\lambda}{\Longrightarrow}} vA_i$ appears within $T_2(s)$.

$F_i(s)$ thus is the list of words which are created within A_i-loops as additional sentential forms and which are not empty.

Example

Let us return to our example of a sequence which had the path ABCDBACDCABDACBDBACD . After applying T_2 we obtained the path ABCDBDBACDCABDACBACD which corresponds to the pictural representation of the sequence s as follows:



If we write $F_A(\ )$ instead of $F_o(\ )$ using the ordered set $N = \{A,B,C,D\}$ of non-terminals, then $F_A(s) = [v_1, v_2]$, where v_1 is that part of the sentential form which is marked by ▬, and v_2 is the part marked by ••••• ; v_1 and v_2 are supposed to be nonempty.
Since there is no D-loop, the list $F_D(s)$ is empty. The other loops are treated similar such that the collection $F(s)$ is of the form $F(s) = ([v_1,v_2],[v_3,v_4],[v_5],[])$.

We want to generalize the term 'collection of free loops' to normal derivations which are compositions of normal sequences. Informally we call a loop $A \overset{*}{\underset{\lambda}{\Longrightarrow}} vA$ within a normal sequence of a normal derivation 'free' if no symbol of the word v is replaced in later steps of the derivation.

Definition 'collection of free loops within normal derivations'

For each normal derivation d the collection of free loops will be denoted by $F(d)$. Thus if d is a normal sequence, then $F(d)$ coincides with the definition of the collection of free loops within a sequence.

Now suppose $d = (A \overset{*}{\underset{\alpha}{\Longrightarrow}} wB\ )$ is a normal derivation for which
$F(d) = (F_o(d),F_1(d), \ldots ,F_n(d))$ is defined and suppose
$s = (B \overset{*}{\underset{x}{\Longrightarrow}} v\ )$ is a normal sequence, then for the normal αx-derivation
$ds := (A \overset{*}{\underset{\alpha}{\Longrightarrow}} wB \overset{*}{\underset{x}{\Longrightarrow}} wv\ )$ the collection of free loops
$F(ds) = (F_o(ds),F_1(ds), \ldots ,F_n(ds))$ is defined as follows:

Case 1: The symbol B of the sentential form wB is not created by any free loop within the normal derivation d. In other words: no word u from the lists $F_i(d)$ that form $F(d)$ contains the symbol B. For all $0 \leqslant i \leqslant n$ $F_i(ds)$ is then defined by: $F_i(ds) := \mathrm{cons}(F_i(d),F_i(s))$.

Case 2: The symbol B is created within some free loop, i.e. there exists an index l, $0 \leqslant l \leqslant n$ such that the list $F_l(d)$ contains a word u, which in turn contains the symbol B. Let l be the smallest index of this kind and let u be the first word in the list $F_l(d)$, which contains the symbol B. Then:

$$F_i(ds) := \text{cons}(\, F_i(d), F_i(s)\,) \quad \text{for all } 0 \leqslant i \leqslant n,\ i \neq l.$$
$$F_l(ds) := \text{cons}(\, \text{delete}(u, F_l(d)),\ F_l(s)\,) \ .$$

We now define a transformation T_3 on normal derivations as follows:

Definition 'the T_3 transformation to delete free loops'

Let d be a normal derivation, then $T_3(d)$ denotes exactly that normal derivation which is obtained by:

(i) deletion of all free loops within d.
(ii) deletion of all the loops $A \overset{*}{\underset{\lambda}{\Longrightarrow}} A$ within d.

The term 'deletion of a loop' should be understood as follows:
If $A \overset{*}{\underset{\alpha}{\Longrightarrow}} uB \overset{*}{\underset{\lambda}{\Longrightarrow}} uvB \overset{*}{\underset{\beta}{\Longrightarrow}} uvw$ is a normal derivation and $B \overset{*}{\underset{\lambda}{\Longrightarrow}} vB$ is a free loop within this derivation d, i.e. $F(d)$ contains v, then
$A \overset{*}{\underset{\alpha}{\Longrightarrow}} uB \overset{*}{\underset{\beta}{\Longrightarrow}} uw$ is the derivation after deletion of the free loop $B \overset{*}{\underset{\lambda}{\Longrightarrow}} vB$.

Clearly $T_3(d)$ is a normal derivation which doesn't have free loops. If d is a normal sequence, then $T_3(d)$ is a sequence, the pathlength of which is at most max. This follows from the fact about $T_2(d)$. Moreover if $d = (A \overset{*}{\underset{\alpha}{\Longrightarrow}} w)$ is a normal derivation, then

$$T_3(d) = (A \overset{*}{\underset{\alpha}{\Longrightarrow}} v) \quad \text{where} \quad \psi(w) = \psi(v) + \sum \psi(u)\ ,$$

the sum is taken over all the words u which are contained in the list $F(d)$. (' u in $F(d)$ ' for short!)

We now prove the following statement:

Statement

Let G and $\tilde{G}$ as in the construction.
If $d = (A \overset{*}{\underset{\alpha}{\Longrightarrow}} w)$ is a normal derivation in G, then there exists a derivation $A \overset{*}{\underset{\alpha}{\Longrightarrow}} v\bar{u}$ in G such that the following holds:

(i) $\psi(w) = \psi(v) + \sum \psi(v')$, v' in $F(d)$.
(ii) $\bar{u}$ contains the symbol $\bar{A}_i$ exactly once iff the list $F_i(d)$ is not the empty list.

If this statement is correct, then for each normal derivation $A_o \overset{}{\underset{\alpha}{\Longrightarrow}} \lambda$ in G there exists a derivation $A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} \lambda$ in $\tilde{G}$, since terminating derivations in G do have only empty lists $F_i(d)$. (Each symbol, generated in some free loop has to be replaced in some further step of the derivation.)

Since normal derivations are sufficient to describe LL(G) we have $LL(G) \subset LL(\tilde{G})$!

We will prove the statement by induction over the number of normal sequences that form the normal derivations in G.

Basic step:

Suppose $s = (A_o \overset{*}{\underset{x}{\Longrightarrow}} w)$ is a normal sequence in G. Then $T_3(s) = (A_o \overset{*}{\underset{x}{\Longrightarrow}} v)$ is a sequence with pathlength at most max. By definition of R_1 in the construction we find a rule $(A_o, x, v\bar{u}) \in R_1$ such that $\bar{A}_i$ is contained in $\bar{u}$ exactly once iff the list $F_i(s)$ is not empty.

This follows from the fact that for free A-loops, even after deletion, the symbol A remains on the path of the sequence. Moreover $\psi(w) = \psi(v) + \sum \psi(v')$, v' in F(s), is true by definition of the transformation T_3.

nduction hypothesis:

The statement is true for all normal derivations $d = (A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} w)$ in G, which are composed by at most k sequences, i.e. $lg(\alpha) \leq k$.

Induction step:

Let $ds = (A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} w_1 A_i \overset{*}{\underset{x}{\Longrightarrow}} w_1 w_2)$ be a normal derivation in G such that

$d = (A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} w_1 A_i)$ is a normal derivation composed by k sequences and

$s = (A_i \overset{*}{\underset{x}{\Longrightarrow}} w_2)$ is a normal sequence.

We have to distinguish two cases:

Case 1:

The symbol A_i within the sentential form $w_1 A_i$ is not created in any free loop of d.

From the induction hypothesis we know that there exists a derivation in $\tilde{G}$ of the form $A_o \overset{}{\underset{\alpha}{\Longrightarrow}} v_1 A_i \bar{u}_1$, where $\psi(w_1 A_i) = \psi(v_1 A_i) + \sum_{v' \text{ in } F(d)} \psi(v')$.

Since $A_i \overset{*}{\underset{x}{\Longrightarrow}} w_2$ is a normal sequence there exists a rule (compare with the basic step) $(A_i, x, v_2 \bar{u}_2)$ such that $\psi(w_2) = \psi(v_2) + \sum_{v' \text{ in } F(s)} \psi(v')$

and $\bar{u}_2$ contains the symbol $\bar{A}_j$ exactly once iff

$\bar{u}_1$ doesn't contain the symbol $\bar{A}_j$ and $F_j(s)$ is not empty or

$F_j(d)$ is empty and $F_j(ds)$ is not empty.

(Note, that u_2 may be chosen as an arbitrary subword of the underlying path!)

The derivation $A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} v_1 A_i \bar{u}_1$ in $\tilde{G}$ and the rule $(A_i, x, v_2 \bar{u}_2)$ of $\tilde{G}$

can now be composed to the derivation

$$A_o \overset{*}{\Longrightarrow} v_1 A_i \bar{u}_1 \overset{*}{\underset{x}{\Longrightarrow}} v_1 v_2 \bar{u}_1 \bar{u}_2 .$$

This derivation fullfills the statement, since:

$$\psi(w_1w_2) = \psi(w_1) + \psi(w_2) = \psi(v_1) + \sum_{v' \text{ in } F(d)} \psi(v') + \psi(v_2) + \sum_{v' \text{ in } F(s)} \psi(v')$$

$$= \psi(v_1v_2) + \sum_{v' \text{ in } F(ds)} \psi(v') \quad .$$

This last equation follows from the definition of F(ds) case 1.
Moreover $\overline{u}_1\overline{u}_2$ contains the symbol $\overline{A}_j$ only if $F_j(ds)$ is not empty. If $\overline{A}_j$ is contained in $\overline{u}_1\overline{u}_2$ then it is contained exactly once: either in $\overline{u}_1$ if d has a free A_j-loop, or in $\overline{u}_2$ if s has a free A_j-loop and d doesn't.

case 2:

The symbol A_i of $ds = (A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} w_1A_i \overset{*}{\underset{x}{\Longrightarrow}} w_1w_2)$ is created within some free loop of d.

By definition of F(ds) we have:

$$F_j(ds) = \begin{cases} \text{cons}(F_j(d), F_j(s)) & \text{for all } j \neq l \\ \text{cons}(\text{delete}(uA_l, F_l(d)), F_l(s)) & \text{for } j = l \text{, where} \\ & uA_i \text{ is the first word in the list } F_l(d) \text{ which contains} \\ & \text{the symbol } A_l\text{, moreover } l \text{ is the smallest index possible.} \end{cases}$$

From this we infer:

$$(*) \qquad \sum_{v' \text{ in } F(ds)} \psi(v') = \sum_{v' \text{ in } F(d)} \psi(v') + \sum_{v' \text{ in } F(s)} \psi(v') - \psi(uA_i) \quad .$$

From the induction hypothesis we know the existence of a derivation in $\tilde{G}$:

$A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} v_1\overline{u}_1\overline{A}_l$ such that:

$$(**) \qquad \psi(w_1A_i) = \psi(v_1) + \sum_{v' \text{ in } F(d)} \psi(v') \quad .$$

$\overline{A}_l$ is contained within the sentential form $v_1\overline{u}_1\overline{A}_l$ since the list $F_l(d)$ is not empty (at least contains uA_i).

Since uA_i is in the list $F_l(d)$ of free A_l-loops, we know that there exists a derivation $A_l \overset{*}{\underset{\lambda}{\Longrightarrow}} uA_lA_i$ in G with pathlength at most card(N).

Since $A_i \overset{*}{\underset{x}{\Longrightarrow}} w_2$ is a sequence, there exists a rule $(A_i, x, v_2\overline{u}_2)$ in R_1 such that:

$$(***) \qquad \psi(w_2) = \psi(v_2) + \sum_{v' \text{ in } F(s)} \psi(v')$$

and u_2 contains symbols from the path of this derivation. We will determine $\overline{u}_2$ later.

Therefore we can find a rule $(\overline{A}_l, xuv_2\overline{u}_2) \in R_2$ and a rule $(\overline{A}_l, x, uv_2\overline{u}_2\overline{A}_l) \in R_3$.

We may combine the α-derivation $A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} v_1\overline{u}_1\overline{A}_l$ in $\tilde{G}$ with each of this rules. This results in two different derivations:

(a) $A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} v_1\overline{u}_1\overline{A}_l \underset{x}{\Longrightarrow} v_1\overline{u}_1uv_2\overline{u}_2$ and

(b) $A_o \overset{*}{\underset{\alpha}{\Longrightarrow}} v_1\overline{u}_1\overline{A}_l \underset{x}{\Longrightarrow} v_1\overline{u}_1uv_2\overline{u}_2\overline{A}_l$.

For each of this derivations the property (i) from the statement is true, as can be seen by:

$$\begin{aligned}
&\psi(w_1) + \psi(w_2) = \psi(v_1) - \psi(A_i) + \sum_{v' \text{ in } F(d)} \psi(v') + \psi(w_2) \text{ , using } (**)\\
&= \psi(v_1) - \psi(A_i) + \sum_{v' \text{ in } F(d)} \psi(v') + \psi(v_2) + \sum_{v' \text{ in } F(s)} \psi(v') \text{ , using } (***)\\
&= \psi(v_1) + \psi(v_2) + \sum_{v' \text{ in } F(ds)} \psi(v') + \psi(uA_i) - \psi(A_i) \text{ , using } (*)\\
&= \psi(v_1) + \psi(v_2) + \psi(u) + \sum_{v' \text{ in } F(ds)} \psi(v')\\
&= \psi(v_1 v_2 u) + \sum_{v' \text{ in } F(ds)} \psi(v')
\end{aligned}$$

We now have to choose derivation (a) or (b) and to find $\overline{u_2}$ such that $\overline{u_1}\overline{u_2}$ (resp. $\overline{u_1}\overline{u_2}\overline{A_1}$) contains the symbol $\overline{A_j}$ exactly once iff the list $F_j(ds)$ is not empty.This can be done as follows:

In the case $j \neq 1$ $F_j(ds)$ is not empty iff $F_j(d)$ or $F_j(s)$ is not empty. Suppose $F_j(d)$ is not empty, then by induction hypothesis $\overline{u_1}$ contains the symbol $\overline{A_j}$ exactly once. Therefore $\overline{u_2}$ can be chosen not to contain $\overline{A_j}$. Suppose $F_j(d)$ is empty, then $F_j(s)$ is not empty and by definition of R_1 we can find $\overline{u_2}$ such that the symbol $\overline{A_j}$ is contained exactly once. This defines $\overline{u_2}$ for all symbols $\overline{A_j} \neq \overline{A_1}$.

In the case $j = 1$ things are different. Since $F_1(d)$ at least contains the word uA_i, the list $F_1(d)$ is not empty. Now if $F_1(s)$ is empty two subcases are possible:

(1) $F_1(ds)$ is empty, since uA_1 is the only word within this list. We can choose the derivation (a) using the rule $(\overline{A_1}, x, uv_2\overline{u_2}) \in R_2$.

(2) $F_1(ds)$ is not empty. We then use the rule $(\overline{A_1}, x, uv_2u_2\overline{A_1}) \in R_3$ and derivation (b).

In both subcases the property (ii) of the statement is fullfilled.

If $F_1(s)$ is not empty, then $F(ds)$ is not empty and as in (2) we use derivation (b).

Since we considered all possibilities we have proven the induction step. By induction the statement is proven for all normal derivations in G and therefore the inclusion $LL(G) \subset LL(\widetilde{G})$ holds true.

As a corollary of our main theorem one may show the following normalform for label grammars. The proof can be done using induction over the length of derivations and is omitted.

Corollary 'normalform for label grammars'

For each label language $L \in \mathcal{H}_\alpha(\mathcal{Y}_\Sigma)$ one can find a label grammar G such that

(i) $LL(G) = L - \{\lambda\}$.

(ii) if (A,x,w) is a rule within G, then $x \in X$ and $lg(w) \leq 2$.

Kindly referring to the literature we list some corollaries which can be obtained using the normalform for label grammars.

Corollary

If L is the homomorphic image of some Szilard language, then $L - \{\lambda\}$ is the image of a Szilard language under a nonerasing homomorphism.

Corollary

For each label grammar G it is decidable whether $LL(G)$ is empty or finite.

Corollary

Each infinite label language contains an infinite regular set.

Corollary

Each label language L contains a letter-equivalent context-free language $\tilde{L}$. That is: $\Psi(L) = \Psi(\tilde{L})$.

Corollary

Each label language can be defined by equations using the operations: union, leftconcatenation with a symbol, and shuffle combined with leftconcatenation.

Corollary

The language $L_1 := \{ a^i b a^j \mid i \geq j \geq 0 \}$ is a label language, whereas the language $L_2 := \{ a^i b a^i \mid i \geq 0 \}$ cannot be generated by a label grammar.

Literature

(74a) M.Höpner, 'Über den Zusammenhang von Szilardsprachen und Matrixgrammatiken', Inst.f.Informatik,Univ. Hamburg, Bericht Nr. 12 (1974).

(74b) M.Höpner, 'Eine Charakterisierung der Szilardsprachen und ihre Verwendung als Steuersprachen',Lect.Notes in Comp.Sc., 26 (1974).

(76) M.Höpner/ M.Opp 'About three equational classes of languages built up by shuffle operations', MFCS 76, Lect. Notes in Comp.Sc., 45 (1976).

(77) M.Höpner/ M.Opp 'Renaming and erasing in Szilard languages' , Inst.f.Informatik, Univ. Hamburg, to be published.

(74) M.Penttonen 'On derivation languages corresponding to context-free grammars', Acta Informatica, 3 (1974).