Languages Generated by Linear Schemas

ASTReNet Workshop 14
King's College London

Sebastian Danicic

March 30, 2007

Finite Paths of A Schema

If we think of the set of finite paths of a schema as a language, I am interested in what sorts of language can be generated by linear scehmas and where, if anwhere, they fit in the Chomsky Hierarchy.

Example

What is the language generated by:

```
while p(j)
{
    if q(k)
    {
      k = f(k);
      j = m(j)
    }
    else
    {
       k = g(k);
       j = h(j);
    }
}
```

Example

What is the language generated by:

```
while p(j)
{
    if q(k)
    {
      k = f(k);
      j = m(j)
    }
    else
    {
       k = g(k);
       j = h(j);
    }
}
```

$$L(S) = (pq(fm|gh))^* p$$

With Linearity Not all Regular Lanuages can be expressed

Can't even do $L(S) = aa$

With Linearity Not all Regular Lanuages can be expressed

Can't even do $L(S) = aa$
because

$$\{ \quad k = a(k); \quad j = a(j) \quad \}$$

is not linear.

Squashing

If we can *squash* different symbols onto the same symbols then we are in better shape.

Squashing

Given a language L we define a *squashing* to be any function on the symbols of L

Squashing a Language

Given a language L and a squashing f, we write

$$L \triangleright f$$

to be the new language obtained by squashing each element of L.

We now can do aa

$$
\begin{aligned}
S = & \\
& \{ \\
& \quad k = a(k); \\
& \quad j = b(j) \\
& \}
\end{aligned}
$$

$$L(S) \rhd \{b \rightarrow a\} = aa$$

What about a^*?

Nope!

What about a^*?

Nope!

Closest is:

while $p(j)$

{

$j = a(j);$

}

What about a^*?

Nope!

Closest is:

while $p(j)$

{

$j = a(j);$

}

$$L(S) = (pa)^* p$$

Restriction

Given a language L and a set of symbols S we write

$$L \restriction S$$

for the set of words in L having had all the symbols not in S removed.

Examples of Restriction

$$(pa)^*p \upharpoonright \{a\} = a^*$$

Interesting Question

What Languages can we define with Linear Schemas plus squashing and restriction?

What Languages can we define with Linear Schemas plus squashing and restriction?

Clearly all regular languages

What Languages can we define with Linear Schemas plus squashing and restriction?

Clearly, all free schemas give rise to regular languages.

What Languages can we define with Linear Schemas plus squashing and restriction?

... so what about non-regular ones.

What Languages can we define with Linear Schemas plus squashing and restriction?

... so non-regular languages can only be generated by non-free schemas

The "Montreal Boat Trip" example

```
while p(j)
 {
    if q(k) k=f(k);
    else
    {
        k=g(k);
        j=h(j);
    }
 }
```

The "Montreal Boat Trip" example

```
while p(j)
 {
    if q(k) k=f(k);
    else
    {
        k=g(k);
        j=h(j);
    }
 }
```

$$L(S) = (pq(f|gh))^* p$$

The "Montreal Boat Trip" example

```
while p(j)
 {
    if q(k) k=f(k);
    else
    {
        k=g(k);
        j=h(j);
    }
 }
```

$$L(S) = (pq(f|gh))^* p$$

No. So what about:

The "Montreal Boat Trip" example

```
while p(j)
 {
    if q(k) k=f(k);
    else
    {
        k=g(k);
        j=h(j);
    }
 }
```

$$L(S) = (pq(f|gh))^* p$$

No. So what about:

$$L(S) = (pq(f|gh))^* pqgh|p$$

The "Montreal Boat Trip" example

```
while p(j)
 {
    if q(k) k=f(k);
    else
    {
        k=g(k);
        j=h(j);
    }
 }
```

$$L(S) = (pq(f|gh))^* p$$

No. So what about:

$$L(S) = (pq(f|gh))^* pqgh|p$$

I think this is right - so non-free does not imply non-regular.

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

What about this one?

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

Notice, after AB the state is always the same, similarly for BA.

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

So we can't have $kABlABm$ for any finite sequences k,l m.

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

Similarly we can't have $kBAlBAm$ for any k,l m.

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

How about $A^*(AB|\lambda)B^*$ | $B^*(BA|\lambda)A^*$?

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

or is it $(A^*|B^*)(AB|BA|\lambda)(A^*|B^*)$?

Another example

```
while p(x,y)
 {
    if q(k)
    {
        x=A1(x);
        y=A2();
    }
    else
    {
        y=B1(y);
        x=B2();
    }
  k=h(k);
}
```

or is it $A^+B^+A^* \mid B^+A^+B^* \mid A^* \mid B^*$?

Schema Languages

Are there any schemas whose language is not regular?

Conjecture

A schema is free if and only if its language can be expressed as a regular expression where each function symbol occurs exactly once.