Re: metrics on context−free languages

Source: http://coding.derkeiler.com/Archive/General/comp.theory/2005−08/msg00085.html

- *From*: sasha mal <sasha.malREMOVEIT@xxxxxxxxxxxxxxxxxxxxxxx>
- *Date*: Mon, 29 Aug 2005 12:11:52 +0200

Thank you very much for giving your approximation schemes. Some questions:

You say:

[

SCHEME #1: Characterization by Subwords
Any regular language can be characterized by the set of n−glyphs (for n = 0, 1, 2, ...) which it does not contain as any subword. A minimal set is always finite. Cutting it off at n = N gives you a family of approximating regular languages R_0, R_1, R_2, ..., that converges to the regular language R.

An example: the language R = (ab)* over the alphabet {a,b,c} has the minimal set {c,aa,bb}.

The characterization is unique, if you also add in extra markers for the word boundaries (^ for beginning, $ for end), in which case the set above is modified to { $$, ^^, $^, $a, $b, $c, a^, b^, c^ }, taken in union with the set { c, aa, bb, ^b, a$ }.

The same is true for context−free languages −− except the sequence is infinite, but the languages R_0, R_1, R_2, ... are still all regular.

]

What exactly ("the same") is true for context−free languages? That the characrerization is unique? Is this not trivial for all languages, since the set of all n−glyphs with ^ and $ contains also all words which don't belong to the language. And the sets {^w$ | w not in L} are different for different L. Or that the languages of the sequence are regular? This is also true, since they are finite.
Or that the minimal set (without ^ and $) is finite in the context−free case? I don't see that. Could you give a reference?

In SCHEME #2, you allow introduction of a star on the right side.
Although one can do it automatically in some cases, i.e. a rule
"A >= term1 + (term2) A" is transformed to "A >= term2* term1"
and

"A >= term1 + A (term2)" is transformed to "A >= term1 term2*"
but I don't see that it introduces all possibles stars. For instance,
"A >= a + aAa" gives the same language as "A >= a(aa)*". Somehow one
needs a notion of a sufficient condition on the occurences of star.

Without star, the second sceme seems to generate languages of words of
bigger and bigger length. Besides, I don't need convergence from below −
it is a simple as generating the words of L of bigger and bigger length.
With star, it improves − but still converges from below.
A simple

SCHEME #5
would give the languages
R_k := (words_{<k}(L) union prefixes_k(L))
of words of length at most k−1 of L and prefixes of length =k of words of L.

Your SCHME #3 is good, but one has to decide whether L = L' for some
context−free languages. I agree with you that one can reduce the general
equivalnce problem to this. By the way, in your example, you probably meant
c\N(n) = S(n+1) (and not =S(n), as you wrote)
since
c\N(n) = c \ N P* (V P*)^n = S P* (V P*)^n = N V P* P* (V P*)^n =
N V P* (V P*)^n = N (V P*)^{n+1} = S(n+1).

Then , you wrote
[
The automaton above, though infinite, is the minimal deterministic
automaton...
]
What do you mean by minimal in the infinite case?
By the way, where do you have SCHEME #3 from?

In SCHME#4, what you define is an automaton that forgets everything that
is deeper in the stack than n.

Thank you for all that.
My intention was to approximate from above and to get the language L as
an element of the approximation sequence if it turns out to be regular
by chance. Although the corresponding decision problem is undecidable,
I'm completely happy with the fact that the next element of the sequence
(after another reresenation of L) would either give the same language L
or would never be computed.

Regards,
sasha.
.

- Prev by Date: ***Roots of a polynomial***
- Next by Date: ***binary trees, successor function***

- Previous by thread: ***Roots of a polynomial***
- Next by thread: ***binary trees, successor function***
- Index(es):
    - ***Date***
    - ***Thread***