Mathematical Methods for Computer Science 1 Spring 2017

Series 10

Sylvain Julmy

1

If all words from $\{0\}^*$ are pairwise L-inequivalent, it means that the total number of L-equivalence classes is infinite, which implies that L is not-regular. In other words, we show that any language over $\{0\}^*$ are either regular or not. If the language is regular, then its number of equivalence classes is finite.
Now we demonstrate that to be non-regular, all words from a language over $\{0\}^*$ are pairwise inequivalent.
In the unary alphabet, any language L can be seen as a subset of $\mathbb{N}$, because only the length of a word inn L is important. Therefore, two number a, b in $\mathbb{N}$ are equivalent if and only if $\forall n.(n + a \in L \leftrightarrow n + b \in L)$. Then, two words from L are equivalent if they are equivalent in some arithmetic modulus k, where $k = b - a$.
If we can define such an arithmetic with modulus, then the language is finite because modulus k over the infinite keep the number in $[0; k - 1]$.
If such an arithmetic can't be found, then each word is inequivalent to each other ones in the language since $\forall n.(n + a \in L \leftrightarrow n + b \in L)$.

2

(a)

The operation L/a remove the element of L where $prefix \cdot a \in L$. For example, let $L = \{aab, aaab\}$, then $L/b = \{aa, aaa\}$.
Clearly, $L/a \cdot \{a\} \subset L$, because from L we remove the all the words that don't end with an "a" and we add an "a" to each element from L that end with an "a" and remove that letter.
Assume $L/a \cdot \{a\} = L$, then its valid for any L, then let $L = \{aab, bba\}$, we have $L/a = \{bb\}$ and finally $L/a \cdot \{a\} = \{bb\} \cdot \{a\} = \{bba\}$, which is not equal to L. Therefore $L = L/a \cdot \{a\}$ is not a tautology.

(b)

$$u \sim_L v \Rightarrow u \sim_{L/a} v$$

Let M be the automaton accepting L, then from

$$u \sim_L v \Leftrightarrow \forall a, b.(aub \in L \leftrightarrow avb \in L)$$

we can characterize the relation in M

$$u \sim_L v \Leftrightarrow \forall q \in Q(M).(\hat{\delta}(q, x) = \hat{\delta}(q, y))$$

It means that u and v are equivalent w.r.t. L if they have the same behavior with M (is this bisimulation ?). We know that L/a is the automaton that accept the words that are constructed from L by either removing their last letter l if $l = a$ or by not taking it if $l \neq a$. Therefore, in the automaton M' accepting L/a, the words u and v would reach the same state in M' and are either accepted or not by M'.
Assume u is accepted by M' and v not, it means that, on a transition, u and v don't reach the same state in M', therefore u has a valid prefix and v does not, which implies that $u \not\sim_L v$ which is a contradiction.

(c)

Given an automaton M accepting the language L, in order to accept L/a, we have to remove the final states of M and then put as final the states of M' which can reach the final states of M by using the transition $\delta(q_i, a) \in F(M)$.
Formally, we have $Q(M') = Q(M) \setminus F(M)$ and $F(M') = \{q \in Q(M) | \delta(q, a) \in F\}$, we don't touch anything else in M'. M' accept only words that concatened with a are in L.

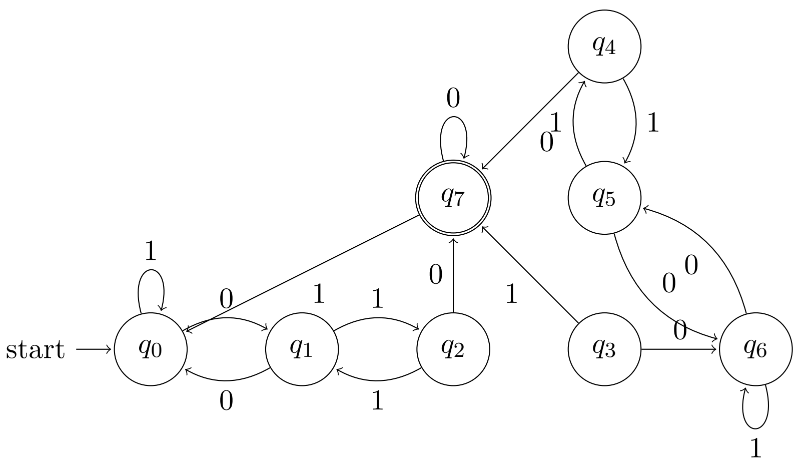
(d)

By the construction use in c), we remove final states from M, in the worst case, we don't have to remove the final states because they can reach them selves with the a letter from L/a, but we never add states.
These two automata does not have the same number of states, for example the automaton accepting the language $L = \{aa\}$ have three state, and the automaton accepting $L/a = \{a\}$ only has one state.

3

4

The automaton :



We can also remove q_3, which cannot be reach from any state.

q_1	$\times$						
q_2	$\times$	$\times$					
q_3	$\times$	$\times$	$\times$				
q_4	$\times$	$\times$		$\times$			
q_5	$\times$		$\times$	$\times$	$\times$		
q_6		$\times$	$\times$	$\times$	$\times$	$\times$	
q_7	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
	q_0	q_1	q_2	q_3	q_4	q_5	q_6