Monadic Second Order logic over higher order Böhm trees

Léo Tible, *under the supervision of* Matthew Hague

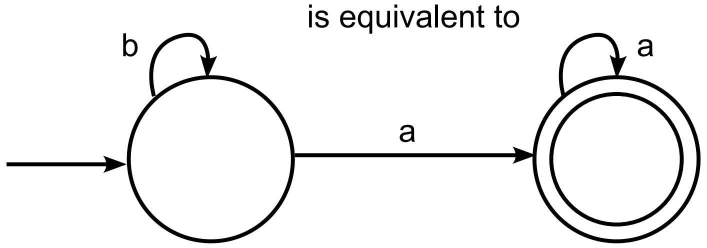
Royal Holloway, University of London

MSO and Automata

A known and celebrated result in automata theory is that Monadic Second Order Logic and non-deterministic automata are equivalent, that is, every language that can be expressed through one can be expressed through the other [1].

$$\exists x, \forall y, \forall z, a(x) \land \big(a(y) \land s(y,z) \Rightarrow a(z)\big)$$

is equivalent to



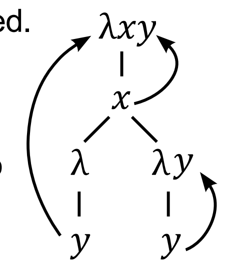
This result holds for words and for trees.

[1] Igor Walukiewicz. Pushdown processes : Games and model-checking.

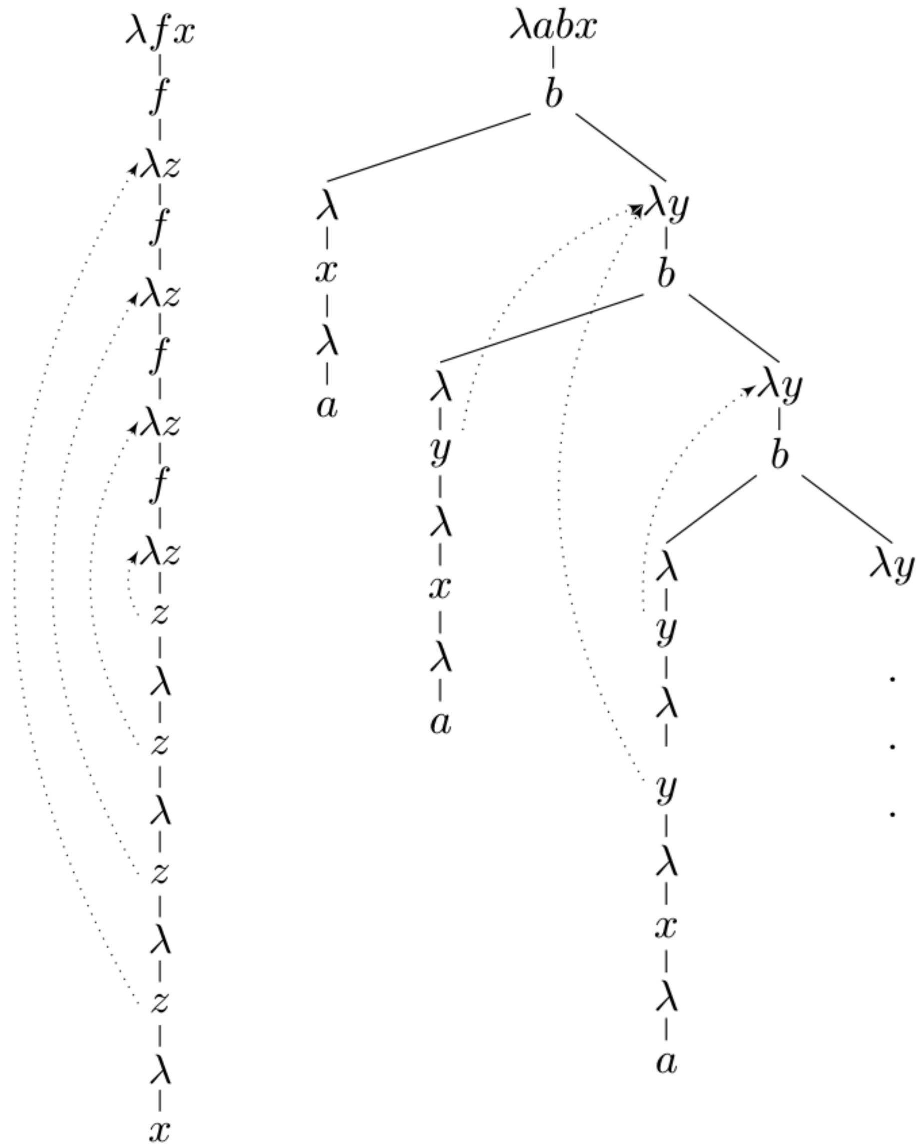
Böhm Trees

Programs can be represented by trees. However, if you only have part of a program, you need a way to represent it. Böhm trees are like program traces for incomplete programs [2]. We use the lambda-calculus notation, and add binder edges from variables to the lambda that introduced them. This allows also a really compact representation of lambda terms as only a finite number of variables is needed.

Example: the lambda term $\lambda xy.xy(\lambda z.z)$ is represented by a Böhm tree with only two different variables



[2] Takeshi Tsukada and C.-H. Luke Ong. Compositional higher-order model checking via Ω-regular games over böhm trees



Example: on the left, the term $\lambda f x. f\left(\lambda z_{1}.\left(f\left(\dots\left(\lambda z_{4}.z_4\left(\dots(z_1 x)\right)\right)\right)\right)\right)$, part of the monster sort, is represented with only 3 variables.
On the right, an infinite term is represented with 4 variables.
Only non-trivial binder edges are shown.

ADTA

Alternating dependency tree automata are the extension to higher order Böhm trees of alternating parity tree automata over trees. There are two types of transitions:

Lambda nodes: $p \xrightarrow{\lambda x_1 x_2 \dots x_n} Q$

On a lambda node in state p labeled by $\lambda x_1 x_2 \dots x_n$, the resulting state is in Q.

Variable nodes: (p,q) $\xrightarrow{x} Q_1, Q_2, \dots, Q_n$

On a variable node in state p, binded by a node in state q, and labeled by x, the i-th child is in state Q_i.

Alternating: The resulting state is chosen by the opponent among the set of state on the right side of the transition.
Dependency : for variable nodes, transitions depend also on the state of the binder node.

MSO extension

Do the celebrated result holds for MSO over Böhm trees ? With the binder predicate $b(x,y)$ that is "the node y is the binder node of the node x", it might. We can prove the following theorem

Given an MSO formula, one can construct an equivalent ADTA.

However, showing the reciprocate is a difficult task as alternation together with binder edges seem to means that the all history of the run must be remembered. That does not seems doable with MSO.
However, no counter-example has been found and therefore, the result might hold anyway.