HINTS FOR HW 4

1 QUESTION ABOUT ALTERNATING FINITE AUTOMATA (afa)

PART A:

Basically, the first part asks you so show how to eliminate those moves in which the input head does not move. This may appear tricky because if you look at the computation tree, there may be infinite paths in which the input head does not move. You may know that a similar problem arise in converting non-deterministic finite automata to deterministic ones.

Although we normally think of a alternating machine as having two kinds of basis functions only (AND and OR), it is clear that to merge several (possibly infinitely many) moves than one move, we need to allow more general Boolean functions which may have arity greater than 2. In fact, I will allow you to attach a **generalized Boolean function** of the form

$$f : \{0, 1, \perp\}^k \to \{0, 1, \perp\}$$

to any state q. The arity of this state is k.

The key idea is to define a generalized Boolean function $f_{q,a}(x_1, \ldots, x_k)$ to each state q and each input symbol a: this function is the least fixed point solution for a finite state diagram corresponding to the ϵ-moves, and whose arguments $x_1, \ldots, x_k$ corresponds to the non-ϵ-moves out of this finite state diagram. For each assignment of values $a_i \in \{0, 1, \perp\}$ to x_i $(i = 0, 1, \ldots, k)$, we can define the least fixed point value for the states of this diagram. Then $f_{q,a}(a_0, \ldots, a_k)$ is just the value of the starting state q in this diagram. NOTE: this least fixed point solution is similar to what we do in space-bounded choice computations.

Another technical problem is this: our definition of generalized afa insists that we attach a function to each state q. We can overcome this restriction as follows: if M causes a non-epsilon transition to state q and the previous symbol read is a, we will enter state designated $[q, a]$. The function attached to $[q, a]$ is $f_{q,a}$.