Let $G$ be a (two-player deterministic finite) reachability game. The underlying graph is $G = (V, E)$; we make the assumption that every vertex has at least one outgoing edge. We write $V = V_{\text{Eve}} \uplus V_{\text{Adam}}$ for the set of vertices controlled by Eve and Adam. The reachability objective is $\text{REACH}(F) = V \ast F V^\omega$, i.e. the set of paths visiting $F \subseteq V$ at least once.

The goal of this problem is to construct efficient algorithms for computing $W_{\text{Eve}}(\text{REACH}(F))$, the set of vertices from which Eve has a winning strategy for the reachability objective. The important parameters here are $n$ the number of vertices and $m$ the number of edges.

For representation purposes, the game is given in the following way: for each vertex $v$, one bit describes whether it is controlled by Eve or Adam, and then we list all the successors of $v$.

The objective $F$ is given as a boolean vector over $V$. To compute $W_{\text{Eve}}(\text{REACH}(F))$ we represent it as well using a boolean vector over $V$.

**Question 1:** We write $\mathcal{P}(V)$ for the set of subsets of $V$. Let us consider the operator $\text{Pre}_F : \mathcal{P}(V) \rightarrow \mathcal{P}(V)$ defined by

$$\text{Pre}_F(X) = F \cup \{ v \in V_{\text{Eve}} : \exists (v, v') \in E, v' \in X \} \cup \{ v \in V_{\text{Adam}} : \forall (v, v') \in E, v' \in X \}.$$ 

Prove that $\text{Pre}_F$ is a monotone operator with respect to inclusion: if $X \subseteq X'$ then $\text{Pre}_F(X) \subseteq \text{Pre}_F(X')$.

**Solution:** “Oh come on!” is an acceptable answer.

**Question 2:** A prefixed point of $\text{Pre}_F$ is a set $X \subseteq V$ such that $\text{Pre}_F(X) \subseteq X$. Prove that:

(i) There exists a prefixed point of $\text{Pre}_F$.

(ii) $W_{\text{Eve}}(\text{REACH}(F))$ is a prefixed point of $\text{Pre}_F$.

(iii) The intersection of two prefixed points of $\text{Pre}_F$ is another prefixed point of $\text{Pre}_F$.

(iv) There exists a least prefixed point of $\text{Pre}_F$.

(v) $W_{\text{Eve}}(\text{REACH}(F))$ is the least prefixed point of $\text{Pre}_F$.

**Solution:**

(i) $V$ is a fixed point of $\text{Pre}_F$.

(ii) Easy.

(iii) Easy.

(iv) Consequence of the previous point.

(v) It is a fixed point. To prove the converse, consider the complement (done in the lecture).
**Question 3:** Construct an algorithm for computing $W_{Eve}(REACH(F))$ based on Knaster - Tarski fixed point theorem, and show that it has complexity $O(n \cdot m)$.

**Solution:** Knaster - Tarski fixed point theorem gives an algorithm for computing the least prefixed point: it says that the sequence $\langle \text{Pre}_k(\emptyset) \rangle_{k \geq 0}$ is eventually constant and its limit is the least prefixed point of $\text{Pre}_F$. For each $k$ we construct $\text{Pre}_k(\emptyset)$ in a naive way, which has complexity $O(m)$. Since there are $n$ iterations we get complexity $O(n \cdot m)$.

We want to improve the complexity to $O(n + m)$. Note that since every vertex has at least one outgoing edge, $n \leq m$, so this is actually $O(m)$.

**Question 4:** Prove that in time $O(m)$ we can get the following equivalent representation of the game: for each vertex $v$, one bit describes whether it is controlled by Eve or Adam, and then we list all the predecessors of $v$.

**Solution:** We go through all edges, and add them to the corresponding list.

**Question 5:** Prove that the algorithm given in pseudo-code below computes $W_{Eve}(REACH(F))$ and that its complexity is $O(m)$.

**Algorithm 1:** The linear time algorithm for reachability games.

**Data:** A reachability game.

**Function** Attractor():

- $A \leftarrow F$
- for $v \in V_{Adam}$ do
  - number-edges($v$) $\leftarrow$ number of outgoing edges of $v$
  - $k \leftarrow 1$
  - $X_k \leftarrow F$
- repeat
  - for $v \in X_k$ do
    - Treat($v$)
    - $k \leftarrow k + 1$
  - until $X_k = X_{k+1}$
- return $A$

**Function** Treat($v$):

- for $e = (u, v) \in E$ do
  - if $u \in V_{Adam}$ and $u \notin A$ then
    - number-edges($u$) $\leftarrow$ number-edges($u$) - 1
    - if number-edges($u$) = 0 then
      - Add $u$ to $A$
      - Add $u$ to $X_{k+1}$
  - if $u \in V_{Eve}$ and $u \notin A$ then
    - Add $u$ to $A$
    - Add $u$ to $X_{k+1}$

**Solution:** To prove correctness we need a suitable invariant: $X_k = \text{Pre}_k(\emptyset)$.

A vertex can be added to $A$ at most once, implying that an edge can be considered at most once, so the complexity is $O(m)$. 