Lecture 9: Büchi Games over Infinite Graphs

We shall first generalize the definition of alternating automata over finite words. Recall that in our definition of alternating automata, the set of states Q is divided into two sets $Q_\forall$ and $Q_\exists$ and every transition out of any $Q_\forall$ state is interpreted as a logical and while transitions out of any $Q_\exists$ state are interpreted as logical ors.

Instead of associating the type of the transition to a state, we could associate it with a (state, letter) pair. In other words, we define an alternating automaton to be a tuple $(Q, \Sigma, \delta, s, F)$, where for each $q \in Q$ and $a \in A$, $\delta(q, a) = \forall S$ or $\delta(q, a) = \exists S$, for some some set S of states. The interpretation of these transitions is the obvious one: if $\delta(q, a) = \forall S$ then, we need to start one copy of the automaton for each state in S to read the rest of the input, and $\delta(q, a) = \exists S$ represents a nondeterministic choice and we need to pick one state from S to read the rest of the input.

For each such automaton A and word w we can associate a game. The only difference w.r.t the game defined in the Lecture 6 is that, whether a state q in the ith copy belongs to the automaton or the pathfinder depends on the input letter a_i. if $\delta(q, a_i)$ is a $\forall$ transition then the state belongs to the pathfinder and it belongs to the automaton otherwise. And with this definition we can reprove all the results in Lecture 6 once again without any difficulty.

We can generalize this further as follows: We define an alternating automaton to be a tuple $(Q, \Sigma, \delta, s, F)$ where $\delta(q, a)$ is positive boolean formula over Q. That is, a formula constructed using elements of Q and the operators $\vee$ and $\wedge$ as well as the constants false and true. In particular, negation is not permitted. Before venturing into the precise definition of runs etc, let me explain this definition informally. Suppose $\delta(q, a) = (q_1 \vee q_2) \wedge q_3$. This means that in order to read the word aw starting at state q, we must start one copy of the automaton at state q_3 to read w and a second copy starting at state q_1 or q_2 to read w. We could also start three copies, one each at states q_1, q_2 and q_3 (recall, that all the copies must accept for a run to be accepting).

Let $S \subseteq Q$ and let ϕ be a formula over Q. We define when S satisfies the formula ϕ as follows:

$$\begin{array}{ll} S \models \mathsf{true} & \text{always} \\ S \models q & \text{if } q \in S \\ S \models \phi_1 \wedge \phi_2 & \text{if } S \models \phi_1 \text{ and } S \models \phi_2 \\ S \models \phi_1 \vee \phi_2 & \text{if } S \models \phi_1 \text{ or } S \models \phi_2 \end{array}$$

We can now define a run of an alternating automaton from state q on a word $w = a_1 a_2 \ldots a_n$ to be a tree labelled by Q satisfying the following properties

1. The tree has $n + 1$ levels.

2. The root is labelled by q.

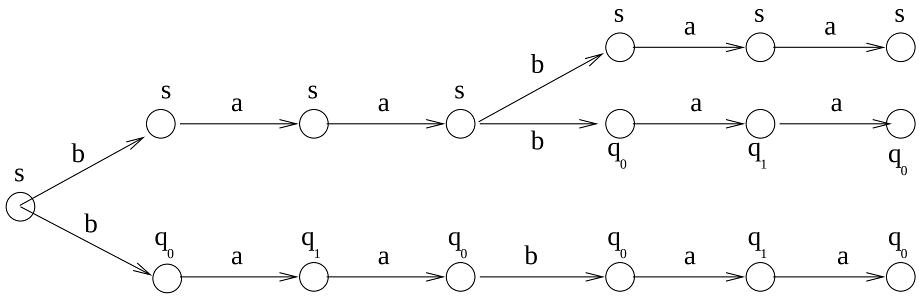
3. If a node at level i is labelled by q and then the labels of its children constitute a set S such that $S \models \delta(q, a_i)$.

The run is accepting if all the leaves of this tree are labelled by states in F and a word w is accepted if there is an accepting run starting at state s on w.

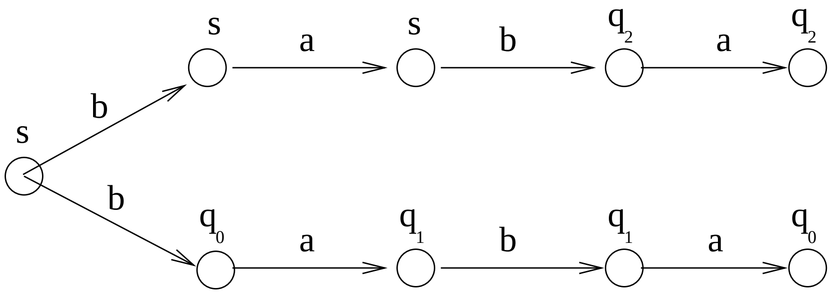
We illustrate these definitions with the automaton $A_1 = (\{s, q_0, q_1, q_2\}, \{a, b\}, \delta, s, \{s, q_0, q_2\})$ where

$$\begin{aligned}
\delta(s,a) &= s \vee q_2 \\
\delta(s,b) &= (s \vee q_2) \wedge (q_0 \vee q_2) \\
\delta(q_0,a) &= q_1 \\
\delta(q_0,b) &= q_0 \\
\delta(q_1,a) &= q_0 \\
\delta(q_1,b) &= q_1 \\
\delta(q_2,a) &= q_2 \\
\delta(q_2,b) &= \mathsf{false}
\end{aligned}$$

Here are is an accepting run of this automaton on the input *baabaa*:



The set $\{s, q_0\}$ satisfies $(s \vee v_2) \wedge (q_0 \vee q_2)$. Here is an accepting run on *baba*.



Here we also use the fact that $q_2 \models (s \vee q_2) \wedge (q_0 \vee q_2)$.

Exercise: What is the language accepted by this alternating automaton?

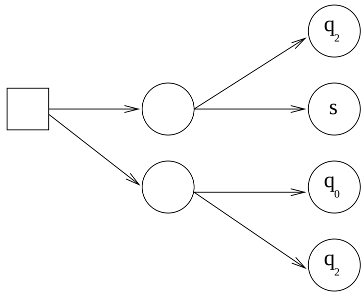
We shall say that a run is *positional*, if for each level i and any two states in level i labelled by the same state, the entire subrun (subtree) rooted at these states is identical.

1 Games from Automata

How do we associate a game with a given automaton A and word w? In the "simpler" version of alternating automaton considered earlier, nodes where the choice is nondeterministic belong to the automaton while nodes with conjunctive choice belong to the pathfinder.

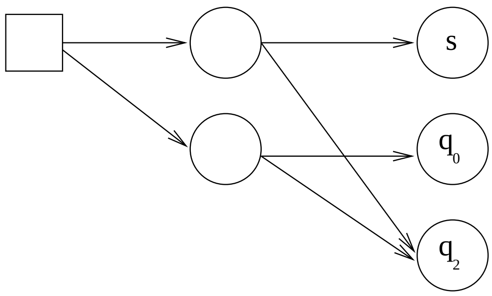
However, in the more elaborate model a transition may involve both nondeterministic and conjunctive choices. (For example consider $\delta(s, b)$ above.)

The idea is to first translate each transition $\delta(q, a)$ into a small game graph. First we construct the tree representation of this formula with the operators in the formula constituting the interior nodes and the states in the formula constituting the leaves. For example, the formula $(s \wedge q_2) \vee (q_0 \wedge q_2)$ gives:

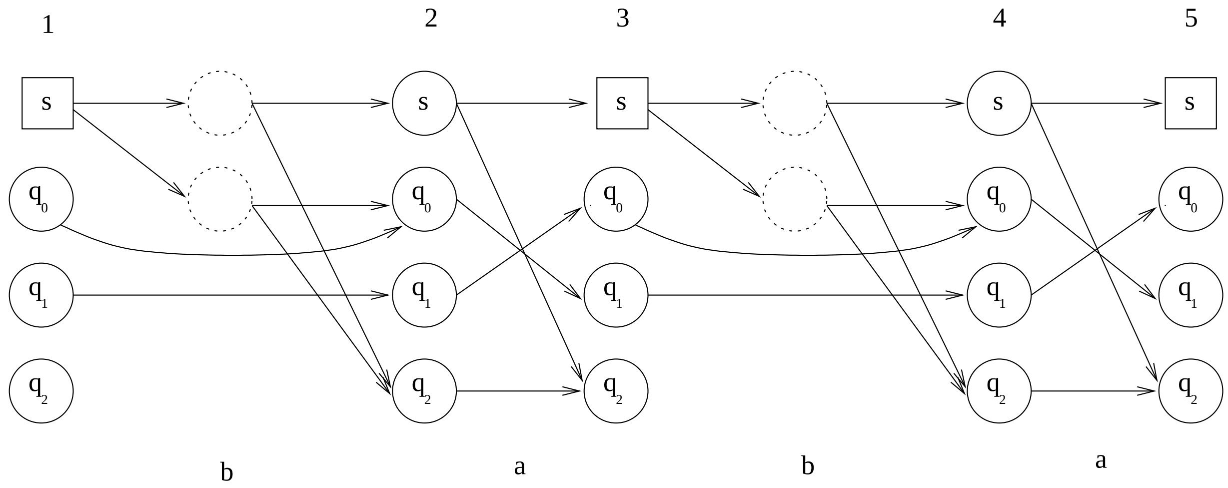


The nodes corresponding to the $\wedge$ operator are treated as pathfinder nodes and and those corresponding to $\vee$ operator belong to the automaton. The leaves of this tree are labelled by states.

We then combine the leaves labelled by the same state. In the above example, we get:



Finally, the game graph corresponding to an automaton A and word $a_1a_2 \ldots a_n$ is constructed using $n + 1$ copies of the set state set Q. The levels are connected up as follows: Pick the game graph associated with $\delta(q, a_i)$, identify its root with the copy of q at level i and its leaves with their corresponding copies in level $i + 1$. Thus, the type of a state q at level i is determined by the outermost operator in $\delta(q, a_i)$. Here is the game graph associated with the automaton A (described earlier) on the input word *baba*.



The state s belongs to the pathfinder at levels 1 and 3 since the first and third letters are b. It belongs to the automaton at levels 2,4 and 5. Also notice that there are states (corresponding to the operators in the transition formulas) that appear in between the levels.

We can show that an automaton A accepts a word w if and only if the player automaton has a positional winning strategy in the game $\mathcal{G}(A, w)$. Recall that in games played on DAGs either the automaton or the pathfinder always has a positional winning strategy. So we might as well restrict our attention to positional winning strategies.

Fix any positional strategy f for the automaton in a game G. By $\mathcal{R}(f, p)$ we shall refer to the subgraph consisting of all positions in the game graph that appear in some play consistent with f and beginning at position p. (We shall write $\mathcal{R}(f)$ if p is the copy of s in level 1.)

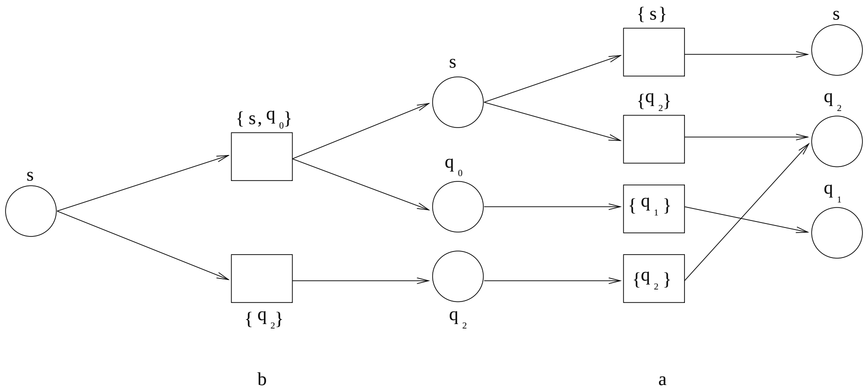
We can show that if f is a positional winning strategy of the automaton then $\mathcal{R}(f)$ corresponds to the folding into a DAG (where nodes with identical subtrees rooted below them have been identified) of some positional accepting run of the automaton.

We wish to reformulate this game into a more useful form. First note that if G_ϕ is the game graph obtained from a formula ϕ, and f is ANY strategy for the automaton in this game then the set of labels of the leaves in $\mathcal{R}(f, r)$, where r is the root (the node corresponding to the top level operator), satifies ϕ. Conversely, note that if a set S satisfies a formula ϕ then there is a strategy f_S for the automaton that ensures that the leaves in $\mathcal{R}(f_S, r)$ are labelled by a subset of S.

Thus a strategy for the automaton over a graph of the form G_ϕ corresponds to restricting the set of possible outcomes (i.e. the state reached at the end of a play) to some subset S that satisfies ϕ (or if ϕ is false, then the automaton is stuck and cannot make a move.) and which of these states in S is reached at the end of a particular play is fixed by the choices made by the pathfinder.

Thus, we can reformulate the game $\mathcal{G}(A, w)$ equivalently as the follows: The game begins at state s at level 1. The automaton picks a subset X_1 of Q that satisfies $\delta(s, a_1)$. The pathfinder then picks an element q_1 of X_1. The automaton then picks a subset satisfying $\delta(q_1, a_1)$ and then the pathfinder picks a state and so on till we reach some state q_n. The automaton wins if q_n is in F and the pathfinder wins otherwise (with the understanding that the pathfinder wins if at any point the automaton is unable to make a move.) Notice that if X satisfies $\delta(q,a)$ then it makes no sense for the pathfinder to play any set X' with $X \subseteq X'$. So we might as well restrict the choices available to the automaton to minimal subsets satisfying ϕ. Henceforth by $\mathcal{G}(A,w)$ we shall refer to this game.

Here is the game corresponding to the earlier defined automaton A on input ba.



In this setting it is quite easy to see that if winning strategies for the automaton on $\mathcal{G}(A,w)$ correspond to accepting runs for A on w and positional winning strategies correspond to positional accepting runs for A on w. Thus we have the following theorem:

Theorem 1 *An alternating automaton A accepts a word w if and only if the player automaton has a positional winning strategy in the game $\mathcal{G}(A,w)$.*

As discussed in Lecture 6, this immediately results in the equivalence of nondeterministic and alternating automata. The nondeterministic automaton simulates the runs of the the alternating automaton level by level and by keeping just one copy of each state that appears at a level.

Corollary 2 *For every alternating automaton A, there is a nondeterministic automaton A' such that $L(A) = L(A')$.*

Is it easy to complement these generalized alternating automata? To complement alternating automata, the standard technique is as follows: First transform the automaton A into another automaton A' so that the game graph $\mathcal{G}(A',w)$ is simply the game graph $\mathcal{G}(A,w)$ with the roles of the two players interchanged. Secondly, complement the accepting condition. A' with the complement accepting condition accepts a word if and only the automaton wins in $\mathcal{G}(A',w)$ if and only if the pathfinder wins in $\mathcal{G}(A,w)$ if and only if A does not accept w.

How to construct such an A'? Here we find the original definition of the game to be more useful. Recall that for $\delta(q,a)$ we simply constructed a game graph where the nodes labelled by $\wedge$ belongs to the pathfinder while $\vee$ nodes belong to the automaton. Thus, by simply interchanging the $\vee$ and $\wedge$ operators we get a game graph where the roles of the pathfinder and automaton are interchanged. Given a formula ϕ we write ϕ^d (i.e. the dual of the formula ϕ) for the one obtained by replacing $\wedge$ and $\vee$ by $\vee$ and $\wedge$ respectively (and false is replaced by true and vice versa). Thus, we have the following theorem:

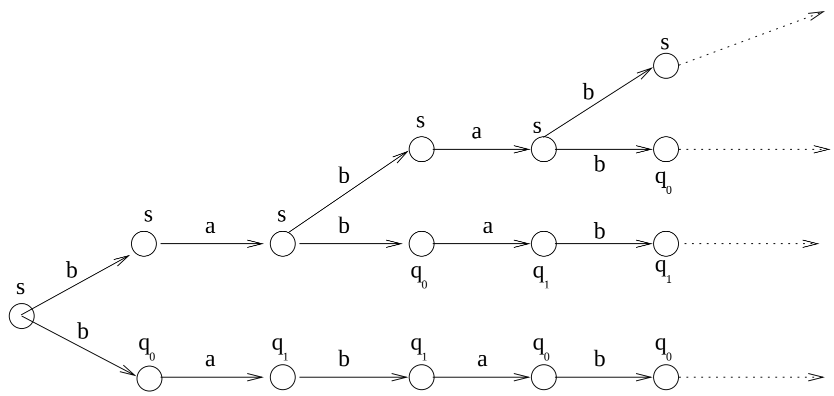
Theorem 3 *Let $A = (Q, \Sigma, \delta, s, F)$ be an alternating automaton. Then, the automaton $A' = (Q, \Sigma, \delta^d, s, Q \setminus F)$, where $\delta^d(q, a) = \delta(q, a)^d$, accepts the complement of the language accepted by A.*

2 Alternating Büchi Automata

An *Alternating Büchi Automaton* is a tuple $A = (Q, \Sigma, \delta, s, F)$ where $\delta(q, a)$ is a positive boolean formula over Q. Thus it is essentially a alternating automaton treated as an automaton over infinite words. The run of such an automaton over an infinite word $a_1 a_2 \ldots a_n \ldots$ is a infinite tree labelled by states from Q satisfying:

1. The root is labelled by q.
2. If a node at level i is labelled by q and then the labels of its children constitute the set S such that $S \models \delta(q, a_i)$.

A run is accepting if every complete path through the run tree is infinite and visits the set F infinitely often. Here is an accepting run of the automaton A_1 on the input *bababa*...



As usual we say that a runtree is positional if it has the following property: for any two nodes labelled by the same state at the same level, the subtrees rooted at these two nodes are identical.

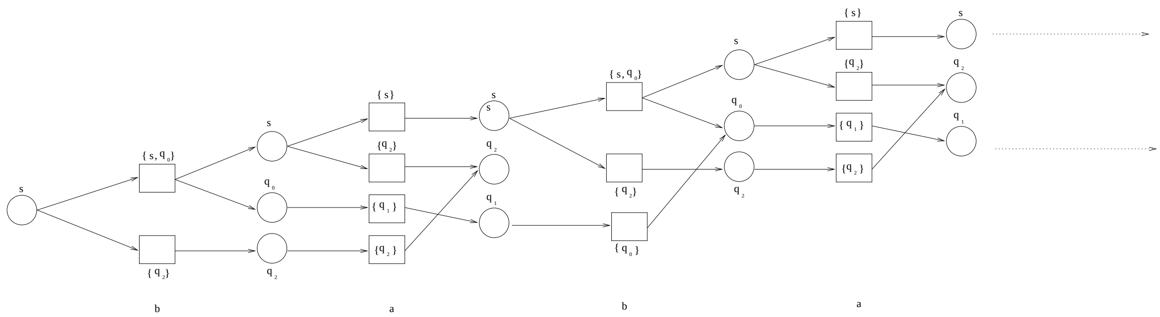
Clearly the class of languages accepted by Alternating Büchi automata subsumes the class of ω-regular languages. What about the converse? Can we simulate every alternating Büchi automaton using a traditional Büchi automaton? In the case of finite words, we simulated the alternating automaton level by level using the fact that we need to keep only one copy of each state at a level. This in turn relied on the fact that alternating finite automata have positional accepting runs on every word they accept and this was proved by establishing that the player automaton has a positional winning strategy in the game (associated with the automaton and any word w).

We shall follow the same route here. First we associate a game with every automaton–word pair. Then we show that the word is accepted if and only if the player automaton has a positional winning strategy in the game. This in turn implies the existence of positional accepting runs, which in turn allows us to simulate alternating automata via nondeterministic automata.

2.1 The Game $\mathcal{G}(A, \sigma)$

Let A be an automaton and a ω-word $\sigma = a_1 a_2 \ldots$ we associate a game as follows. The nodes are classified as those in level $1, 2, \ldots$. At level $2i - 1$ there are nodes labelled by the elements of Q. At level $2i$ there are nodes labelled by elements of 2^Q. Nodes in the odd levels belong to the automaton and the nodes at the even levels belong to the pathfinder. From a node labelled q at level $2i-1$, there is an edge to a state labelled X in level $2i$ if and only if $X \models \delta(q, a_i)$. From a node labelled X at level $2i$ there is an edge to a node labelled q at level $2i+1$ if and only if $q \in X$. Thus, the automaton picks sets of states that satisfy the transition formula and the pathfinder picks a state from this set.

Since we are interested only in the result of games starting at the state s at level 1 we might as well omit all the states that are not reachable in any play starting at this state. We may also assume that the automaton will never play an X' if there is an $X \subseteq X'$ that also satisfies the transtion formula. (Thus, this is exactly the same game as defined for the finite case, however, we have not stated the winning condition yet.) Here is the game graph $\mathcal{G}(A_1, bababa \ldots)$:



In this game, the winning criterion is as follows: Any play that is finite is winning for the pathfinder (this happens if the play reaches a state q at some level i with if $\delta(q, a_i) = \mathsf{false}$). An infinite play is winning for the automaton if it visits states in F infinitely often. This game is what is called a *Büchi Game* (we define and analyse Büchi games in the next section). Thus, a winning strategy for the automaton is one in which, every play consistent with the strategy is infinite and visits the set F infinitely often. A winning strategy for the pathfinder is one in which every play consist with the strategy is either finite or visits elements of F only finitely often.

For any strategy f of the automaton we set $\mathcal{R}(f)$ to be the subgraph of the game graph that is reached by some play consistent with the strategy f. It is quite easy to see that we can unfold the DAG $\mathcal{R}(f)$ (i.e. expand the DAG into a tree, duplicating vertices whenver necessary in the obvious manner) for any strategy f for the automaton to get a run for A on σ. This unfolding yields a positional run whenever the strategy is positional. And clearly, the strategy is winning implies that the unfolded runtree is accepting.

Conversely, given an accepting run for A on w, we can easily construct a winning strategy in the game: For any position $s \rightarrow X_1 \rightarrow x_1 \rightarrow X_2 \rightarrow x_2 \ldots \rightarrow X_i \rightarrow x_i$, if $s \rightarrow x_1 \rightarrow x_2 \ldots x_i$ is a path in the accepting run and if X is the set of labels of the children of this x_i, then play $x_i \rightarrow X$. If this path does not appear in the accepting run, play anything. It is quite easy to verify that if the automaton plays this strategy, then any play will stay within a path that appears in the accepting tree and hence visit the set F infinitely often and will therefore be winning for the automaton. Further, notice that if the accepting run was positional then this construction yields a positional winning strategy for the automaton. Thus we have the following theorem:

Theorem 4 *Let A be an alternating Büchi automaton and let σ be an ω-word. A has an (positional) accepting run on w if and only if the player automaton has a (positional) winning strategy in the game $\mathcal{G}(A, \sigma)$.*

3 Reachability Games

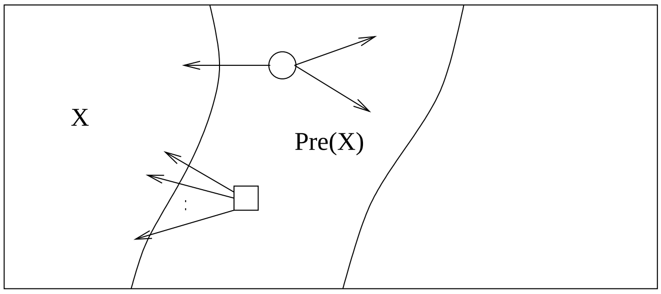
We begin by considering a simple game called the *reachability game*. In a reachability game we are given a graph $G = (V_0, V_1, \rightarrow)$ (where V_0 is the set of nodes from where player 0 makes moves and V_1 is the set of vertices from where player 1 makes moves) and a set $X \subseteq V = V_0 \cup V_1$. The aim of player 0 is to ensure that the game enters some vertex in X while the aim of player 1 is to ensure that this does not happen. For the moment, we shall assume that every vertex has at least one out going edge.

We would like to calculate the set of nodes W_0 from which the player 0 can force the game to enter X. Clearly $X \subseteq W_0$. What else? Consider any vertex $v \in V_0$ with at least one outgoing edge into X. Clearly player 0 can win from v too. Moreover, if $v \in V_1$ and all the outgoing edges from v go into X then once again player 0 will win from v (he needs do nothing, the first move by player 1 will force the game into X.)

For any set U let us define $\mathsf{pre}(U)$ to be the following set:

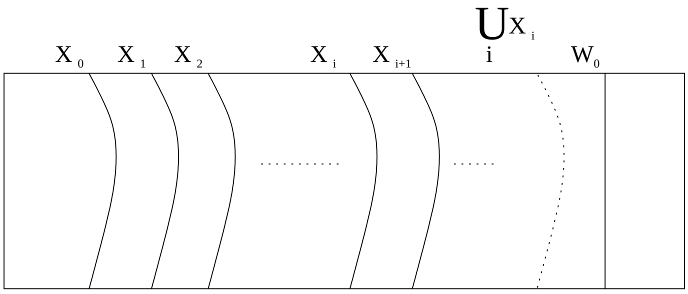
$$\mathsf{pre}(U) \quad = \quad \{v \in V_0 \mid \exists w.\ v \rightarrow w \ \wedge \ w \in U\} \bigcup \{v \in V_1 \mid \forall w.\ v \rightarrow w \ \Rightarrow \ w \in U\}$$

The above argument says that $X \subseteq W_0$ and $\mathsf{pre}(X) \subseteq W_0$ and moreover player 0 has a positional strategy to win from all the nodes in $X \cup \mathsf{pre}(X)$.

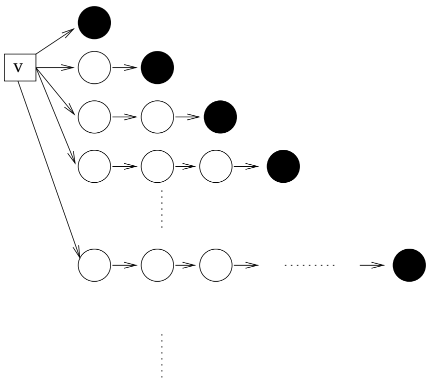


We can generalise this to say that whenvever $U \subseteq W_0$ then $\mathsf{pre}(U) \subseteq W_0$ and if player 0 has a positional strategy that ensures that he wins the game starting at any position in U then he also has a positional winning strategy that wins the game starting at any position in $U \cup \mathsf{pre}(U)$. The strategy is the following: Within U play the (positional) winning strategy that is promised by the hypothesis. For any node $q \in \mathsf{pre}(U) \cap V_0$, we are assured that there is at least one neighbour w in U. Fix such a w for each q and the strategy plays $q \to w$ at q. Thus, after the first move the game enters U and in U the strategy is already assured force the game into X after some sequence of moves. For any $v \in \mathsf{pre}(U) \cap V_1$, the first move by player 1 will move the game into U where the winning strategy for player 0 will force the game to enter X. Thus, player 0 has a winning strategy from all of $U \cap \mathsf{pre}(U)$ and in particular has a positional winning strategy if he has a postional strategy to win from U.

Therefore, if we set $X_0 = X$ and $X_{i+1} = \mathsf{pre}(X_i) \cup X_i$ then, for all i, $X_i \subseteq W_0$.



However, there may be nodes in W_0 that are not in $\bigcup_{i<\omega} X_i$. This is because we have placed no restrictions on our game graphs. They could be of any cardinality, and the degree of a vertex could be infinite or even uncountable. In the following game (where the nodes in X are the dark coloured ones), the node $v \notin \bigcup_{i<\omega} X_i$ but player 0 wins from v.



But note that $\mathsf{pre}(\bigcup_{i<\omega} X_i) \subseteq W_0$ and we could continue our iterations to higher ordinals. We define the sequence X_i as follows:

$$\begin{array}{lcll} X_0 & = & X & \\ X_{i+1} & = & \mathsf{pre}(X_i) \cup X_i & i+1 \text{ is a successor ordinal} \\ X_\beta & = & \bigcup_{i<\beta} X_i & \text{when } \beta \text{ is a limit ordinal} \end{array}$$

We know that there is a smallest ordinal κ such that $X_\kappa = X_{\kappa+1}$. We shall now show that $W_0 = X_\kappa$. We define the positional winning strategy over X_κ now. Suppose $v \in X_\kappa \cap V_0$ then there is a smallest ordinal β such that $v \in X_\beta$. We call this ordinal $\mathsf{ord}(v)$. From the inductive construction it is clear that $\beta = i+1$ for some i (or $\beta = 0$ which means that player 0 has already won. So we may pick any outgoing edge as the strategy.) Therefore there is some $w \in X_i$ and $v \to w$. Fix such a w and define the positional strategy at v to be $v \to w$. We call this strategy $\mathsf{reduce}(X)$ and we shall refer to X_κ as $\mathsf{wforce}(X)$.

Also note that if $v \in X_\kappa \cap V_1$ and $\mathsf{ord}(v) = i+1$ then every edge out of v leads to a node in X_i. Thus, if the game starts in X_κ and player 0 plays the strategy $\mathsf{reduce}(X)$ then every move in the play reduces the value of ord. But ordinals are wellfounded and so within finite number of moves the play must enter some node node w with $\mathsf{ord}(w) = 0$, in other words, the game must enter the set X. Thus, this positional strategy is winning for player 0 starting at any node in X_κ.

Suppose $v \in V \setminus X_\kappa$. Since $v \notin \mathsf{pre}(X_\kappa)$, it follows that either $v \in V_0$ and $\forall w.\ (v \to w) \Rightarrow w \in V \setminus X_\kappa$ or $v \in V_1$ and $\exists w.\ (v \to w)\ \wedge\ w \in V \setminus X_\kappa$. Thus, from a node in $V \setminus X_\kappa$, player 0 cannot move the game into X_κ.

Here is a positional strategy $\mathsf{avoid}(X)$ for player 1: Given any $v \in V_1 \setminus X_\kappa$, pick any $v \to w$ such that $w \in V \setminus X_\kappa$ and play $v \to w$ as the move at v. (For nodes in X_κ the moves for player 1 are defined arbitrarily.) This positional strategy ensures that player 1 never moves the game into X_κ from any vertex in $V \setminus X_\kappa$.

Combining the conclusions of the previous two paragraphs we see that player 1 has a positional strategy that ensures that a game starting in $V \setminus X_\kappa$ stays within $V \setminus X_\kappa$ for ever and thus results in a win for player 1 (since $X \subseteq X_\kappa$). This gives us the following *determinacy theorem* for reachability games:

Theorem 5 *Let $(V_0, V_1, \to)$ and X specify a reachability game. We can partition the set V as W_0 and W_1 such that player 0 has a positional winning strategy that wins the game starting at any position in W_0 and player 1 has a postional winning strategy that wins the game starting at any position in W_1.*

Also observe that if G is a finite graph then the sets W_0 and W_1 can be computed and the positional winning strategies for player 0 and 1 can also be computed.

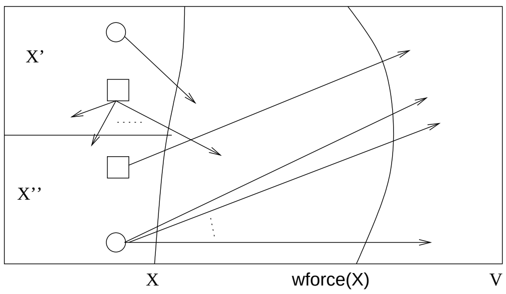
An useful fact: If the game starts within $V \setminus \mathsf{wforce}(X)$ and player 1 plays $\mathsf{avoid}(X)$ then the play never enters $\mathsf{wforce}(X)$ (or equivalently stays within $V \setminus \mathsf{wforce}(X)$.

Exercise: Given a finite reachability game G what is the complexity of computing the winning sets for player 0 and 1 ?

3.1 A useful variant

Suppose we modified that the reachability game to demand that the game visit X after at least one move has been made (i.e. player 0 does not win immediately if the game starts in a vertex in X.). Then what are the winning sets for player 0 and player 1?

Notice that any node in $\mathsf{wforce}(X) \setminus X$ is still winning for player 0. Because, starting at such a node , the strategy $\mathsf{reduce}(X)$ forces the game to visit the set X. Also, every node in $V \setminus \mathsf{wforce}(X)$ is still winning for player 1 as he can keep the game within $V \setminus \mathsf{wforce}(X)$. So the only nodes which may change hands are those within X. States in X can be divided into two sets X' and X'' where $X' = X \cap \mathsf{pre}(\mathsf{wforce}(X))$ and $X'' = X \setminus \mathsf{pre}(\mathsf{wforce}(X))$.



We claim that player 0 has a positional winning strategy that wins from all the vertices in $\mathsf{wforce}(X) \setminus X \cup X'$. At vertices in $\mathsf{wforce}(X) \setminus X$ it plays the usual $\mathsf{reduce}(X)$ strategy. If $v \in X' \cap V_0$ then, there is a $w \in \mathsf{wforce}(X)$ such that $v \to w$. Fix such a w and player 0 plays $v \to w$ at v. It is quite easy to check that this is a winning strategy for player 0 starting at any position in $\mathsf{wforce}(X) \setminus X \cup X'$ and we shall henceforth refer to this set as $\mathsf{force}(X)$.

Player 1 has a positional winning strategy that wins from all the vertices in $V \setminus \mathsf{wforce}(X) \cup X''$. The strategy for player 1 is the usual $\mathsf{avoid}(X)$ strategy at vertices in $V \setminus \mathsf{wforce}(X)$. For a vertex v in $X'' \cap V_1$ notice that there is at least one edge $v \to w$ with $w \in V \setminus \mathsf{wforce}(X)$. Fix such a w and player 1 player $v \to w$ at this v. When player 1 plays this strategy, after the first move the game never enters the set $\mathsf{wforce}(X)$ and thus is winning for player 1.

We shall refer to these strategies for player 0 and 1 as $\mathsf{reduce}(X)$ and $\mathsf{avoid}(X)$ (since they continue to be winning positional winning strategies for 0 and 1 in the simple reachability game though over different winning sets).

4 Büchi Games

A Büchi game consists of game graph $(V_0, V_1, \to)$, a finite set of colours C, a labelling function λ assigning a colour to each vertex, and a set $F \subseteq C$. A play in this game is winning for player 0 if it visits vertices coloured with colours from F infinitely often. For any Büchi automaton A and word w, the game $\mathcal{G}(A, w)$ is a Büchi game. Take the colouring set to be

$Q \cup \{\perp\}$. Colour every vertex at the even levels by $\{\perp\}$ and colour a vertex q at any odd level by q.

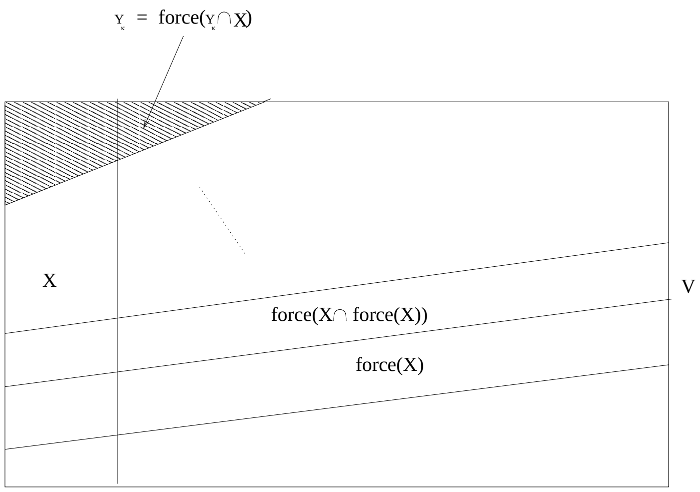
We wish to establish that in every Büchi game, the set of nodes can be divided into to sets such that player 0 has a postional winning strategy for games starting in one set and player 1 has a positional winning strategy starting in the other.

So, we are interested in vertices from where player 0 can force the game to enter the set $X = \{v \mid \lambda(v) \in F\}$ infinitely often. We know that $\mathsf{force}(X)$ is the set of vertices from where he can force the game to enter X at least once (not including the start vertex). From where can we force the game to visit vertices in X at least twice? If we start from a vertex in $\mathsf{force}(X \cap \mathsf{force}(X))$ then clearly we can force the game to enter some vertex $v \in X \cap \mathsf{force}(X)$ and from such a vertex we can clearly force the game to enter X once more. Similarly, from a vertex in $\mathsf{force}(X \cap \mathsf{force}(X \cap \mathsf{force}(X)))$ player 0 can force the game to enter X at least three times and so on.

If we can find a set Y such that $Y = \mathsf{force}(X \cap Y)$ then clearly, player 0 would win starting at any vertex in Y. He simply plays the $\mathsf{reduce}(X \cap Y)$ strategy and this would force the play to enter the set $X \cap Y$. But $Y = \mathsf{force}(X \cap Y)$ and so the play would be forced to enter $Y \cap X$ again (Note that this argument would not work if we had used wforce instead) and so on. Thus the play would enter X infinitely often and player 0 wins.

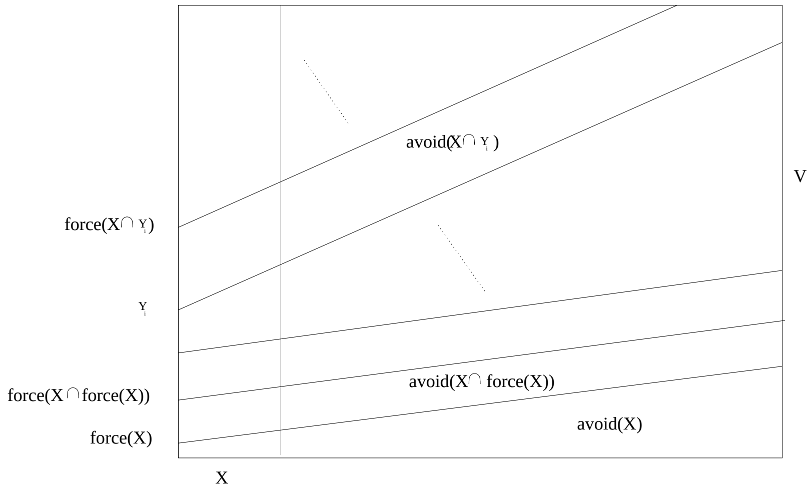
The Tarski-Knaster theorem suggests how to find the most generous such Y and here is how: Define a sequence $Y_0, Y_1, \ldots$ as follows:

$$\begin{array}{lcll} Y_0 & = & V & \\ Y_{i+1} & = & \mathsf{force}(X \cap Y_i) & i+1 \text{ is a successor ordinal} \\ Y_\beta & = & \bigcap_{i<\beta} Y_i & \text{when } \beta \text{ is a limit ordinal} \end{array}$$



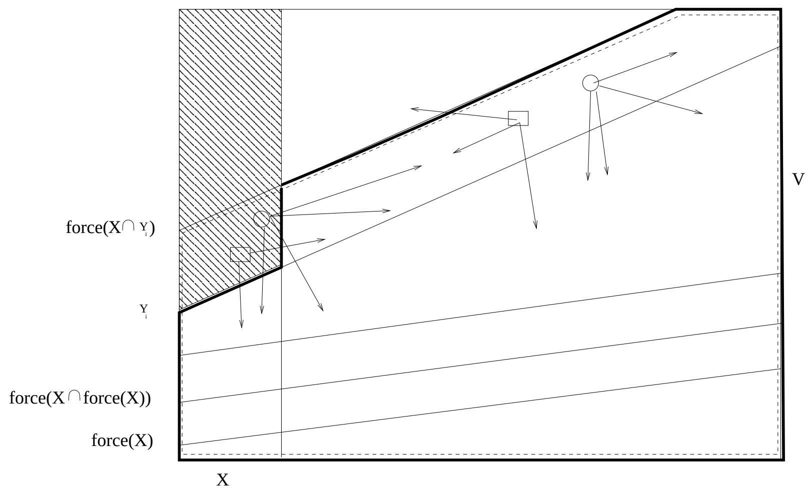
Notice that the function force is monotone (i.e. if $S \subseteq T$ then $\mathsf{force}(S) \subseteq \mathsf{force}(T)$). And since $Y_0 = V$, $Y_1 \subseteq Y_0$. Thus, for all i $Y_{i+1} \subseteq Y_i$ and consequently $Y_i \subseteq Y_j$ for any $i > j$. Thus, there must a least ordinal κ such that $Y_{\kappa+1} = Y_\kappa$. Thus $Y_\kappa = \mathsf{force}(X \cap Y_\kappa)$ and thus player 0 has a positional winning strategy (i.e. $\mathsf{reduce}(X \cap Y_\kappa)$). that wins from all positions in Y_κ.

What about vertices not in Y_κ? Suppose $x \in V \setminus Y_\kappa$. $Y_0 = V$, therefore there is a least i such that $x \in Y_i$ and $x \notin Y_{i+1}$ for some successor ordinal $i+1$. We define $\mathsf{ord}(x)$ to be the least ordinal i such that $x \notin Y_{i+1}$. If $\mathsf{ord}(x) = i$ then $x \in V \setminus \mathsf{force}(X \cap Y_i)$. Within the set $V \setminus \mathsf{force}(X \cap Y_i)$ player 1 can play $\mathsf{avoid}(X \cap Y_i)$ to ensure that the play never enters $X \cap Y_i$. We define a positional strategy for player 1 as follows: at any $x \in V \setminus Y_\kappa$ play $\mathsf{avoid}(X \cap Y_{\mathsf{ord}(x)})$.



Notice that as long as player 1 plays $\mathsf{avoid}(X \cap Y_i)$ the game can never enter Y_{i+1} and thus all vertices visited in such a play have ord values less than or equal to i. If at some point the ord value j becomes strictly less than i then our strategy would play $\mathsf{avoid}(X \cap Y_j)$ and consequently the game would never visit a vertex in Y_{j+1} and so on . Thus, in any play consistent with this strategy for player 1, the ord values of the vertices visisted forms a nonincreasing sequence.

Now let us examine what happens whenever a play enters a vertex $x \in X$. Suppose $\mathsf{ord}(x) = i$.



In this figure, the area enclosed by the thick boundary is $V \setminus \mathsf{wforce}(X \cap Y_i)$ while the area enclosed by the dotted boundary is $V \setminus \mathsf{force}(X \cap Y_i)$ (i.e. $V \setminus Y_{i+1}$). If $x \in X$ and $\mathsf{ord}(x) = i$, then $x \in X \cap Y_i$ (and therefore $x \in \mathsf{wforce}(X \cap Y_i)$). On the other hand, $x \notin \mathsf{force}(X \cap Y_i)$. Therefore, by the discussion in section 3.1, the $\mathsf{avoid}(X \cap Y_i)$ strategy for player 1 ensures that the play never returns to $X \cap Y_i$ after the first move. In particular, the first move results in the game moving to $V \setminus \mathsf{wforce}(X \cap Y_i)$. Thus as long as the game stays within positions with ord equal to i, the game will never return to $X \cap Y_i$. And since every vertex $v \in X$ with $\mathsf{ord}(v) = i$ is in Y_i this means that game never returns to X as long as the ord value stays at i. This together with the fact that the ord values are nonincreasing along any play consistent with our strategy ensures that X is visited only finitely often.

Thus, we have the following theorem:

Theorem 6 *In any Büchi game the set of vertices can be partitioned in two sets W_0 and W_1 such that player 0 has a positional winning strategy that wins all plays starting from any position in W_0 and player 1 has a positional winning strategy that wins all plays starting at any position in W_1.*