

Reachability and Büchi games

Yu Cong

March 23, 2024

Overview

1. Motivation & References

2. Reachability Game

3. Büchi Game

Motivation & References

Motivation: Reachability and Büchi games are important in system verification and testing. Computing the winning set of Büchi games is a central problem in computer aided verification with a large number of applications.

References:



John Smith (2012)

Title of the publication

Journal Name 12(3), 45 – 678.

Reachability Game

A reachability game is a 2-player (namely P0 and P1) game on a directed finite graph.

Game graph: directed graph $G(\{V_0 \cup V_1\}, E)$. ($\{V_0, V_1\}$ is a partition of V)

Target set: target set is $T \subseteq \{V_0 \cup V_1\}$.

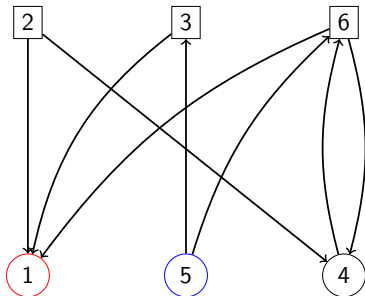
A play P is a (finite or infinite) path in the game graph beginning at the initial vertex s . If $v \in V_0$, P0 moves along an outgoing edge of v . Otherwise, P1 takes the move.

Definition of winning: P0 wins if $T \cap P \neq \emptyset$, otherwise P1 wins.

Memoryless strategy: a strategy for P0 is a mapping $\alpha : V_0 \rightarrow V$ that defines how P0 should extend the current play.

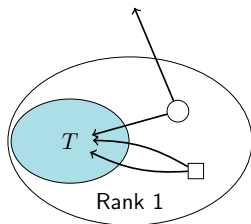
Example for Reachability Game

Rectangle vertices are in V_1 , circles are in V_0 ;
Vertices in T are red, the initial vertex v_I is blue.



A winning play for P0 is $\{5, 3, 1\}$

Algorithm for Reachability Game



- if s is in T , P0 wins;
- if $s \in V_0$ and s has at least one outgoing edge to $u \in T$, P0 wins in one step;
- if $s \in V_1$ and all of s 's outgoing edges go to $u \in T$, P0 wins in one step;

Algorithm for Reachability Game

We defined Rank 0 and Rank 1 already, now we define Rank i .
 $R_i := \{v \in V \mid \text{P0 can force a visit from } v \text{ to a vertex in } T \text{ in } i \text{ steps}\}$

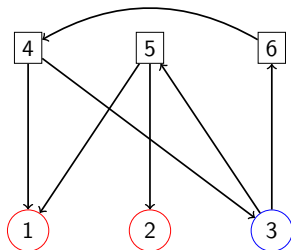
Define Reachability set of T for P0, $Reach(T, 0) := \bigcup_{i=1}^{n-1} R_i$

A vertex $v \in R_i$:

if $v \in V_0$ and there is an edge $e(v, u)$ $u \in R_{i-1}$;

if $v \in V_1$ and for every edge $e(v, u)$ we have $u \in \bigcup_{j=0}^{i-1} R_j$;

Algorithm for Reachability Game



- $R_0 = \{1, 2\};$
- $R_1 = \{5\};$
- $R_2 = \{3\};$
- $R_3 = \{4\};$
- $R_4 = \{6\};$

For simplicity, denote $u \in R_k$ by $\text{Rank}[u]=k$.

An $O(m)$ Algorithm for Reachability Game

Algorithm 1: Reachability for P0

Data: game graph G , target set T

Result: Rank[$|V|$]

```

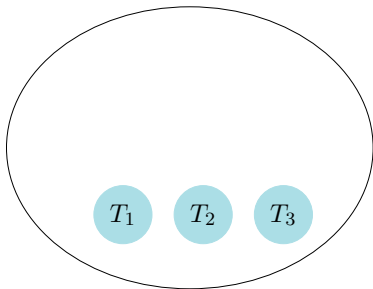
1  Q:= an empty queue;
2  Rank[ $|V|$ ],count[ $|V|$ ]:= all 0s array;
3  Q.push(T);
4  while Q is not empty do
5      u:=Q.front,Q.pop;
6      for  $e(v,u) \in E$  do
7          if  $v \in V_0$  and  $v$  has not been visited then
8              Rank[v]:=Rank[u]+1; Q.push( $\{v\}$ )
9          else if  $v \in V_1$  then
10             count[v]:=count[v]+1;
11             if count[v]=Out Degree of  $v$  then Rank[v]:=Rank[u]+1;
12                 Q.push( $\{v\}$ ) ;
13             end
14         end
15     end
16 end

```

Every edge is used at most once.

Type

T_1, T_2, \dots, T_k are disjoint subsets of V , now we want to compute Reachability of each one of them.



Definition A type of vertex x is a tuple (y_1, \dots, y_k) , where each $x_i \in \{0, 1\}$, such that $y_i = 1$ iff x is in $Reach(T_i, 0)$.

Compute Types

- Run reachability algorithm for every T_i , $O(km)$;
- Compute simultaneously.
- Can it be done in linear or nearly linear time?

Minimum Base

The minimum base of T is the minimum subset of T which can generate the same Reachability set as T .

Computing the minimum base is NP-hard.

Set cover problem: Given a set S of n elements, a collections S_1, S_2, \dots, S_m of subsets of S , and a number K , does there exists a collection of at most k of these sets whose union is equal to all of S .

Minimum Base

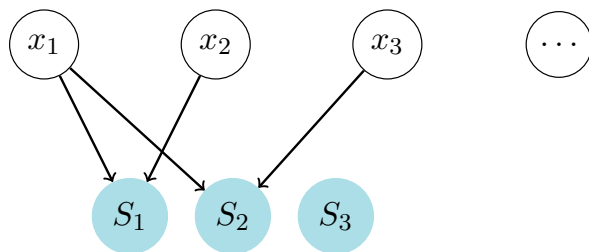
Proof:

We prove that the decision problem for minimum base is NP-Complete.

The decision problem L is can we find a base with at most k vertices.

- 1 L is in NP.
- 2 set cover problem(which is NP-Complete) can be reduced to L in polynomial time.
 - Construct a Reachability game graph $G(V_0, E)$. There are m vertices in T representing m subsets in set cover problem, n vertices not in T representing n elements in S .
 - If subset S_i contains element x_j , connect an edge from vertex representing S_i to vertex representing x_j in T .

Minimum Base



$$S_1 = \{x_1, x_2\}$$

$$S_2 = \{x_1, x_3\}$$

So L is NP-Complete. The minimum base problem is NP-Hard.

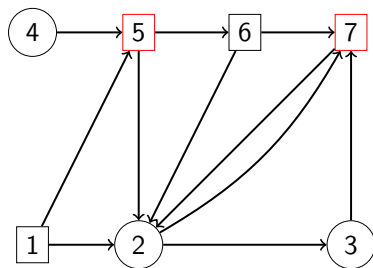
Büchi Game

Definition A **Büchi game** is a game $\mathcal{G} = (G, s, T)$ where G is the Reachability game graph, V_i is an initial vertex, $T \subseteq V$ is the target set as in Reachability game.

Play: The definition of play in Büchi Game is the same as in Reachability game.

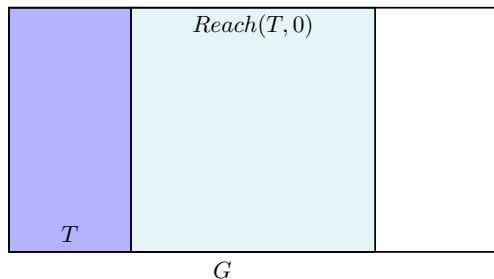
Definition of winning: We assume the play P is infinite here. if there exists infinite many vertices $v \in T$ in P , P0 wins. Otherwise P1 wins.

Example for Büchi Game



P0 is always winning on this game graph.

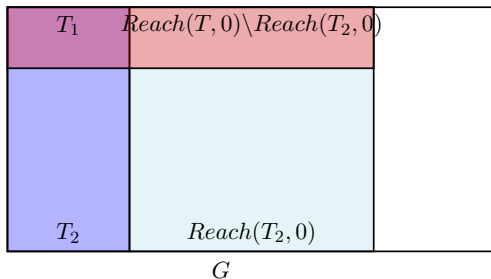
Algorithm for Büchi Game 1



If $v \notin Reach(T, 0) \cup T$,
 v can not reach T , P0
will lose.

Some vertices in T can
not reach
 $Reach(T, 0) \cup T$, P0
will also lose on these
vertices.

Algorithm for Büchi Game 1

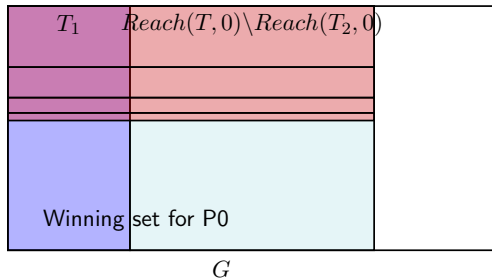


$T_1 = \{v \in T \mid v \text{ can't reach } T \cup Reach(T, 0)\}$

Some vertices in T_2 can only reach $Reach(T, 0) \setminus Reach(T_2, 0)$

We find $T_3 = \{v \in T_2 \mid v \text{ can't reach } T_2 \cup Reach(T_2, 0)\}$

Algorithm for Büchi Game 1



We repeat this process until T_k does not shrink.

The remaining part of $T_k \cup Reach(T_k, 0)$ is the winning set for P0.

Algorithm for Büchi Game 1

- How to find T_1

$$T_1 = \{v \in T \mid v \text{ can't reach } T \cup \text{Reach}(T, 0)\}$$

$$T_1 = \{v \in T \mid v \text{ can only reach } V \setminus \{T \cup \text{Reach}(T, 0)\}\}$$

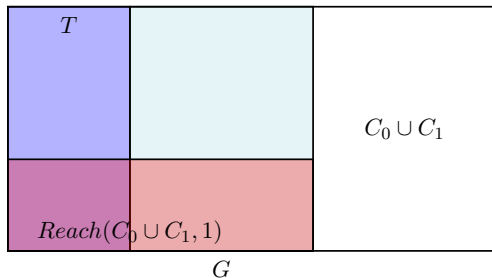
P1 wants to reach $V \setminus \{T \cup \text{Reach}(T, 0)\}$, P0 tries to avoid $V \setminus \{T \cup \text{Reach}(T, 0)\}$.

compute $\text{Reach}(V \setminus \{T \cup \text{Reach}(T, 0)\}, 1)$

- Time complexity

$O(m)$ to find T_i , at most $O(n)$ times. Worst-case $O(nm)$.

Algorithm for Büchi Game 2



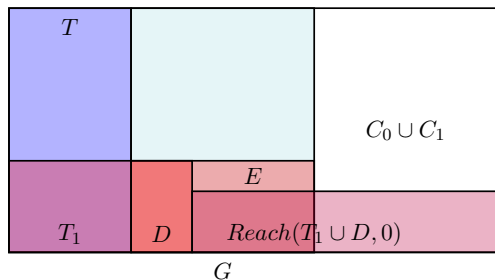
Compute C_0 and C_1 .

C_0 is a set of vertices in $V_0 \setminus T$ having all outgoing edges to vertices in $V \setminus T$.

C_1 is a set of vertices in $V_1 \setminus T$ having an outgoing edge to vertices in $V \setminus T$.

Compute
 $Reach(C_0 \cup C_1, 1)$

Algorithm for Büchi Game 2



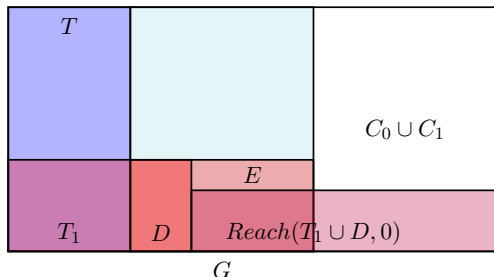
Some vertices in $Reach(C_0 \cup C_1, 1)$ can "reach" T_1 . (D in the left picture)

Compute $Reach(T_1 \cup D, 0)$.

$E = Reach(C_0 \cup C_1, 1) \setminus \{T_1 \cup D \cup Reach(T_1 \cup D, 0)\}$

$\{E \cup C_0 \cup C_1\} \setminus Reach(T_1 \cup D, 0)$ is the set of vertices which can't "reach" T .

Algorithm for Büchi Game 2



$S = \{E \cup C_0 \cup C_1\} \setminus Reach(T_1 \cup D, 0)$ is the same as $V \setminus \{T \cup Reach(T)\}$ in Algorithm 1.

Then we can compute $Reach(S, 1)$ to delete some losing vertices for P0 in T .

Repeat the same process on $G \setminus \{T \setminus Reach(S, 1)\}$

Algorithm for Büchi Game 2

- Time complexity

Finding S needs $O(m)$ time.

Also in the worst case we need to compute S $O(n)$ times.
worst case $O(nm)$.