MEMORY

General strategy: $\sigma : \text{Play}_{Eve} \to E$

Positional strategy: $\sigma : V_{Eve} \to E$

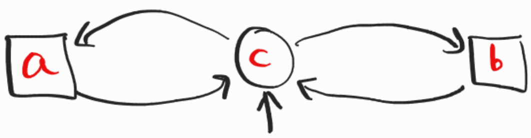
Finite memory strategies

Memory structure: $\begin{cases} m_0 \in M \\ \delta : M \times E \to M \end{cases}$ deterministic automaton reading edges

$\sigma : V_{Eve} \times M \to E$

Example:

Reach (a) $\wedge$ Reach (b)



requires two memory states !

$\mathcal{G}$ game $\quad\quad \mathcal{M}$ memory structure

$$\mathcal{G} = (V = V_{Eve} \uplus V_{Adam}, E, col : V \to C, \Omega \subseteq C^\omega)$$

$$\mathcal{M} = (m_{init} \in M, \delta : M \times C \to M)$$

We define $\mathcal{G} \times \mathcal{M}$ synchronised product:

$V \times M$ set of vertices

$V_{Eve} \times M$ vertices controlled by Eve

$V_{Adam} \times M$ vertices controlled by Adam

$$(v, m) \longrightarrow (v', m') \quad \text{if} \quad (v, v') \in E \text{ and } \delta(m, (v, v')) = m'$$

We often use $\mathcal{G} \times \mathcal{M}$ for reductions

GENERALIZED BÜCHI GAMES

Objectives : $\bigwedge_{i=0}^{k-1} \text{Büchi}(F_i) \qquad F_i \subseteq V$

Question 1 :

How much memory does Eve need (as a function of k) ?

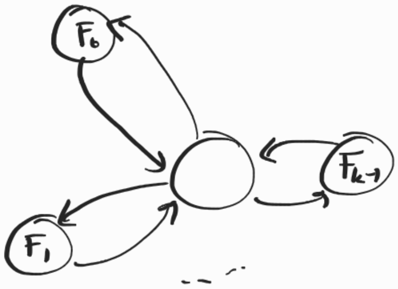
Question 2 :

Construct an algorithm for solving generalized Büchi games

Question 3 :

How much memory does Adam need (as a function of k) ?

Question 1:



Colours : $\{-\} \cup \{0, \dots, k-1\}$

uncoloured

Eve needs k memory states to win

If she uses less, there's one colour that she will never see.

Here's a sufficient memory structure:

$$\mathcal{M} = \left(\{0, \dots, k-1\},\ \delta : M \times C \longrightarrow M \right)$$

$$\delta(i, -) = i$$

$$\delta(i, i) = i+1 \mod k$$

$$\delta(i, j) = i \qquad \text{if } i \neq j$$

Construct $\mathcal{G} \times \mathcal{M}$ with objective

$$\text{Büchi}\left(V \times \{0\}\right)$$

Eve wins from v in $\mathcal{G}$ $\iff$ Eve wins from $(v, 0)$ in $\mathcal{G} \times \mathcal{M}$

σ positional in $\mathcal{G} \times \mathcal{M}$ $\Rightarrow$ σ using memory $\mathcal{M}$ in $\mathcal{G}$

$\mathcal{G} \times \mathcal{M}$ Büchi, so positionally determined.

Question 2: solving $\mathcal{G} \times \mathcal{U}\mathcal{B}$ is enough for solving $\mathcal{G}$

$$\mathcal{G} \begin{cases} n & \text{vertices} \\ m & \text{edges} \\ k+1 & \text{colours} \end{cases} \qquad \mathcal{G} \times \mathcal{U}\mathcal{B} \begin{cases} n \times k \\ m \times k \\ \text{Büchi} \end{cases}$$

$$O(n\, m\, k^2)$$

Question 3

Adam has positional winning strategies.

To see this, we need the following properties:

(*) If $\forall i \quad W_{Eve}(\text{Büchi}(F_i)) = V$

then $W_{Eve}(\text{GenBüchi}(F_0, \dots, F_{k-1})) = V$

(**) If $\exists i \quad W_{Adam}(\text{Büchi}(F_i)) \neq \emptyset$

then $\mathcal{G}' = \mathcal{G}[V \setminus \text{Attr}_{Adam}(W_{Adam}(\text{Büchi}(F_i)))]$

$W_{Eve}(\mathcal{G}) = W_{Eve}(\mathcal{G}')$

This induces a recursive algorithm.

Each time we use (**) we use a positional strategy for Adam on disjoint subsets of vertices.

Therefore Adam has positional winning strategies.

GENERALIZED REACHABILITY GAMES

Objectives : $\bigwedge_{i=1}^{k} \text{Reach}(F_i)$ $\quad F_i \subseteq V$

Question 1 :

How much memory does Eve need (as a function of k) ?

Question 2 :

Construct an algorithm for solving generalized reachability games

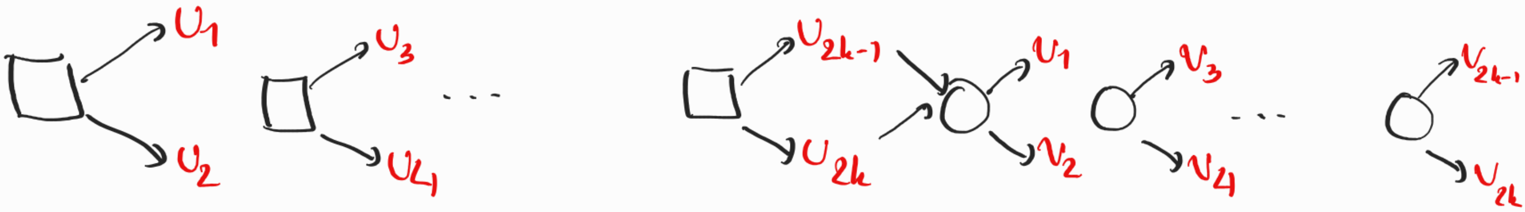
Question 3 :

How much memory does Adam need (as a function of k) ?

Question 1:

Lower bound

$F_i = \{u_i, v_i\}$ $\quad$ $2k$ colours



Eve needs to revert all choices of Adam

$\rightarrow$ 2^k memory states.

This $2k$ colours: $\quad 2^{\#\text{colours}/2}$

Can be improved to $2^k - 1$

Upper bound

$C = \{\underset{\text{Uncoloured}}{-}\} \cup \{1, \dots, k\}$

Construct $\mathcal{M} = (\underbrace{\mathcal{P}(\{0, \dots, k-1\})}_{M}, \delta : M \times C \to M)$

$$\delta(S, -) = S$$
$$\delta(S, i) = S \cup \{i\}$$

We equip $\mathcal{G} \times \mathcal{M}$ with $\mathrm{Reach}(V \times \{1, \dots, k\})$

Eve wins from v in $\mathcal{G}$ $\Leftrightarrow$ Eve wins from $(v, \emptyset)$ in $\mathcal{G} \times \mathcal{M}$

σ positional in $\mathcal{G} \times \mathcal{M} \Rightarrow \sigma$ with memory $\mathcal{M}$ in $\mathcal{G}$

Since $\mathcal{G} \times \mathcal{M}$ is a reachability game
it is positionally determined
So in $\mathcal{G}$, $\mathcal{M}$ is sufficient

We get 2^k. With some twist we get $2^k - 1$

Question 2 :

solving $\mathcal{G} \times \mathcal{M}$ is enough for solving $\mathcal{G}$

$$\mathcal{G} \begin{cases} n & \text{Vertices} \\ m & \text{edges} \\ k+1 & \text{colours} \end{cases} \qquad \mathcal{G} \times \mathcal{M} \begin{cases} n \times 2^k \\ m \times 2^k \\ \text{Reachability} \end{cases}$$

$$O(m 2^k)$$

We can show that the problem is actually PSPACE complete

QBF Quantified Boolean Formulas:

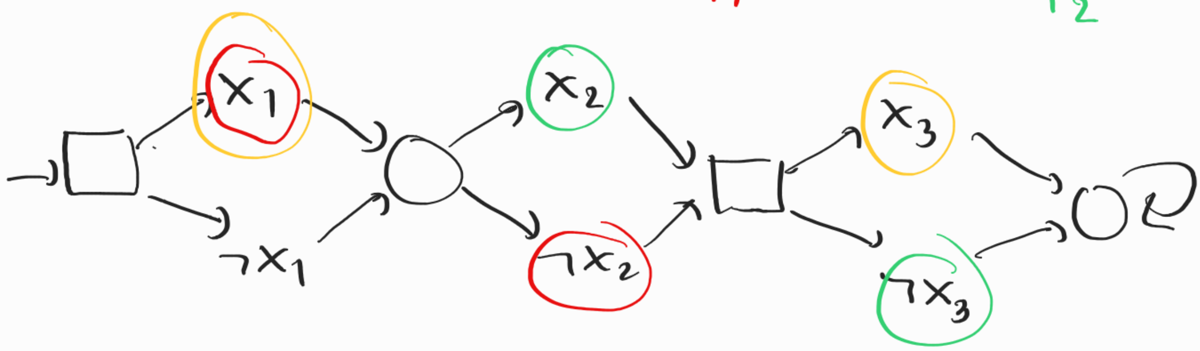
<u>IN:</u> $\forall x_1 \exists x_2 \forall x_3 \exists x_4 \dots \phi(x_1 \dots x_n)$

Boolean formula in CNF

<u>OUT:</u> is the formula true?

PSPACE-complete

$\forall x_1 \exists x_2 \forall x_3 \quad \underbrace{(x_1 \vee \neg x_2)}_{F_1} \wedge \underbrace{(x_2 \vee \neg x_3)}_{F_2} \wedge \underbrace{(x_1 \vee x_3)}_{F_3}$



Formula true $\iff$ Eve wins in the game

$\rightarrow$ PSPACE-hard

Question 3

Lower bound

Consider the following game $\quad k = 2p+1$

- Eve chooses p colours
- Adam chooses p colours
- Eve chooses p colours

Adam needs $\binom{2p+1}{p}$ memory states:

he needs to replicate exactly Eve's choices

Upper bound

Fix $v \in V$. For $S \subseteq S' \subseteq \{1, \dots, k\}$

$$(v, S') \in W_{Adam}(\mathcal{G} \times \mathcal{M}) \Rightarrow (v, S) \in W_{Adam}(\mathcal{G} \times \mathcal{M})$$

Let $W(v) = \{ S \subseteq \{1, \dots, k\} : (v, S) \in W_{Adam}(\mathcal{G} \times \mathcal{M}) \}$ is downward-closed.

Let $S_1(v), \dots, S_p(v)$ maximal subsets of $W(v)$.

Since they are incomparable, we have $p \leq \binom{k}{k/2}$

they form an antichain

Size of largest antichain in $(\mathcal{P}(\{1, \dots, k\}), \subseteq)$

Let $\mathcal{M} = \left(\left\{ 1, \dots, \binom{k}{k/2} \right\}, 1, \delta \right)$

$\delta : M \times E \longrightarrow M$ defined by

$$\delta(i, (v,v')) = j \quad \text{such that}$$

$$S_i(v) \cup \{col(v')\} \subseteq S_j(v') \quad \text{if it exists}$$

Define $\sigma : V \times M \longrightarrow E$ by

$$\sigma(v, i) = (v, v') \text{ such that } \delta(i, (v,v')) \text{ is defined}$$

Claim: for all play π consistent with σ,

write $col(\pi) \subseteq \{1, \dots, k\}$ the set of colours seen in π

let π ends in (v, i)

vertex memory state

Then $\quad col(\pi) \subseteq S_i(v)$