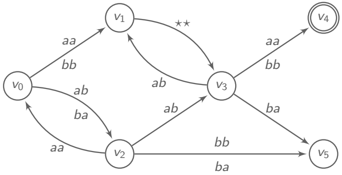
Games with arbitrarily many players

Nathalie Bertrand - Inria Rennes
joint work with Patricia Bouyer and Anirban Majumdar

CFV online seminar - June 5th 2020

2-player concurrent games



How to play?

- token is initially in vertex v_0
- Player 1 and Player 2 choose actions simultaneously
- next vertex is determined by the combination of actions

Player 1 has a winning strategy if she can win whatever Player 2 does

Motivations for parameterized concurrent games

a distinguished agent
trying to achieve a goal against
arbitrarily many adversaries

Eve vs Rest of the world

Motivations for parameterized concurrent games

a distinguished agent
trying to achieve a goal against
arbitrarily many adversaries

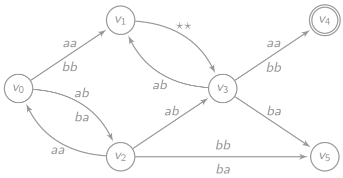
Eve vs Rest of the world

arbitrarily many agents
trying to achieve a goal
as a coalition

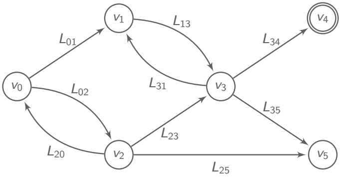
Strategy synthesis for coalition

Framework for parameterized concurrent games

From 2 players to arbitrarily many

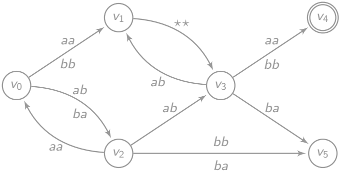


L_{ij} languages of finite words

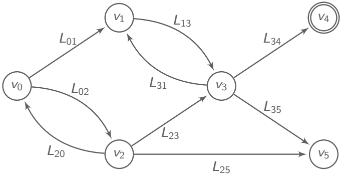


Framework for parameterized concurrent games

From 2 players to arbitrarily many



L_{ij} languages of finite words

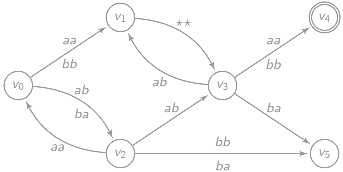


How to play?

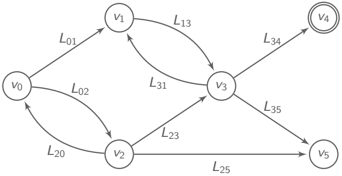
- number of players **k** is fixed initially, yet unknown to them
- players know their "position" (*e.g.* Player 3 is third in list)
- they observe the sequence of vertices
- each player chooses an action, forming altogether a finite word
  $\forall i$ Player i choosing a_i yields the word $\mathbf{w} = a_1 \cdots a_{\mathbf{k}}$;
- to which language **w** belongs determines the next vertex

Framework for parameterized concurrent games

From 2 players to arbitrarily many



L_{ij} languages of finite words



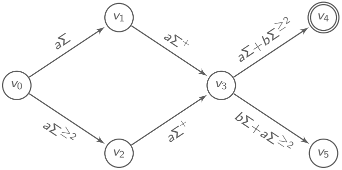
How to play?

- number of players **k** is fixed initially, yet unknown to them
- players know their "position" (*e.g.* Player 3 is third in list)
- they observe the sequence of vertices
- each player chooses an action, forming altogether a finite word
  $\forall i$ Player i choosing a_i yields the word $\mathbf{w} = a_1 \cdots a_{\mathbf{k}}$;
- to which language **w** belongs determines the next vertex

Rk: choice of **k** and resolution of non-determinism is **adversarial**

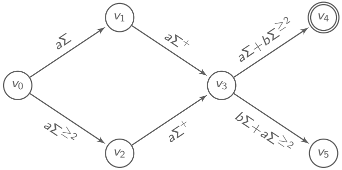
A first parameterized reachability game

Eve vs Rest of the world



A first parameterized reachability game

Eve vs Rest of the world

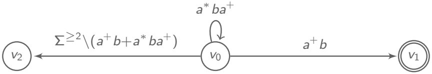


- game starts at v_0
- the number of players **k** is fixed but unknown to the players
- Player 1 plays a, other players each choose an action in Σ
- if $\mathbf{k} = 2$, the token moves to v_1, otherwise, it moves to v_2
- in v_3, Player 1 can ensure to reach v_4:
  choose a (resp. b) if the play went to v_1 (resp. v_2)
- $v_0 \xrightarrow{aa} v_1 \xrightarrow{ab} v_3 \xrightarrow{aa} v_4 \in \mathrm{Plays}_2$ $\quad$ $v_0 \xrightarrow{aab} v_2 \xrightarrow{abb} v_3 \xrightarrow{baa} v_4 \in \mathrm{Plays}_3$

Player 1 can reach v_4 independently of the number of opponents

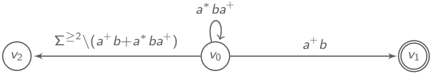
A second parameterized reachability game

Strategy synthesis for coalition



A second parameterized reachability game

Strategy synthesis for coalition



- game starts at v_0
- the number of players $\mathbf{k}$ is fixed but unknown to the players
- as a coalition all players can ensure to reach v_1
  at step i, Player i plays b and all others play a
- $\mathsf{Play}_{\mathbf{k}} = v_0 \xrightarrow{ba^{\mathbf{k}-1}} v_0 \xrightarrow{aba^{\mathbf{k}-2}} v_0 \cdots v_0 \xrightarrow{a^{\mathbf{k}-1}b} v_1$

Players can collectively reach v_1 independently of their number

Formalization of our two problems of interest

Eve vs Rest of the world 

Input: a parameterized arena, a winning objective **Win**
Output: whether Eve has a winning strategy to achieve **Win** independently of the number of her opponents

$$\exists\sigma_E\ \forall\mathbf{k}\ \forall\sigma_2\cdots\sigma_{\mathbf{k}}\ \mathsf{Plays}(\sigma_E,\sigma_2,\cdots,\sigma_{\mathbf{k}})\subseteq\mathbf{Win}?$$

Formalization of our two problems of interest

Eve vs Rest of the world 

Input: a parameterized arena, a winning objective **Win**
Output: whether Eve has a winning strategy to achieve **Win** independently of the number of her opponents

$$\exists \sigma_E \ \forall \mathbf{k} \ \forall \sigma_2 \cdots \sigma_{\mathbf{k}} \ \mathsf{Plays}(\sigma_E, \sigma_2, \cdots, \sigma_{\mathbf{k}}) \subseteq \mathbf{Win}?$$

Strategy synthesis for coalition 

Input: a parameterized arena, a winning objective **Win**
Output: whether there is a coalition strategy to achieve **Win** independently of the number of players

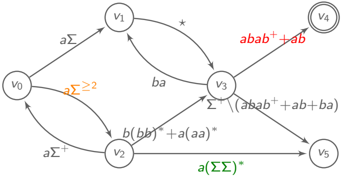
$$\exists \sigma_1 \sigma_2 \cdots \ \forall \mathbf{k} \ \mathsf{Plays}(\sigma_1, \sigma_2, \cdots, \sigma_{\mathbf{k}}) \subseteq \mathbf{Win}?$$

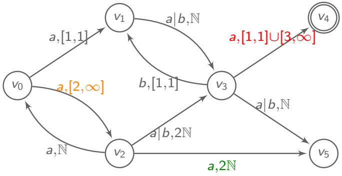
Eve vs Rest of the world

...

Reduction to simpler games: counting is enough

Observation Eve's opponents act as a coalition
$\rightarrow$ only their number matter, not their actions

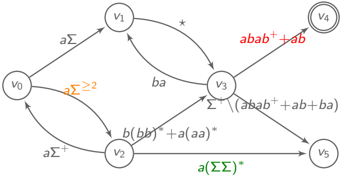


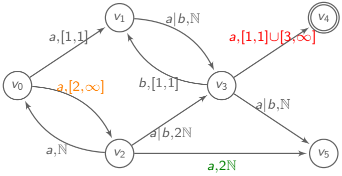


Reduction to simpler games: counting is enough

Observation Eve's opponents act as a coalition

$\rightarrow$ only their number matter, not their actions





How to play?

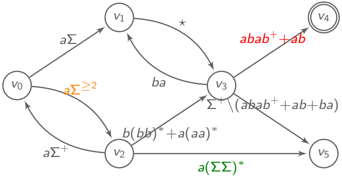
- environment chooses number of players $\mathbf{k}$, unknown to Eve
- at vertex v, Eve chooses action a

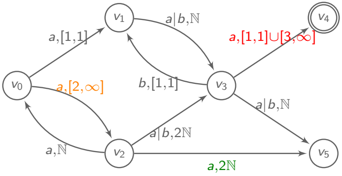
  environment chooses edge $v \xrightarrow{a,S} v'$ with $\mathbf{k} \in S$
- game proceeds from v'

Reduction to simpler games: counting is enough

Observation Eve's opponents act as a coalition

$\rightarrow$ only their number matter, not their actions





How to play?

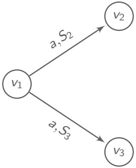
- environment chooses number of players **k**, unknown to Eve
- at vertex v, Eve chooses action a

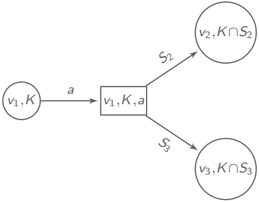
  environment chooses edge $v \xrightarrow{a,S} v'$ with $\mathbf{k} \in S$
- game proceeds from v'

Rk: for a regular language L, $\mathbf{count}(L)$ is semi-linear

Knowledge game

2-player turn-based game encoding Eve's knowledge on nb of opponents

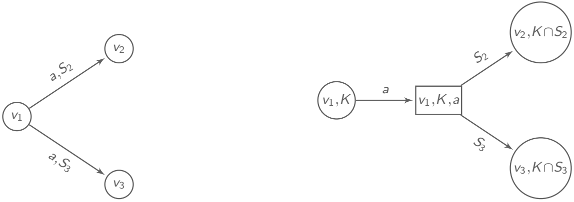




- ○ chooses actions, □ chooses next vertex
- initial vertex $(v_0, \mathbb{N})$ owned by ○
- knowledge of Eve is updated according to moves

Knowledge game

2-player turn-based game encoding Eve's knowledge on nb of opponents

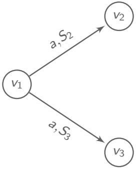


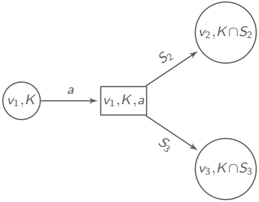
- ○ chooses actions, □ chooses next vertex
- initial vertex $(v_0, \mathbb{N})$ owned by ○
- knowledge of Eve is updated according to moves

Eve wins the parameterized game $\iff$ ○ wins the knowledge game

Knowledge game

2-player turn-based game encoding Eve's knowledge on nb of opponents



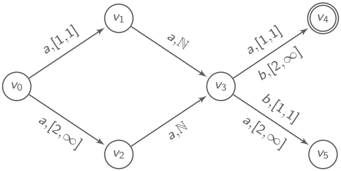


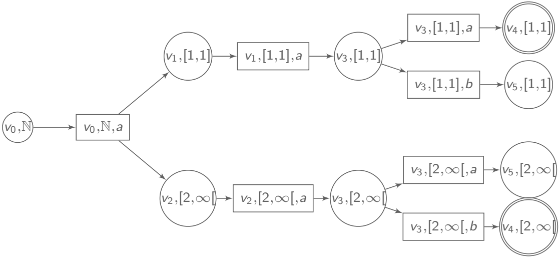
- ○ chooses actions, □ chooses next vertex
- initial vertex $(v_0, \mathbb{N})$ owned by ○
- knowledge of Eve is updated according to moves

Eve wins the parameterized game $\iff$ ○ wins the knowledge game

Knowledge game can be solved in polynomial time in **its size**

Knowledge game on an example





Resolution of concurrent parameterized games

Decidability and complexity

The parameterized game problem for **reachability objectives** is decidable, with the following complexities

	Deterministic	Non-deterministic
Intervals	PTIME-complete	
Finite unions of intervals	NP-complete	PSPACE-complete
Semilinear sets	PSPACE-complete	

Resolution of concurrent parameterized games

Decidability and complexity

The parameterized game problem for **reachability objectives** is decidable, with the following complexities

	Deterministic	Non-deterministic
Intervals	PTIME-complete	
Finite unions of intervals	NP-complete	PSPACE-complete
Semilinear sets	PSPACE-complete	

Simple case of intervals

knowledge game is quadratic in the number of end-points

Resolution of concurrent parameterized games

Decidability and complexity

The parameterized game problem for **reachability objectives** is decidable, with the following complexities

	Deterministic	Non-deterministic
Intervals	PTIME-complete	
Finite unions of intervals	NP-complete	PSPACE-complete
Semilinear sets	PSPACE-complete	

Simple case of intervals

knowledge game is quadratic in the number of end-points

General case: semi-linear sets

knowledge game is at most exponential in the number of semilinear sets
but there is a polynomial space algorithm

PSPACE upper bound for semilinear constraints

Parameterized game problem for reachability objectives is in PSPACE

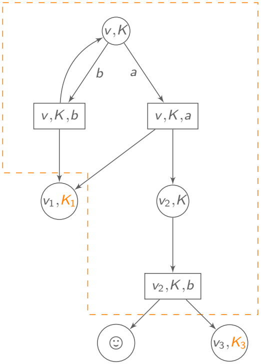
Proof idea

- decompose the knowledge game into subgames
  with objective to reduce the knowledge while remaining winning
- DFS algorithm tagging states (v, K) with ✓/✗ up to $(v_0, \mathbb{N})$

Close-up on subgames

for every ○ vertex (v, K)
restriction of the knowledge game

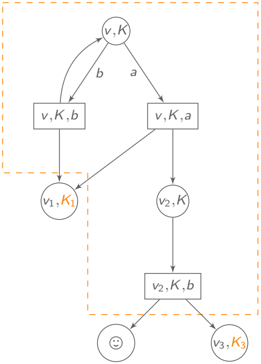
- starting at (v, K)
- stopping at any (v', K') with $K' \subsetneq K$
  or at the target ☺

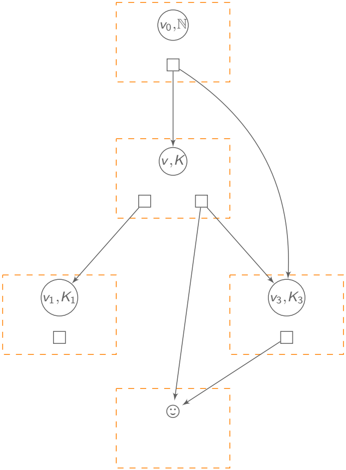


Close-up on subgames

for every ○ vertex (v, K)
restriction of the knowledge game

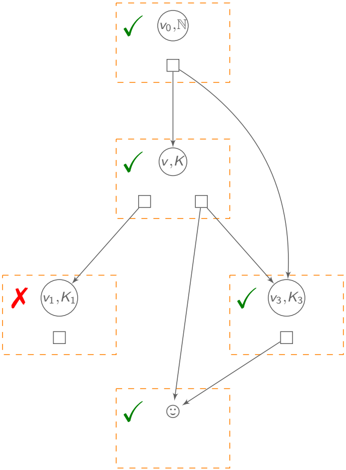
- starting at (v, K)
- stopping at any (v', K') with $K' \subsetneq K$
  or at the target ☺





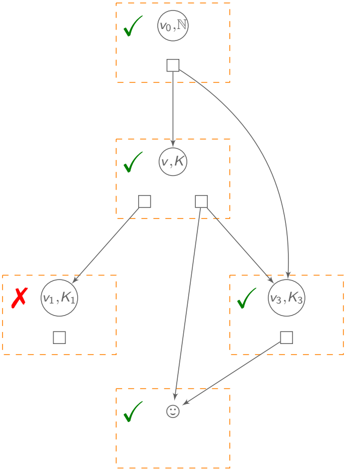
DAG of subgames

Close-up on tagging algorithm



- tag ☺ with ✓other leaves with ✗
- tag (v, K) with ✓if in the subgame starting at (v, K) ○ has a strategy to reach ✓

Close-up on tagging algorithm



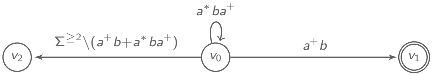
- tag ☺ with ✓other leaves with ✗
- tag (v, K) with ✓if in the subgame starting at (v, K) ○ has a strategy to reach ✓

How to do this in PSPACE?

- in a DFS, store only subgames and tags that are relevant
- any subgame for (v, K) is of polynomial size and has polynomially many exits (v', K')
- the height of the DAG is polynomial
- once a tag is computed, one can forget the whole sub-DAG

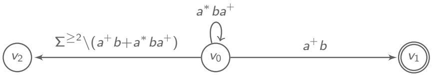
Strategy synthesis for coalition

Strategy synthesis for coalition of arbitrarily many players



$$\exists \sigma_1 \sigma_2 \cdots \ \forall \mathbf{k} \ \text{Plays}(\sigma_1, \sigma_2, \cdots, \sigma_{\mathbf{k}}) \subseteq \mathbf{Win}?$$

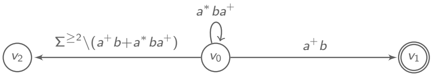
Strategy synthesis for coalition of arbitrarily many players



$$\exists \sigma_1 \sigma_2 \cdots \ \forall \mathbf{k} \ \text{Plays}(\sigma_1, \sigma_2, \cdots, \sigma_{\mathbf{k}}) \subseteq \mathbf{Win}?$$

At step i, Player i plays b and all others play a
is a winning coalition strategy to reach v_1

Strategy synthesis for coalition of arbitrarily many players



$$\exists \sigma_1 \sigma_2 \cdots \ \forall \mathbf{k} \ \text{Plays}(\sigma_1, \sigma_2, \cdots, \sigma_{\mathbf{k}}) \subseteq \mathbf{Win}?$$

At step i, Player i plays b and all others play a
is a winning coalition strategy to reach v_1

Collective strategies map histories to ω-words $\vec{\sigma}(v_0^n) = a^{n-1} b a^{\omega}$

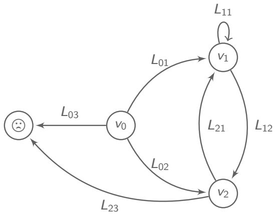
How to play?

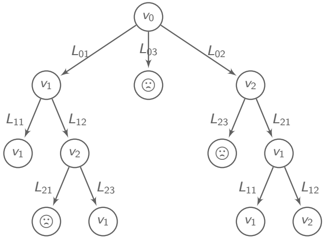
- environment chooses number of players $\mathbf{k}$, unknown to them
- at vertex v, players collectively choose an ω-word $\mathbf{w}$
  environment chooses edge $v \xrightarrow{L} v'$ with $\mathbf{w}_{\leq \mathbf{k}} \in L$
- players may learn some info about their number
- game proceeds from v'

Synthesis of collective strategy for safety objectives

From game arena build **tree unfolding** and stop

- either if the same label already appears for an ancestor
- or when label is ☹



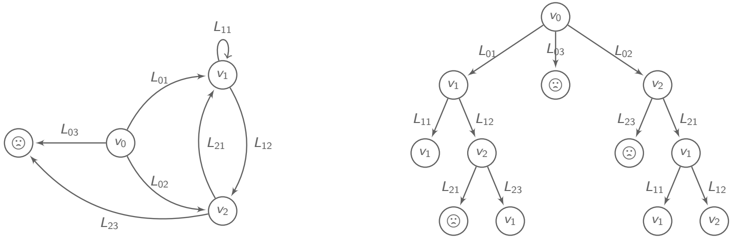


equivalently, coalition strategies map inner nodes of the tree to ω-words

Synthesis of collective strategy for safety objectives

From game arena build **tree unfolding** and stop

- either if the same label already appears for an ancestor
- or when label is ☹



equivalently, coalition strategies map inner nodes of the tree to ω-words

One can build a doubly exponential deterministic safety automaton over Σ^m ($m = \sharp$ inner nodes) that accepts winning strategies.

Existence of a winning coalition strategy is in **EXPSPACE** (and **PSPACE-hard**)

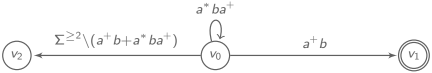
Contributions

- Definition of concurrent games with arbitrary many players
- Eve vs Rest of the world 
  - reduction to knowledge game (2-player and turn-based)
  - reachability objectives are PSPACE-complete
- Strategy synthesis for coalition 
  - safety objective are in EXPSPACE and PSPACE-hard

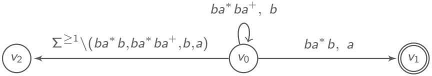
On-going work

Strategy synthesis for coalition: reachability

A positive instance



A negative instance



- even for very basic arenas, the problem seems non trivial
- challenge: acceleration techniques seem needed both on knowledge and on ω-words