Nash equilibrium in quantitative games played on graphs

Thomas BRIHAYE* Véronique BRUYÈRE* Julie DE PRIL*
Hugo GIMBERT†

† LABRI, Bordeaux (France)

* University of Mons (Belgium)

LogICCC Final Conference

GASICS project

Computer programming

Computer programming is a difficult task:

- understand deeply the initial problem;
- find a solution;
- write the program correctly.

Computer programming

Computer programming is a difficult task:

- understand deeply the initial problem;
- find a solution;
- write the program correctly.

$\hookrightarrow$ Does the program behave as expected in every possible situation that can happen during its use?

Computer programming

Computer programming is a difficult task:

- understand deeply the initial problem;
- find a solution;
- write the program correctly.

↪ Does the program behave as expected in every possible situation that can happen during its use?

Definition

A software bug is an error, a failure in a computer program or system that induces an incorrect result.

Software bugs

Bug example: In August 2005, a Malaysian Airlines Boeing 777 that was on autopilot suddenly ascended 3,000 feet.

Software bugs

Bug example: In August 2005, a Malaysian Airlines Boeing 777 that was on autopilot suddenly ascended 3,000 feet.

Bug consequences:

- loss of confidence from users' point of view,
- loss of credibility from institutions' point of view,
- large financial loss,
- human loss,...

Software bugs

Bug example: In August 2005, a Malaysian Airlines Boeing 777 that was on autopilot suddenly ascended 3,000 feet.

Bug consequences:

- loss of confidence from users' point of view,
- loss of credibility from institutions' point of view,
- large financial loss,
- human loss,...

$\Rightarrow$ Real need to **verify** the correctness of a program!

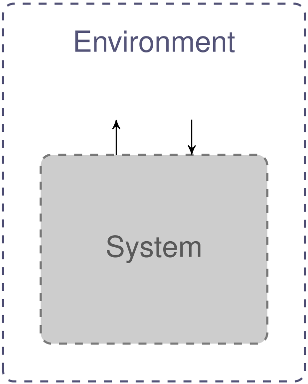
The autopilot case

Requirement: to arrive safe and sound in every weather conditions.

Difficulty: more than 18 million lines of code contained in the computer system of a plane!

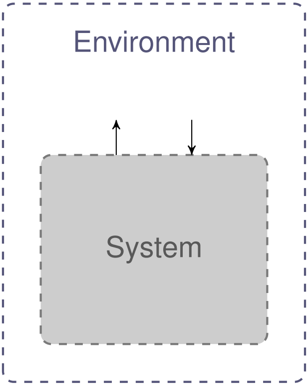
Consequence: for practical reasons (deadlines, costs), it is impossible to test such a system in every situation that it might encounter.

Modelisation



Expected functionality

Modelisation

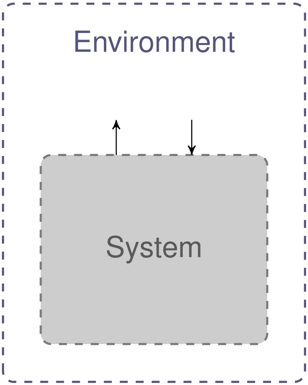


$\rightsquigarrow$ Mathematical model

? satisfies ?

Expected functionality $\rightsquigarrow$ Logical formula

Modelisation



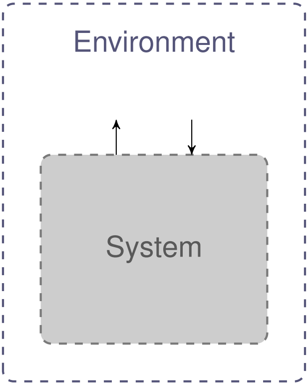
$\rightsquigarrow$ Mathematical model

GAME

? satisfies ?

Expected functionality $\rightsquigarrow$ Logical formula

Modelisation



$\rightsquigarrow$ Mathematical model

2-player, zero-sum, qualitative GAME

? satisfies ?

Expected functionality $\rightsquigarrow$ Logical formula

The autopilot case

Restrictive model because:

- there are several planes in the air,
- planes do not have antagonist objectives,
- it does not enable to verify a functionality while minimizing the fuel consumption.

The autopilot case

Restrictive model because:

- there are several planes in the air,
- planes do not have antagonist objectives,
- it does not enable to verify a functionality while minimizing the fuel consumption.

$\Rightarrow$ Release the constraints of 2-player, zero-sum, qualitative games.

$\Rightarrow$ Consider and study n-player, non zero-sum, quantitative games.

(Part of the GASICS project)

From winning strategies to equilibria

- 2-player, zero-sum, qualitative games:

  $\hookrightarrow$ looking for a winning strategy for one player.

- n-player, non zero-sum, quantitative games:

  $\hookrightarrow$ need for a different concept.

  $\hookrightarrow$ looking for a Nash equilibrium for all players.

A toy example: the printer battle

In a publishing house, **3 persons** intend to print their own document.

Gaston
Office worker

Jeanne
Secretary

Fantasio
Editor-in-chief

(Some characters of the belgian comic strip Gaston)

A toy example: the printer battle

In a publishing house, **3 persons** intend to print their own document.

Gaston
Office worker

Jeanne
Secretary

Fantasio
Editor-in-chief

(Some characters of the belgian comic strip Gaston)

2 printers: printer P_1 and printer P_2.

The printer battle as a game...

3 persons	$\hookrightarrow$	3 players
2 printers	$\hookrightarrow$	2 choices for each player

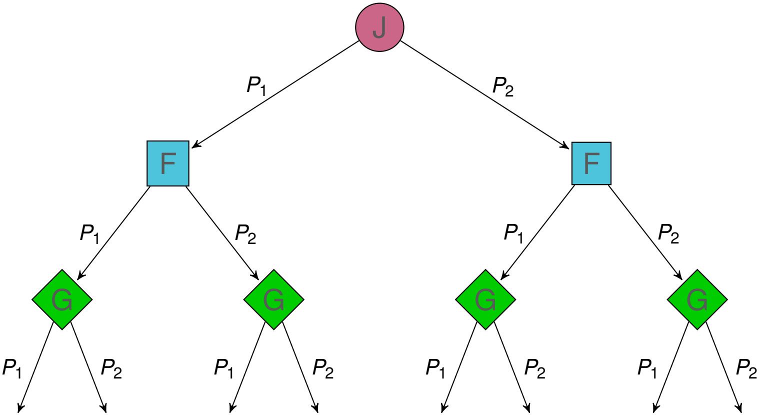
Particularity of printer P_1: paper jam from 2 sent documents.

Aim of each player: manage to print her/his document.

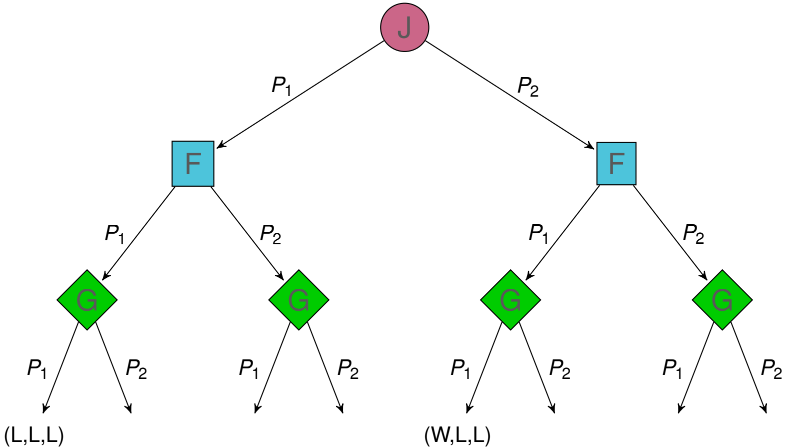
$\hookrightarrow$ A player is winning if her/his document is printed.

$\boxed{\Rightarrow \textbf{3-player, non zero-sum, qualitative game.}}$

Qualitative modelisation of the printer battle



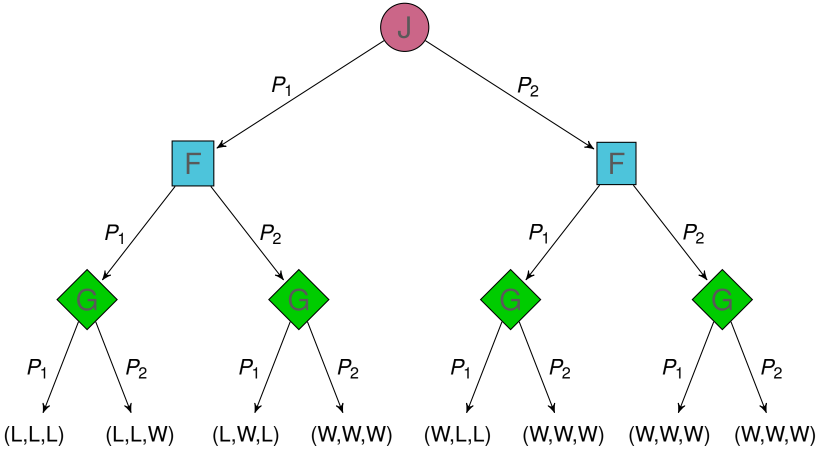
Qualitative modelisation of the printer battle



All of them are losing.

Jeanne is winning, men are losing.

Qualitative modelisation of the printer battle



All of them are losing.

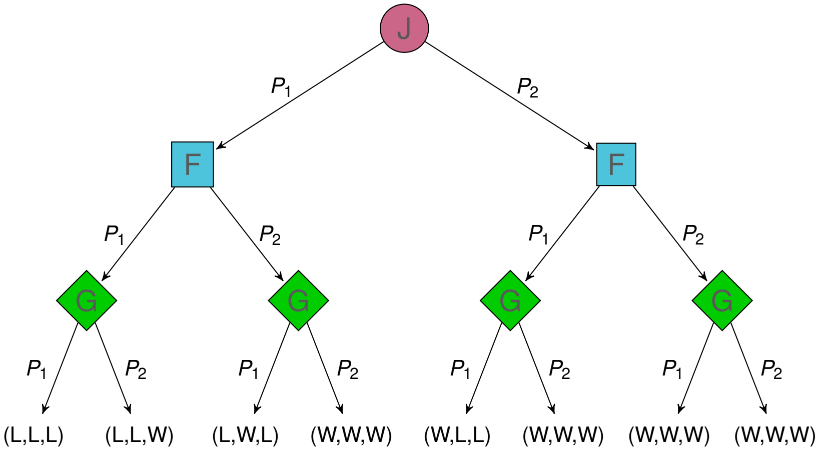
Jeanne is winning, men are losing.

Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.

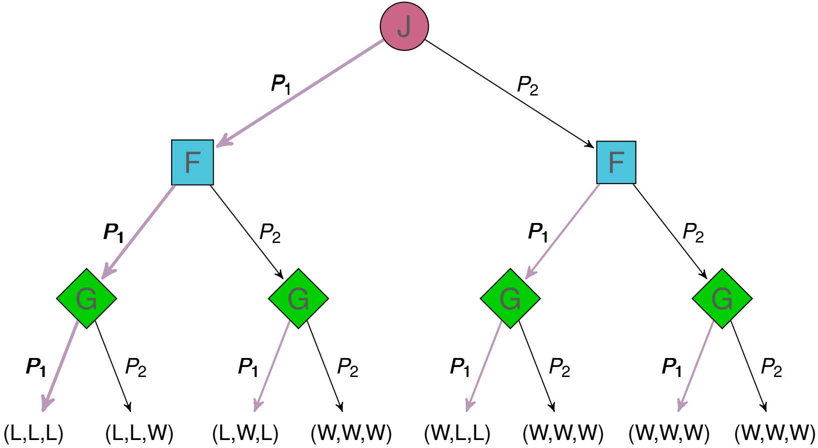
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



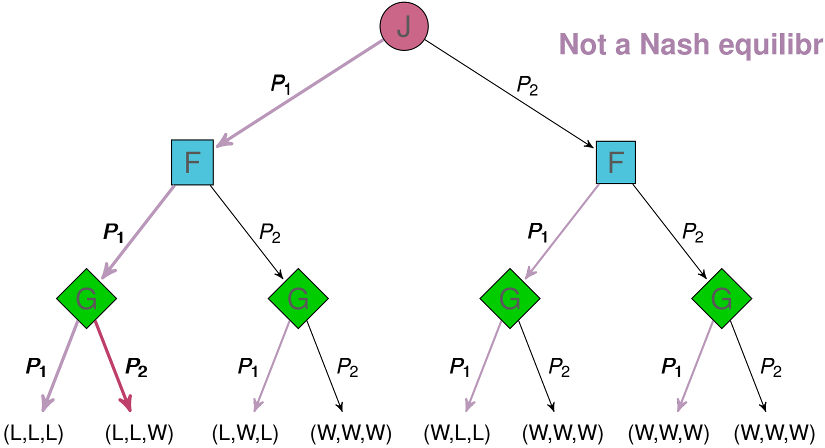
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



Nash equilibrium

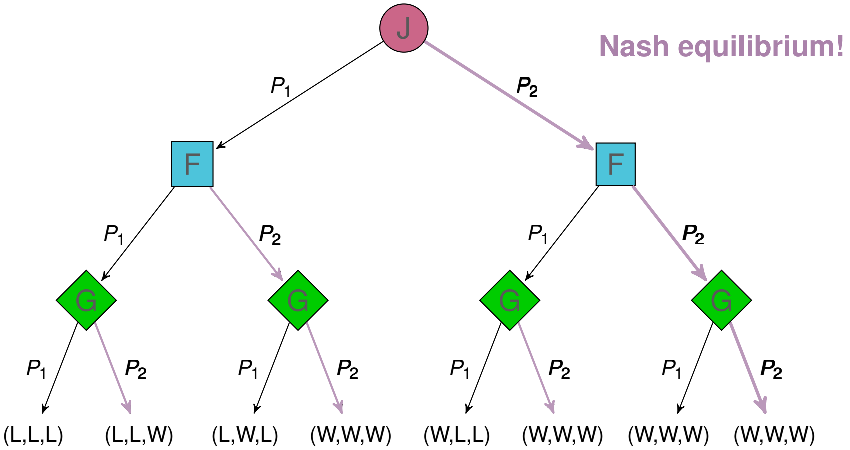
Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



Not a Nash equilibrium!

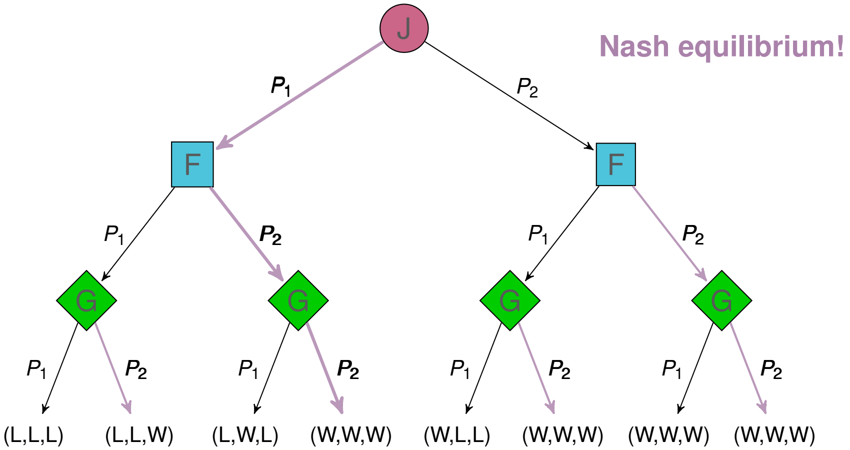
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



Let us add time constraint...

Jeanne, Fantasio and Gaston are all in a hurry.

Other particularity of printer P_1: two times faster than printer P_2.

Printer P_1 $\hookrightarrow$ 1 second/document

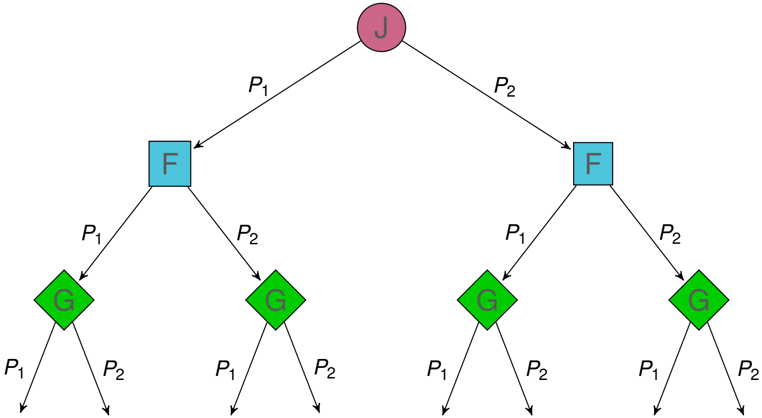
Printer P_2 $\hookrightarrow$ 2 seconds/document

Aim of each player: minimize the time to print one's document.

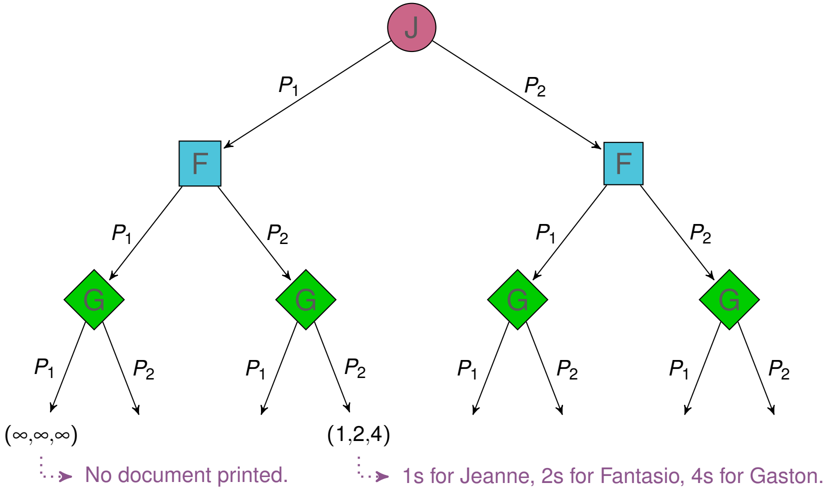
$\hookrightarrow$ The less time it takes, the happier a player is.

$\Rightarrow$ **3-player, non zero-sum, quantitative game.**

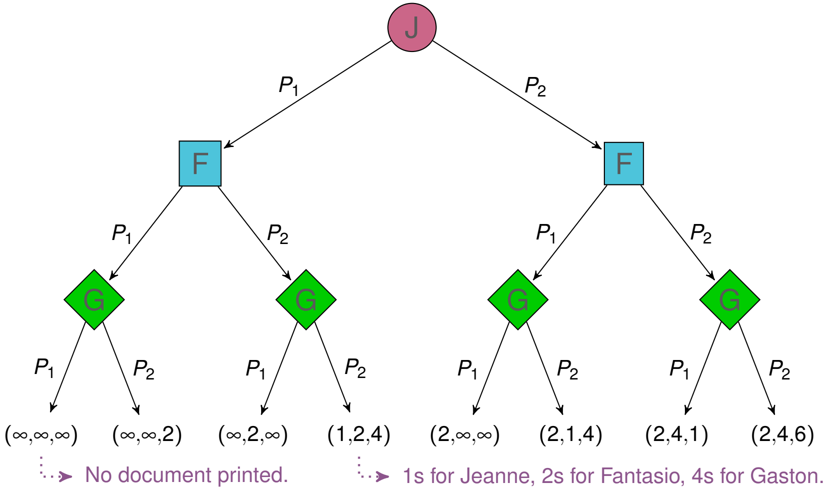
Quantitative modelisation of the printer battle



Quantitative modelisation of the printer battle



Quantitative modelisation of the printer battle

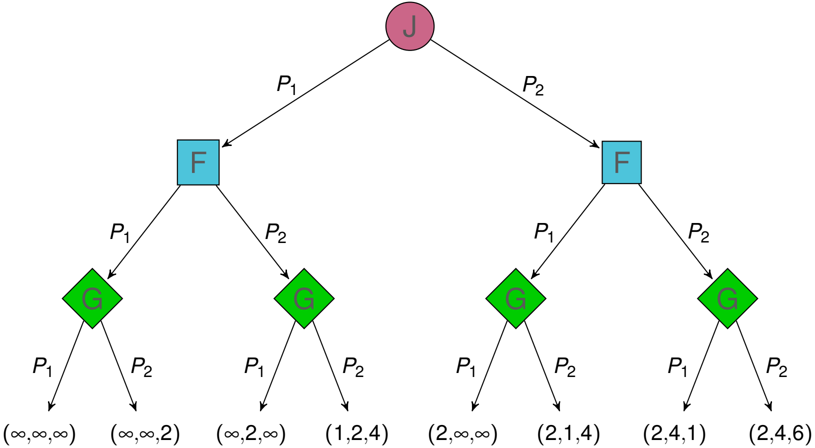


Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.

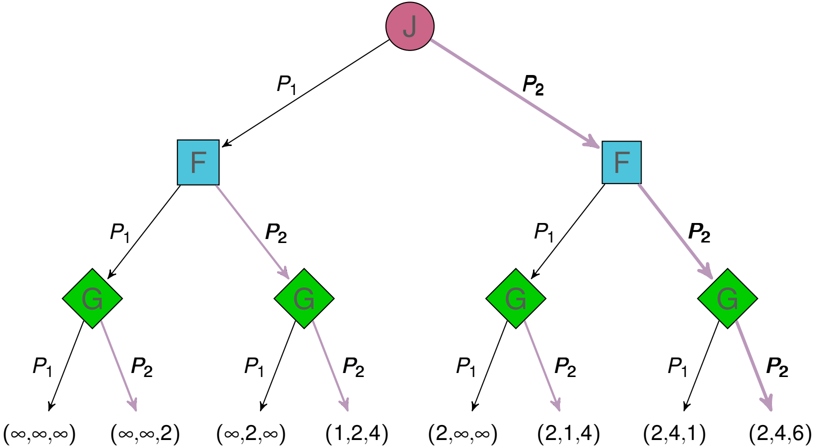
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



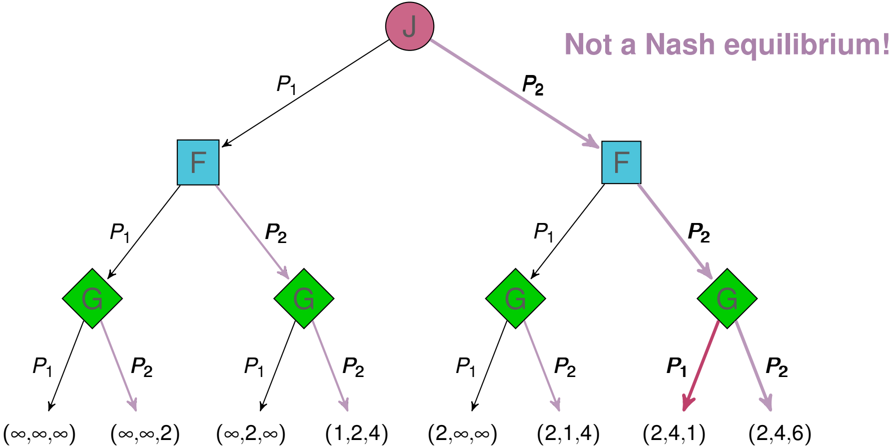
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



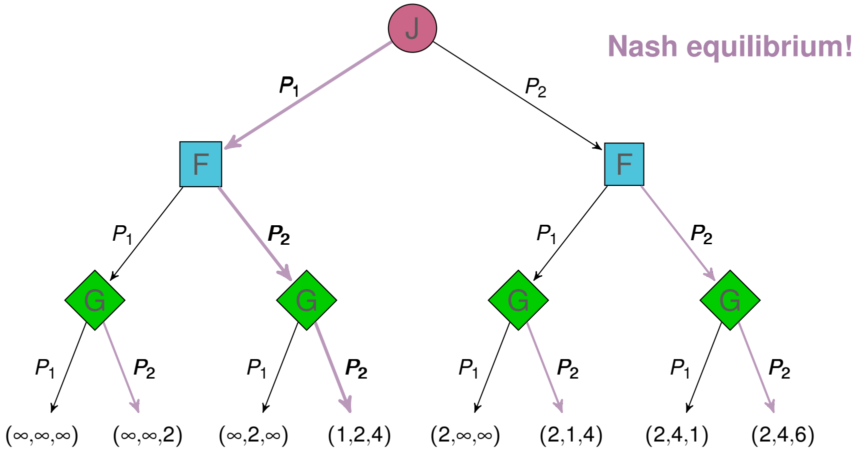
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



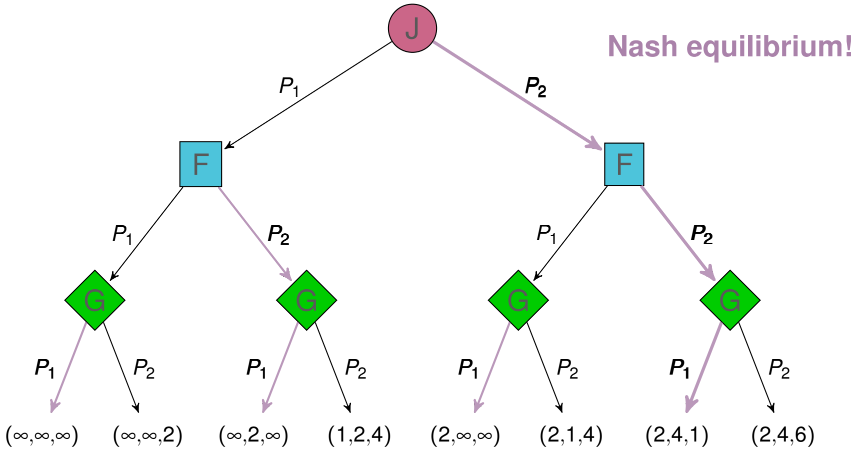
Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



Nash equilibrium

Idea: A strategy profile where no player has an incentive to deviate from the strategy chosen, given the choices of the other players.



Some of our results in reachability games

Theorem ([BBD10])

There exists a finite-memory Nash equilibrium in every n-player quantitative reachability game.

Theorem ([BBD10])

Given a NE in a n-player quantitative reachability game, there exists a finite-memory NE with the same set of "winning" players.

[BBD10] T. Brihaye, V. Bruyère, J. De Pril, Equilibria in Quantitative Reachability Games, CSR, vol. 6072 of *LNCS*, 72-83. Springer 2010.

More recent work

In *n*-player quantitative reachability games [BBDG11] :

- Existence of a subgame perfect equilibrium.
- Algorithm to decide existence of a secure equilibrium.

In *concurrent* *n*-player quantitative reachability games [KLST11] :

- Algorithm to extract strategies of Nash equilibria.
- Complexity of the decision variant of the problem.

[BBDG11] Ongoing work of T. Brihaye, V. Bruyère, J. De Pril and H. Gimbert.

[KLST11] M. Klimos, K. G. Larsen, F. Stefanak, J. Thaarup, Finding Nash Equilibria in Concurrent Priced Games.

Thank you for your attention!