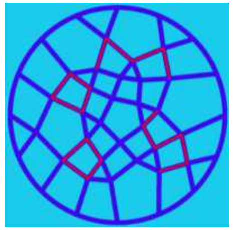
Some Properties of Switching Games on oriented Matroids.



Adrien Vieilleribière.
LRI Université Paris-Sud.

joint work with
David Forge.
LRI Université Paris-Sud.

Combinatorial Geometries and Applications: Oriented Matroids and Matroids
Marseille-Luminy, November 8 2005

Initial Switching Game

- The Shannon Switching Game : [S60]
  - On undirected graphs
  - Maker / Breaker game
  - Construct a s-t path
  - Who win ? How ?

- Complete Solution on Graphs :
  - A. Lehman : A solution to the Shannon Switching Game, [SIAM J. 1964]
  - simplified proof using Nash William's Theorem [Edmonds]

Solve the game on graphs.

- Maker wants to Mark a s-t path
  - Induction on spawning chains

- Maker wins

⇕

- the graph has a subgraph connecting s to t with two edge-disjoint spanning trees

Various Switching Games.

- Change who begin
- Give more than one play by turn (biased version)
- Add previous claimed elements
- Other substructure as Goal
  - Cut, Spanning tree, Circuit
- Other structures :
  - Directed graphs
  - Matroids
  - Oriented Matroids

Outline of the Talk

- Playing on a matroid
  - Graphs
  - Oracles and Matroids
- Playing on oriented matroids
  - Some oriented games with comparison
  - Weak maps and uniform matroids
- General representation by a game tree
  - Examples with known families
    - Cases 2-4, 6-3 and 8-4
- Playing on a Lawrence matroid
  - Make explicit a winning strategy
  - Biased extensions
- Rank complexity
- Conclusion

Representing matroids.

- On a ground set E
- Exponential space in an explicit description
- Can be defined by oracles :
  - Independence oracle

    $X \subseteq P(E)$,

    is X independent ?
  - Port oracle

    $e \in E \quad X \subseteq P(E)$,

    does X contain a circuit containing e ?

Solving circuit game on matroids.

- Maker wants to Mark a circuit on M
  - Characterized by block-matroids [Hamidoune,Las Vergnas 84]
  - Time : Poly(|E|) with $|E|^2$ independent oracle calls [Kelmans,Polesskii 94]

  - With an independence oracle in time polynomial, we can compute winning strategies in PTIME.
    - Cover Graphs by an easy PTIME oracle.
    - Cover matroids defined by submodular functions. Ellipsoid algorithm leading to strongly PTIME

Extending to threshold computation on matroids.

- Biased : Maker has q choices by turn.
- threshold computation : finding the smallest q such that Maker wins → trivially harder than deciding who wins.
  - e-based ear decomposition of M.
  - with $O(|E|^2)$ ports oracles in time $O(|E|^3)$.
- with a port oracle in time polynomial, we can decide who wins in PTIME.
  - Extend efficient decision on graphs in the biased game.
    - Applications (goal=integer density subgraph)

[Bednarska,Pikhurko 05]

Independence Oracle Vs Port Oracle.

- With $O(|E|^4)$ calls to port oracle and $O(|E|^5)$ time, we can simulate an independence oracle.

- Port Oracle are meaningful if Maker has to build a circuit through a given element.
  - It abstracts the cost of final position validation.

- In the undirected case, efficient port (or independence) oracle leads to efficient solution for these switching games.

Outline of the Talk

- Playing on a matroid.
- Playing on oriented matroids.
  - Some oriented games with comparison.
  - Weak maps and uniform matroids.
- General representation by a game tree.
  - Examples with known families.
    - Cases 2-4, 6-3 and 8-4.
- Playing on a Lawrence matroid.
  - Make explicit a winning strategy.
  - Biased extensions.
- Rank complexity.
- Conclusion.

Game 1 on oriented matroid :

Maker orients one element by turn.
Maker begins.

Breaker orients one element by turn.

- Game 1 : $OC^{(1^{*},\,1^{+})}_{(\{e\},\,\emptyset)}$

No restriction on the circuit to build

One first element given.

Goal : Build a Circuit of the inputed oriented matroid

Game 3 on oriented matroid :

Maker orients one element by turn.
Maker begins.

Breaker removes one element by turn.

- Game 3 : OC $\begin{pmatrix} 1^*, 1 \end{pmatrix}$ ({e} , {e})

Circuit has to contain e.

One first element given.

Goal : Build a Circuit of the inputed oriented matroid

Comparing Games :

For Maker

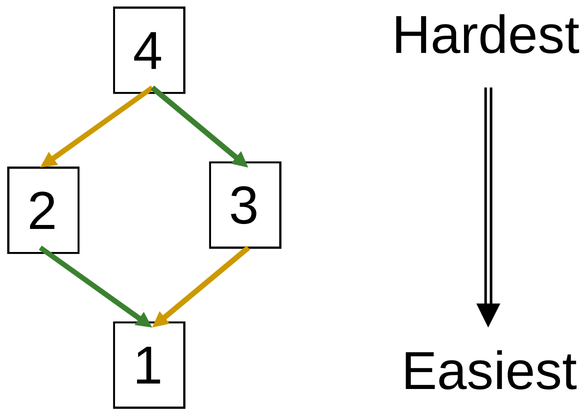
- Game 1 : $OC^{(1^*,\ 1^+)}_{(\{e\},\ \emptyset)}$
- Game 2 : $OC^{(1^*,\ 1^+)}_{(\{e\},\ \{e\})}$
- Game 3 : $OC^{(1^*,\ 1)}_{(\{e\},\ \emptyset)}$
- Game 4 : $OC^{(1^*,\ 1)}_{(\{e\},\ \{e\})}$

$(1^*, 1) \rightarrow (1^*, 1^+)$

3->1 , 4->2

$(\{e\}, \{e\}) \rightarrow (\{e\}, \emptyset)$

2->1 , 4->3



Weak maps :

- Let M_1 and M_2 be two oriented matroids on the same ground set.
  - There is a weak map from M_1 to M_2 when every signed circuit of M_1 contains a circuit of M_2.
  - If M_1 and M_2 have the same rank, a weak map from M_1 to M_2 is said rank preserving.
  - Notation : $M_1 \prec M_2$.

- Let $\mathbf{\Phi_1}$ and $\mathbf{\Phi_2}$ be two family of matroids on the same ground set.
  - Notation : $\mathbf{\Phi_1} \prec \mathbf{\Phi_2}$ iff

    $\forall\ M_2 \in \mathbf{\Phi_2}\ \exists\ M_1 \in \mathbf{\Phi_1}$ such that $M_1 \prec M_2$.

Extending winning strategies with weak maps

- A matroid is said to be winning if Maker have a winning strategy when he begins.

- If M is winning and M_2 is such that $M \prec M_2$, then M_2 is winning.

- If Φ is winning and $\Phi \prec \Phi_2$ then Φ_2 is winning.

Considering uniform matroids is sufficient

- Every Oriented matroid is the weak map of a uniform oriented matroid of the same rank.

  [Björner,Las Vergnas & al. 1999]

- With the previous slide : Corollary :

  If every uniform matroid of $\mathbf{M}_{n,r}$ is winning,

  then every matroid of rank k on n elements is winning.

- This technique can be used in every Game OC $\genfrac{}{}{0pt}{}{(\ \_\ ,\ \_\ )}{(\ \_\ ,\ \emptyset\ )}$

On uniform matroids:

- If 2r > n then Maker looses.
  - All circuits have the same cardinality r+1.
  - Maker finishes Game 3 with $\lfloor n/2 \rfloor$+1 elements.
  - If maker doesn't choose enough elements, he cannot win.

- Interesting matroids :
  - 2r=n.

Outline of the Talk

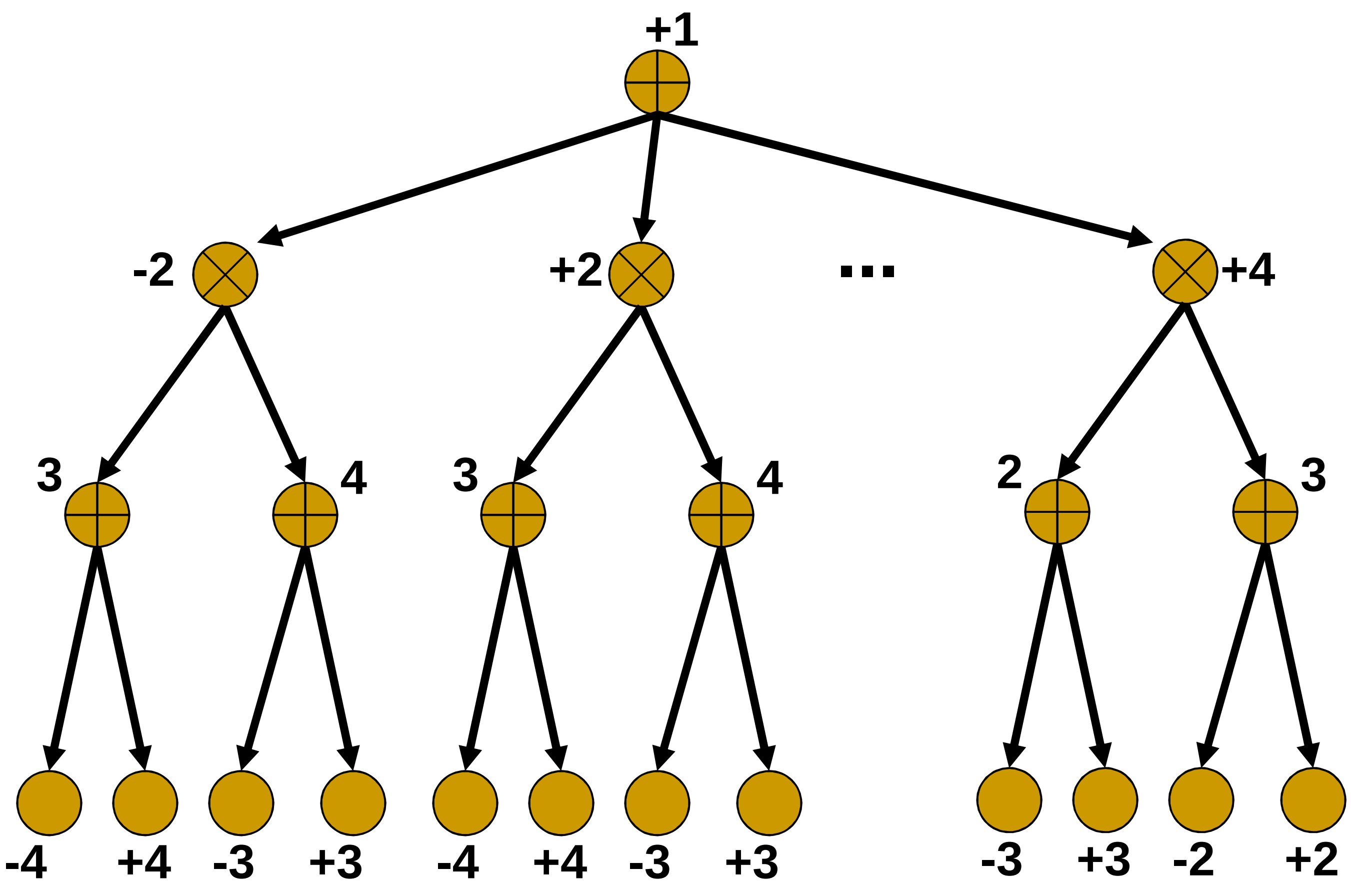
- Playing on a matroid.
- Playing on oriented matroids.

- General representation by a game tree.
  - Examples with known families.
    - Cases 2-4, 6-3 and 8-4.

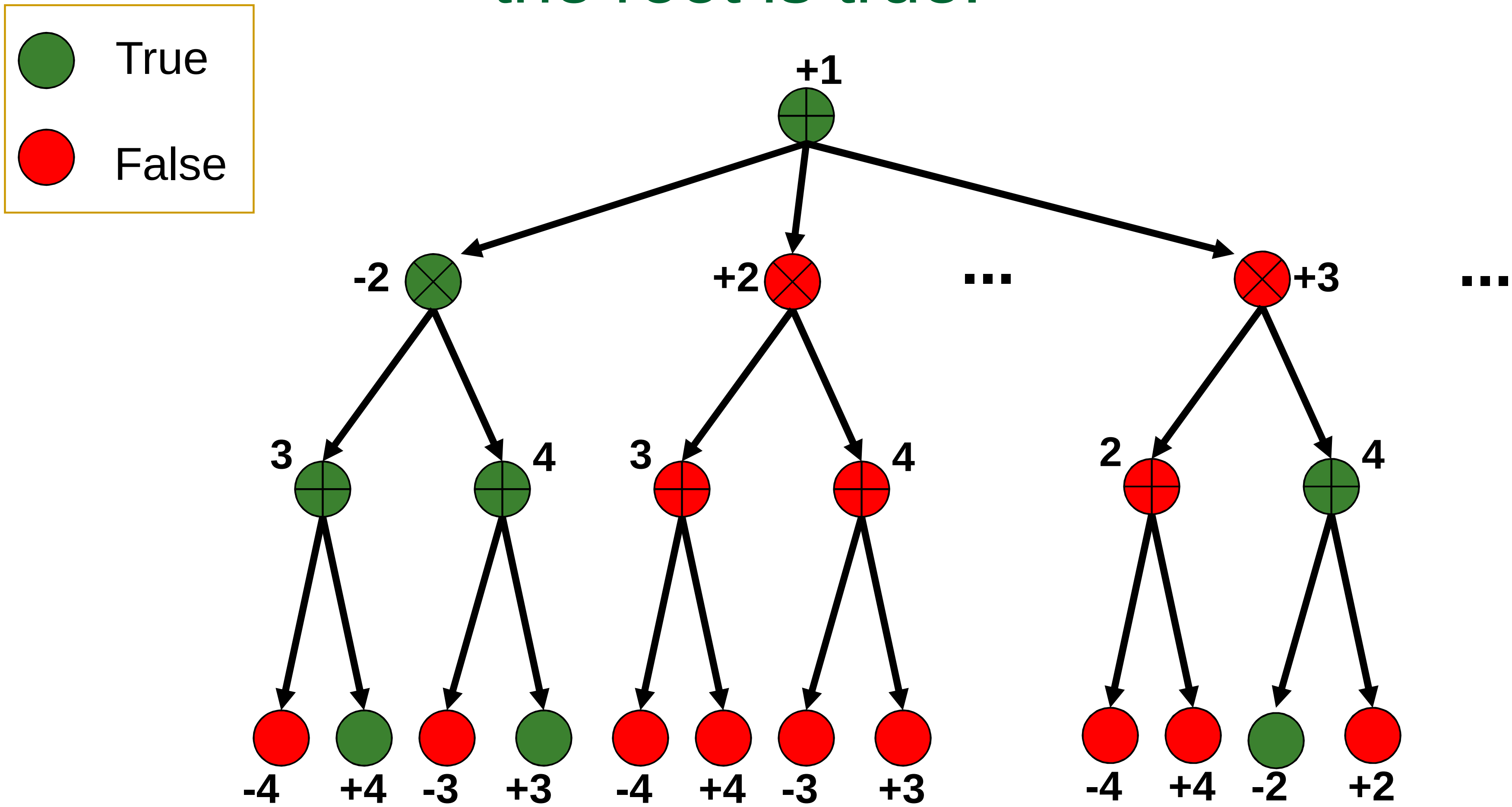
- Playing on a Lawrence matroid.
  - Make explicit a winning strategy.
  - Biased extensions.
- Rank complexity.
- Conclusion.

Tree representation. Game 3.

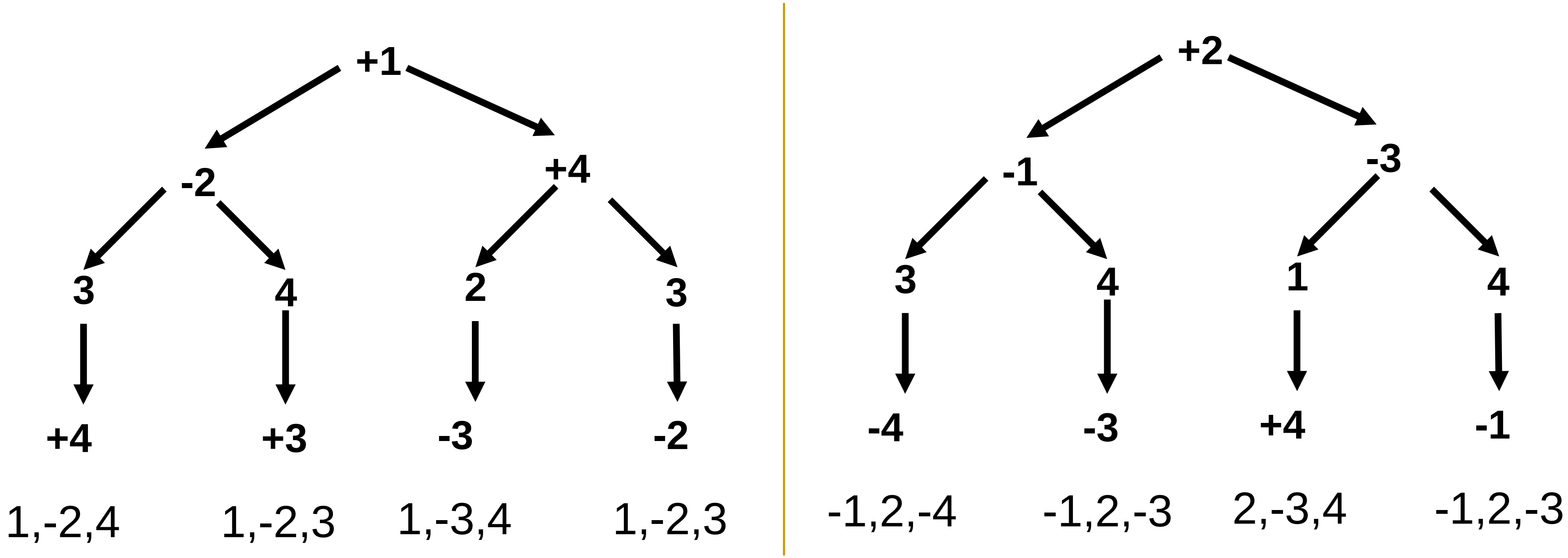
- Root : {e}.
- Maker : OR node.
- Breaker : AND node.
- Leaf : Circuit oracle.



There is a winning strategy for Maker iff the root is true.



Example : Game 3 or 4, 4 elements, rank 2



Only one matroid up to isomorphism :

The alternated matroid : ++++++ : 1-2-3-4.

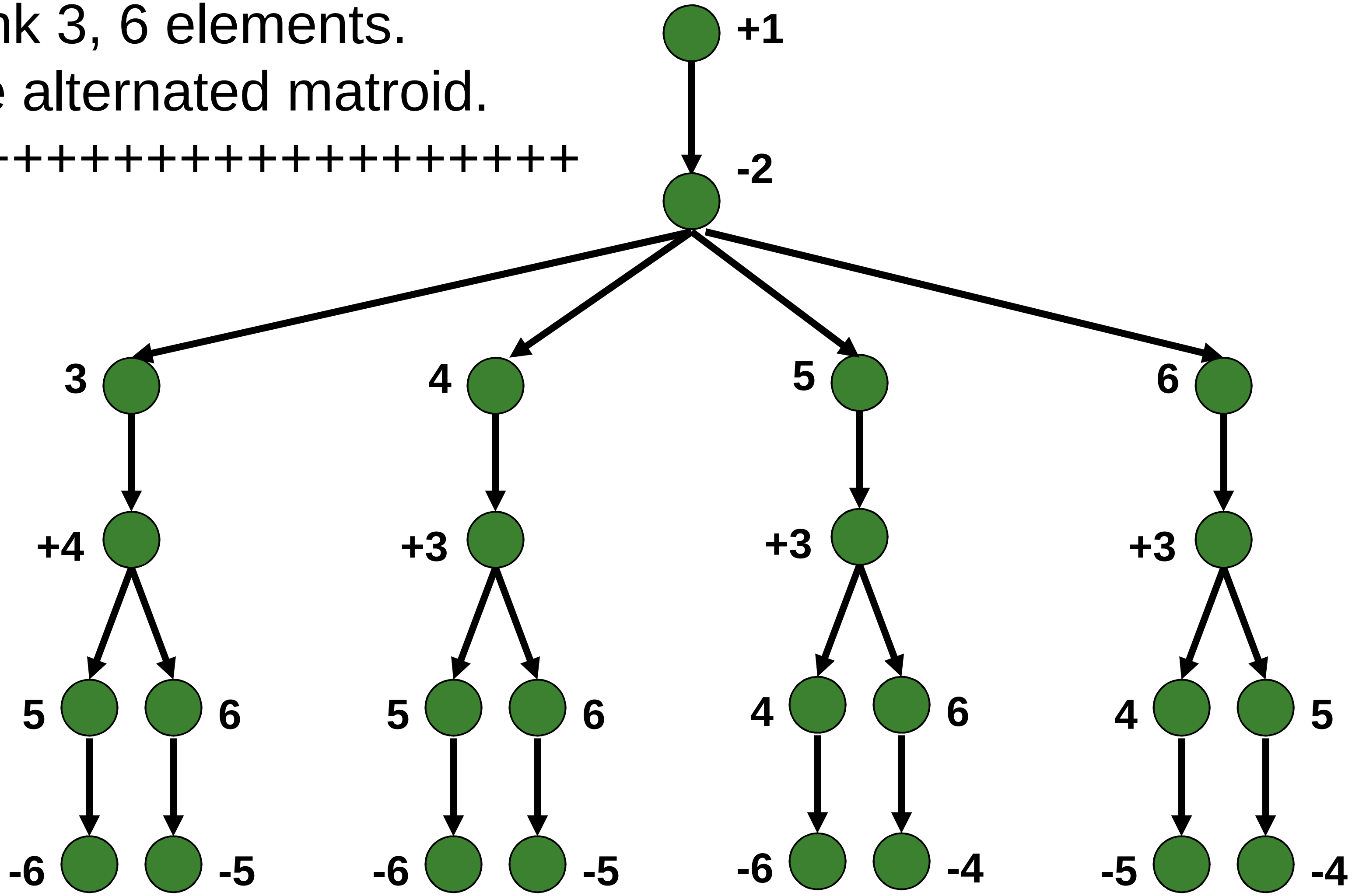
For any {e}, Maker wins iff he begins.

Example 1 : 1/3

Rank 3, 6 elements.

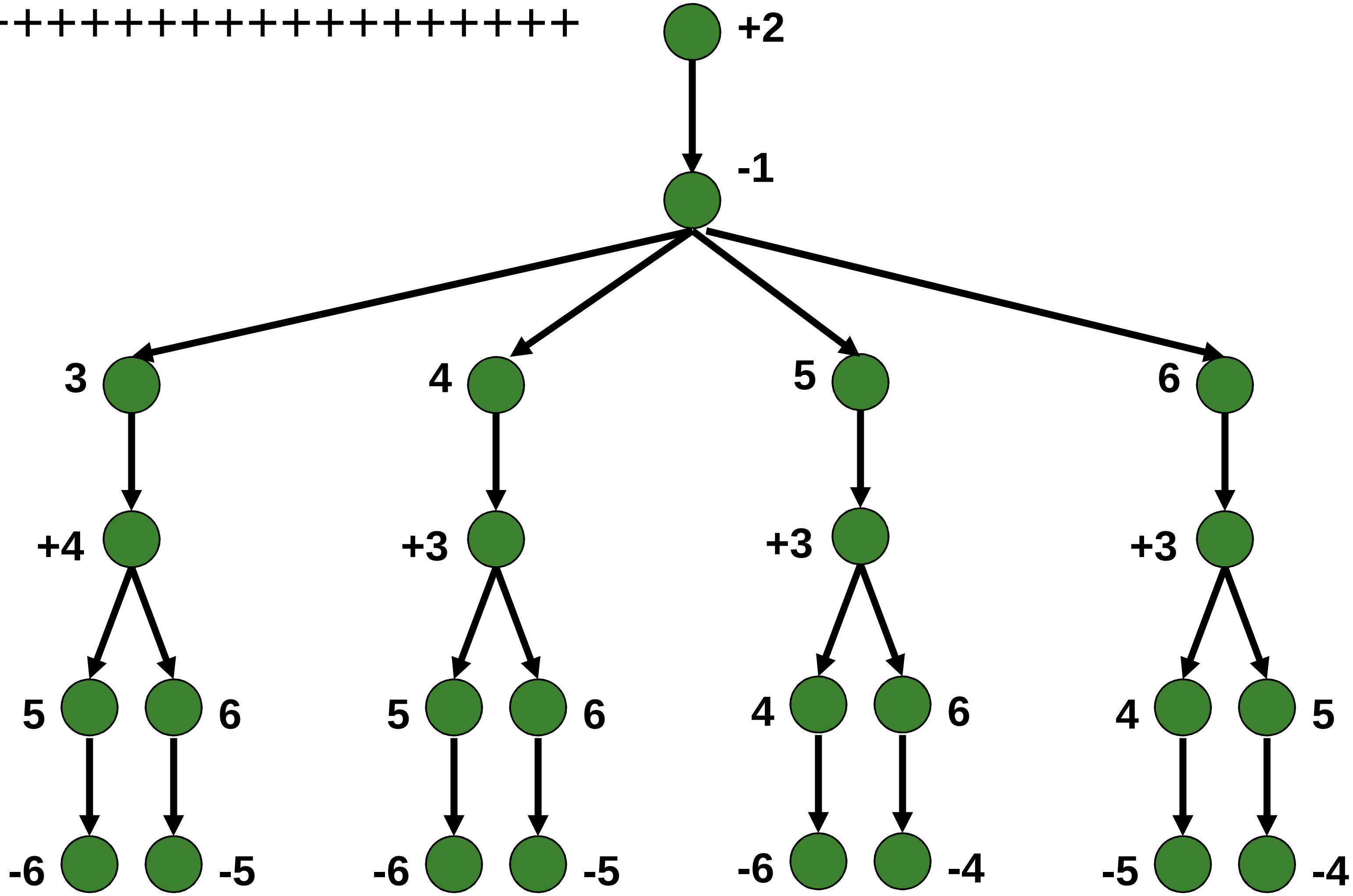
The alternated matroid.

++++++++++++++++++++



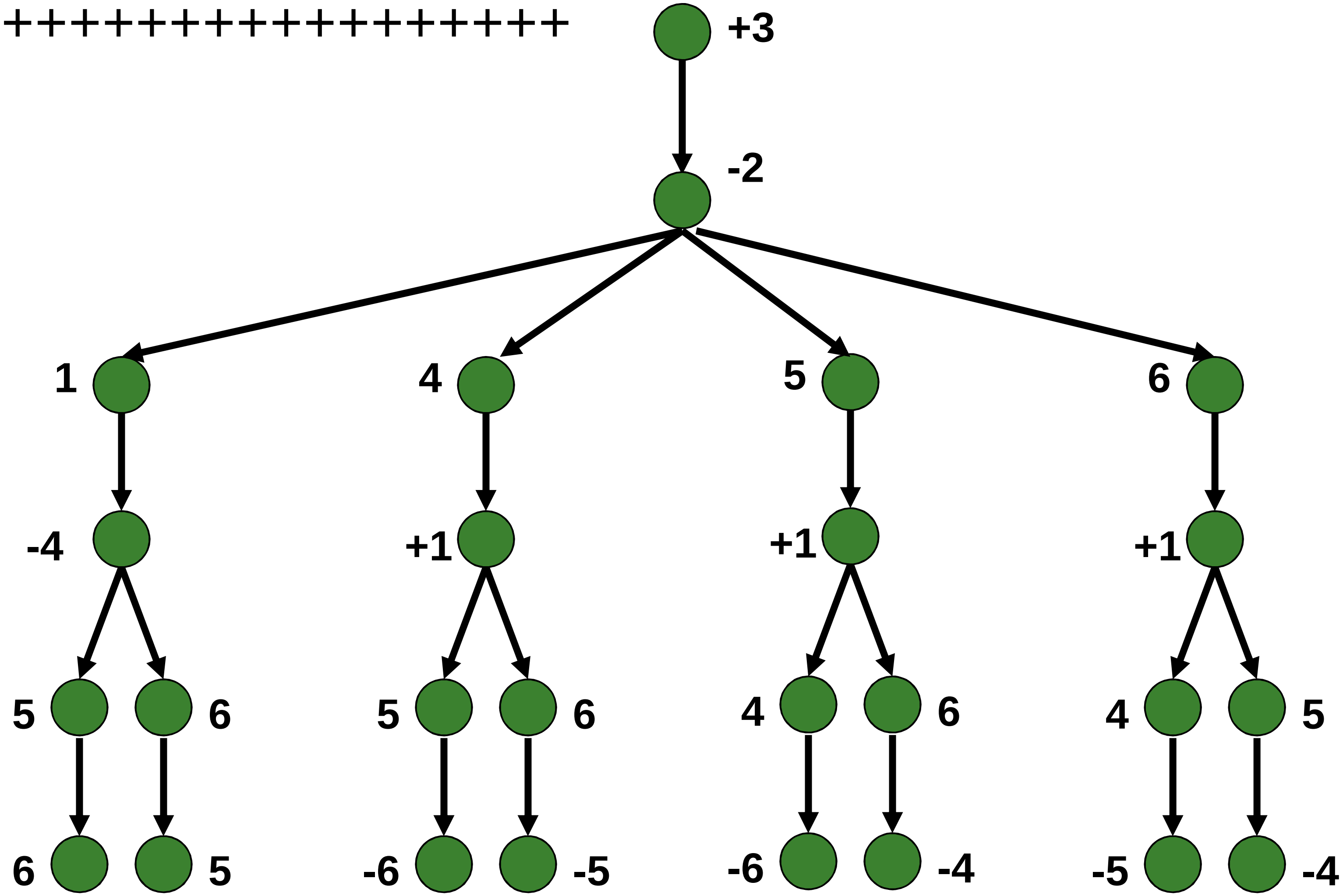
Example 1 : 2/3

++++++++++++++++++++



Example 1 : 3/3

++++++++++++++++++++++



Lawrence Matroid

- Let $A = (a_{i,j})$ a $r \times n$ matrix of 1 and –1
- The chirotope of L_A, the associated Lawrence matroid, is

  $\chi(i_1 < \ldots < i_r) = \Pi_{k=1..r}\, a_{k,ik}$.

- Properties :
  - Uniform matroid.
  - Vectorial matroid.
  - Exactly n simplicial cells.
  - Let $C=\{ i_1 < i_2 < \ldots < i_{r+1} \}$ be a circuit of the matroid.

    A signature of C is given by $C(i_1)=+$ and recursively

    $C(i_{j+1}) = - C(i_j) * a_{j,ij} * a_{j,ij+1}$.

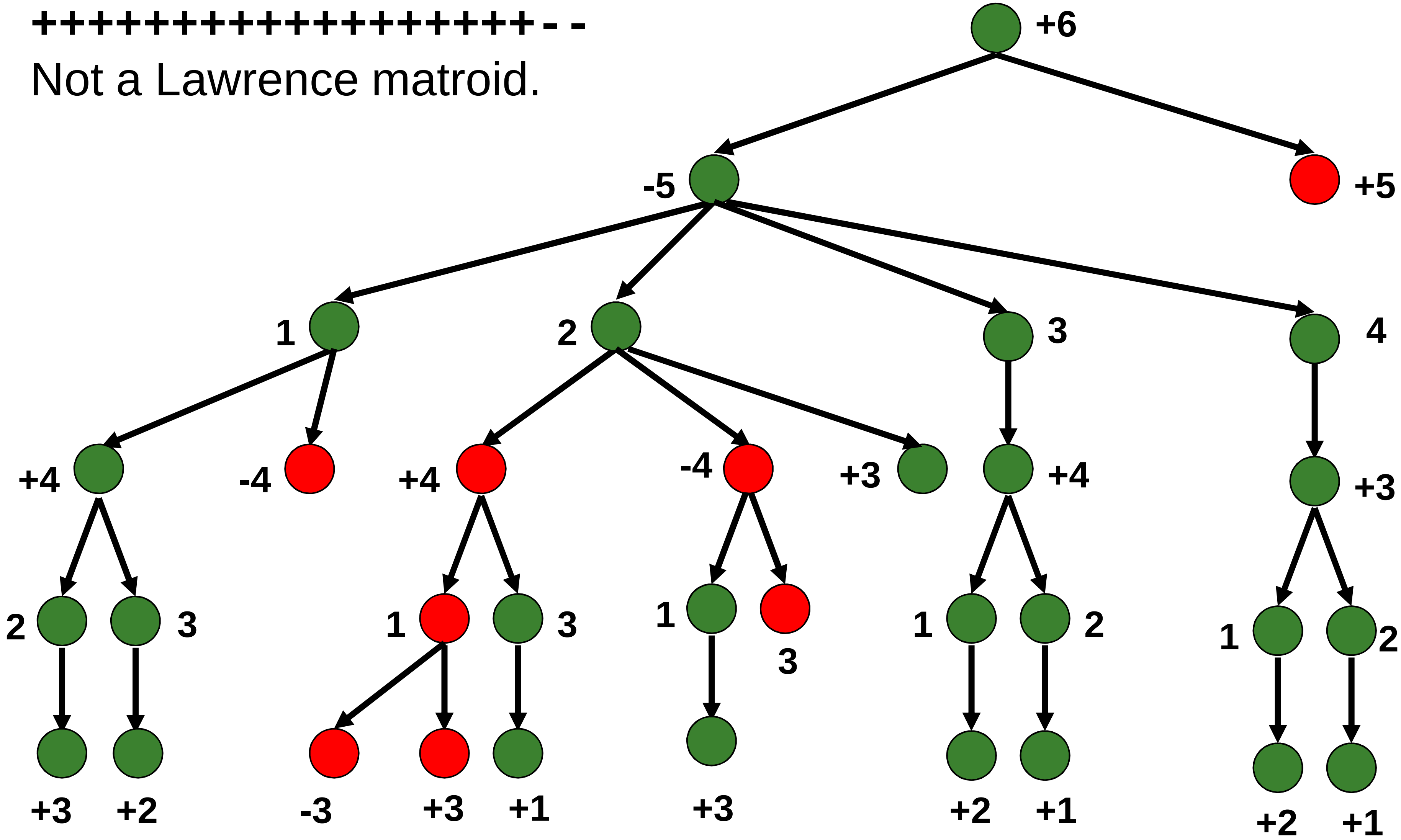
Example 2 with 6 elements and rank 3.

- **+++++++++++++++++++-**
  - Particular case of Lawrence matroids
    - Signing correctly elements is simple
    - Clear extensions for all the four games

Example 3 :

++++++++++++++++++++- -

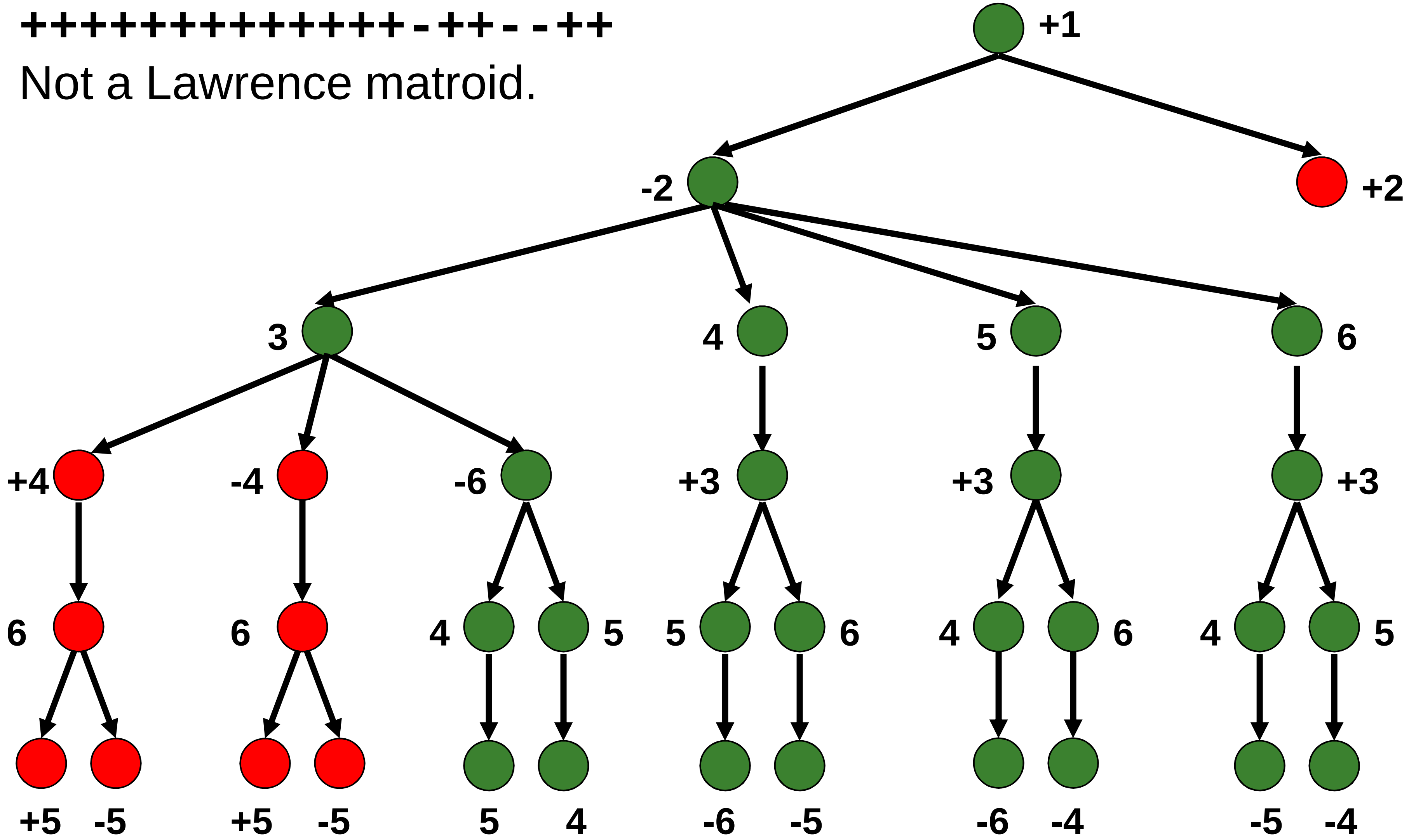
Not a Lawrence matroid.



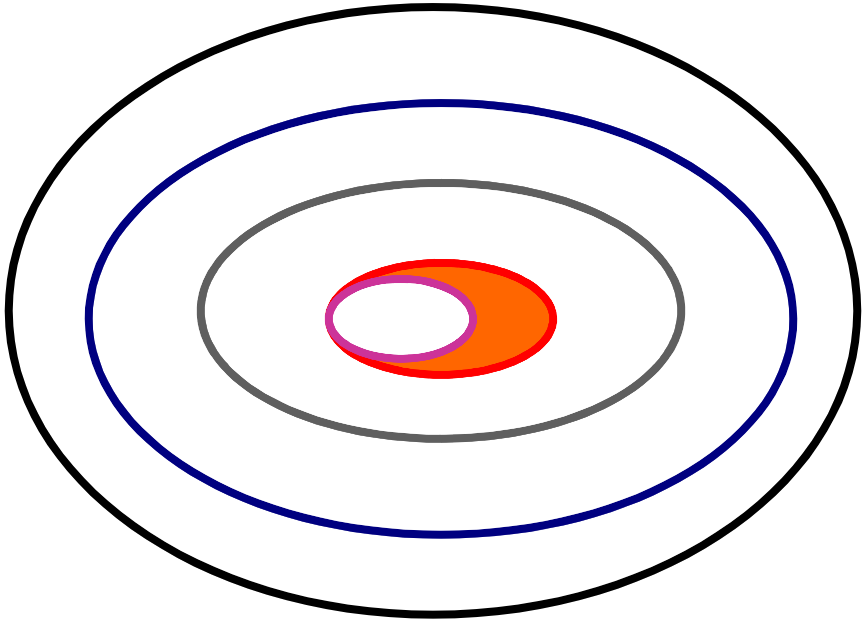
Example 4 :

++++++++++++++-++--++

Not a Lawrence matroid.

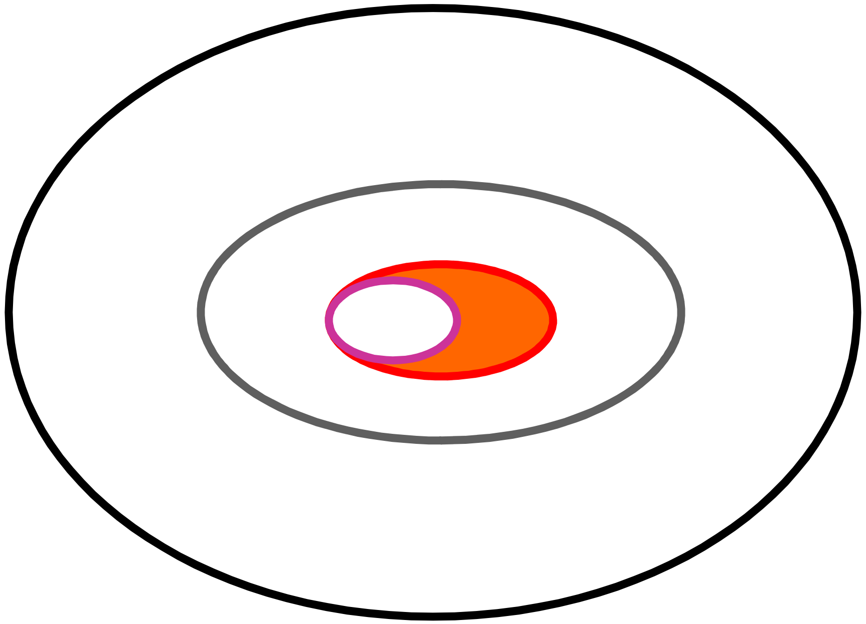


3-6 Summary.



- **2^{20} = 1048576 sign vectors**
- **23808 matroids**
- **23808 winning vectors**
- **17 Isomorphic classes**
- **4 uniform representants**
- **2 Lawrence**

8-4 summary



- **2^{70} = 1180591620717411303424 ≈ 10^{21} sign vectors**
  - **? Matroids**
  - **? winning vectors**
- **181472 Isomorphic classes**
- **2628 uniform representants**
  - **? Lawrence (soon)**

Outline of the Talk

- Playing on a matroid.
- Playing on oriented matroids.
- General representation by a game tree.

- Playing on a Lawrence matroid.
  - Making explicit a winning strategy.
  - Biased extensions.

- Rank complexity.
- Conclusion.

Making explicit a winning strategy on Lawrence matroids.

- Winning strategy : general principle
  - x_0 is given (+)
  - Maker orients r elements $x_1,\ldots,x_r$
  - When maker orients an element x_i, he knows its final position in the circuit and a previous claimed neighbor x_j.
  - He can sign x_i using the relation
    - $S(x_i) = -S(x_j) * a_{k,x_i} * a_{k,x_j}$ such that x_i and x_j will be in position k and k+1 in the circuit.

Example

1	2	3 -	4 +	5	6
1	-1	1	-1	1	1
1	1	-1	-1	1	1
1	1	1	-1	1	-1

Example

1	2	3	4	5	6
		-	+		-
1	-1	1	-1	1	1
1	1	-1	-1	1	1
1	1	1	-1	1	-1

Example

1	2	3	4	5	6
+		-	+		-
1	-1	1	-1	1	1
1	1	-1	-1	1	1
1	1	1	-1	1	-1

Simple characterization of the existence of a Maker winning strategy on Lawrence matroids.

- Theorem :

The game 3 is winning on a Lawrence matroid of rank r and of order n if and only if $n \geq 2r$.

Extending the threshold computation on Lawrence matroids.

- Theorem :

  The biased (1*, q) version of game 3 is winning on a Lawrence matroid of rank r and of order n

  if and only if $n \geq r(q+1)$.

- With a similar strategy.
- Undirected version : $n \geq (r-1)(q+1)+2$
- q-1 more elements needed in the directed case.

Outline of the Talk

- Playing on a matroid.
- Playing on oriented matroids.
- General representation by a game tree.
- Playing on a Lawrence matroid.

- Rank complexity.

- Conclusion .

Space complexity needed for the rank.

- $C_=L$: Languages such that it exists a non deterministic logarithmic space-bounded machine such that $x \in L$ iif for input x,

  #accepting paths = # rejecting paths.

- $\text{Ver.RANK}^{-} = \{ (A,r) \mid A \in Z^{m \times n}, n \in N, \text{rank}(A) < r \}$
- $\text{Ver.RANK} = \{ (A,r) \mid A \in Z^{m \times n}, n \in N, \text{rank}(A) = r \}$

- Ver.RANK^{-} is complete for $C_=L$ and Ver.RANK is complete for the second level of the boolean hierarchy above $C_=L$

- Implies hardness for NL and membership in TC^1 and NC^2.

Time and operations bounds for the rank.

- Parallel model : Rank for $n \times n$ matrices in polylogtime $O^*(1)$ using $n^{2.376}$ processors.

- The rank r of $A \in K[x]^{n \times n}$ degree d and n-r independent nullspace vectors can be computed in
  - Time $O^*(n \times MM(n,\log||A||))$
    [Moenck,Carter 1979]
  - $O(n^\alpha + n^2 \log ||A||)$ operations in K
    [Bürgisser,Clausen,Shokrollahi 1997]
  - $O^*(MM (n,d)) = O^*(n^\alpha d)$ operations in K
    [Storjohann,Villard 2005]

Reduction Conclusions.

- Computing the rank is easy over a field
  - → a representation can help
  - → space complexity from rank approach is

    $O(n^2) + \text{SpaceCost}(C_{=}L) = O(n^{2.7})$

- Focus on uniform oriented matroids.
- On Lawrence matroid, studies all the four games collapse.
- Focus on non Lawrence matroids.

Space Conclusions.

- A PSPACE oracle for circuits leads to
  - a PSPACE decision for “who wins”
  - a PSPACE algorithm to choose of a good “move” if one exists
  - This space upper bound can be easily extended for PSPACE oracle deciding final positions

- Since PSPACE hardness is already known for game 4 on oriented graphs, we expect this exact space complexity for simple extensions of the game 3.

Future Experimental Work.

- Use of duality remarks.

- Intensive study for 4-8 case : 70 signs
  - Complete checkout seems too hard (2^{70})
  - Testing can help

- Toward the case 5-10 : 252 signs
  - Isomorphic classes for the 5-10 case are not known
  - Without a complete matroid generator, testing is inescapable

- Implementation of weak maps

- Algorithm to get a Lawrence representation if possible

Future Theoretical Work.

- Explain the experimental coincidence between existence of a winning strategy and the 3-term Grassmann Plucker relations.

- Find a larger class of oriented matroids upon with linear (or polynomial, non greedy) strategies exist.

- Precise the cost of weak map argument for a effective uniform standardisation.

- Precise the hardness increase for breaker signing or previous given elements in general ?

Thank you for your attention.

Questions ?

vieille@lri.fr