Proving combinatorial properties of graphs using model theory

Pierre Ohlmann, Michał Pilipczuk, Wojciech Przybyszewski, Szymon Toruńczyk

University of Warsaw

Highlights 2023

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

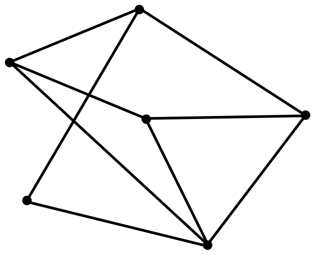
Splitter wins once he deletes the last vertex.

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.
Example play of the treedepth game:



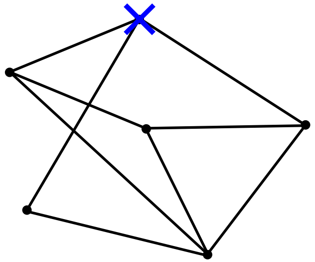
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:



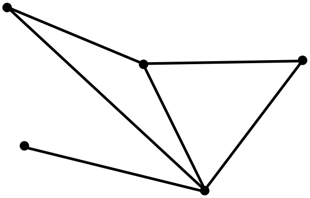
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:



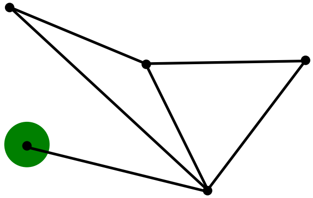
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:

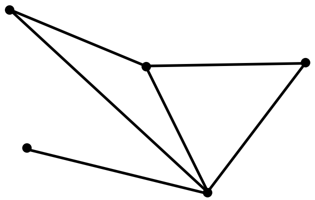


Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.
Example play of the treedepth game:



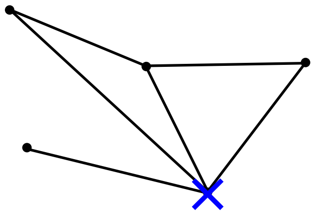
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:



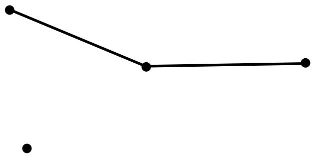
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:



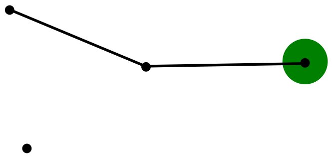
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:



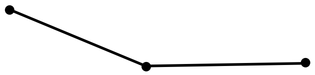
Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:

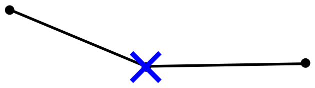


Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.
Example play of the treedepth game:



Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.
Example play of the treedepth game:

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Example play of the treedepth game:

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.
Example play of the treedepth game:

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Definition

A treedepth of a graph G is the minimum number of rounds that are enough for Splitter to always win the treedepth game, no matter how Connector is playing.

Treedepth game

The treedepth game is played on a graph G_1. In round i

1. Splitter chooses a vertex v to delete
2. Connector chooses G_{i+1} as a connected component in $G_i - v$.

Splitter wins once he deletes the last vertex.

Definition

A treedepth of a graph G is the minimum number of rounds that are enough for Splitter to always win the treedepth game, no matter how Connector is playing.

Observation: We don't need to assume that G is a finite graph for this definition to make sense.

Progressing moves in the treedepth game

Theorem. [Ohlmann, Pilipczuk, P., Toruńczyk, 2023]

There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that if a graph G has treedepth d then Splitter has at most $f(d)$ progressing moves[1].

[1] A vertex v is a progressing move for Splitter if every connected component C of $G - \{v\}$ has strictly smaller treedepth than G.

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Consider the following theory over the signature that consists of constant symbols $\{v_i : i \in I\} \cup \{v_\infty\}$ and one binary relation E:

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Consider the following theory over the signature that consists of constant symbols $\{v_i : i \in I\} \cup \{v_\infty\}$ and one binary relation E:

- $v_i \neq v_j$ for every $i, j \in I$;

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Consider the following theory over the signature that consists of constant symbols $\{v_i : i \in I\} \cup \{v_\infty\}$ and one binary relation E:

- $v_i \neq v_j$ for every $i, j \in I$;
- $E(v_i, v_j)$ for every $(v_i, v_j) \in E(G)$ and $\neg E(v_i, v_j)$ for every $(v_i, v_j) \notin E(G)$;

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Consider the following theory over the signature that consists of constant symbols $\{v_i : i \in I\} \cup \{v_\infty\}$ and one binary relation E:

- $v_i \neq v_j$ for every $i, j \in I$;
- $E(v_i, v_j)$ for every $(v_i, v_j) \in E(G)$ and $\neg E(v_i, v_j)$ for every $(v_i, v_j) \notin E(G)$;
- Splitter wins the treedepth game in d rounds if he plays optimally;

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Consider the following theory over the signature that consists of constant symbols $\{v_i : i \in I\} \cup \{v_\infty\}$ and one binary relation E:

- $v_i \neq v_j$ for every $i, j \in I$;
- $E(v_i, v_j)$ for every $(v_i, v_j) \in E(G)$ and $\neg E(v_i, v_j)$ for every $(v_i, v_j) \notin E(G)$;
- Splitter wins the treedepth game in d rounds if he plays optimally;
- $v_\infty \neq v_i$ for every $i \in I$;

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.

Proof.

Assume statement doesn't hold. Denote $V(G) = \{v_i : i \in I\}$.

Consider the following theory over the signature that consists of constant symbols $\{v_i : i \in I\} \cup \{v_\infty\}$ and one binary relation E:

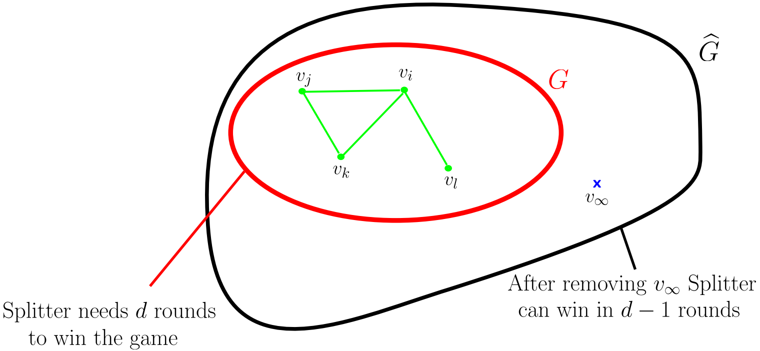
- $v_i \neq v_j$ for every $i, j \in I$;
- $E(v_i, v_j)$ for every $(v_i, v_j) \in E(G)$ and $\neg E(v_i, v_j)$ for every $(v_i, v_j) \notin E(G)$;
- Splitter wins the treedepth game in d rounds if he plays optimally;
- $v_\infty \neq v_i$ for every $i \in I$;
- v_∞ is a progressing move.

□

Progressing moves in infinite graphs

Lemma.

For every infinite graph G of treedepth d Splitter has finitely many progressing moves.



Progressing moves in infinite graphs

Theorem. [Ohlmann, Pilipczuk, P., Toruńczyk, 2023]

There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that if a graph G has treedepth d then Splitter has at most $f(d)$ progressing moves.

Progressing moves in infinite graphs

Theorem. [Ohlmann, Pilipczuk, P., Toruńczyk, 2023]

There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that if a graph G has treedepth d then Splitter has at most $f(d)$ progressing moves.

Proof.

Assume the statement is not true. Consider the following theory:

- Splitter can win the treedepth game in at most d rounds;

Progressing moves in infinite graphs

Theorem. [Ohlmann, Pilipczuk, P., Toruńczyk, 2023]

There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that if a graph G has treedepth d then Splitter has at most $f(d)$ progressing moves.

Proof.

Assume the statement is not true. Consider the following theory:

- Splitter can win the treedepth game in at most d rounds;
- there are at least m progressing moves for every $m \in \mathbb{N}$.

Progressing moves in infinite graphs

Theorem. [Ohlmann, Pilipczuk, P., Toruńczyk, 2023]

There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that if a graph G has treedepth d then Splitter has at most $f(d)$ progressing moves.

Proof.

Assume the statement is not true. Consider the following theory:

- Splitter can win the treedepth game in at most d rounds;
- there are at least m progressing moves for every $m \in \mathbb{N}$.

Compactness yields a model that contradicts the previous lemma. □

Summary

Techniques from model theory can be applied in finite model theory – in this way we get model theoretic proofs of combinatorial theorems.

Summary

Techniques from model theory can be applied in finite model theory – in this way we get model theoretic proofs of combinatorial theorems.

- Cannonical decompositions of graphs of bounded treedepth/shrubdepth. [Ohlmann, Pilipczuk, P., Toruńczyk, '23]

Summary

Techniques from model theory can be applied in finite model theory – in this way we get model theoretic proofs of combinatorial theorems.

- Cannonical decompositions of graphs of bounded treedepth/shrubdepth. [Ohlmann, Pilipczuk, P., Toruńczyk, '23]
- Game characterization of monadically stable classes of graphs. [Gajarský, Mählmann, McCarty, Ohlmann, Pilipczuk, P., Siebertz, Sokołowski, Toruńczyk, '23]

Summary

Techniques from model theory can be applied in finite model theory – in this way we get model theoretic proofs of combinatorial theorems.

- Cannonical decompositions of graphs of bounded treedepth/shrubdepth. [Ohlmann, Pilipczuk, P., Toruńczyk, '23]
- Game characterization of monadically stable classes of graphs. [Gajarský, Mählmann, McCarty, Ohlmann, Pilipczuk, P., Siebertz, Sokołowski, Toruńczyk, '23]

Thank you for your attention!