# A Note on Restricted Forms of LGG

Ondřej Kuželka[1] and Jan Ramon[2]
[1]Cardiff University, [2]KU Leuven

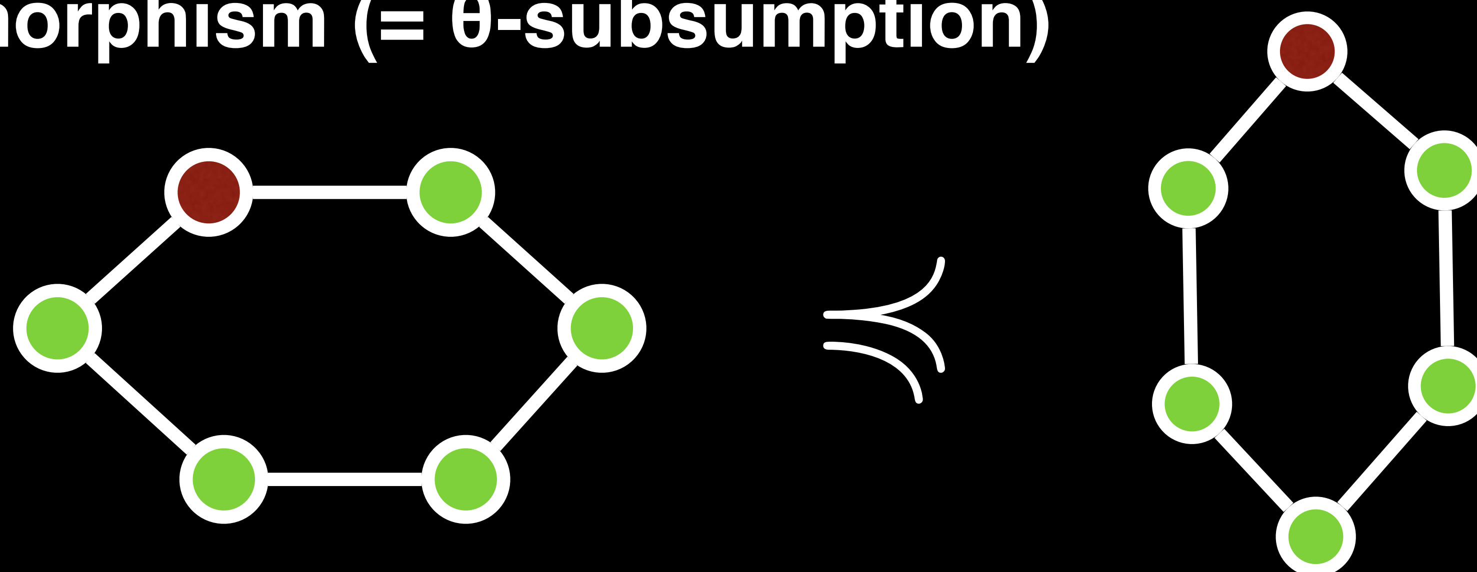
# What is this talk about?

- It is about a **negative answer** to a conjecture which we had and which has consequences for bottom-up learning in ILP.
- The negative answer strongly suggests optimality of the notion of ***bounded least general generalization*** [Kuželka, Szabóová & Železný, ILP'12]

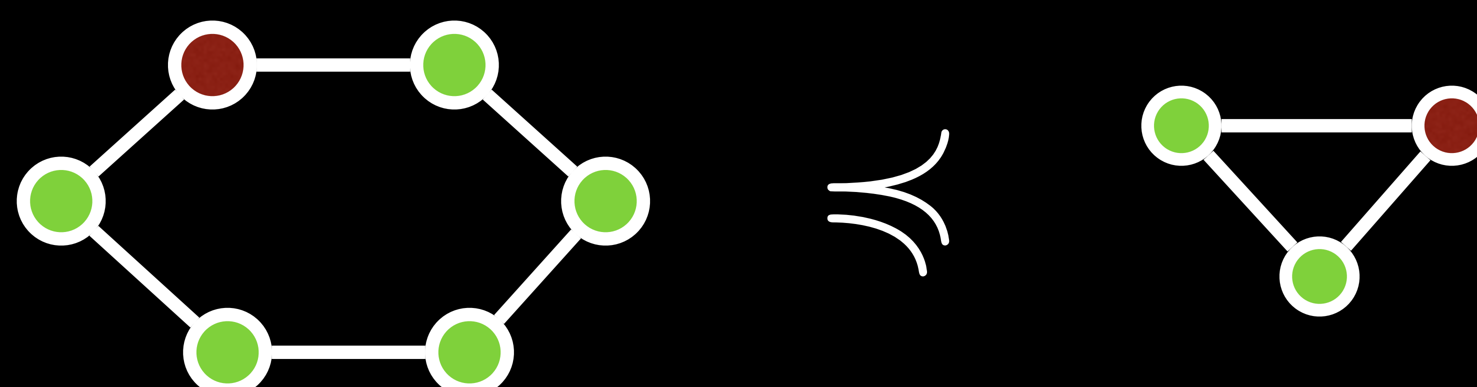
# Preliminaries

# Homomorphism

- **Homomorphism (= θ-subsumption)**

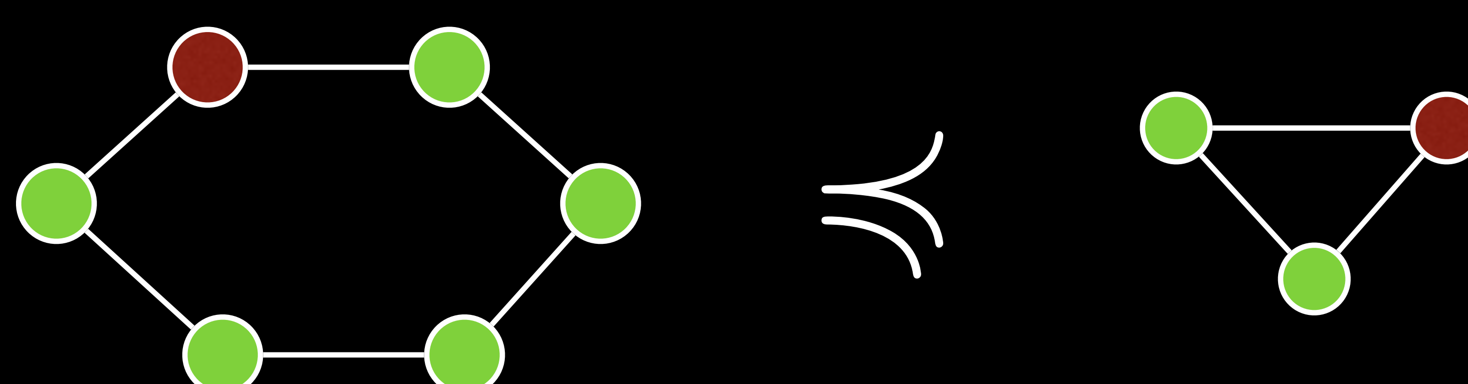


- **but also…**



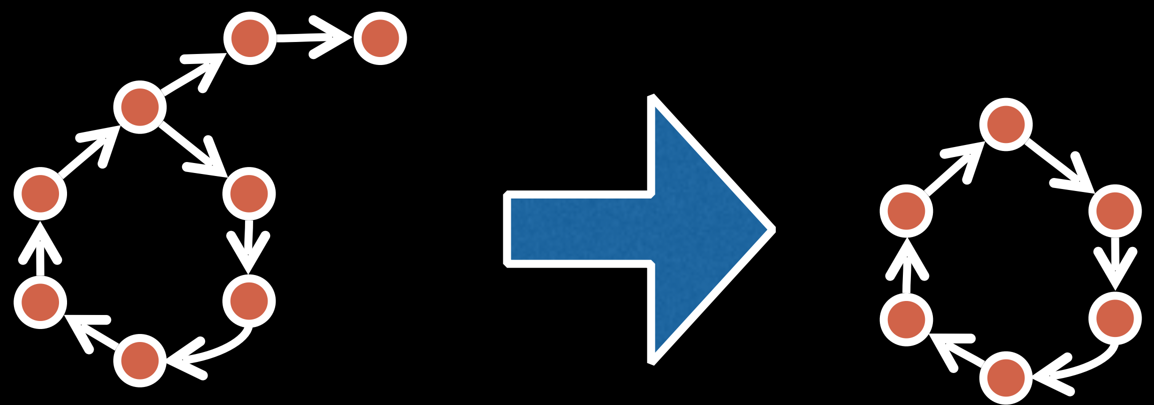
# θ-subsumption

- Essentially the same "thing" as homomorphism…
- Clause *A* θ-subsumes clause *B* if there is a substitution $\theta$ such that $A\theta \subseteq B$.
- Example:
  - $A = e(A,B) \vee e(B,C) \vee e(C,D) \vee e(D,E) \vee e(E,F) \vee e(F,A) \vee red(A) \vee e(B,A) \vee e(C,B) \vee e(D,C) \vee e(E,D) \vee e(F,E) \vee e(A,F)$
  - $B = red(X) \vee e(X,Y) \vee e(Y,Z) \vee e(Z,X) \vee e(Y,X) \vee e(Z,Y) \vee e(X,Z)$
  - Then $A\theta \subseteq B$, $\theta = \{A/X, B/Y, C/Z, D/Y, E/Z, F/Y\}$



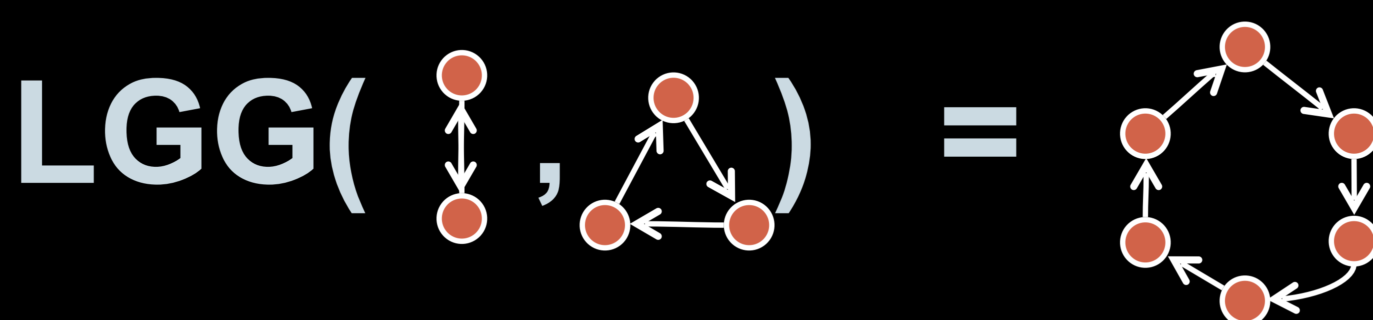
# Core (= θ-reduction)

- A graph G is a *core* if there is no smaller graph homomorphically equivalent to it.
- θ-reduction of a clause C is a clause R which is θ-equivalent to C and there is no smaller clause θ-equivalent to it.
- Deciding if a graph is a core is coNP-complete.



# Plotkin's Least General Generalization (LGG)

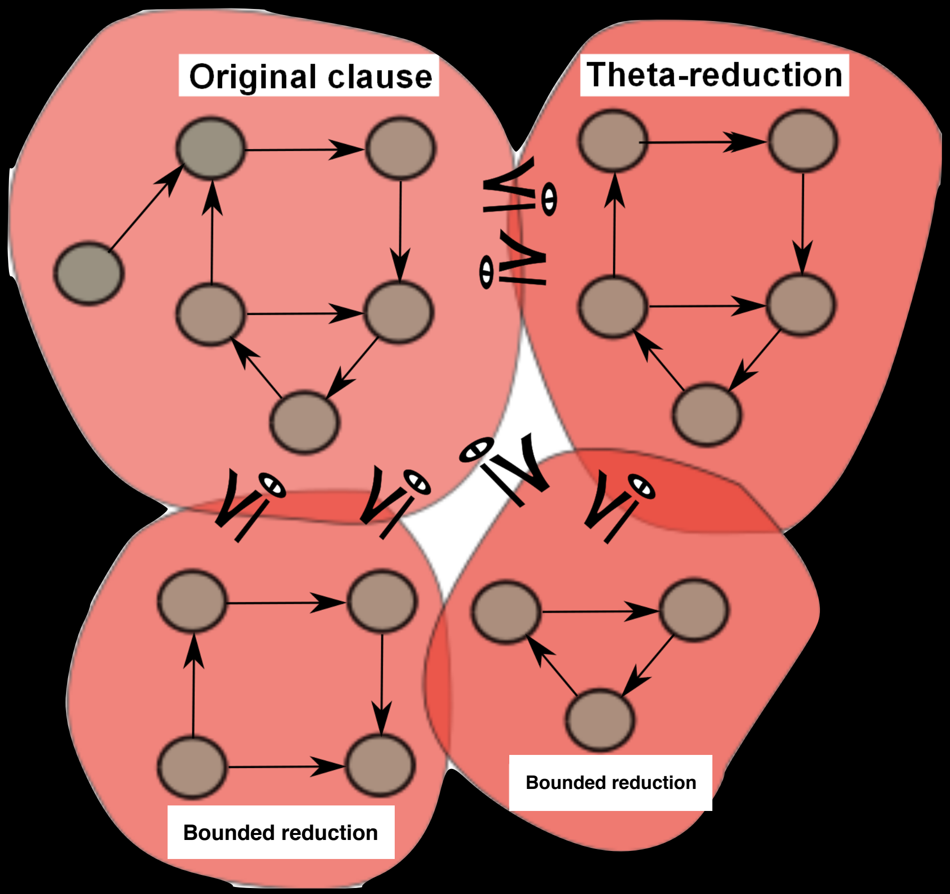
- Clause *C* is an LGG of clauses *A* and *B* if $C \leq A$, $C \leq B$ and, for any clause *D* such that $D \leq A$, $D \leq B$, it holds $D \leq C$.
- LGG is used for learning (new hypotheses are created as LGGs of examples).
- θ-reduction is used for reducing LGGs (θ-reduction of an LGG is still an LGG).
- Corresponds to tensor products of graphs.



# Bounded LGG

- Let X be a set of clauses. A clause *B* is said to be a bounded least general generalization w.r.t. the set *X* of clauses $A_1, A_2, \ldots, A_n$ (denoted by B = $LGG_X(A_1,A_2,\ldots,A_n)$) if and only if B ≼ $A_i$ for all $i \in \{1,2,\ldots,n\}$ and if for every other clause $C \in X$ such that $C \preccurlyeq A_i$ for all $i \in \{1,2,\ldots,n\}$, it holds $C \preccurlyeq B$.

- It is a generalization/relaxation of conventional LGG
- Introduced in order to alleviate computational difficulties related to intractability of θ-subsumtpion and θ-reduction
- It uses polynomial-time so-called bounded reduction instead of θ-reduction

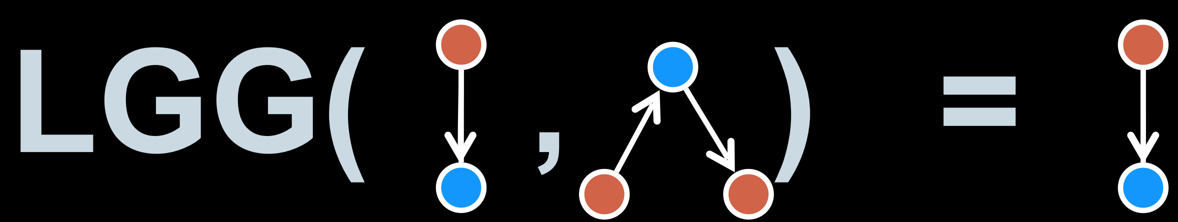


# A Bit Inconvenient Property of Bounded LGG

- **There are cases when:**
  - The set *X* has reasonable properties (e.g. *X* may consist of bounded-size or bounded-treewidth clauses)
  - *A* and *B* are clauses such that none of their bounded LGGs belongs to the set *X*.
  - (This does not affect any of the provable desirable properties of bounded LGGs.)

# On the other hand… LGGs of Forests

- If *X* is the set of directed forests, [Horváth, AIJ 2001] notes that if *A* and *B* are from *X* then LGG(A,B) ∊ X as well.



# The Conjecture

# LGG in a Set *X*

- A stronger variant of bounded LGG

**Only defined for clauses from X!**

- LGG in a set X, of clauses A and B from the set X is a clause from the set $LGG_X(A,B) \cap X$.

**It may also not exist.**

**Recall that $LGG_X(A,B)$ is a set.**

(like bounded LGG, it does not have to be least general, but only in the set X)

# The Conjecture

- **LGG *in a set X* always exists if *X* is the set of clauses of tree width at most *k*.**

*The conjecture holds for forests by Horvath's result.*

*If true, it would imply mildly positive complexity results for learning from bounded-treewidth clauses.*

# Results

# What would not work…

- In order to prove that *LGG in a set X* does **not** exist, it is not enough to show that (θ-reduction of) LGG of some clauses from *X* is not from *X.*

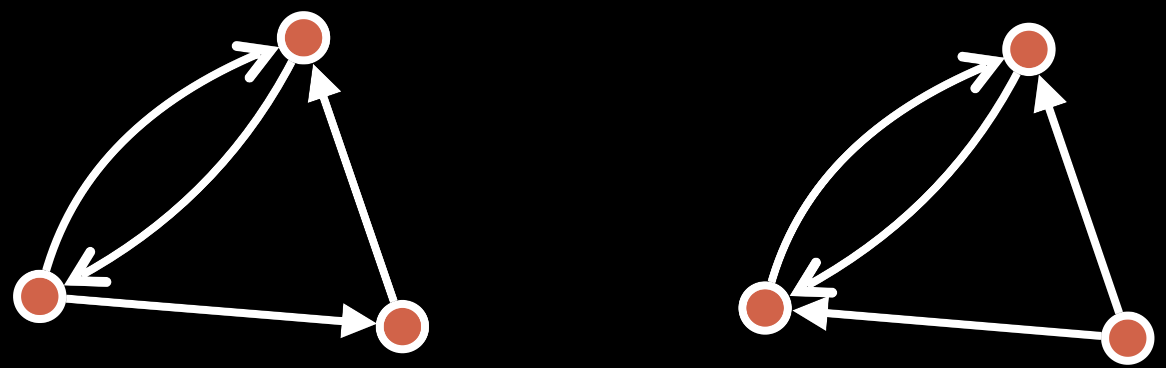
- **Example:**

*X* = clauses with at most 3 literals
*A* = e(X, Y ) ∨ e(Y, X)
*B* = e(X, Y ) ∨ e(Y, Z) ∨ e(Z, X)
*LGG*(*A*, *B*) = e(X1, X2) ∨ e(X2, X3) ∨ e(X3, X4) ∨ e(X4, X5) ∨ e(X5, X6) ∨ e(X6, X1), thus LGG(*A*, *B*) ∩ X = ∅.
However, $\mathrm{LGG}^{in}$ (*A*, *B*) = e(W,X)∨e(X,Y)∨e(Y,Z).

# Example

## A simpler illustrating result:

If n ≥ 4 then there is no LGG operator in the set *X* of clauses with at most n atoms based on one binary predicate.

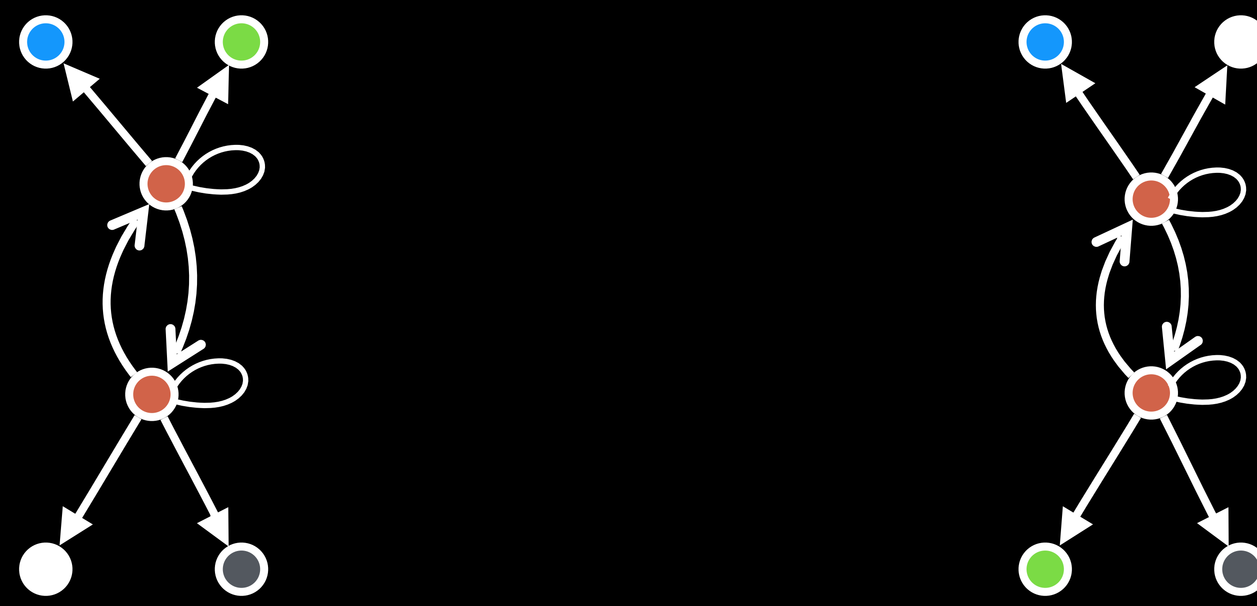


By enumerating all graphs with at most 4 edges, we can show that these two graphs have no LGG in *X*.

# The Negative Result

**Theorem:** There is no LGG operator in the set of clauses with treewidth 1.

*Graphs used in the proof:*



The problem is more difficult than on the previous slide because the set X is infinite in this case (so enumeration would not help).

We can show that these two graphs have no LGG in the set of tree width 1 clauses.

Note: This does not contradict Horvath as our proof requires loops (which are forbidden in forests).

# Conclusions

- We have provided a negative answer to a natural question *that someone would probably sooner or later have to ask*.
- **Open questions:**
  - Are there interesting sets of clauses with *LGG in set*?
  - Are there classes of clauses with bounded LGGs with slowly growing sizes/treewidths?