Hereditary FO and Extensional ESO

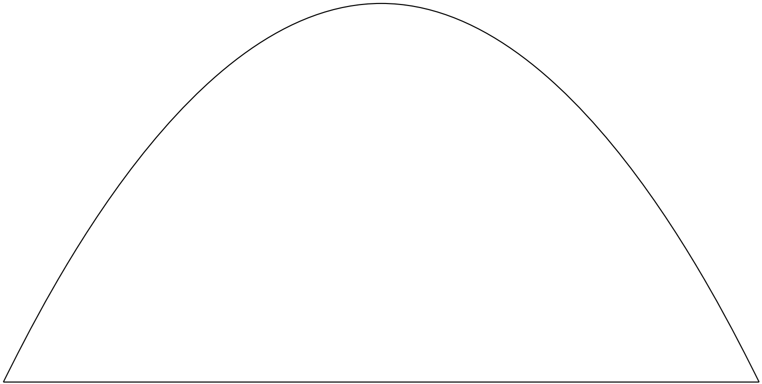
Santiago Guzmán-Pro
joint work with Manuel Bodirsky

Institute of Algebra
TU Dresden

Midsummer Combinatorial Workshop XXX

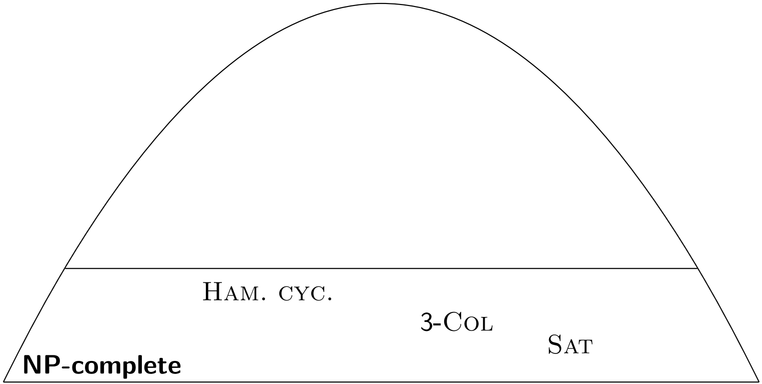
ERC Synergy Grant POCOCOP (GA 101071674)

Decision problems



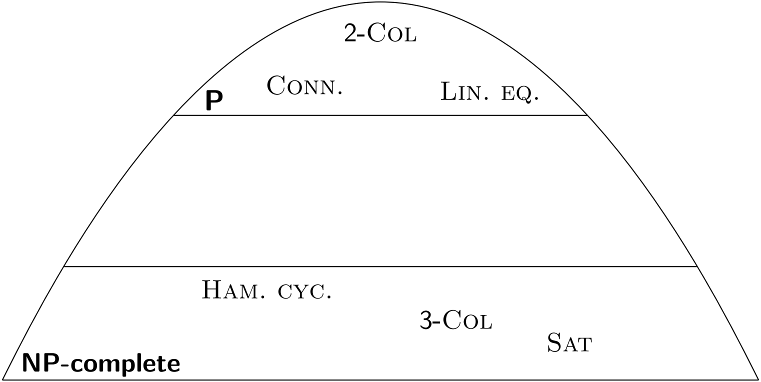
NP = ESO
(Fagin 1973)

Decision problems



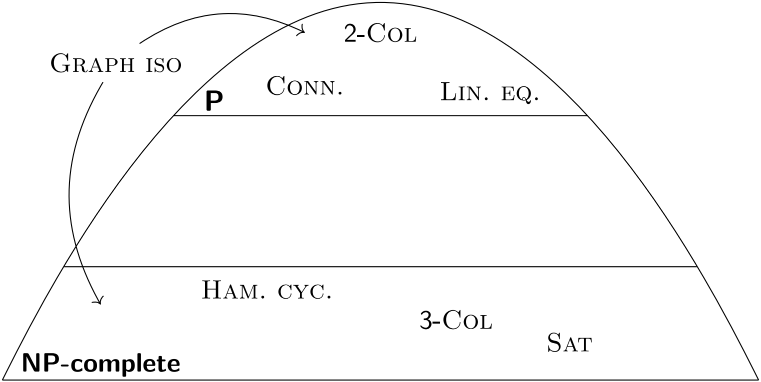
NP = **ESO**
(Fagin 1973)

Decision problems



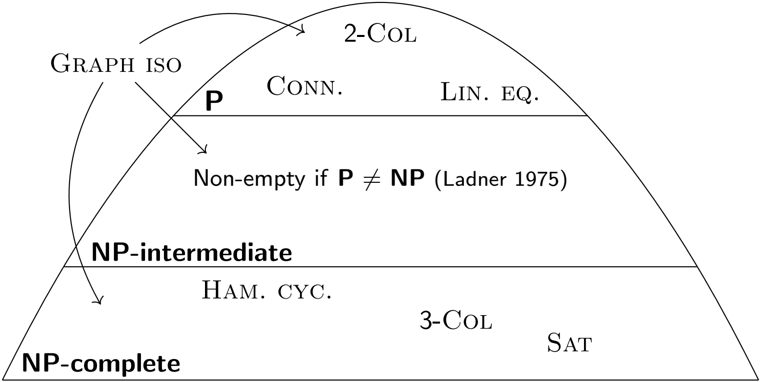
NP = ESO
(Fagin 1973)

Decision problems



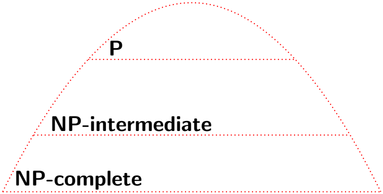
NP $=$ **ESO**
(Fagin 1973)

Decision problems



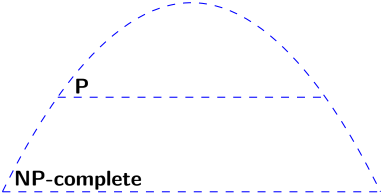
NP = ESO
(Fagin 1973)

Decision problems



Subclass of **NP**

Full computational power of **NP**
(up to P-time eq.)

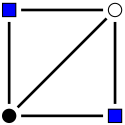


Subclass of **NP**

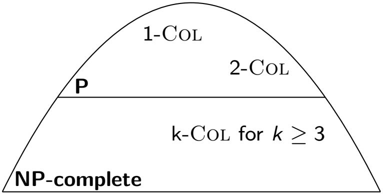
P vs. **NP-complete** dichotomy

Dichotomies

k-Col: On input graph G decide if there is a k-vertex colouring without monochromatic edges



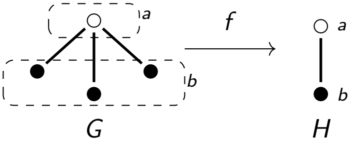
G

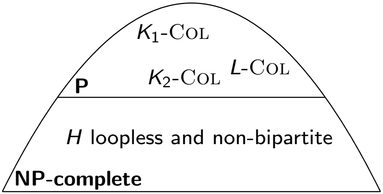


P vs. **NP-complete** dichotomy

Dichotomies

H-Col: On input graph G decide if there is a homomorphism $f: G \to H$

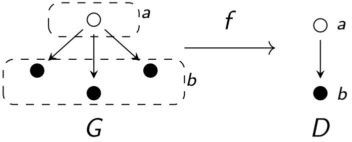


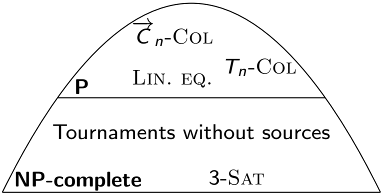


P vs. **NP-complete** dichotomy
(Hell-Nešetřil Theorem)

Dichotomies

D-COL: On input digraph (structure) G decide if there is a homomorphism $f\colon G \to D$

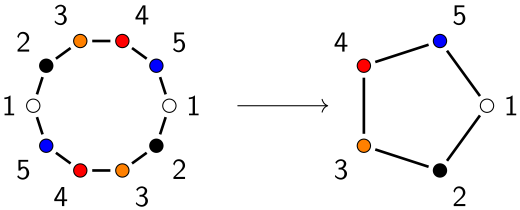




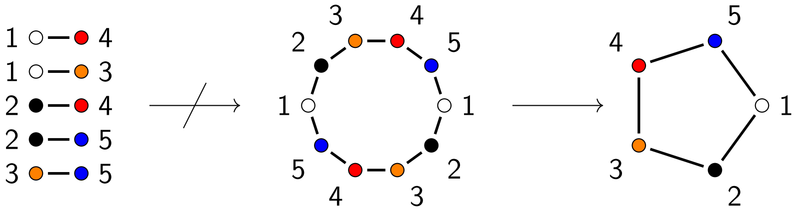
P vs. **NP-complete** dichotomy

Bulatov, Zhuk (2017)

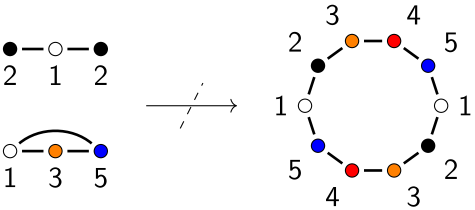
NP-richness



NP-richness

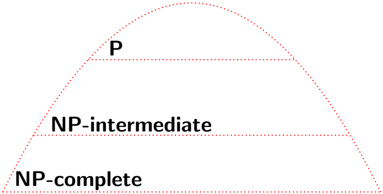


NP-richness



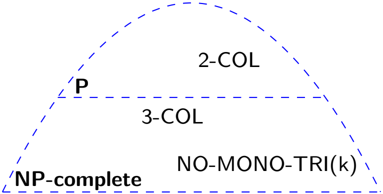
NP-richness

Forbidden vertex-coloured pattern problems



Forbid induced subgraphs

Full computational power of **NP**
Feder-Vardi (1999)
Kun-Nešetřil (2025)

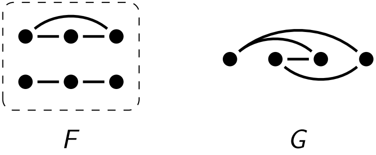


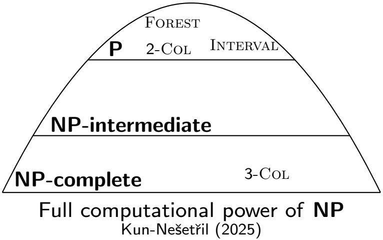
Forbid homomorphically

P vs. **NP-complete** dichotomy
Feder-Vardi (1999) + Bulatuv, Zhuk (2017)
Bodirsky-Madelain-Mottet (2021)

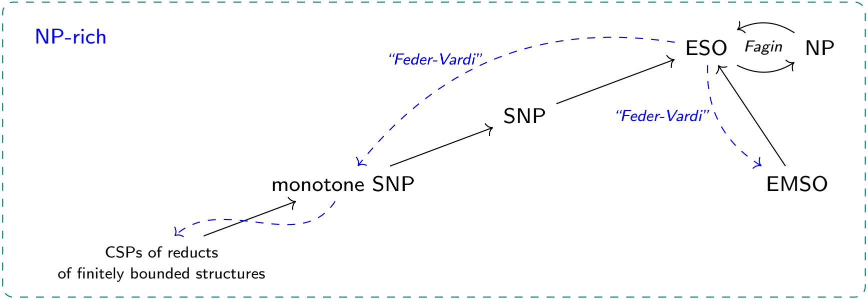
NP-richness

ORD(F): On input graph G decide if there is a linear ordering $<$ such that $(G, <)$ is F-free

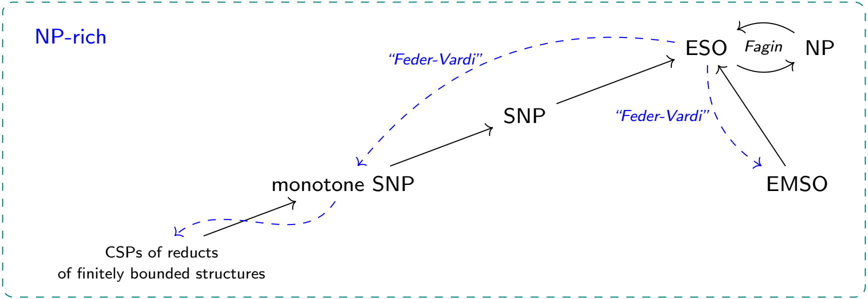




NP-richness


NP-rich
ESO
Fagin
NP
"Feder-Vardi"
SNP
"Feder-Vardi"
monotone SNP
EMSO
CSPs of reducts
of finitely bounded structures

NP-richness



What makes a fragment $\mathcal{L}$ of ESO have the full-computational power of NP?

Is the tractability problem for $\mathcal{L}$ decidable?

Is there a "natural" fragment of ESO that is not NP-rich but has no dichotomy?

Graph modification problems

Graph modification problems

Edge-modification

For a fixed graph class $\mathcal{C}$ and an input graph G determine the minimum number of edge modifications to G so that it belongs to $\mathcal{C}$.

- Edge-deletion problems (Yanakkakis 1981).
- Hardness of Edge-Modification problems (Alon, Stav 2009).
- Dichotomy Results on the Hardness of H-free Edge Modification Problems (Aravind, Sandeep, Sivadasan 2017).
- Hardness of approximation for H-free edge modification problems (Bliznets, Cygan, Komosa, Pilipczuk 2018)

Graph modification problems

Resilience problems

For a fixed query μ, determine the resilience of μ in an input database $\mathbb{D}$.

- The Complexity of Resilience and Responsibility for Self-Join-Free Conjunctive Queries (Freire, Gatterbauer, Immerman, Meliou 2015)
- New Results for the Complexity of Resilience for Binary with Self-Joins (Freire, Gatterbauer, Immerman, Meliou 2020).
- A Unified Approach for Resilience and Causal Responsibility with Integer Linear Programming (ILP) and LP Relaxations. (Makhija, Gatterbauer 2023).
- The Complexity of Resilience Problems via Valued Constraint Satisfaction (Bodirsky, Semanišinová, Lutz 2024).

Graph modification problems

Vertex-deletion

For a fixed graph class $\mathcal{C}$ and an input graph G determine the minimum k so that $G - U$ belongs to $\mathcal{C}$ for some $|U| \leq k$.

- Node-Deletion NP-Complete Problems (Krishnamoorthy, Deo 1979)
- The node-deletion problem for hereditary properties is NP-complete (Lewis, Yanakkakis 1980).
- Finding odd-cycle transversals (Reed, Smith, Vetta 2004)
- On the Descriptive Complexity of Vertex Deletion Problems (Bannach, Chudigiewitsch, Tantau 2024)

Graph modification problems

Modification to first-order logic

Edge-modification: Given a graph G and a positive integer k test whether it is possible to modify at most k edges so that it satisfies ϕ.

Edge-completion: Given a graph G and a positive integer k test whether it is possible to add at most k edges from G so that it satisfies ϕ.

Edge-deletion: Given a graph G and a positive integer k test whether it is possible to remove at most k edges from G so that it satisfies ϕ.

Vertex-deletion: Given a graph G and a positive integer k test whether it is possible to remove at most k vertices from G so that it satisfies ϕ.

On the parameterized complexity of graph modification to first-order logic properties (Fomin, Golovach, Thilikos 2020)

Graph modification problems

Modification to first-order logic (without parameter k)

Edge-modification: Given a graph G test whether we can modify the edges of G so that it satisfies ϕ.

Edge-completion: Given a graph G test whether we can add edges to G so that it satisfies ϕ.

Edge-deletion: Given a graph G test whether we can remove edges from G so that it satisfies ϕ.

Vertex-deletion: Given a graph G test whether we can remove vertices of G so that it satisfies ϕ.

Graph modification problems

Modification to first-order logic (without parameter k)

Edge-modification: Given a graph G test whether we can modify the edges of G so that it satisfies ϕ.

Edge-completion: Given a graph G test whether we can add edges to G so that it satisfies ϕ.

Edge-deletion: Given a graph G test whether we can remove edges from G so that it satisfies ϕ.

Vertex-deletion: Given a graph G test whether we can remove vertices of G so that it satisfies ϕ.

Graph modification problems

Modification to first-order logic (without parameter k)

Edge-modification: Given a graph G test whether we can modify the edges of G so that it satisfies ϕ.

Edge-completion: Given a graph G test whether we can add edges to G so that it satisfies ϕ.

Edge-deletion: Given a graph G test whether we can remove edges from G so that it satisfies ϕ.

Vertex-deletion: Given a graph G test whether we can remove vertices of G so that it satisfies ϕ.

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{B}$ of $\mathbb{A}$ satisfies ϕ.

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{B}$ of $\mathbb{A}$ satisfies ϕ.

$\textsc{Her}(\phi)$: On input $\mathbb{A}$ decide if every substructure $\mathbb{B}$ of $\mathbb{A}$ satisfies ϕ

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{B}$ of $\mathbb{A}$ satisfies ϕ.

$\textsc{Her}(\phi)$: On input $\mathbb{A}$ decide if every substructure $\mathbb{B}$ of $\mathbb{A}$ satisfies ϕ

Ex. 1 $\phi :=$ exists a vertex of degree at most 1

Ex. 2 $\phi :=$ exists a simplicial vertex

Ex. 3 $\phi :=$ exists a source.

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{A}'$ of $\mathbb{A}$ satisfies ϕ.

$\textsc{Her}(\phi)$: On input $\mathbb{A}$ decide if every substructure $\mathbb{B}$ of $\mathbb{A}$ satisfies ϕ

Ex. 1 $\phi :=$ exists a vertex of degree at most 1

Ex. 2 $\phi :=$ exists a simplicial vertex

Ex. 3 Acyclic digraphs: $\phi :=$ exists a source

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{A}'$ of $\mathbb{A}$ satisfies ϕ.

Ex. 1 $\phi :=$ exists a vertex of degree 1

Ex. 2 $\phi :=$ exists a simplicial vertex

Ex. 3 $\phi :=$ exists a source

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{A}'$ of $\mathbb{A}$ satisfies ϕ.

Ex. 1 Forests: $\phi :=$ exists a vertex of degree 1.

Ex. 2 Chordal graphs: $\phi :=$ exists a simplicial vertex (Rose, 1970).

Ex. 3 Acyclic digraphs: $\phi :=$ exists a source.

Qst. Is every problem in HerFO solvable in polynomial-time?

Hereditary first-order logic

A structure $\mathbb{A}$ *hereditarily satisfies* ϕ if every substructure $\mathbb{A}'$ of $\mathbb{A}$ satisfies ϕ.

Ex. 1 Forests: $\phi :=$ exists a vertex of degree 1.

Ex. 2 Chordal graphs: $\phi :=$ exists a simplicial vertex (Rose, 1970).

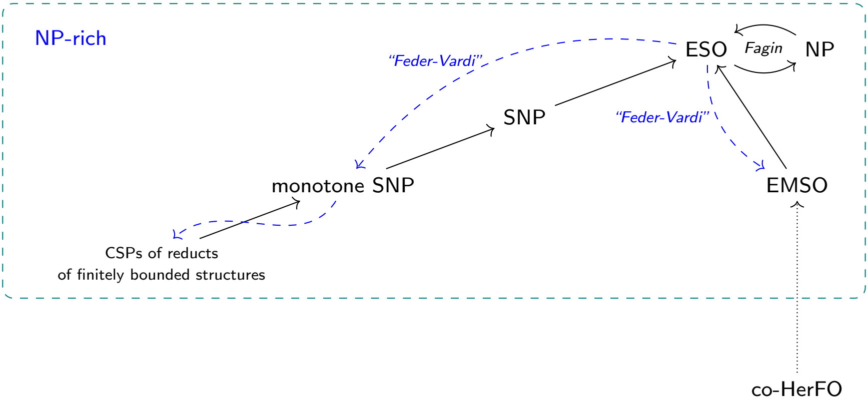
Ex. 3 Acyclic digraphs: $\phi :=$ exists a source.

Qst. Is every problem in HerFO solvable in polynomial-time?

Thm. The tractability problem for HerFO is undecidable.

Hereditary first-order logic

"Qst." What is the computational power of HerFO?



Extensional ESO

Extensional ESO

F-free edge-completion: Given a graph G test whether it is possible to add edges to G so that it becomes F-free.

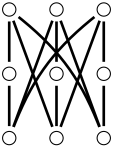
Ex 1. Acyclic digraphs: extend the edge relation to a (strict) linear order.

F

Extensional ESO

Edge-completion to ϕ: Given a graph G test whether it is possible to add edges so that it satisfies ϕ.

Ex 2. (Pach, 1971) A graph G has circular chromatic number < 3 iff G can be extended to a maximal triangle-free graphs that avoids:



Petersen minus vertex

Extensional ESO

Extensional ESO is the fragment of ESO

$$\exists R_1 \ldots, R_k . \left[\bigwedge_{i \in [n]} \forall \bar{x} R_i'(x) \implies R(\bar{x}) \right] \land \phi$$

Extensional ESO

Extensional ESO is the fragment of ESO

$$\exists E.\ [\forall x, y\ E'(x,y) \implies E(x,y)] \ \wedge\ (V, E) \text{ is } F\text{-free}$$

Ex 1 F-free edge completion problems (e.g., acyclic digraphs)

Extensional ESO

Extensional ESO is the fragment of ESO

$$\exists E.\ \left[\forall x, y E'(x,y) \implies E(x,y)\right]\ \wedge\ \phi$$

Ex 1 *F*-free edge completion problems (e.g., acyclic digraphs)

Ex 2. Edge-completion to ϕ

Extensional ESO

Extensional ESO is the fragment of ESO

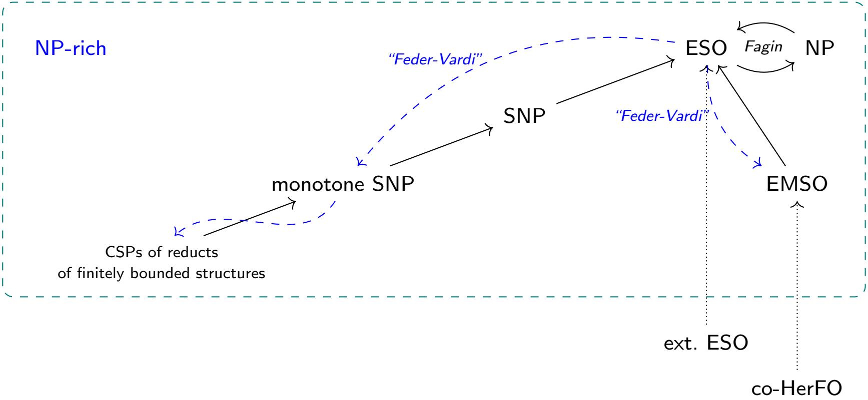
$$\exists R, B, G.\ \left[\bigwedge_{C \in \{R,B,G\}} \forall x.\ C'(x) \implies C(x) \right] \wedge (R, B, G) \text{ is a proper 3-colouring}$$

Ex 1 *F*-free edge completion problems (e.g., acyclic digraphs)

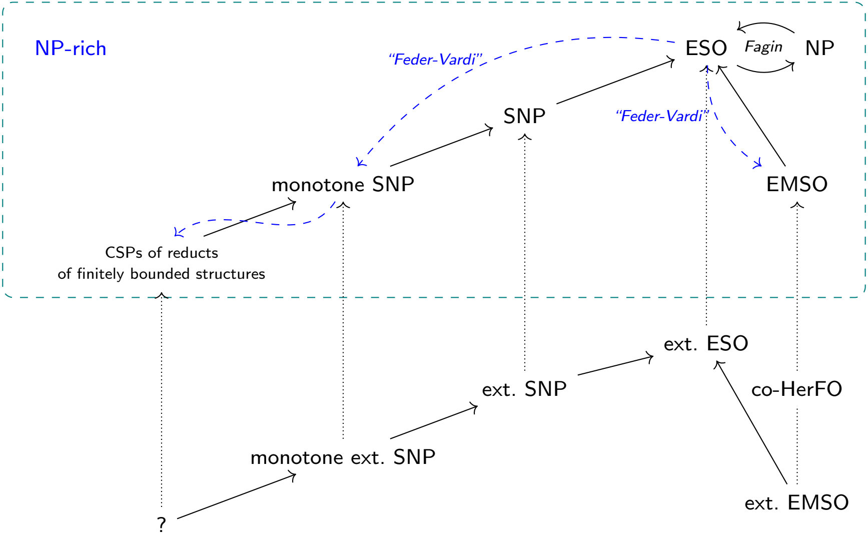
Ex 2. Edge-completion to ϕ

Ex 4. Pre-coloured 3-$\textsc{Col}$ (more generally, pre-coloured H-$\textsc{Col}$)

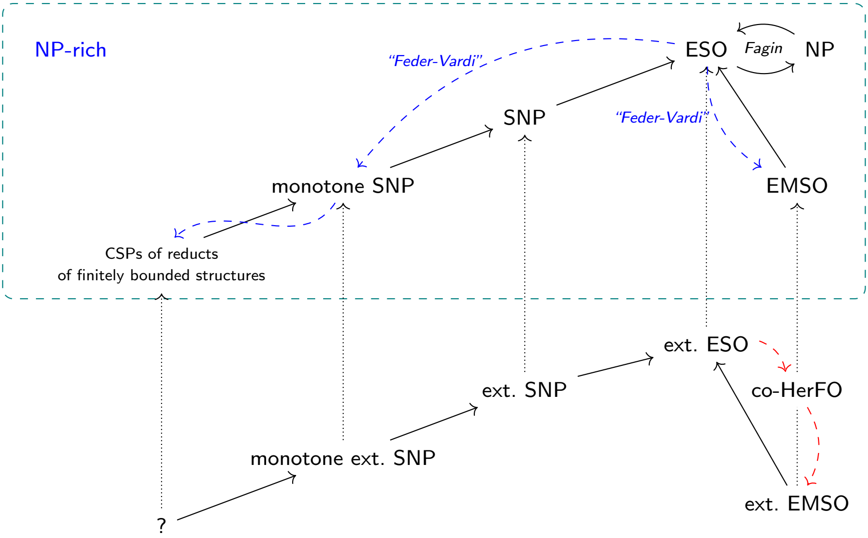
Extensional ESO



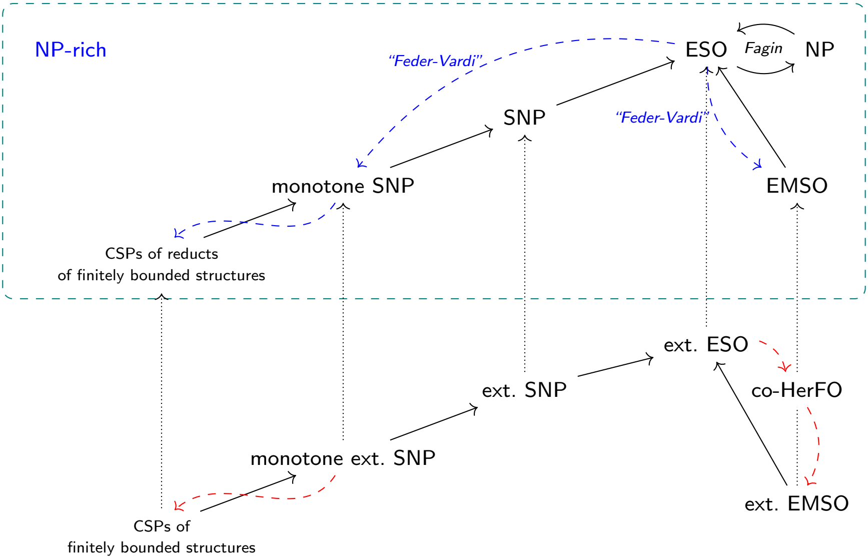
Extensional ESO



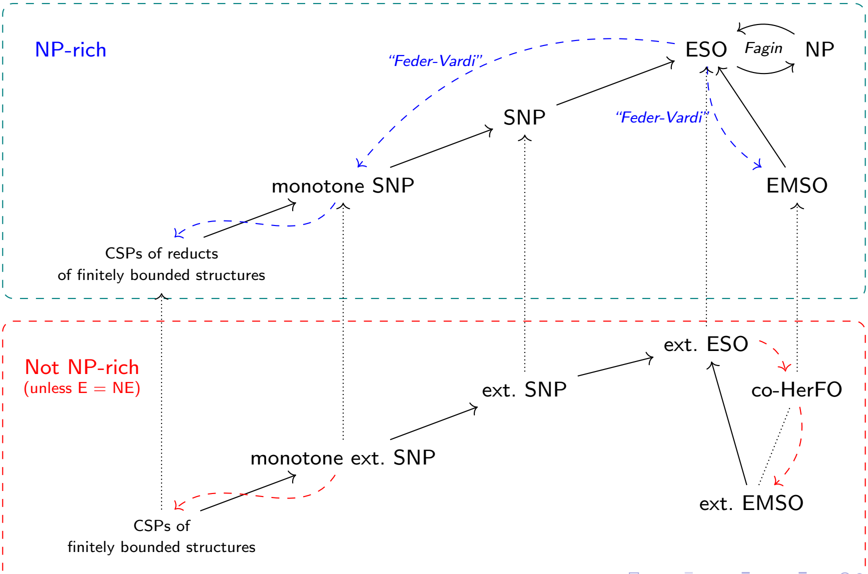
Extensional ESO



Extensional ESO



Extensional ESO



Thank you for your attention!