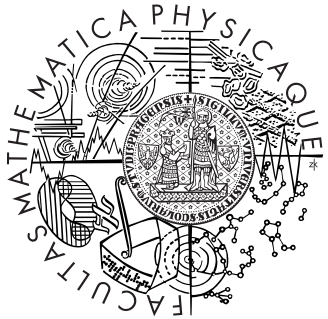
Extending Partial Representations of Subclasses of Chordal Graphs

Pavel Klavík

joint work with J. Kratochvíl, Y. Otachi and T. Saitoh



Department of Applied Mathematics,
Faculty of Mathematics and Physics,
Charles University in Prague

Japan Advanced Institute of Science and Technology,
Ishikawa, Japan

Graduate School of Engineering, Kobe University,
Kobe, Japan

ISAAC 2012, Taipei

We study a generalization of the RECOGNITION problem called the PARTIAL REPRESENTATION EXTENSION problem.

For example for planar graphs:

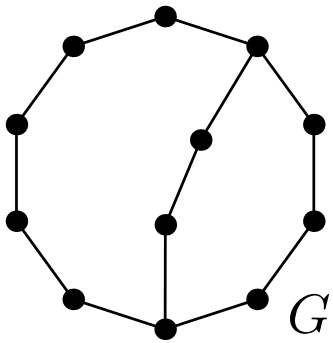
RECOG

REPEXT

We study this problem for chordal graphs (subtree-in-tree graphs).

We study a generalization of the RECOGNITION problem called the PARTIAL REPRESENTATION EXTENSION problem.

For example for planar graphs:



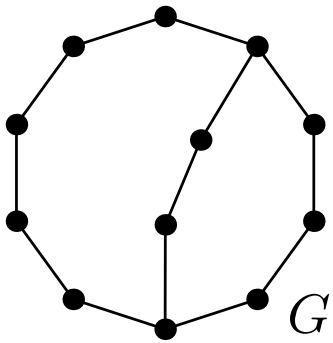
RECOG

REPEXT

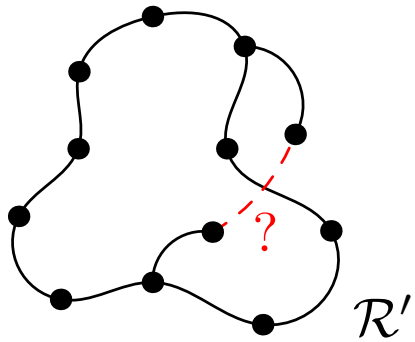
We study this problem for chordal graphs (subtree-in-tree graphs).

We study a generalization of the RECOGNITION problem called the PARTIAL REPRESENTATION EXTENSION problem.

For example for planar graphs:



RECOG

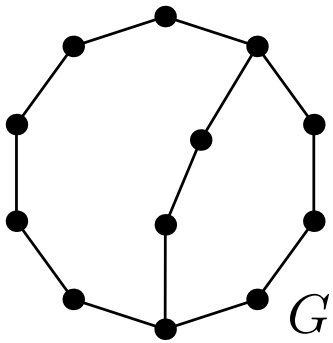


REPEXT

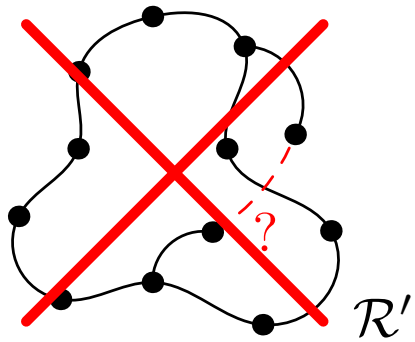
We study this problem for chordal graphs (subtree-in-tree graphs).

We study a generalization of the RECOGNITION problem called the PARTIAL REPRESENTATION EXTENSION problem.

For example for planar graphs:



RECOG

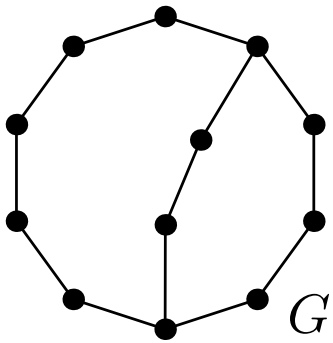


REPEXT

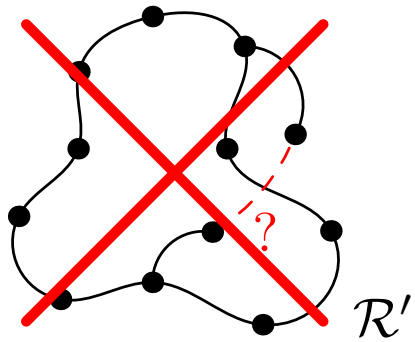
We study this problem for chordal graphs (subtree-in-tree graphs).

We study a generalization of the RECOGNITION problem called the PARTIAL REPRESENTATION EXTENSION problem.

For example for planar graphs:



RECOG



REPEXT

We study this problem for chordal graphs (subtree-in-tree graphs).

We study intersection representations:

$$\mathcal{R} = \{R_v : v \in V\} \quad \text{such that} \quad uv \in E \iff R_u \cap R_v \neq \emptyset.$$

When we restrict the sets R_v, we get some nice classes:

INT: $K_{1,3}$

PINT/UINT: K_4^-

FUN: C_4

CIRCLE: C_5

We study intersection representations:

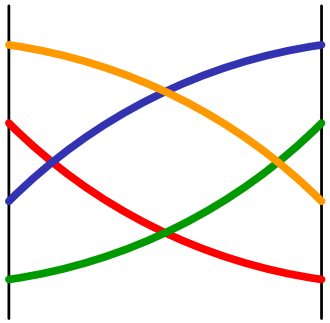
$$\mathcal{R} = \{R_v : v \in V\} \quad \text{such that} \quad uv \in E \iff R_u \cap R_v \neq \emptyset.$$

When we restrict the sets R_v, we get some nice classes:

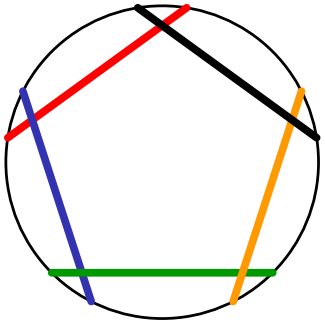
INT: $K_{1,3}$

PINT/UINT: K_4^-

FUN: C_4



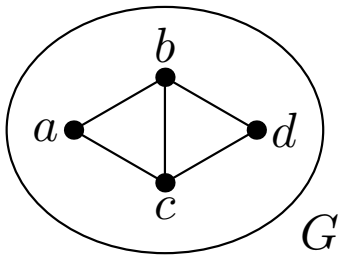
CIRCLE: C_5

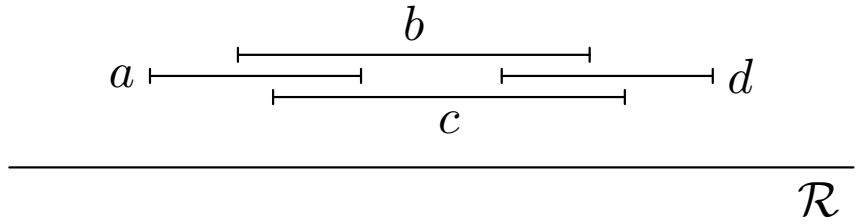


- A partial representation $\mathcal{R}'$ is a representation of an induced subgraph G'.
- A representation $\mathcal{R}$ extends $\mathcal{R}'$ if it assigns the same sets to G' as $\mathcal{R}'$.

RECOG:

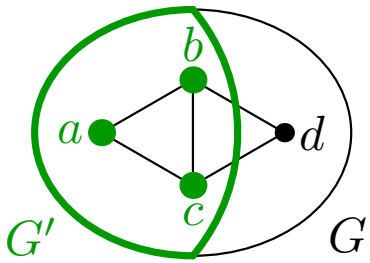
$G \xrightarrow{?} \mathcal{R}$

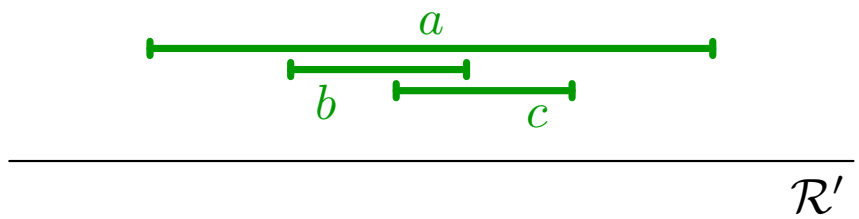




REPEXT:

$G + \mathcal{R}' \xrightarrow{?} \mathcal{R}$

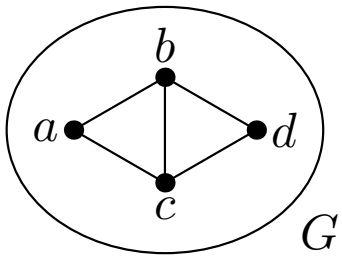


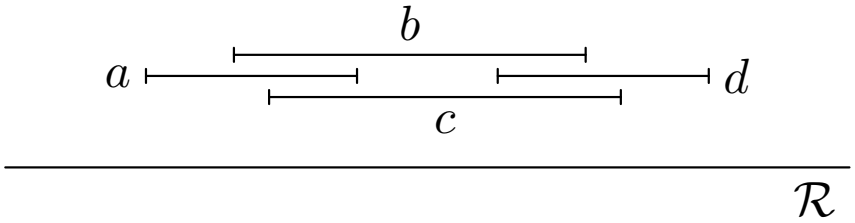


- A partial representation $\mathcal{R}'$ is a representation of an induced subgraph G'.
- A representation $\mathcal{R}$ extends $\mathcal{R}'$ if it assigns the same sets to G' as $\mathcal{R}'$.

RECOG:

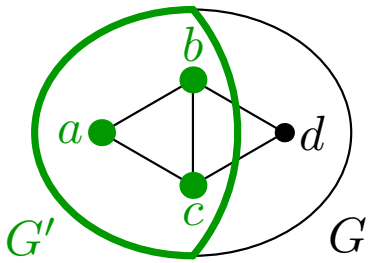
$G \xrightarrow{?} \mathcal{R}$

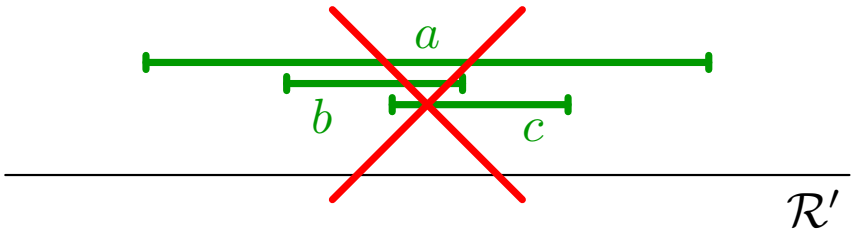




REPEXT:

$G + \mathcal{R}' \xrightarrow{?} \mathcal{R}$





Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011), Bläsius, Rutter (SODA 2013), K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh, Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011), Bläsius, Rutter (SODA 2013), K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh, Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011),
  Bläsius, Rutter (SODA 2013),
  K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh,
  Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011),
  Bläsius, Rutter (SODA 2013),
  K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh,
  Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011),
  Bläsius, Rutter (SODA 2013),
  K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh,
  Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011),
  Bläsius, Rutter (SODA 2013),
  K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh,
  Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Previous polynomial and NP-completeness results:

- Planar graphs: Angelini et al. (SODA 2010), Patrignani (2006).
- Interval graphs: K., Kratochvíl, Vyskočil (TAMC 2011),
  Bläsius, Rutter (SODA 2013),
  K., Kratochvíl, Otachi, Saitoh, Vyskočil (2013?).
- Fun/perm graphs: K., Kratochvíl, Krawczyk, Walczak (ESA 2012).
- Proper/unit int: K., Kratochvíl, Otachi, Rutter, Saitoh,
  Saumell, Vyskočil (2013?).

Question: Is almost everything solvable polynomially?
Not in the case of chordal graphs.

Motivation for the partial representation extension problem:

1. The problem is very natural and applicable.
2. Similar types of extension problems are frequently studied:
   - For example a very hard pre-coloring extension problem.
3. The problem forces better understanding of studied classes:
   - For interval graphs or function graphs, we can use known structure.
   - For unit interval graphs, we had to develop new structure.
4. It is related to other restricted representation problems:
   - For example to the problem of simultaneous representations.
   - Many of our techniques can be used elsewhere.
   - We need to understand structure of all representations.

Motivation for the partial representation extension problem:

1. The problem is very natural and applicable.
2. Similar types of extension problems are frequently studied:
   - For example a very hard pre-coloring extension problem.
3. The problem forces better understanding of studied classes:
   - For interval graphs or function graphs, we can use known structure.
   - For unit interval graphs, we had to develop new structure.
4. It is related to other restricted representation problems:
   - For example to the problem of simultaneous representations.
   - Many of our techniques can be used elsewhere.
   - We need to understand structure of all representations.

Motivation for the partial representation extension problem:

1. The problem is very natural and applicable.
2. Similar types of extension problems are frequently studied:
   - For example a very hard pre-coloring extension problem.
3. The problem forces better understanding of studied classes:
   - For interval graphs or function graphs, we can use known structure.
   - For unit interval graphs, we had to develop new structure.
4. It is related to other restricted representation problems:
   - For example to the problem of simultaneous representations.
   - Many of our techniques can be used elsewhere.
   - We need to understand structure of all representations.

Motivation for the partial representation extension problem:

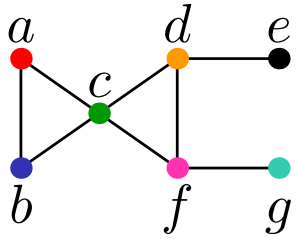
1. The problem is very natural and applicable.
2. Similar types of extension problems are frequently studied:
   - For example a very hard pre-coloring extension problem.
3. The problem forces better understanding of studied classes:
   - For interval graphs or function graphs, we can use known structure.
   - For unit interval graphs, we had to develop new structure.
4. It is related to other restricted representation problems:
   - For example to the problem of simultaneous representations.
   - Many of our techniques can be used elsewhere.
   - We need to understand structure of all representations.

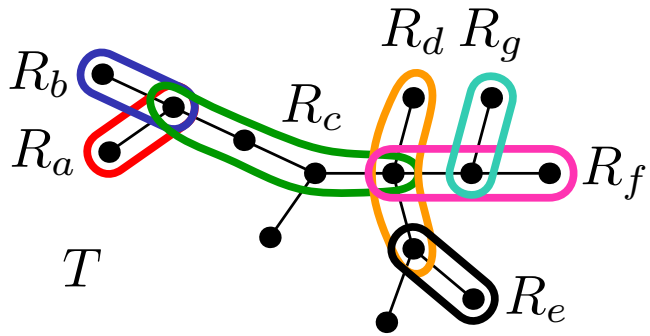
Motivation for the partial representation extension problem:

1. The problem is very natural and applicable.
2. Similar types of extension problems are frequently studied:
   - For example a very hard pre-coloring extension problem.
3. The problem forces better understanding of studied classes:
   - For interval graphs or function graphs, we can use known structure.
   - For unit interval graphs, we had to develop new structure.
4. It is related to other restricted representation problems:
   - For example to the problem of simultaneous representations.
   - Many of our techniques can be used elsewhere.
   - We need to understand structure of all representations.

Extending chordal graphs

A chordal graph is an intersection graphs of subtrees of a tree T.





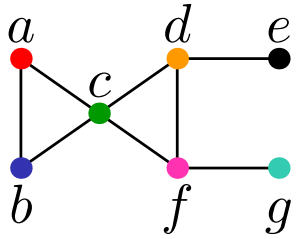
We also consider three subclasses of CHOR:

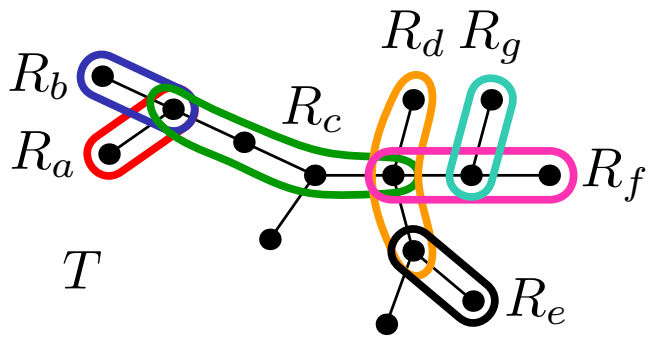
- PATH: Every subtree is a path.
- INT: The tree T is a path.
- PINT: No subpath is a proper subset of another subpath, i.e.,

$$R_u \subseteq R_v \implies R_u = R_v.$$

$$\text{PINT} \subsetneq \text{INT} \subsetneq \text{PATH} \subsetneq \text{CHOR}.$$

A chordal graph is an intersection graphs of subtrees of a tree T.

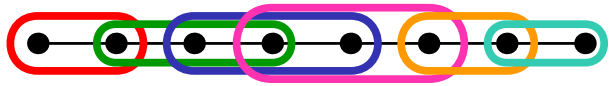




We also consider three subclasses of CHOR:

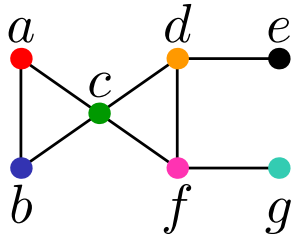
- PATH: Every subtree is a path.
- INT: The tree T is a path.
- PINT: No subpath is a proper subset of another subpath, i.e.,

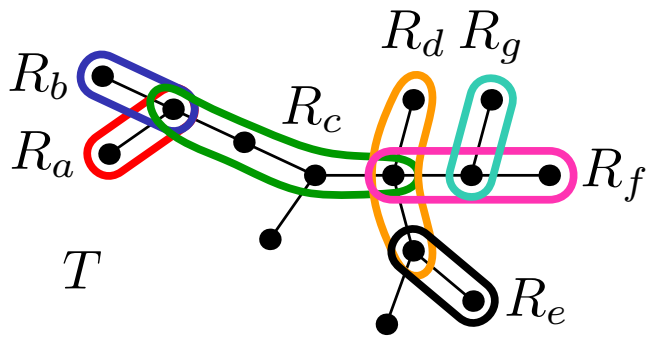
$$R_u \subseteq R_v \implies R_u = R_v.$$



$$\text{PINT} \subsetneq \text{INT} \subsetneq \text{PATH} \subsetneq \text{CHOR}.$$

A chordal graph is an intersection graphs of subtrees of a tree T.

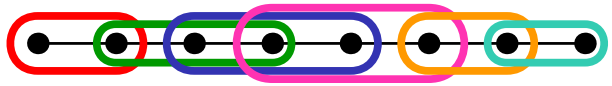




We also consider three subclasses of CHOR:

- PATH: Every subtree is a path.
- INT: The tree T is a path.
- PINT: No subpath is a proper subset of another subpath, i.e.,

$$R_u \subseteq R_v \implies R_u = R_v.$$



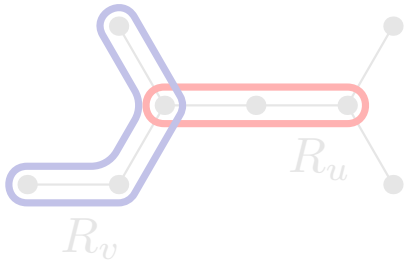
$$\text{PINT} \subsetneq \text{INT} \subsetneq \text{PATH} \subsetneq \text{CHOR}.$$

What is a partial representation of a chordal graph?

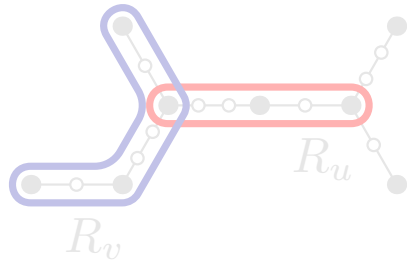
$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.

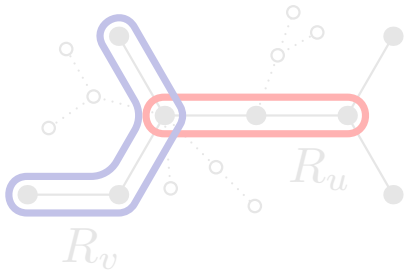
FIXED:



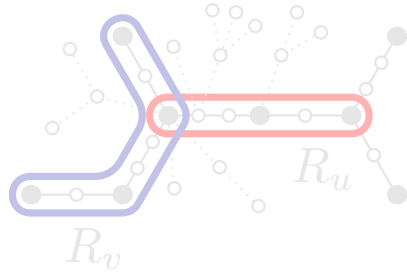
SUB:



ADD:



BOTH:

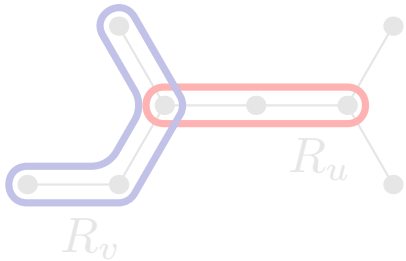


What is a partial representation of a chordal graph?

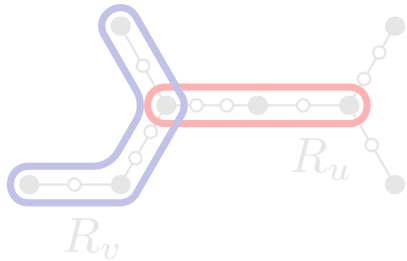
$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.

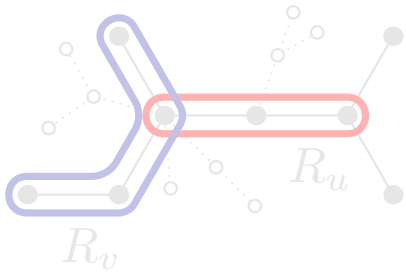
FIXED:



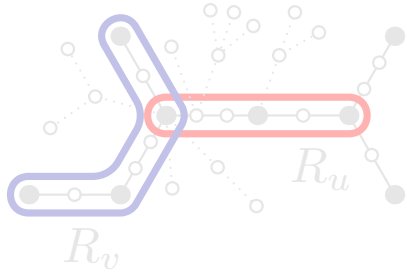
SUB:



ADD:



BOTH:

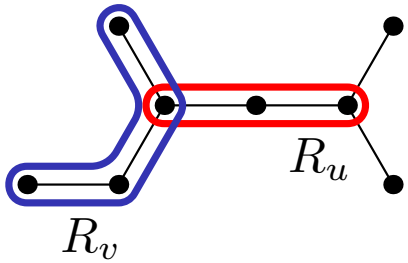


What is a partial representation of a chordal graph?

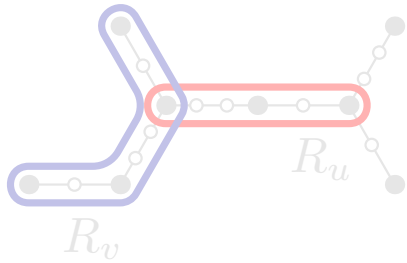
$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.

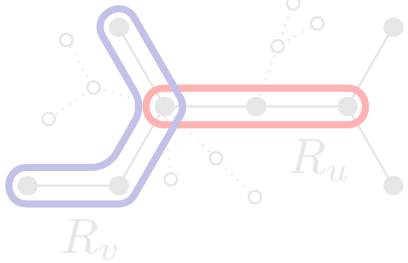
FIXED:



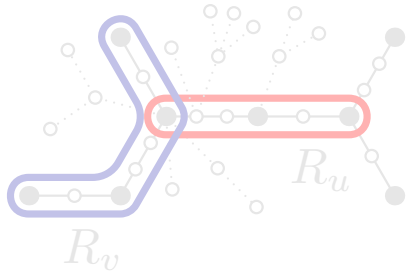
SUB:



ADD:



BOTH:

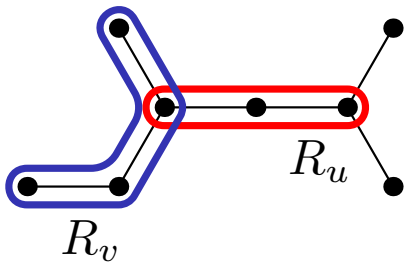


What is a partial representation of a chordal graph?

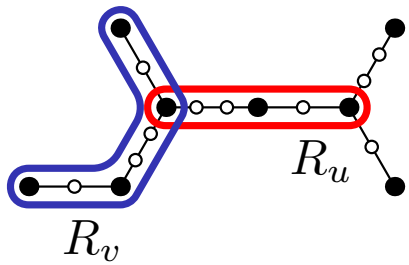
$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.

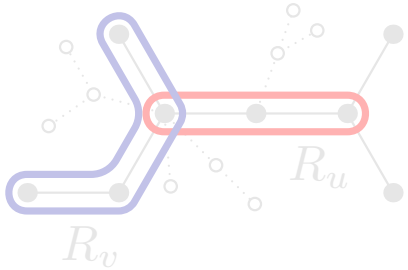
FIXED:



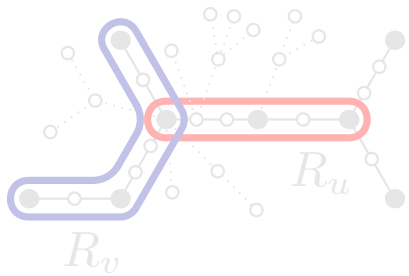
SUB:



ADD:



BOTH:

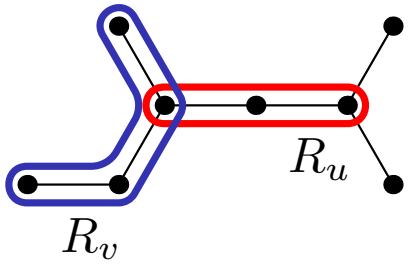


What is a partial representation of a chordal graph?

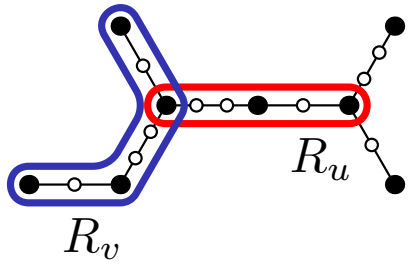
$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.

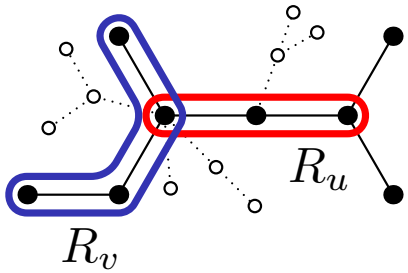
FIXED:



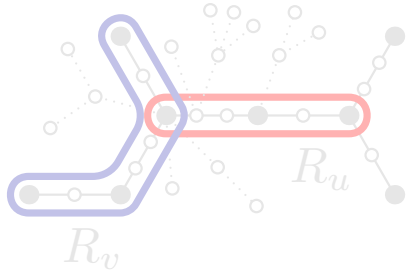
SUB:



ADD:



BOTH:

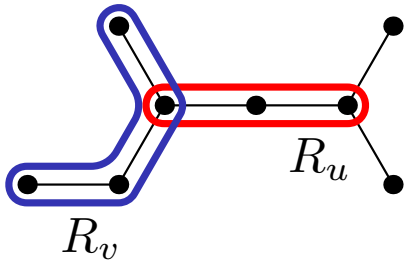


What is a partial representation of a chordal graph?

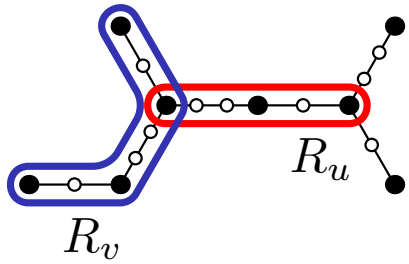
$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.

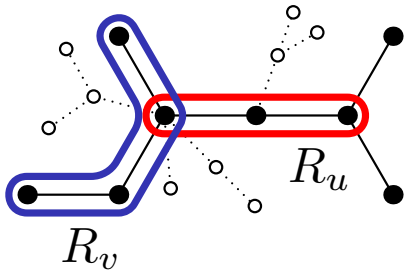
FIXED:



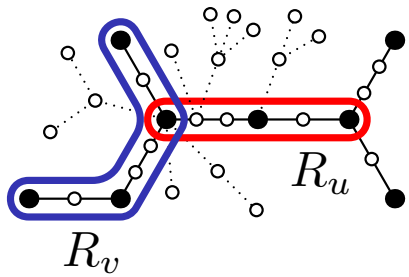
SUB:



ADD:



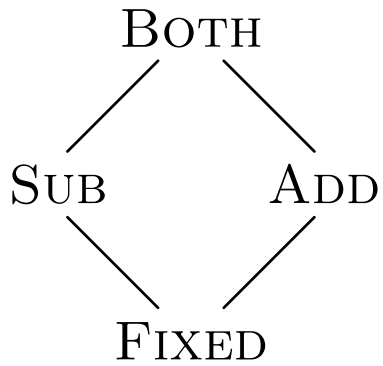
BOTH:



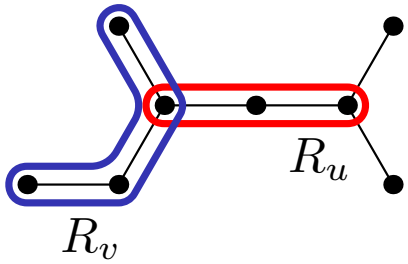
What is a partial representation of a chordal graph?

$\mathcal{R}'$ gives some tree T' which is modified in $\mathcal{R}$ to T:

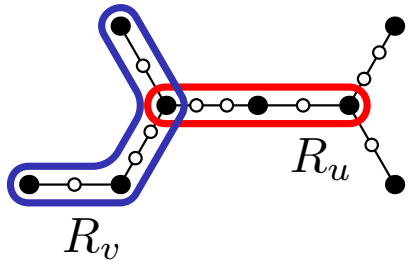
- FIXED: the tree T' can not be modified, $T = T'$.
- SUB: the tree T' can be subdivided.
- ADD: we can add branches to T'.
- BOTH: the combination of SUB and ADD.



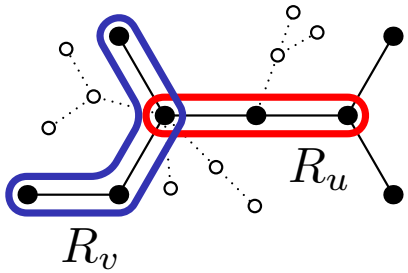
FIXED:



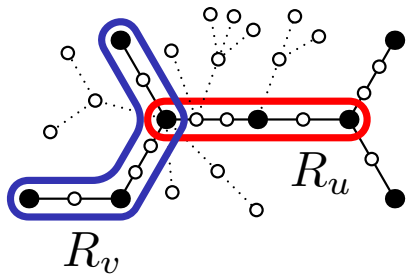
SUB:



ADD:



BOTH:



We consider the problems RepExt and Recog* (with $G' = \emptyset$).

		PINT	INT	PATH	CHOR
Fixed	Recog*	$\mathcal{O}(n+m)$	$\mathcal{O}(n+m)$	NP-complete	NP-complete
	RepExt	NP-complete	NP-complete	NP-complete	NP-complete
Sub	Recog*	$\mathcal{O}(n+m)$	$\mathcal{O}(n+m)$	NP-complete	NP-complete
	RepExt	$\mathcal{O}(n+m)$	$\mathcal{O}(n+m)$	NP-complete	NP-complete
Add	Recog*	$\mathcal{O}(n+m)$	$\mathcal{O}(n+m)$	$\mathcal{O}(nm)$	$\mathcal{O}(n+m)$
	RepExt	$\mathcal{O}(n+m)$	NP-complete	NP-complete	NP-complete
Both	Recog*	$\mathcal{O}(n+m)$	$\mathcal{O}(n+m)$	$\mathcal{O}(nm)$	$\mathcal{O}(n+m)$
	RepExt	$\mathcal{O}(n+m)$	$\mathcal{O}(n+m)$	**open**	NP-complete

standard recognition algorithms

previous RepExt results

NP-completeness results

All reductions are from the 3-PARTITION problem:

Problem: 3-PARTITION
Input: Integers k, M and $A_1, \ldots, A_{3k}$ with $\frac{M}{4} < A_i < \frac{M}{2}$.
Output: Is it possible to partition $A_1, \ldots, A_{3k}$ into k triples such that each triple sums to M?

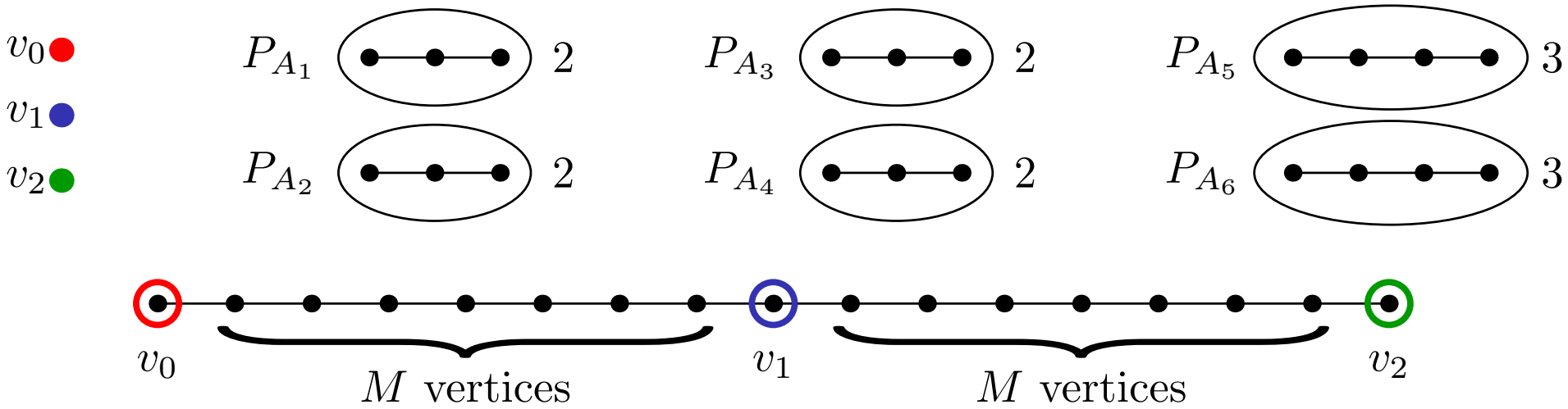
The reduction for REPEXT(INT, FIXED):
Input of 3-PARTITION: $k = 2$, $M = 7$ and A_i's are 2, 2, 2, 2, 3 and 3.

All reductions are from the 3-PARTITION problem:

Problem: 3-PARTITION
Input: Integers k, M and $A_1, \ldots, A_{3k}$ with $\frac{M}{4} < A_i < \frac{M}{2}$.
Output: Is it possible to partition $A_1, \ldots, A_{3k}$ into k triples such that each triple sums to M?

The reduction for REPEXT(INT, FIXED):

Input of 3-PARTITION: $k = 2$, $M = 7$ and A_i's are 2, 2, 2, 2, 3 and 3.

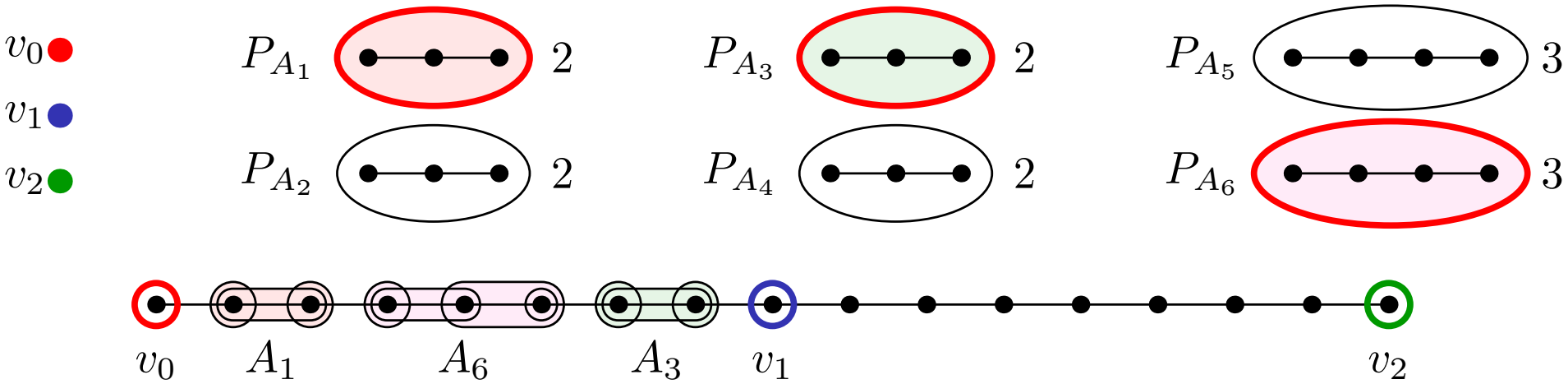


All reductions are from the 3-PARTITION problem:

Problem: 3-PARTITION
Input: Integers k, M and $A_1, \ldots, A_{3k}$ with $\frac{M}{4} < A_i < \frac{M}{2}$.
Output: Is it possible to partition $A_1, \ldots, A_{3k}$ into k triples such that each triple sums to M?

The reduction for REPEXT(INT, FIXED):

Input of 3-PARTITION: $k = 2$, $M = 7$ and A_i's are 2, 2, 2, 2, 3 and 3.

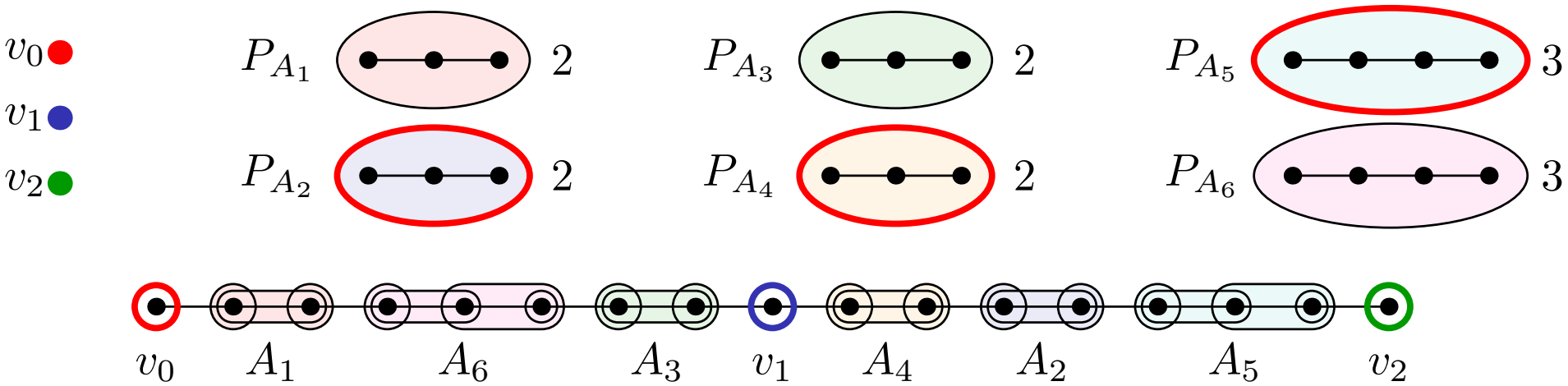


All reductions are from the 3-PARTITION problem:

Problem: 3-PARTITION
Input: Integers k, M and $A_1, \ldots, A_{3k}$ with $\frac{M}{4} < A_i < \frac{M}{2}$.
Output: Is it possible to partition $A_1, \ldots, A_{3k}$ into k triples such that each triple sums to M?

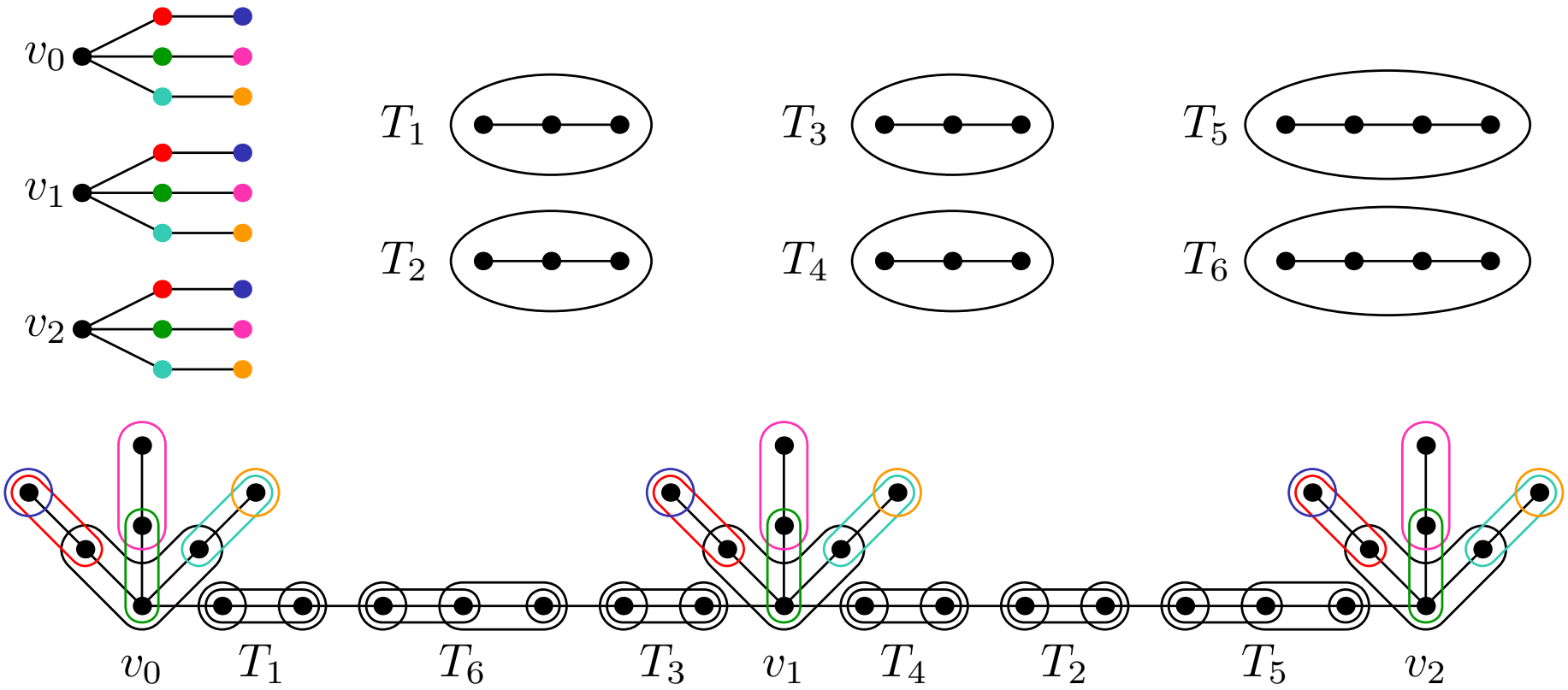
The reduction for REPEXT(INT, FIXED):

Input of 3-PARTITION: $k = 2$, $M = 7$ and A_i's are 2, 2, 2, 2, 3 and 3.

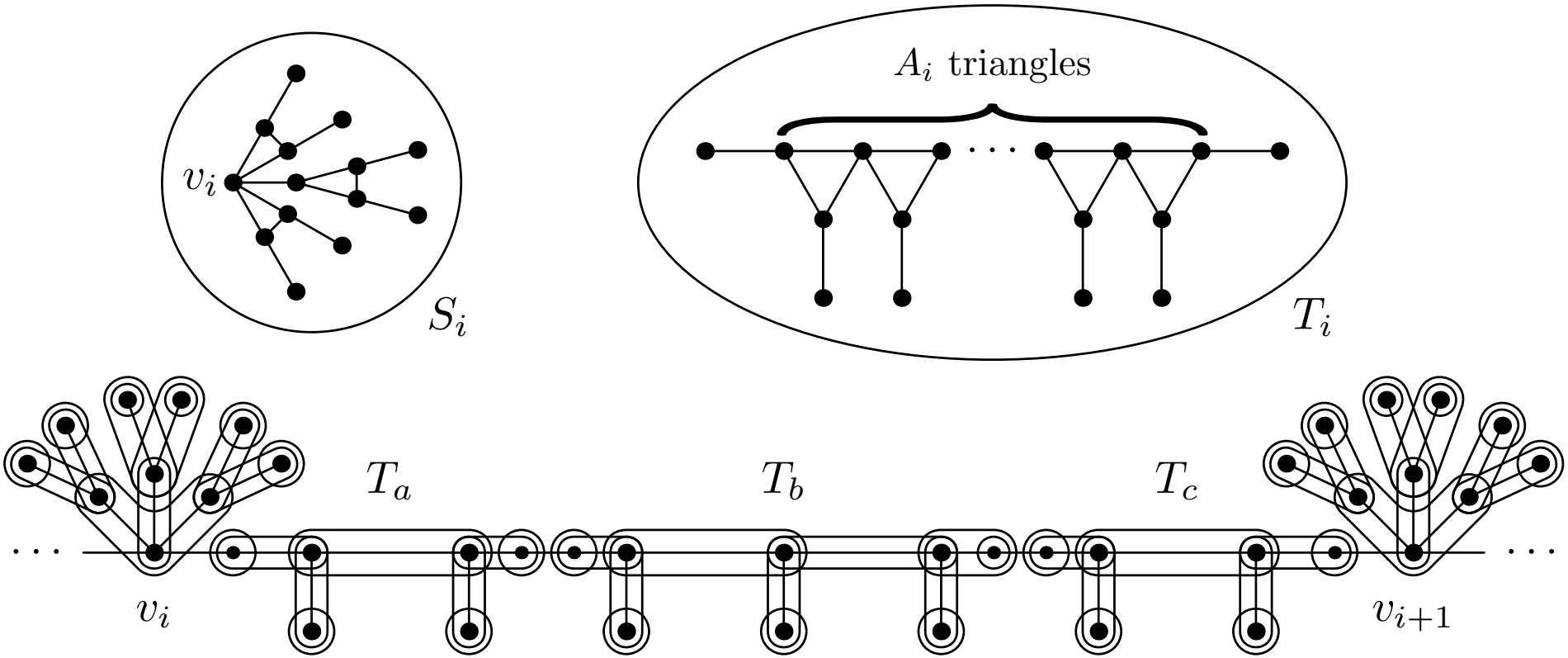


The reduction for RECOG*(PATH, FIXED) and RECOG*(CHOR, FIXED):

Input of 3-PARTITION: $k = 2$, $M = 7$ and A_i's are $2, 2, 2, 2, 3$ and 3.

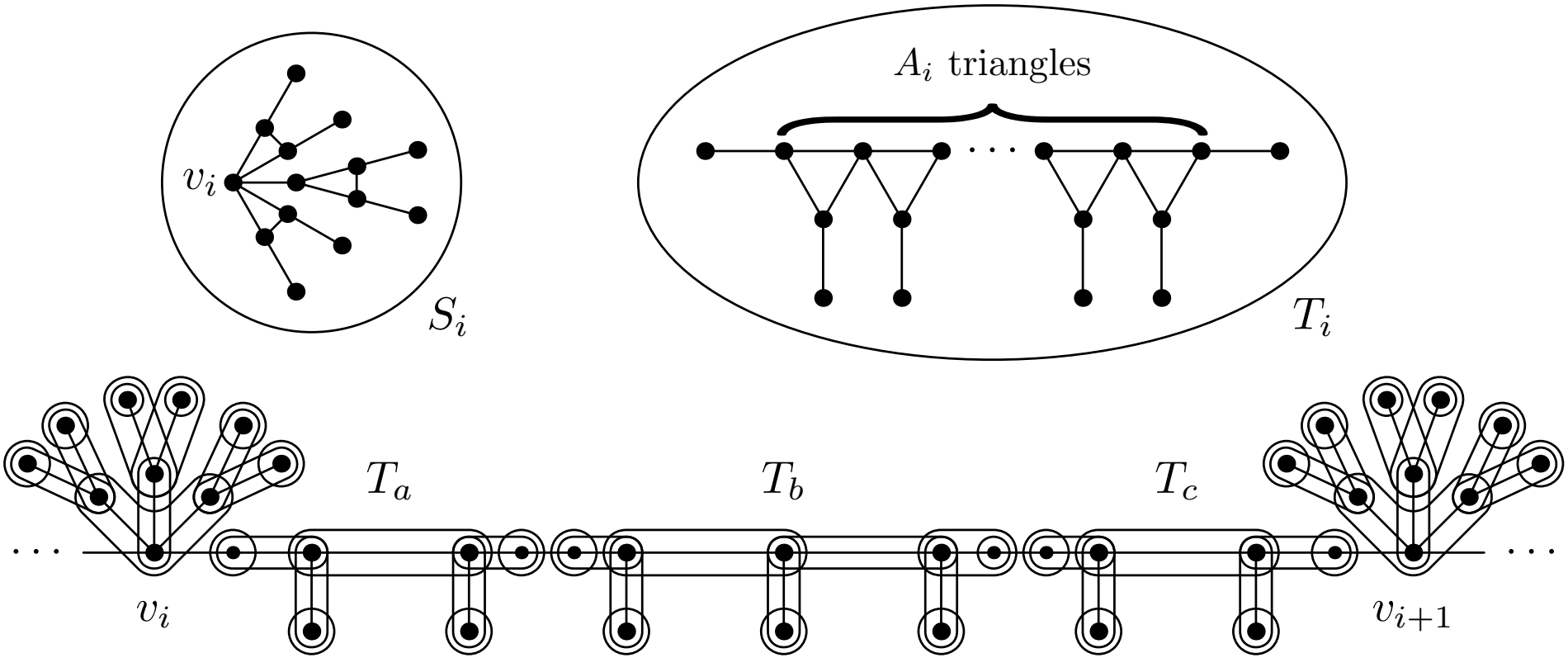


The reduction for RECOG*(PATH, SUB) and RECOG*(CHOR, SUB):



For ADD type: We have one big pre-drawn subtree covering whole T'.

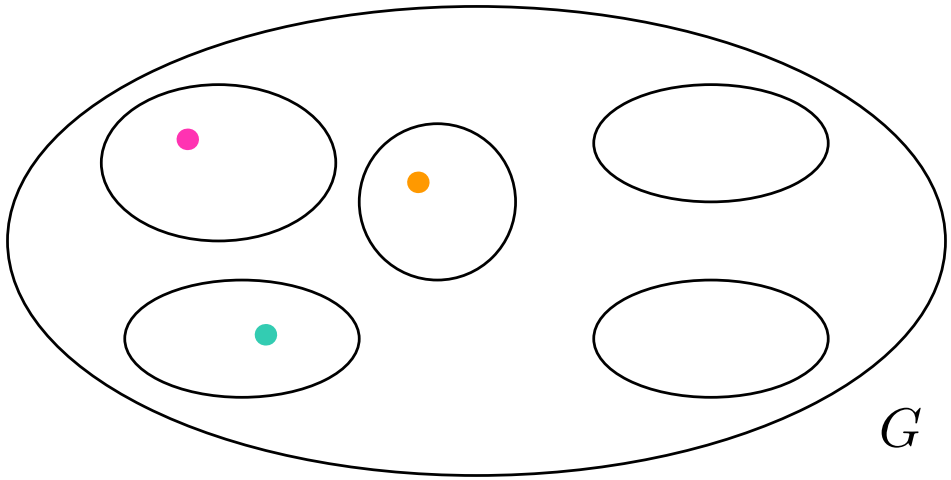
The reduction for RECOG*(PATH, SUB) and RECOG*(CHOR, SUB):



For ADD type: We have one big pre-drawn subtree covering whole T'.

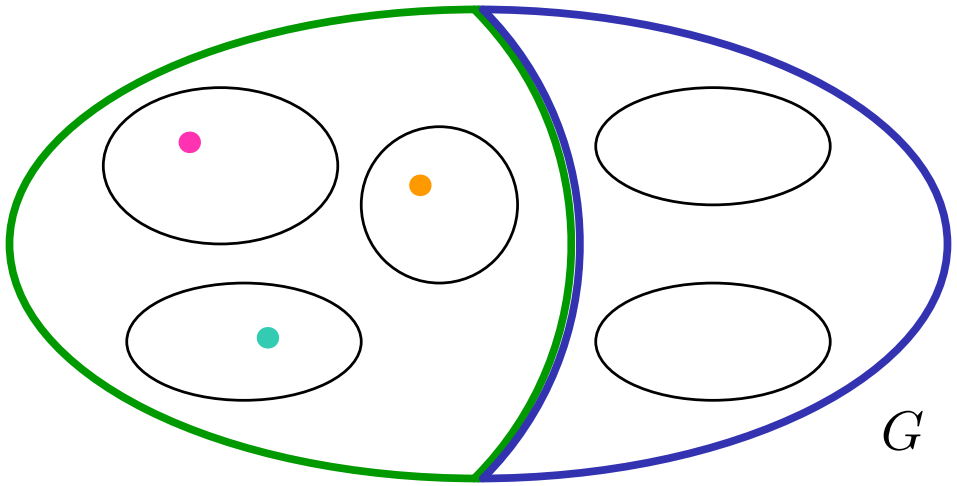
Polynomial results

General ideas for interval graphs:



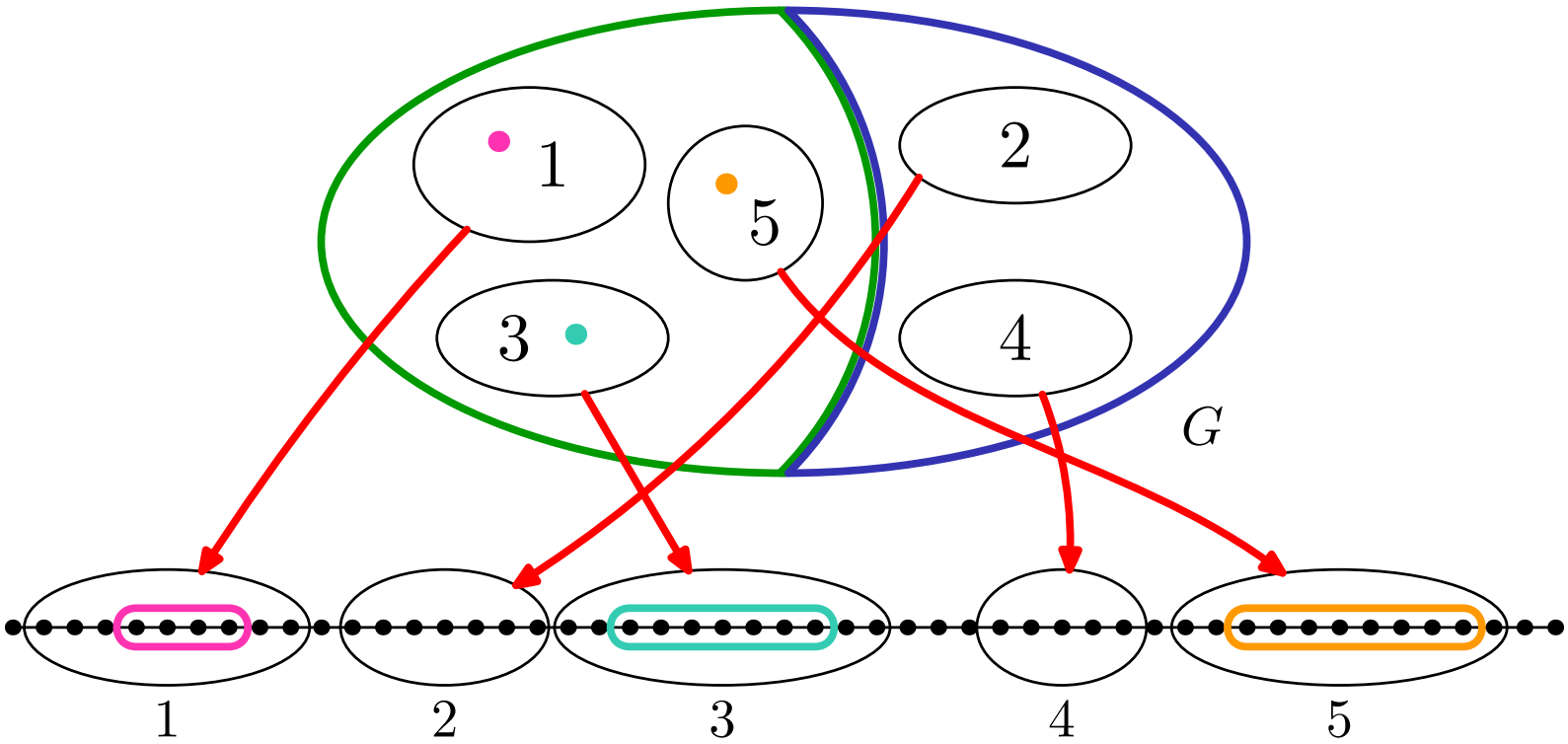
- There are two types of components: located and unlocated.
- We need to deduce an order of components from left to right.
- We process components from left to right.
- We place them greedily as far to the left as possible.

General ideas for interval graphs:



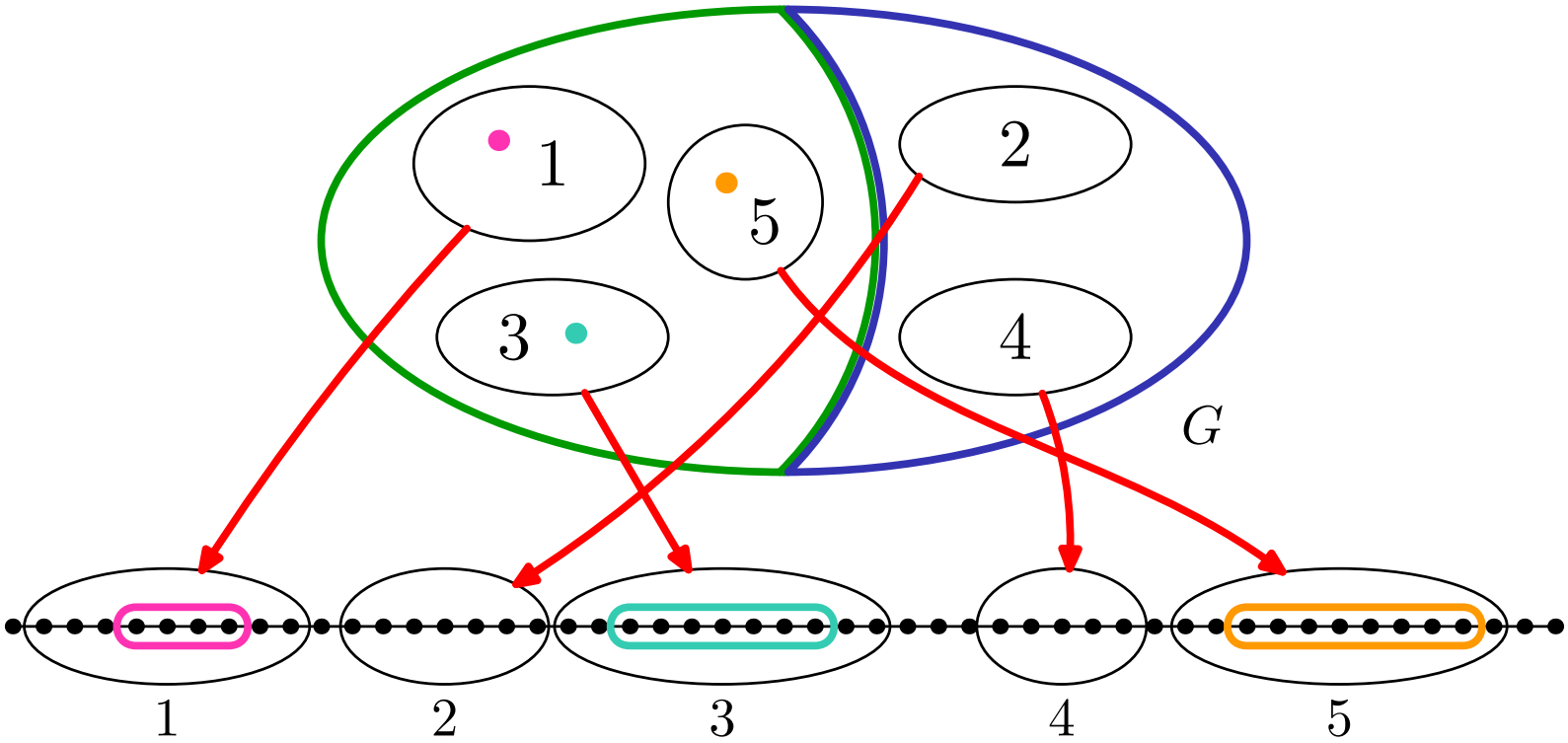
- There are two types of components: located and unlocated.
- We need to deduce an order of components from left to right.
- We process components from left to right.
- We place them greedily as far to the left as possible.

General ideas for interval graphs:

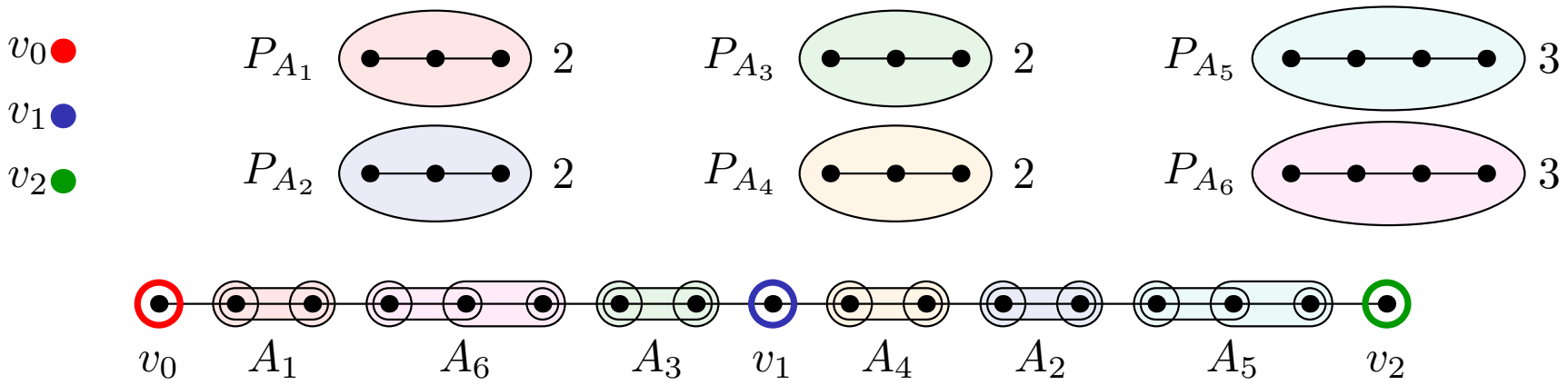


- There are two types of components: located and unlocated.
- We need to deduce an order of components from left to right.
- We process components from left to right.
- We place them greedily as far to the left as possible.

General ideas for interval graphs:



- There are two types of components: located and unlocated.
- We need to deduce an order of components from left to right.
- We process components from left to right.
- We place them greedily as far to the left as possible.

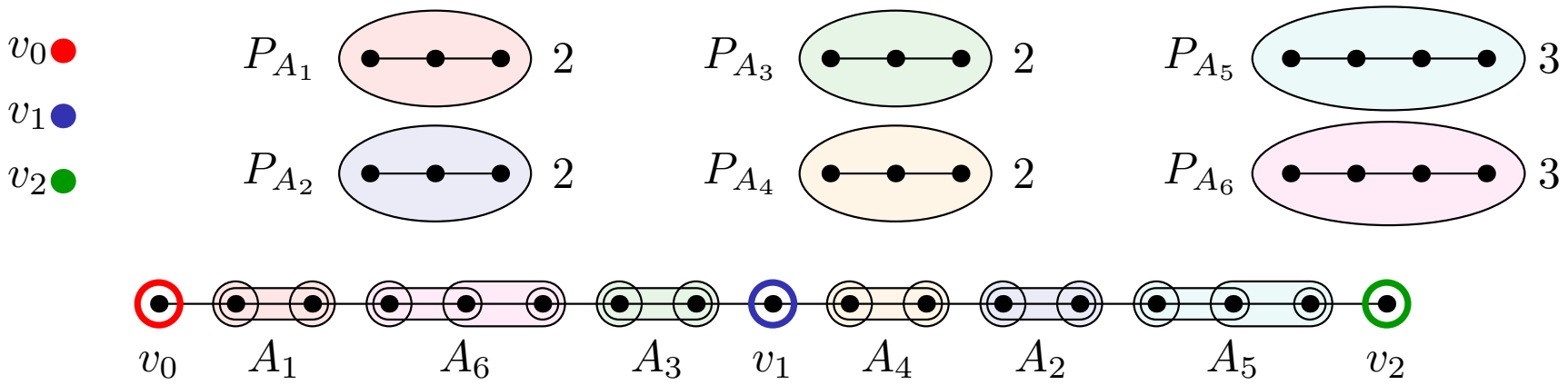


Problem: BINPACKING
Input: Integers k, ℓ, $V_1, \ldots, V_k$ and $A_1, \ldots, A_\ell$.
Output: Does there exist a k-partition $\mathcal{P}_1, \ldots, \mathcal{P}_k$ of $A_1, \ldots, A_\ell$ such that $\sum_{A_i \in \mathcal{P}_j} A_i \leq V_j$ for every $\mathcal{P}_j$.

Theorem

The problems BINPACKING *and* REPEXT(PINT, FIXED) *are "equivalent":*

$$\text{BINPACKING} \leq \text{REPEXT(PINT, FIXED)} \leq_{wtt} \text{BINPACKING}.$$



Problem: BINPACKING

Input: Integers k, ℓ, $V_1, \ldots, V_k$ and $A_1, \ldots, A_\ell$.

Output: Does there exist a k-partition $\mathcal{P}_1, \ldots, \mathcal{P}_k$ of $A_1, \ldots, A_\ell$ such that $\sum_{A_i \in \mathcal{P}_j} A_i \leq V_j$ for every $\mathcal{P}_j$.

Theorem

The problems BINPACKING *and* REPEXT(PINT, FIXED) *are "equivalent":*

$$\text{BINPACKING} \leq \text{REPEXT(PINT, FIXED)} \leq_{wtt} \text{BINPACKING}.$$

Open Problem

What is the complexity of RepExt(PATH, Both)?

Open Problem

How hard is RepExt(INT, Fixed) with respect to the number k of pre-drawn intervals?

Open Problem

What is the complexity of the RepExt problem for other classes?

CIRCLE, CIRCULAR ARC, ...

Open Problem

What is the complexity of RepExt(PATH, Both)?

Open Problem

How hard is RepExt(INT, Fixed) with respect to the number k of pre-drawn intervals?

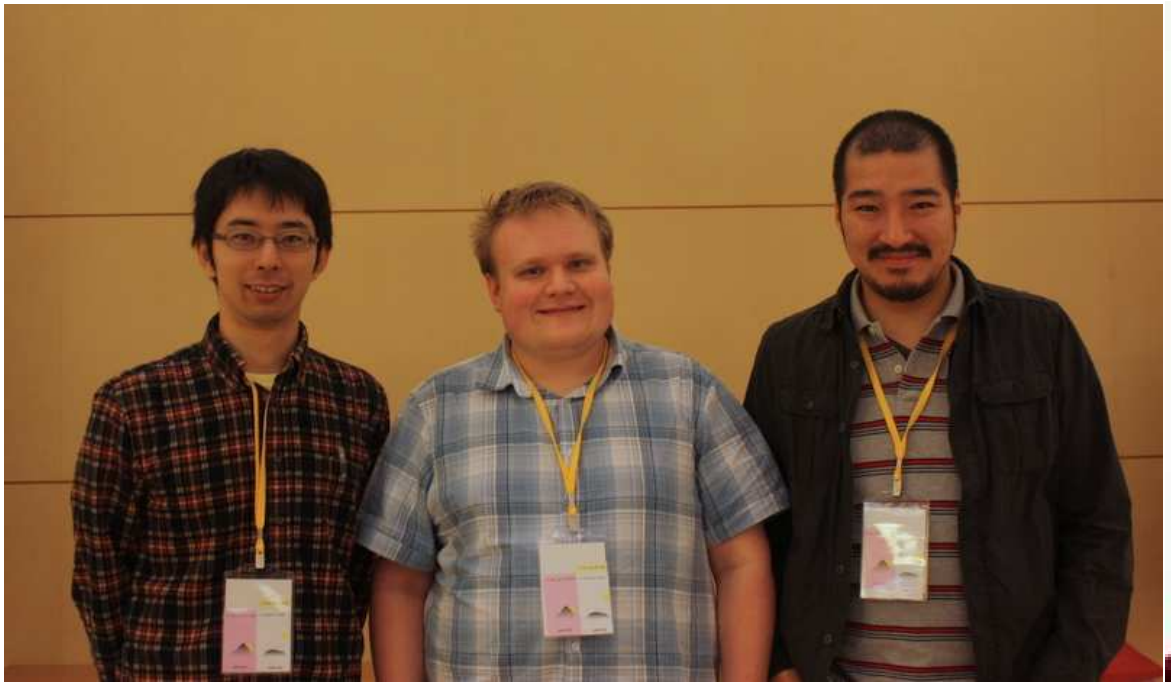
Open Problem

What is the complexity of the RepExt problem for other classes?

CIRCLE, CIRCULAR ARC, ...

Thank you!

谢谢



Marry Christmas from Toshiki, Pavel, Yota and Jan.