Unit Disk Representations

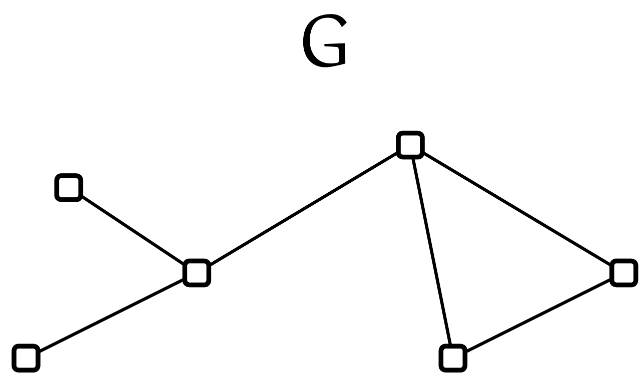
of Embedded Trees, Outerplanar and Multi-Legged Graphs

Sujoy Bhore, Maarten Löffler, Soeren Nickel and Martin Nöllenburg

16.09.2021 · GD 2021

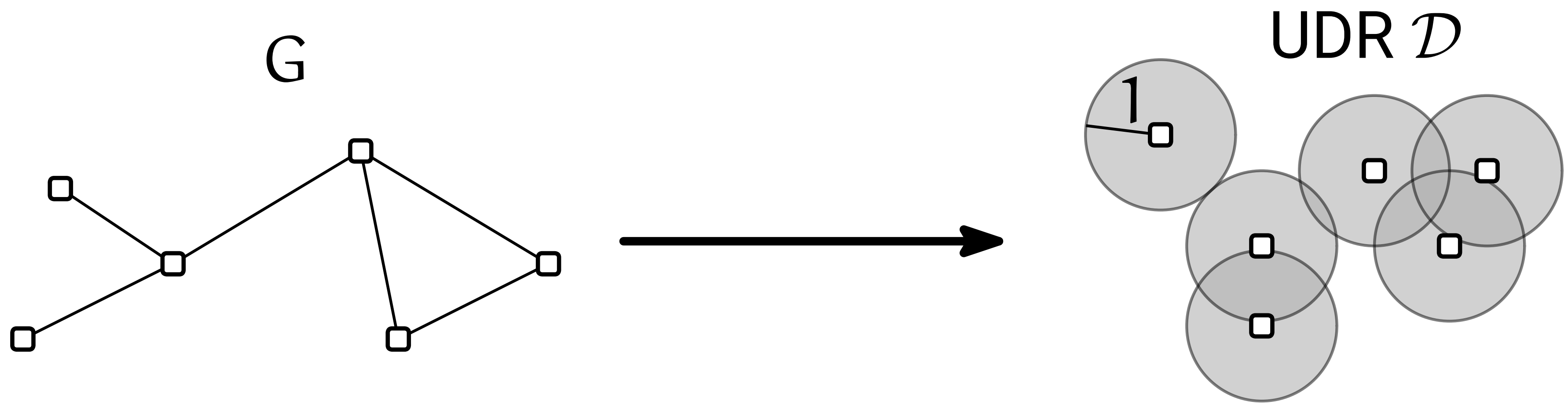
Unit Disk Graphs

A graph G is a **unit disk graph**



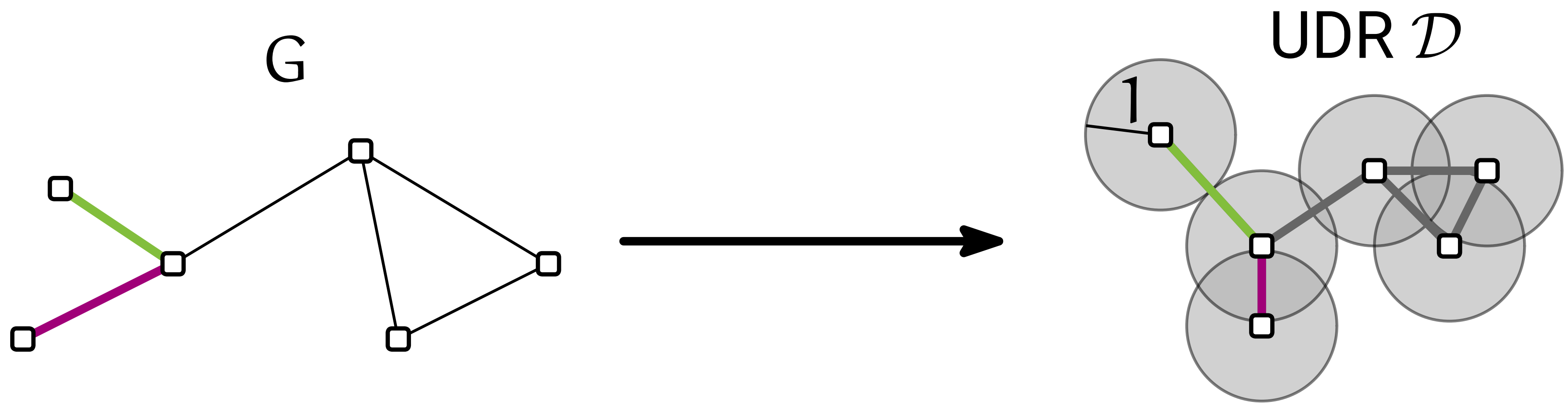
Unit Disk Graphs

A graph G is a **unit disk graph** if it has a **unit disk representation** $\mathcal{D}$.



Unit Disk Graphs

A graph G is a **unit disk graph** if it has a **unit disk representation** $\mathcal{D}$.

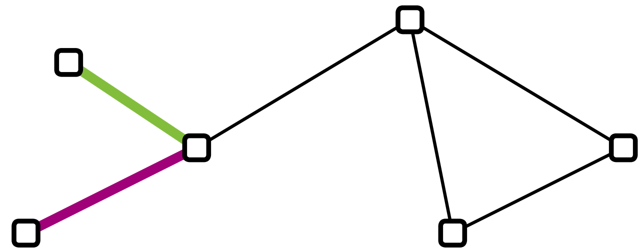


- Disks can be in **contact** or **intersect**.

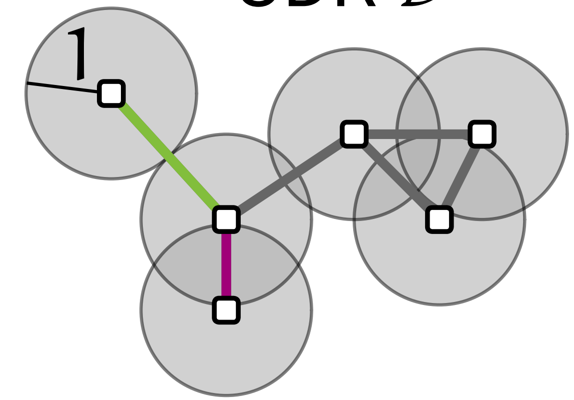
Unit Disk Graphs

A graph G is a **unit disk graph** if it has a **unit disk representation** $\mathcal{D}$.

G



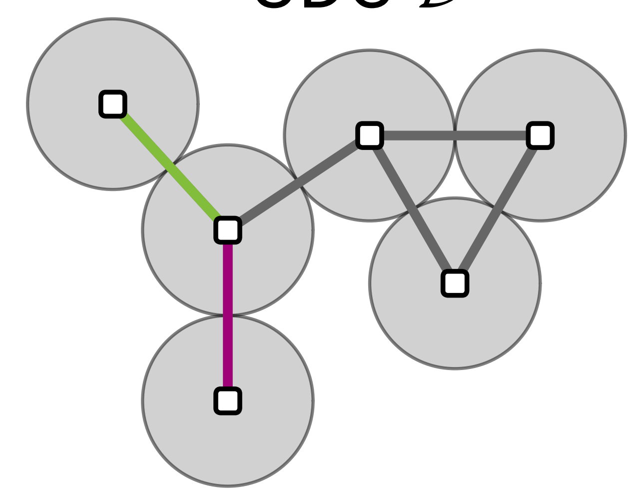
UDR $\mathcal{D}$



- Disks can be in **contact** or **intersect**.

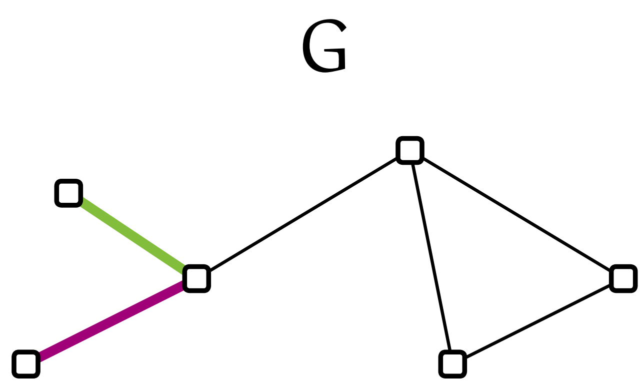
- If all disks are **interior-disjoint**, then $\mathcal{D}$ is a unit disk **contact** representation.

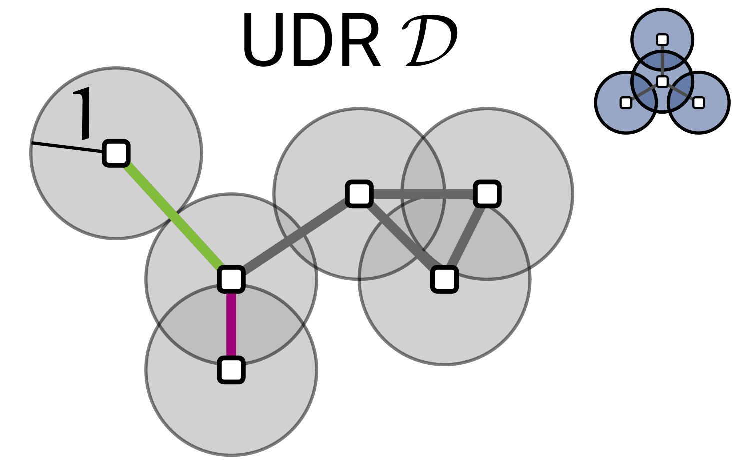
UDC $\mathcal{D}$



Unit Disk Graphs

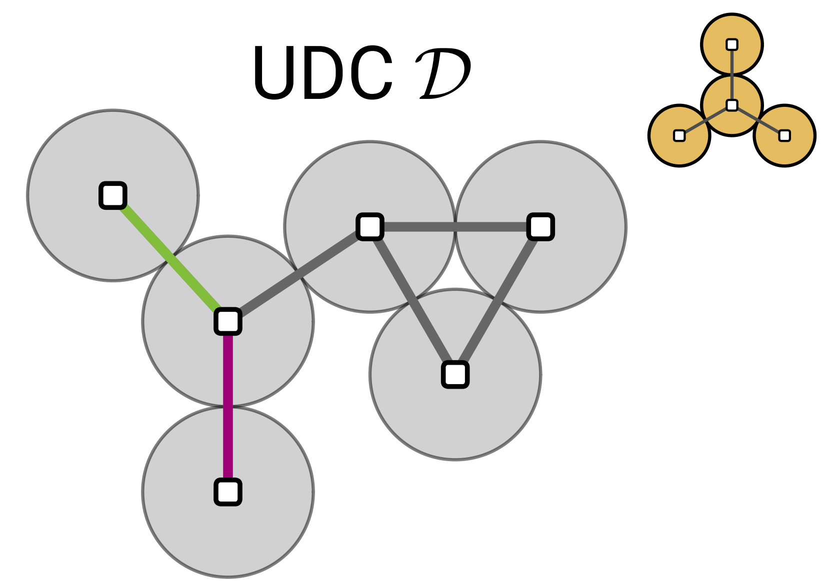
A graph G is a **unit disk graph** if it has a **unit disk representation** $\mathcal{D}$.





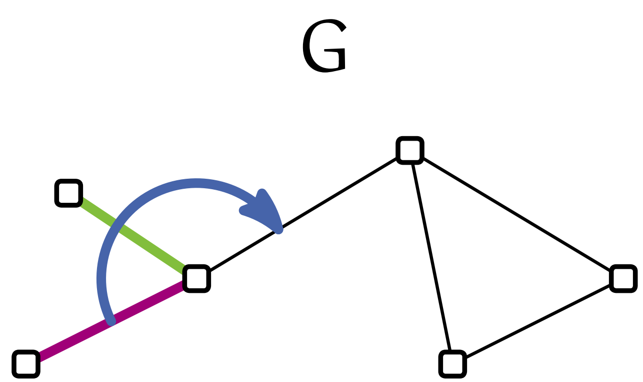
- Disks can be in **contact** or **intersect**.

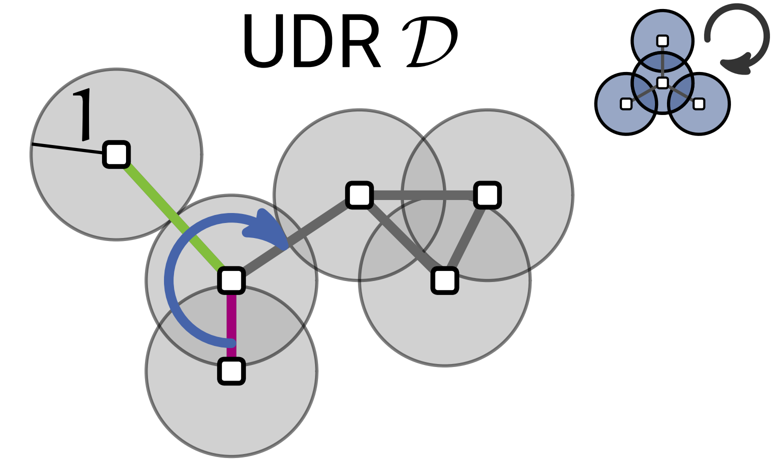
- If all disks are **interior-disjoint**, then $\mathcal{D}$ is a unit disk **contact** representation.

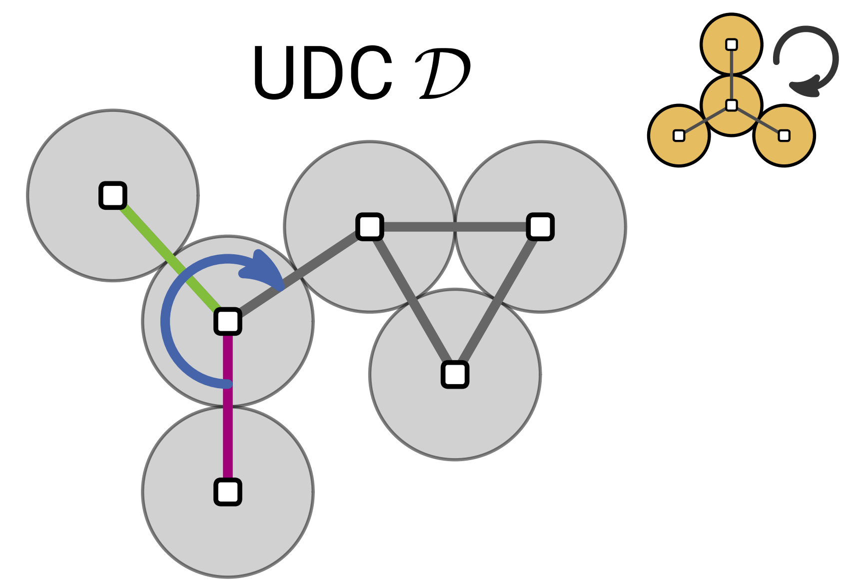


Unit Disk Graphs

A graph G is a **unit disk graph** if it has a **unit disk representation** $\mathcal{D}$.



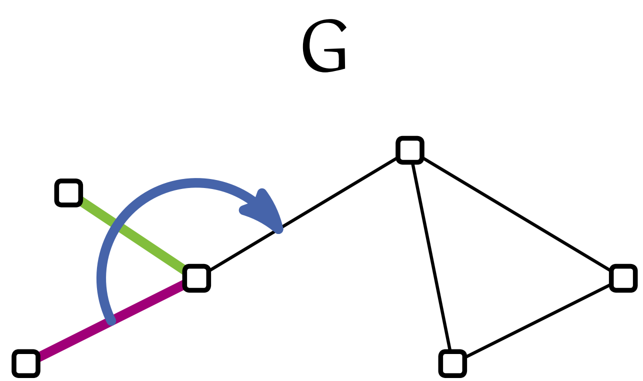


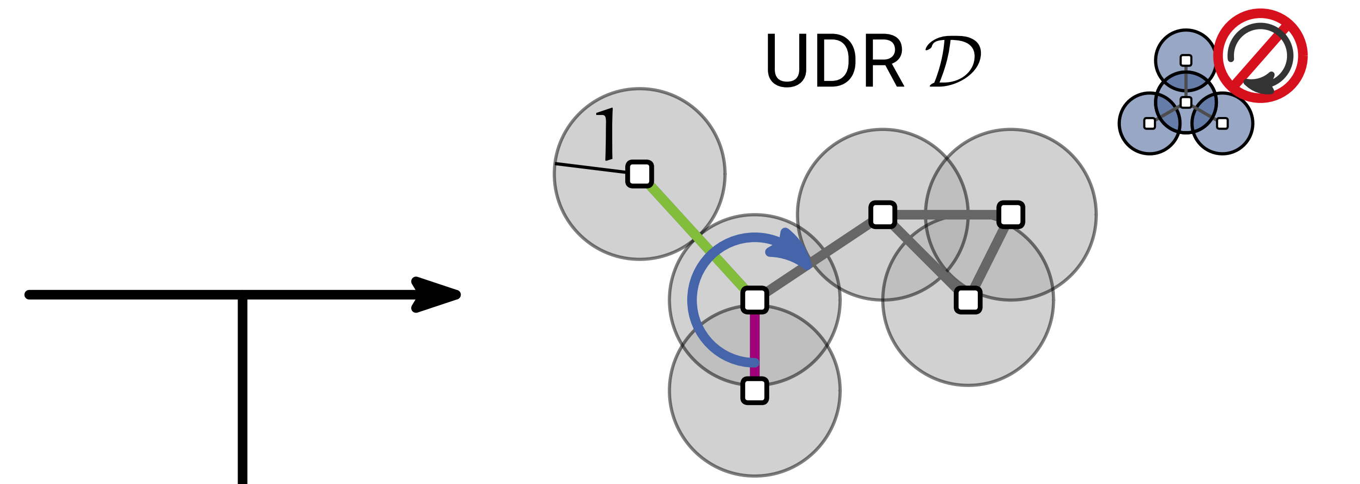


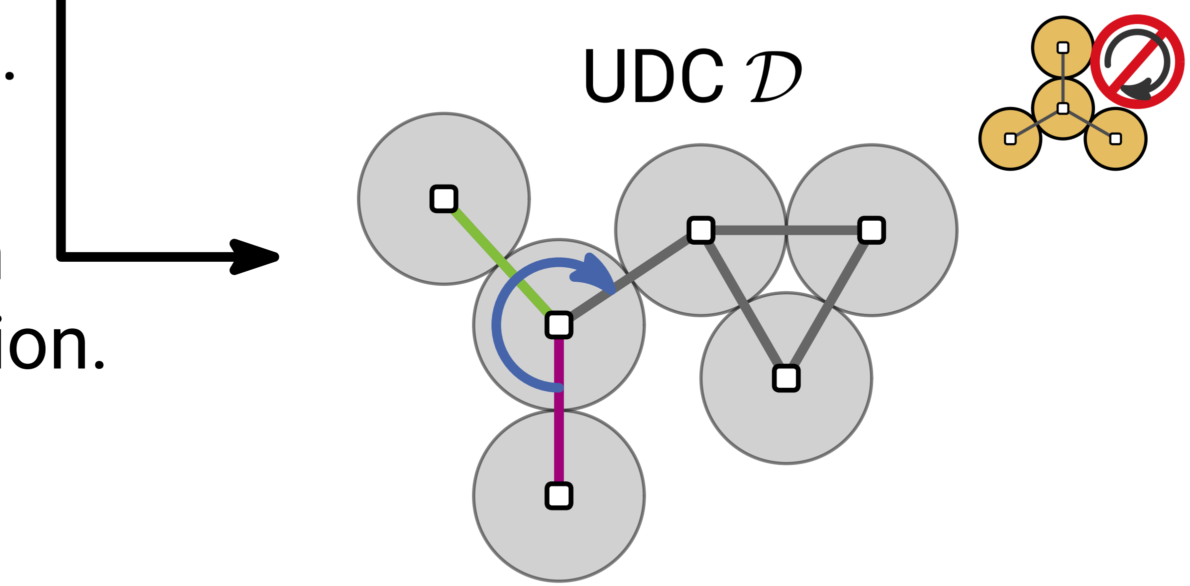
- Disks can be in **contact** or **intersect**.
- If all disks are **interior-disjoint**, then $\mathcal{D}$ is a unit disk **contact** representation.
- If G is embedded $\mathcal{D}$ needs to induce the **same rotation system**.

Unit Disk Graphs

A graph G is a **unit disk graph** if it has a **unit disk representation** $\mathcal{D}$.







- Disks can be in **contact** or **intersect**.
- If all disks are **interior-disjoint**, then $\mathcal{D}$ is a unit disk **contact** representation.
- If G is embedded $\mathcal{D}$ needs to induce the **same rotation system**.

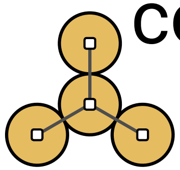
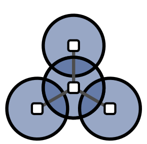
Recognition Problem

Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

[1] [Breu & Kirkpatrick, 1998]

Recognition Problem

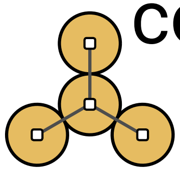
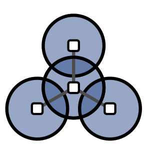
Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

	planar	outerplanar	tree	caterpillar
 contact graphs	NP-hard[1]			
embedded				
 disk graphs	NP-hard[1]			
embedded				

[1] [Breu & Kirkpatrick, 1998]

Recognition Problem

Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

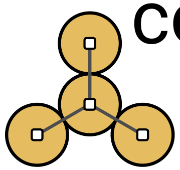
	planar	outerplanar	tree	caterpillar
 contact graphs	NP-hard[1]	NP-hard[2]		$\mathcal{O}(n)$ [2]
embedded				
 disk graphs	NP-hard[1]			
embedded				

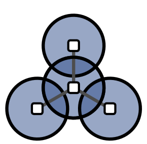
[1] [Breu & Kirkpatrick, 1998] [2] [Klemz et al., 2015]

Recognition Problem

Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

	planar	outerplanar	tree	caterpillar
contact graphs	NP-hard[1]	NP-hard[2]		$\mathcal{O}(n)$ [2]
embedded			NP-hard[3]	
disk graphs	NP-hard[1]			
embedded				

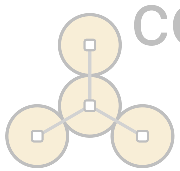


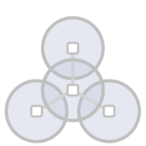


[1] [Breu & Kirkpatrick, 1998] [2] [Klemz et al., 2015]
[3] [Bowen et al., 2015]

Recognition Problem – Our Results

Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]





	planar	outerplanar	tree	caterpillar
contact graphs	NP-hard[1]	NP-hard[2]		$\mathcal{O}(n)$ [2]
embedded			NP-hard[3]	
disk graphs	NP-hard[1]	NP-hard		
embedded			NP-hard	

[1] [Breu & Kirkpatrick, 1998] [2] [Klemz et al., 2015]
[3] [Bowen et al., 2015]

Recognition Problem – Our Results

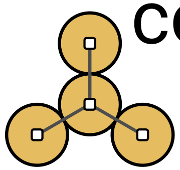
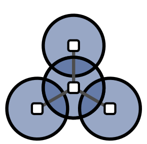
Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

	planar	outerplanar	tree	caterpillar
contact graphs	NP-hard[1]	NP-hard[2]		$\mathcal{O}(n)$ [2]
embedded	NP-hard[3]			
disk graphs	NP-hard[1]	NP-hard		$\mathcal{O}(n)$
embedded	NP-hard			

[1] [Breu & Kirkpatrick, 1998] [2] [Klemz et al., 2015]
[3] [Bowen et al., 2015]

Recognition Problem – Our Results

Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

	planar	outerplanar	tree	caterpillar
 contact graphs	NP-hard[1]	NP-hard[2]	?	$\mathcal{O}(n)$ [2]
embedded	NP-hard[3]			?
 disk graphs	NP-hard[1]	NP-hard	?	$\mathcal{O}(n)$
embedded	NP-hard			?

[1] [Breu & Kirkpatrick, 1998]
[2] [Klemz et al., 2015]
[3] [Bowen et al., 2015]

Recognition Problem – Our Results

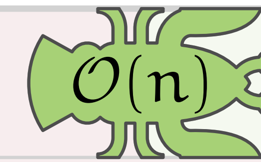
Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

	planar	outerplanar	tree	caterpillar
weak contact			NP-hard[4]	$\mathcal{O}(n)$ [4]
embedded				NP-hard[5]
contact graphs	NP-hard[1]	NP-hard[2]	?	$\mathcal{O}(n)$ [2]
embedded			NP-hard[3]	?
disk graphs	NP-hard[1]	NP-hard	?	$\mathcal{O}(n)$
embedded			NP-hard	?

[1] [Breu & Kirkpatrick, 1998] [2] [Klemz et al., 2015] [3] [Bowen et al., 2015] [4] [Cleve, 2020] [5] [Chiu et al., 2019]

Recognition Problem – Our Results

Recognition of unit disk (contact) graphs is NP-hard for planar graphs.[1]

	planar	outerplanar	tree	caterpillar
weak contact	NP-hard[4]			 $\mathcal{O}(n)$ [4]
embedded	NP-hard[5]			
contact graphs	NP-hard[1]	NP-hard[2]	?	$\mathcal{O}(n)$ [2]
embedded	NP-hard[3]			?
disk graphs	NP-hard[1]	NP-hard	?	$\mathcal{O}(n)$
embedded	NP-hard			?

[1] [Breu & Kirkpatrick, 1998] [2] [Klemz et al., 2015] [3] [Bowen et al., 2015] [4] [Cleve, 2020] [5] [Chiu et al., 2019]

NP-Hardness Overview

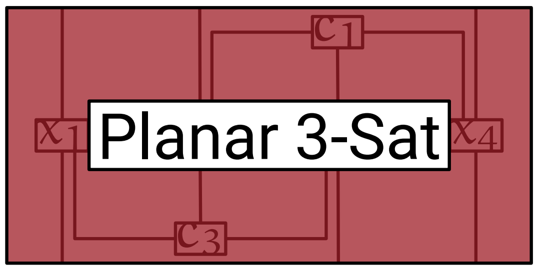
Recognition of unit disk contact graphs is NP-hard for embedded trees.

[Bowen et al., 2015]

NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

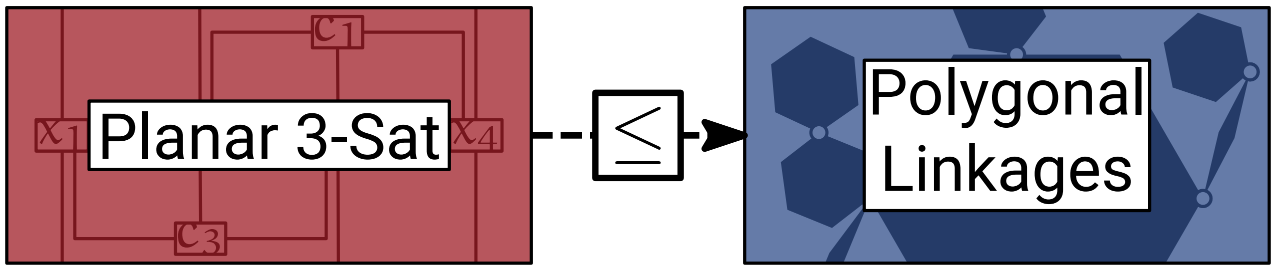
Reduction from planar 3-Sat [Bowen et al., 2015]



NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

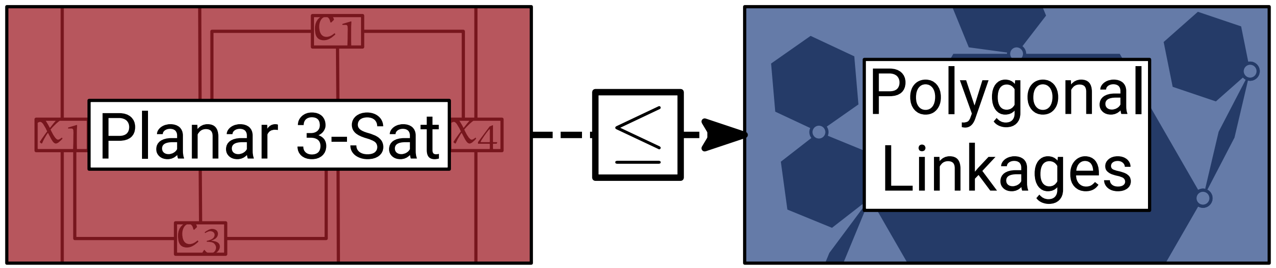
Reduction from planar 3-Sat [Bowen et al., 2015]

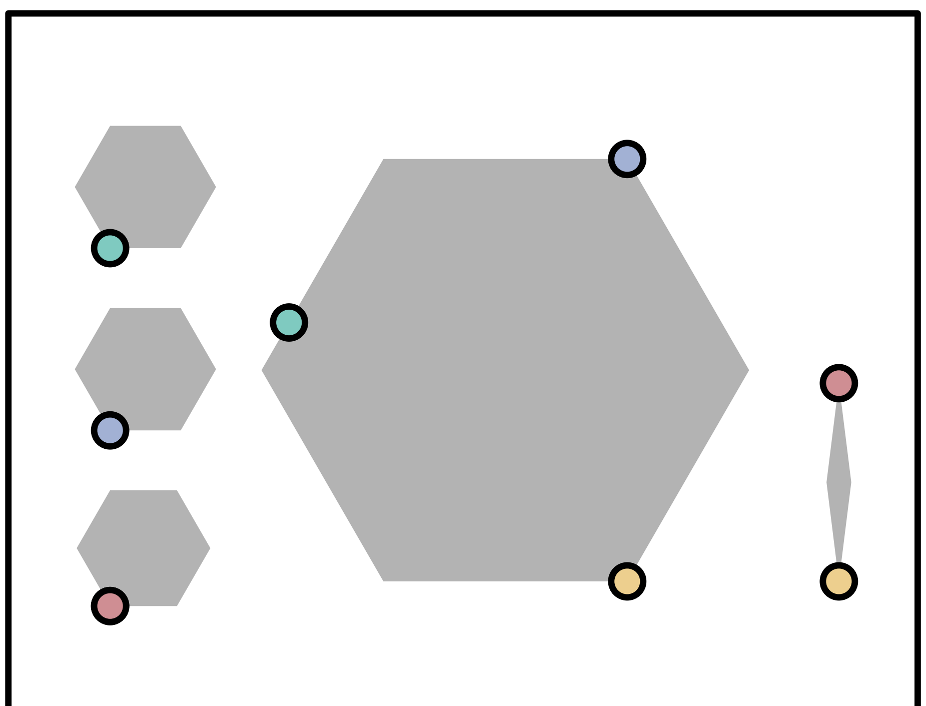


NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

Reduction from planar 3-Sat [Bowen et al., 2015]

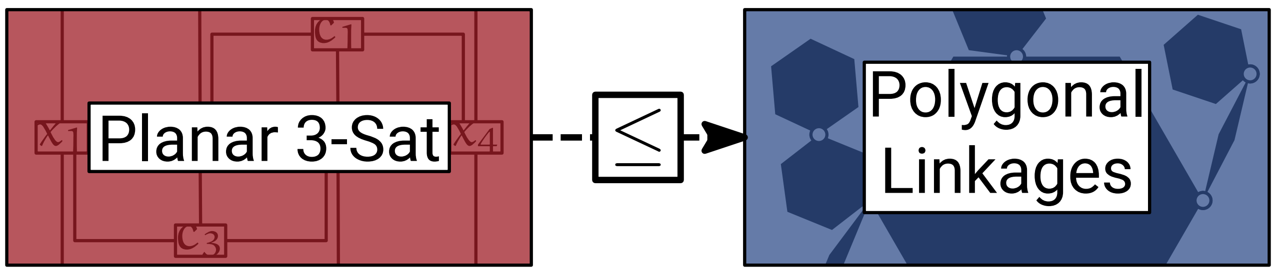


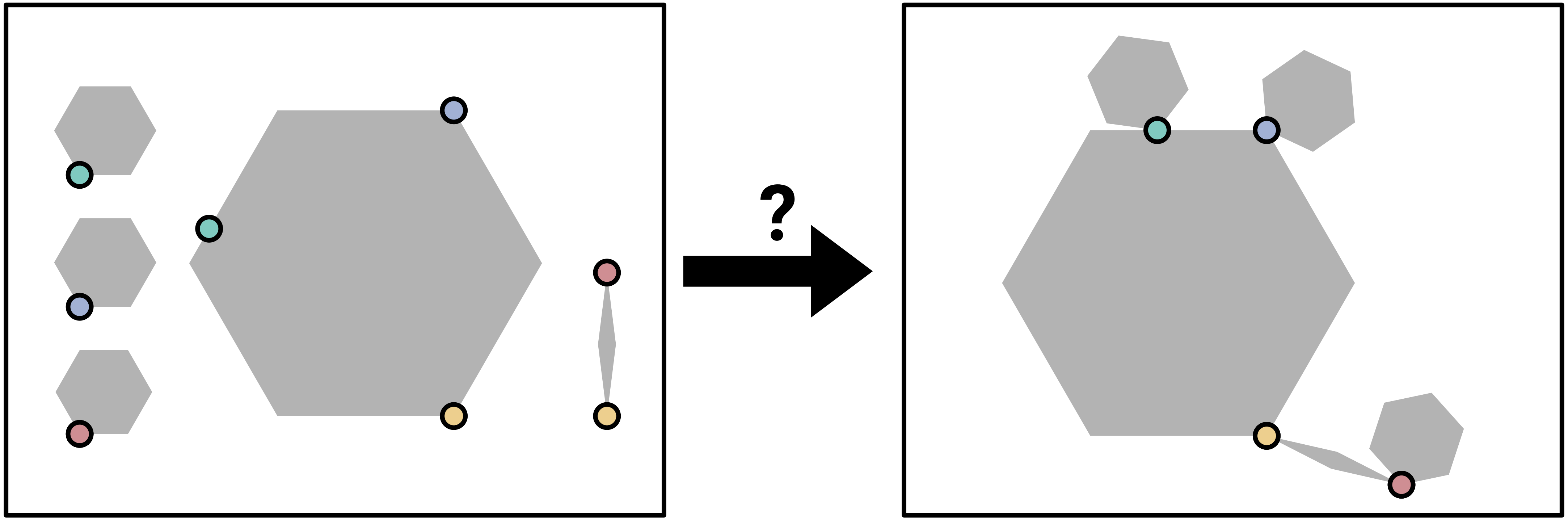


NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

Reduction from planar 3-Sat [Bowen et al., 2015]

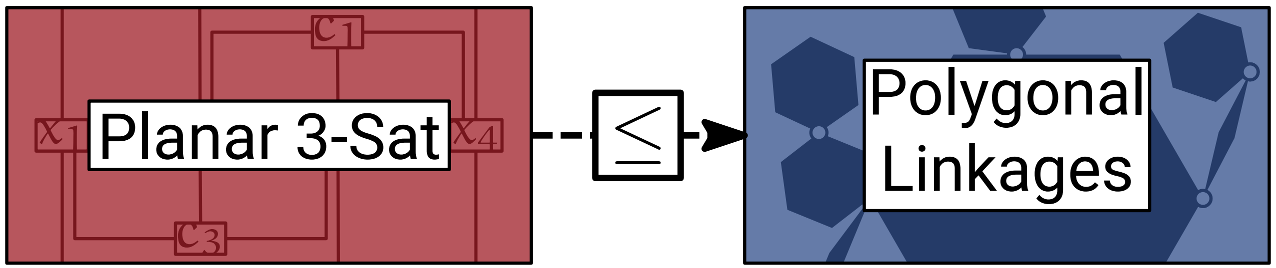


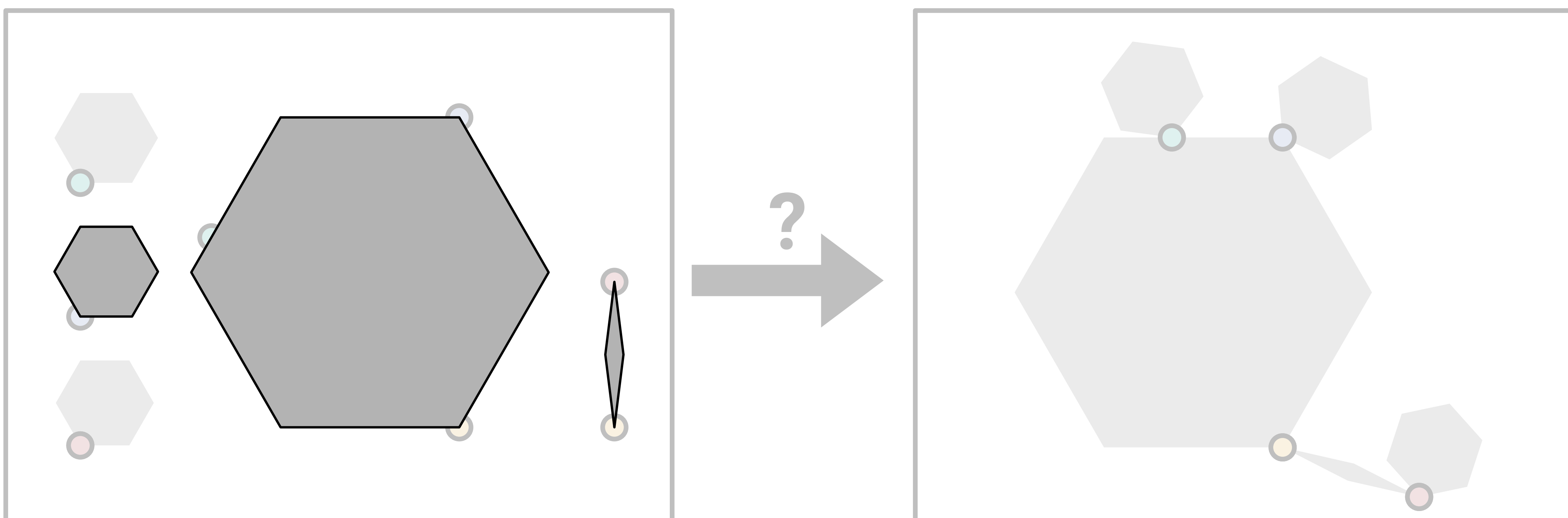


NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

Reduction from planar 3-Sat [Bowen et al., 2015]

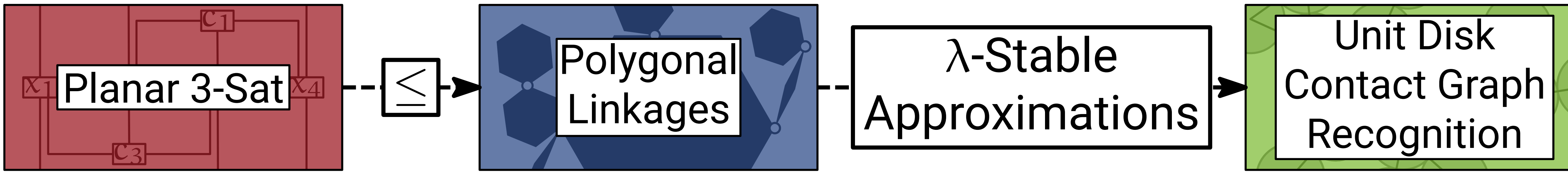


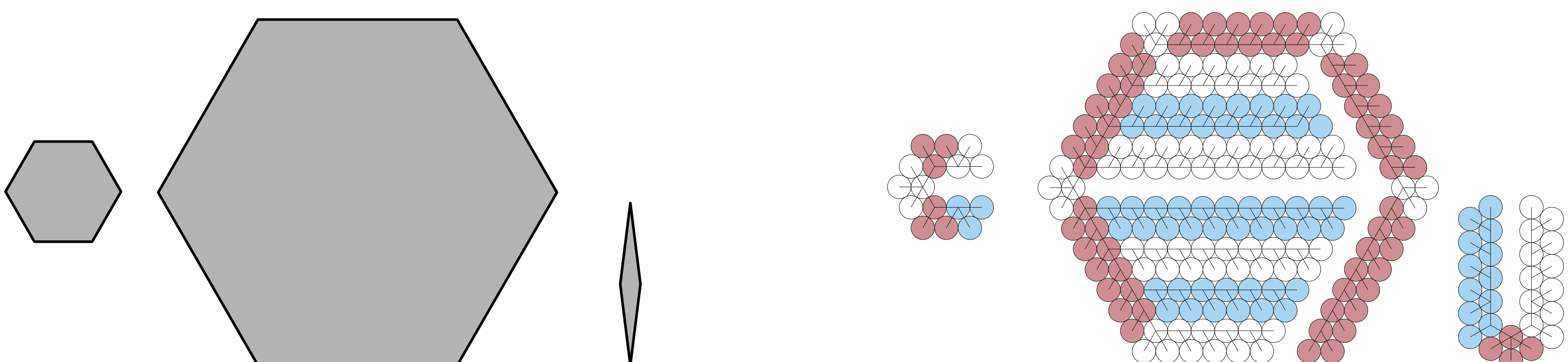


NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

Reduction from planar 3-Sat [Bowen et al., 2015]

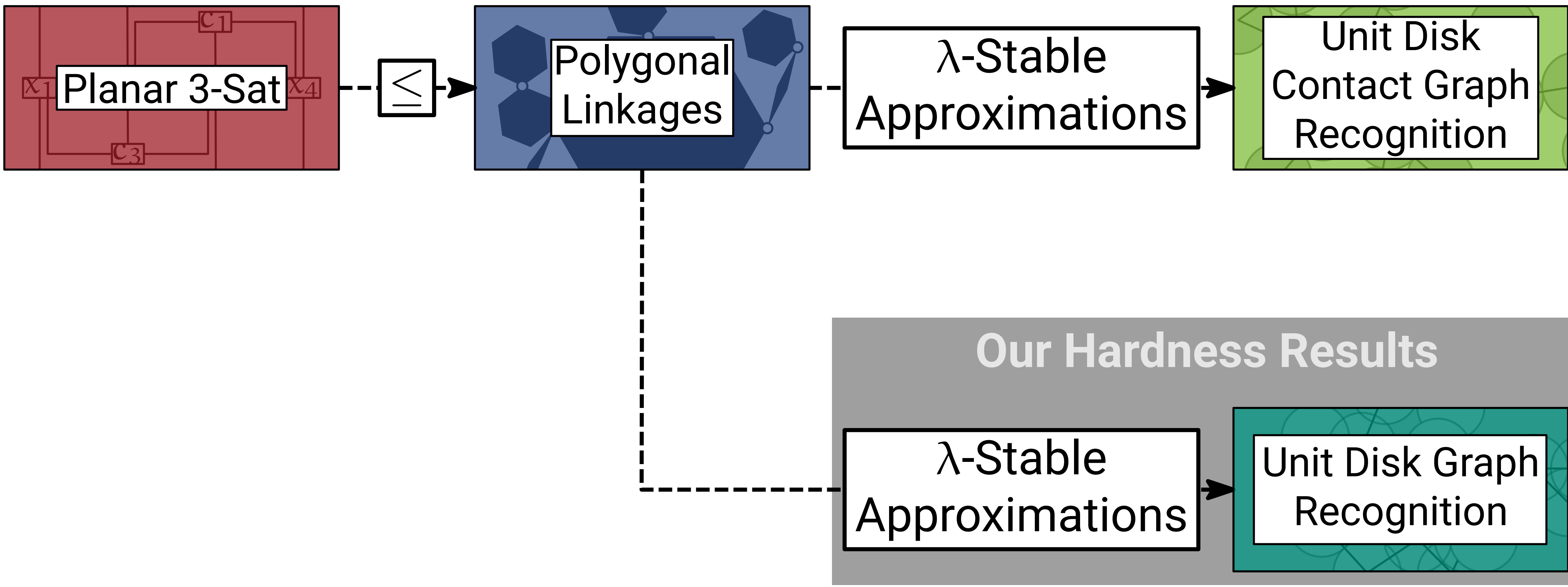




NP-Hardness Overview

Recognition of unit disk contact graphs is NP-hard for embedded trees.

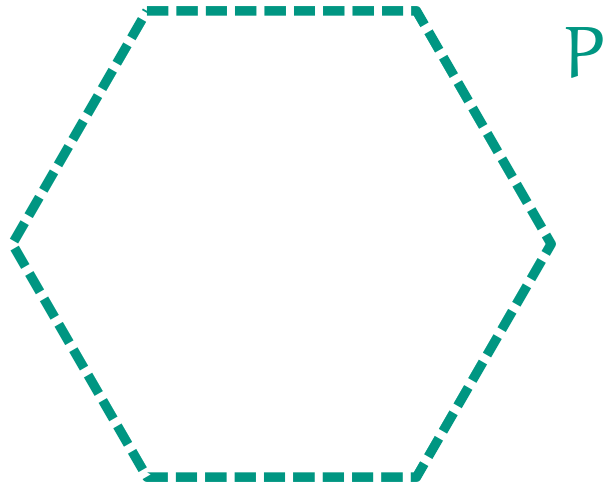
Reduction from planar 3-Sat [Bowen et al., 2015]



λ-Stable Approximations of Polygons

a **λ-stable approximation** of a polygon P

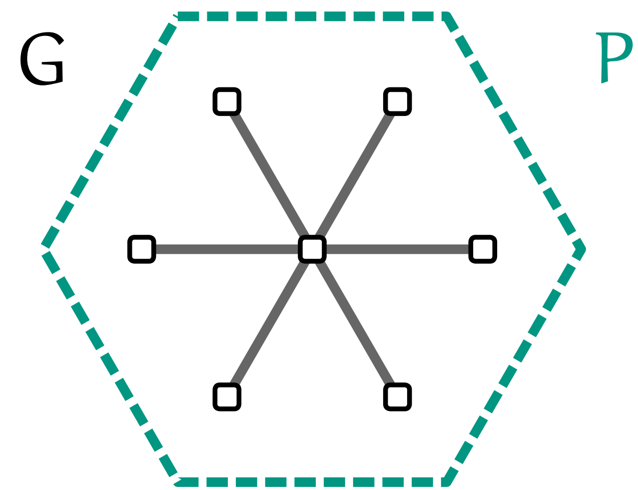
[Bowen et al., 2015]



λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P

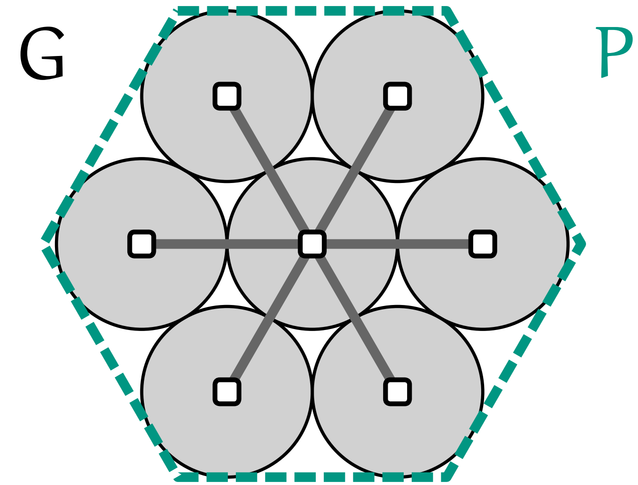
[Bowen et al., 2015]



λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if the union of all disks in every UDC of G

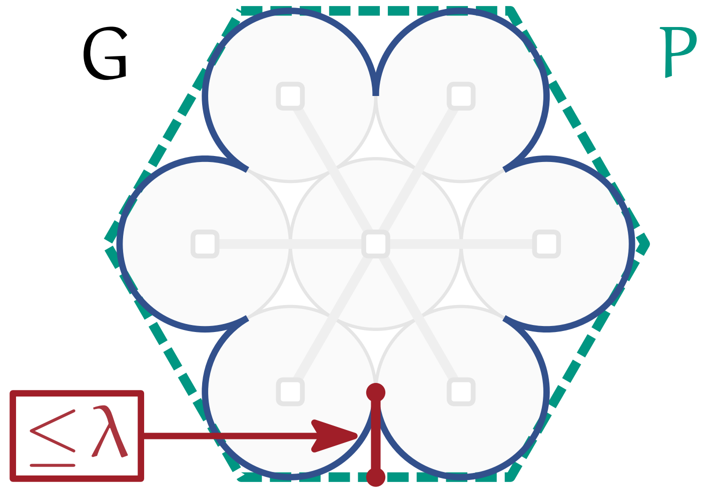
[Bowen et al., 2015]



λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if
the union of all disks in every UDC of G has a Hausdorff distance $\leq \lambda$ to P.

[Bowen et al., 2015]

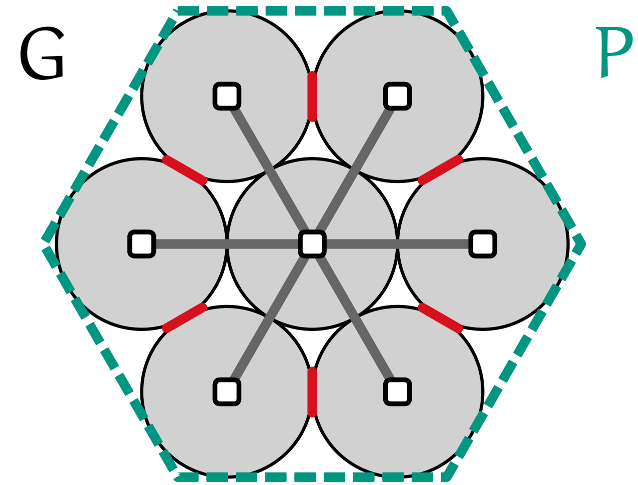


(directed) Hausdorff-Distance $\approx$ How far do you have to travel from any point on P to reach the UDC boundary of G

λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if
the union of all disks in every UDC of G has a Hausdorff distance $\leq \lambda$ to P.

[Bowen et al., 2015]



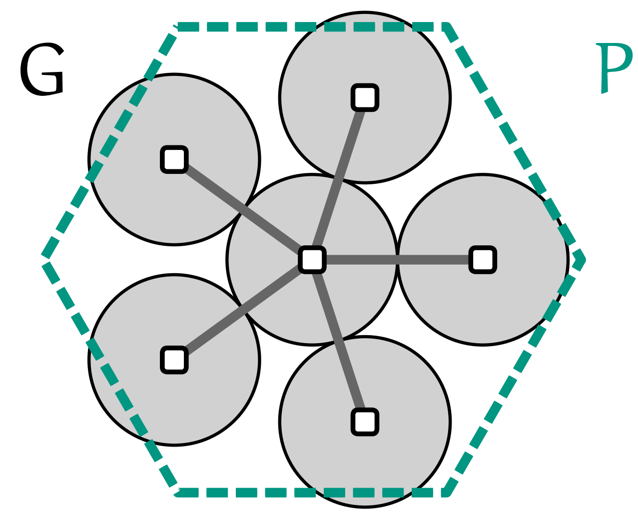
But:

- False adjacencies are not allowed

λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if the union of all disks in every UDC of G has a Hausdorff distance $\leq \lambda$ to P.

[Bowen et al., 2015]

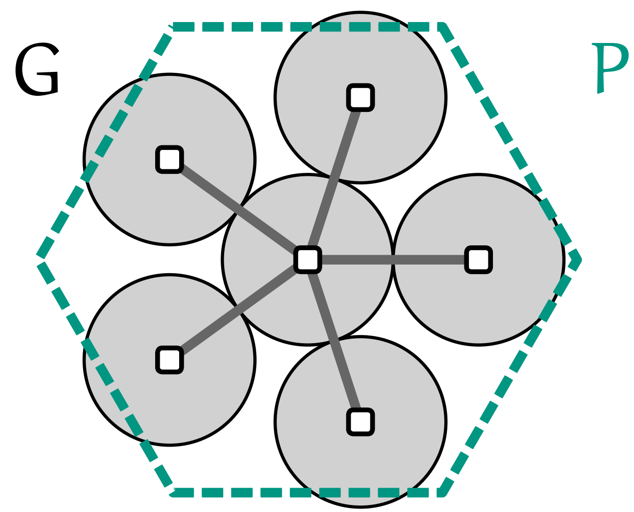


But:

- False adjacencies are not allowed

λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if the union of all disks in every **UDR** of G has a Hausdorff distance $\leq \lambda$ to P.

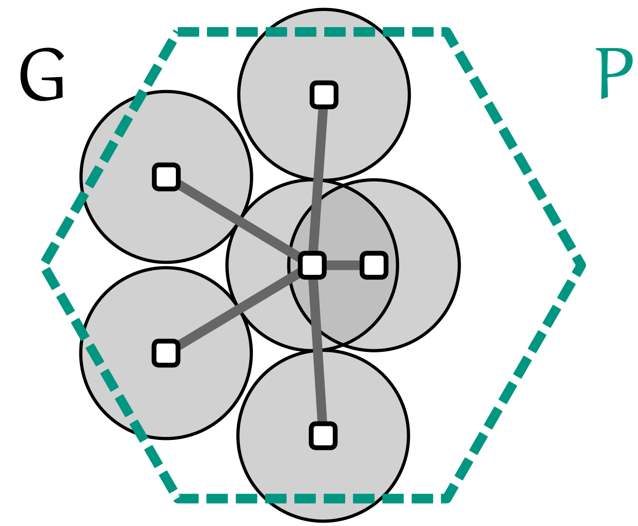


But:

- False adjacencies are not allowed
- UDR allow for overlap

λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if the union of all disks in every **UDR** of G has a Hausdorff distance $\leq \lambda$ to P.

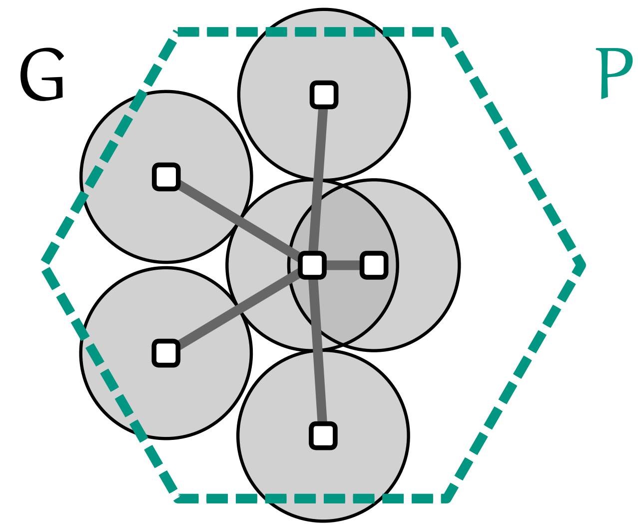


But:

- False adjacencies are not allowed
- UDR allow for overlap

λ-Stable Approximations of Polygons

A graph G is a **λ-stable approximation** of a polygon P, if the union of all disks in every **UDR** of G has a Hausdorff distance $\leq \lambda$ to P.



But:

- False adjacencies are not allowed
- UDR allow for overlap
- Must hold for all UDRs

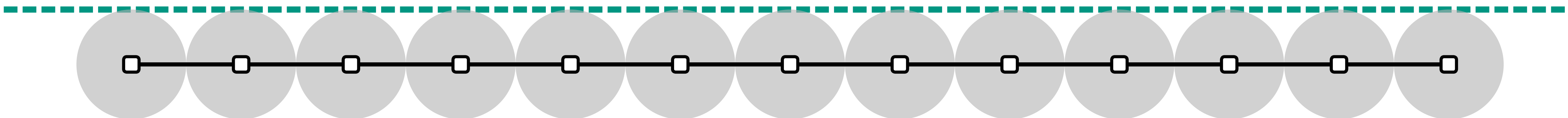
$\Longrightarrow$ We have to force a very restricted placement in the UDR.

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bend Restrictions in Embedded UDRs

What about tracing a straight line?



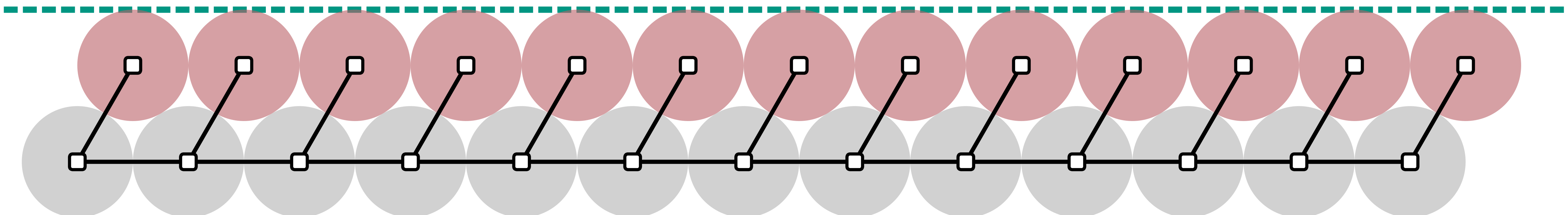
Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

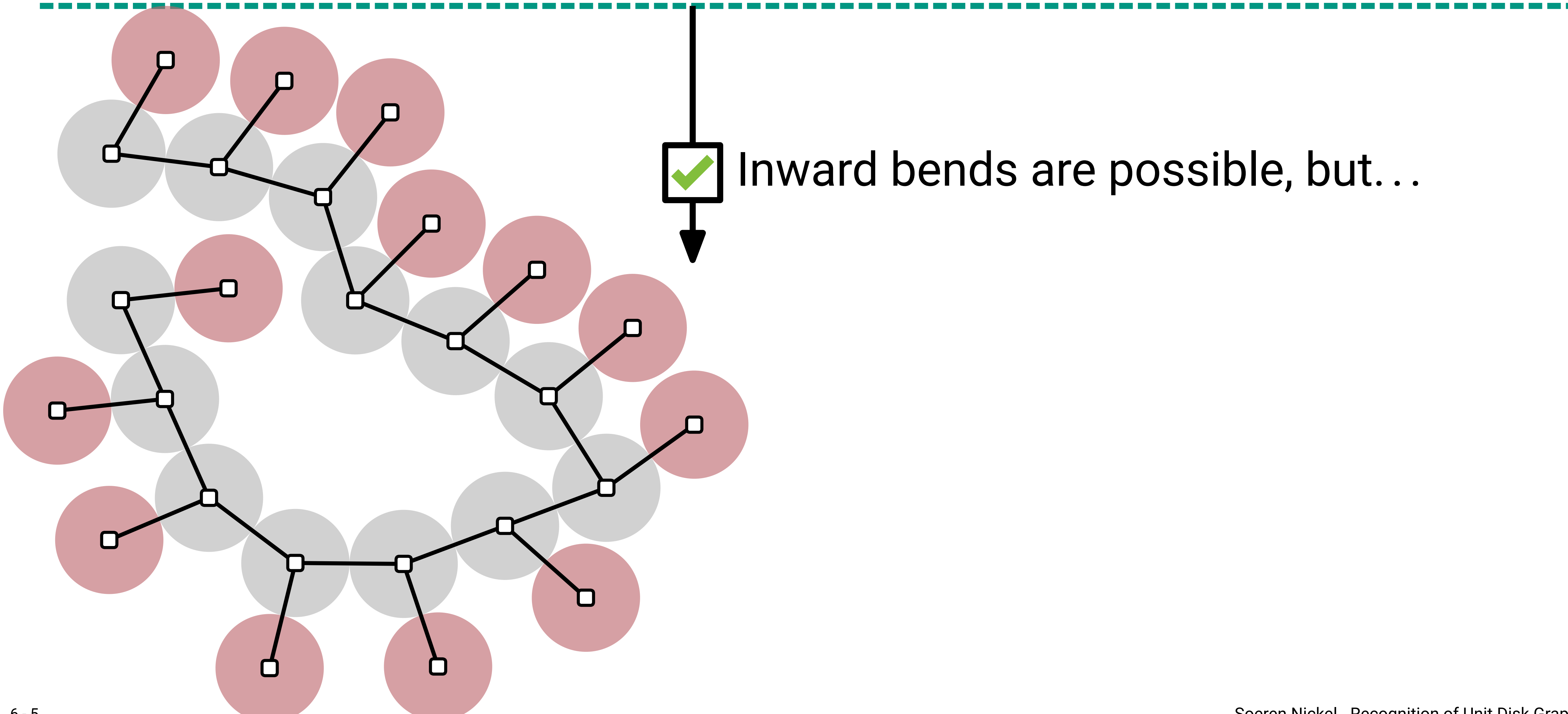
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)

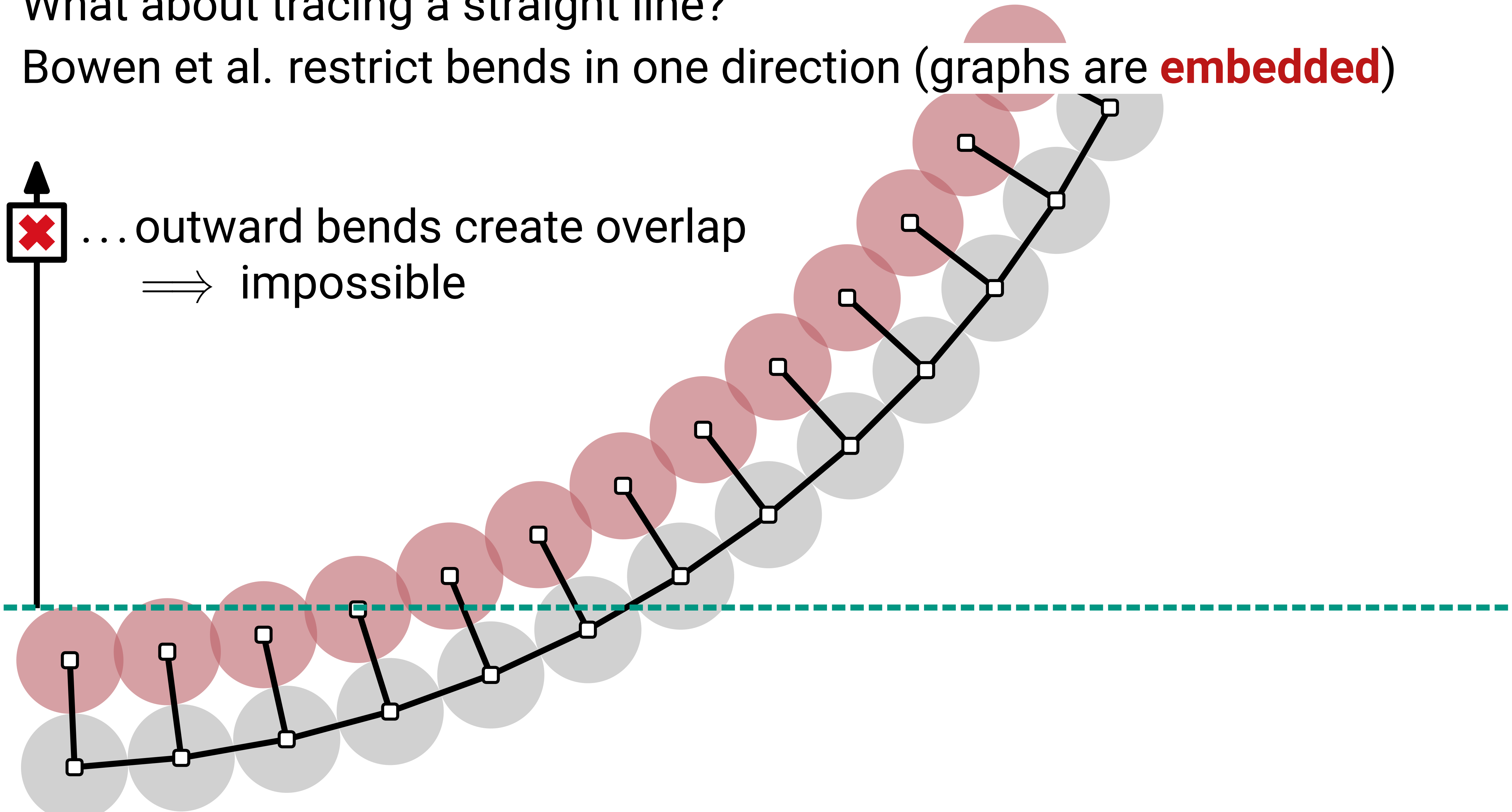


Inward bends are possible, but...

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



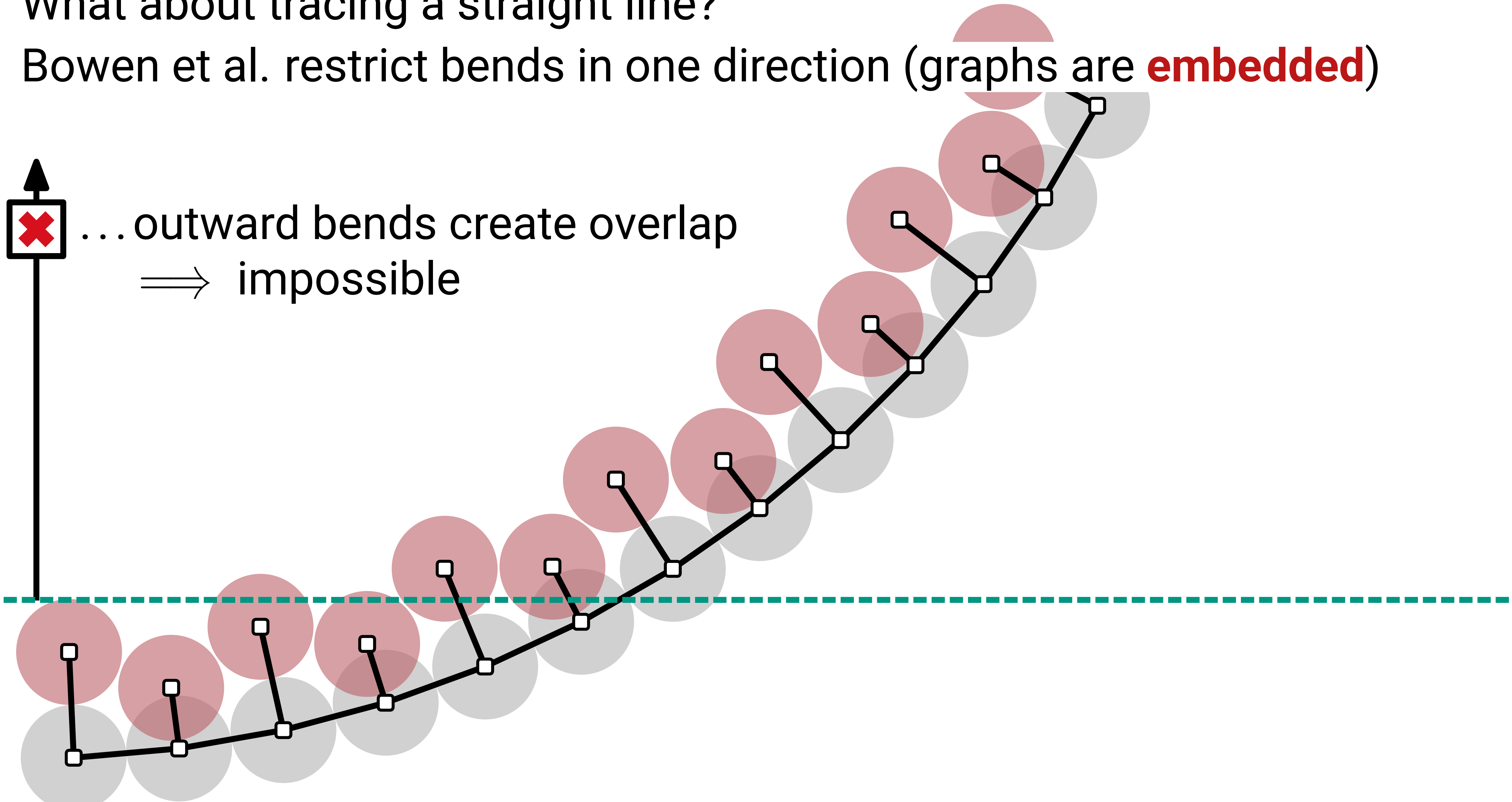
. . . outward bends create overlap

$\Longrightarrow$ impossible

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



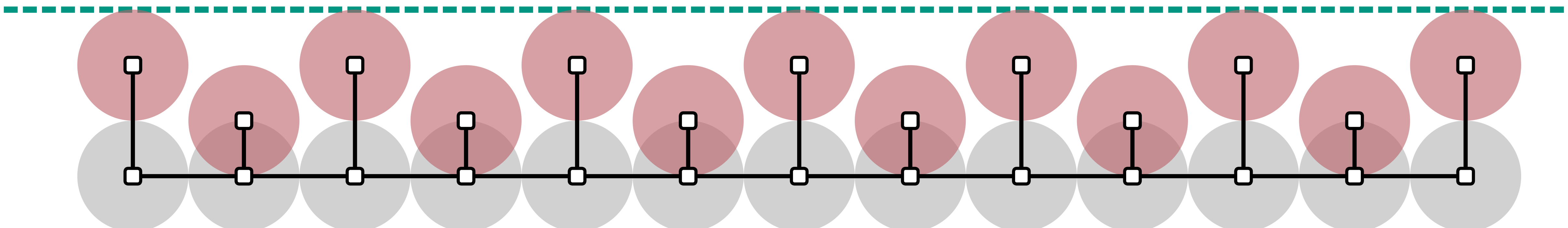
...outward bends create overlap

$\implies$ impossible

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



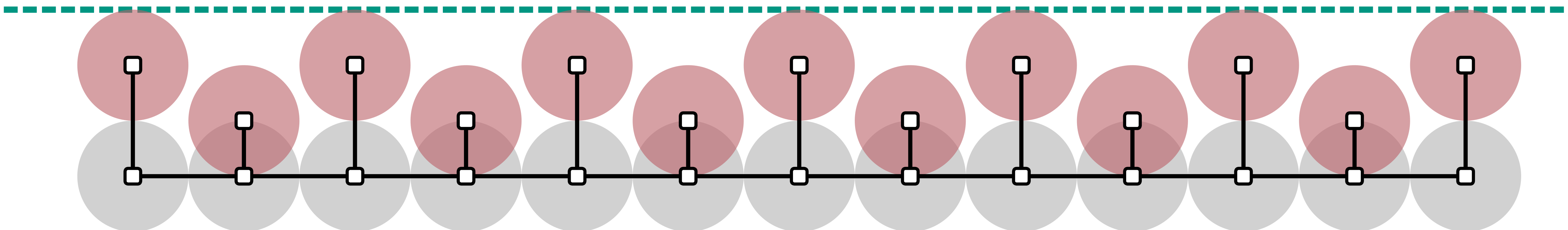
Does not work immediately in a **UDR**

Overlaps create space for outward bends

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

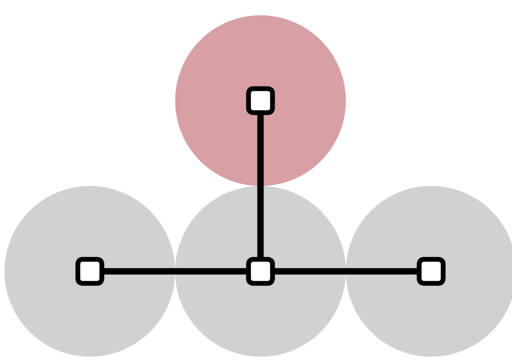
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Does not work immediately in a **UDR**

Overlaps create space for outward bends

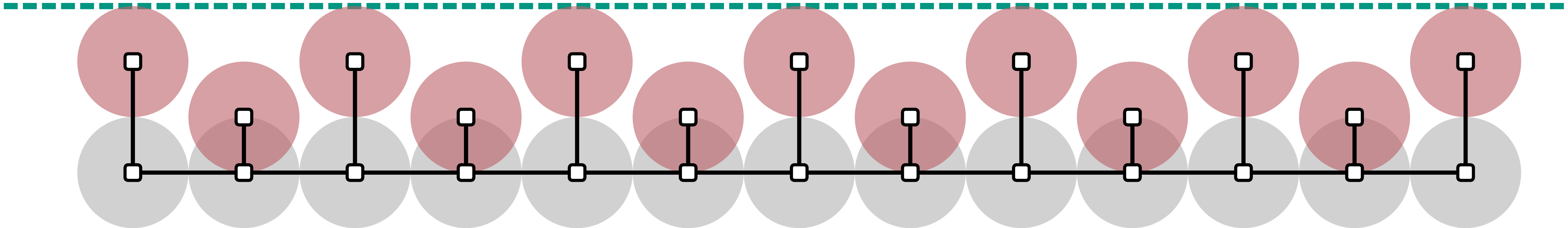
But non-connected disks need to be disjoint:



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

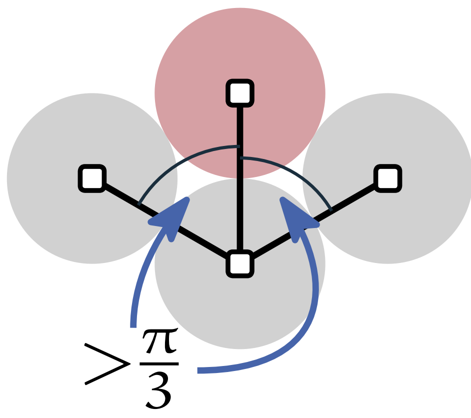
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Does not work immediately in a **UDR**

Overlaps create space for outward bends

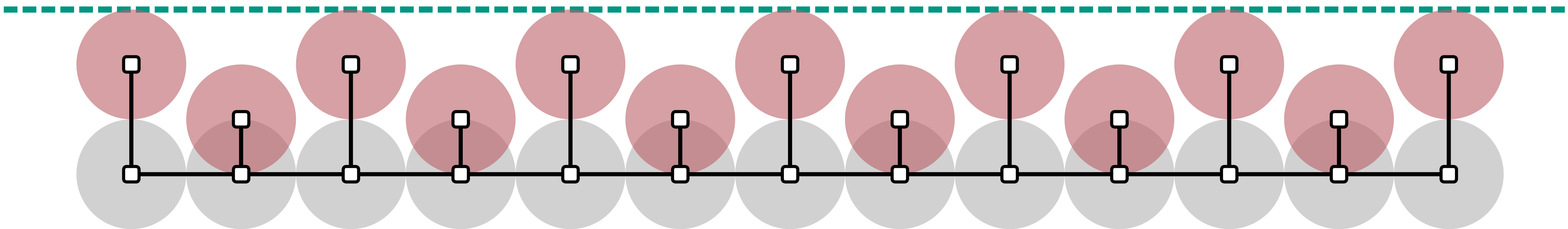
But non-connected disks need to be disjoint:



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

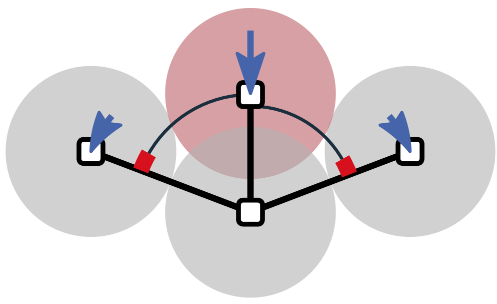
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Does not work immediately in a **UDR**

Overlaps create space for outward bends

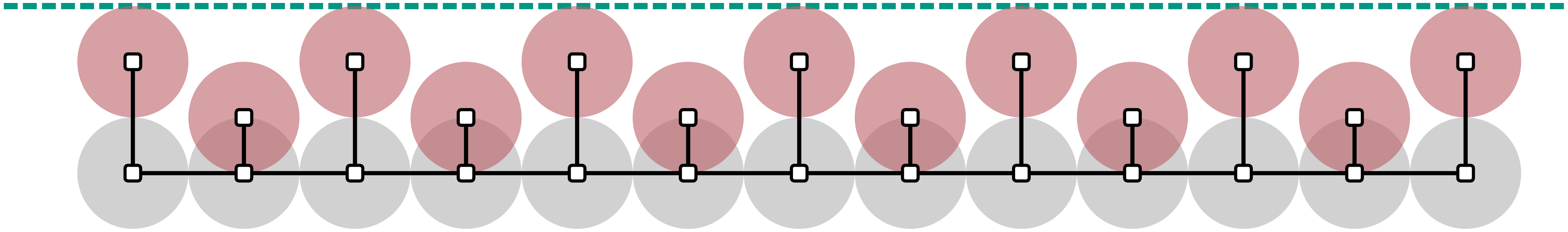
But non-connected disks need to be disjoint:



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

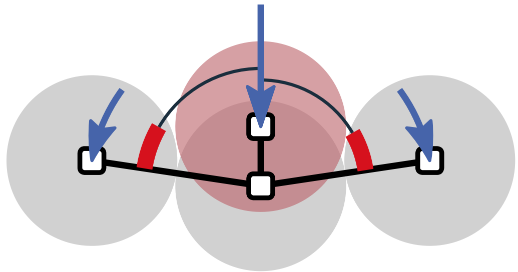
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Does not work immediately in a **UDR**

Overlaps create space for outward bends

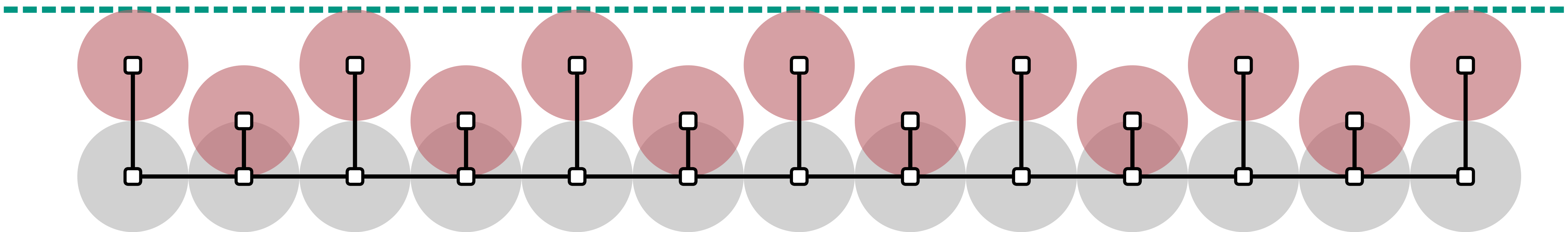
But non-connected disks need to be disjoint:



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

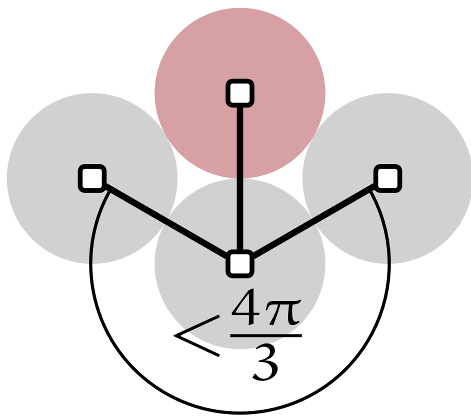
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Does not work immediately in a **UDR**

Overlaps create space for outward bends

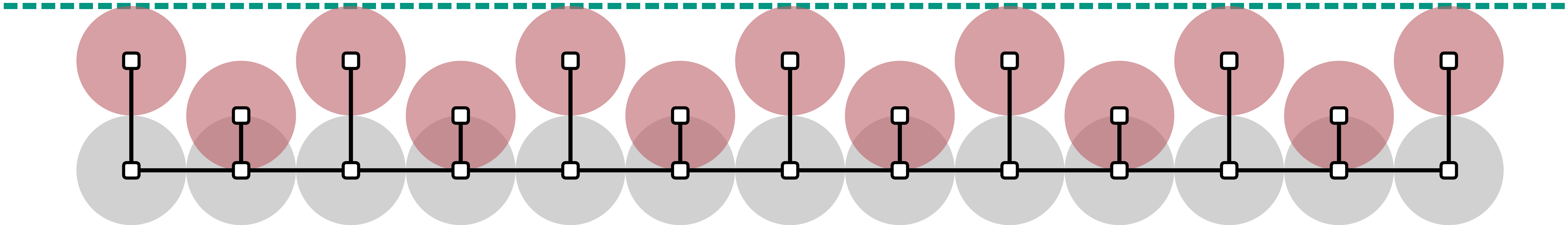
But non-connected disks need to be disjoint:



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

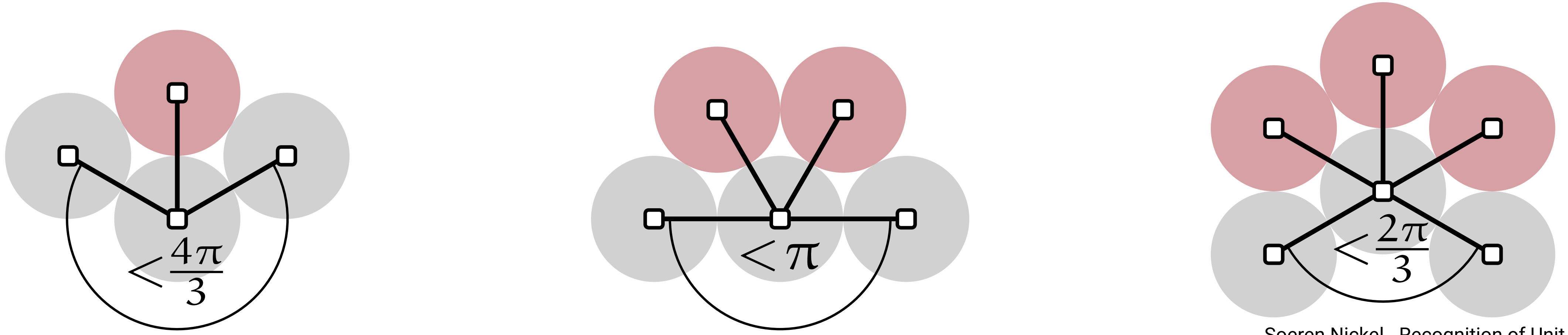
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Does not work immediately in a **UDR**

Overlaps create space for outward bends

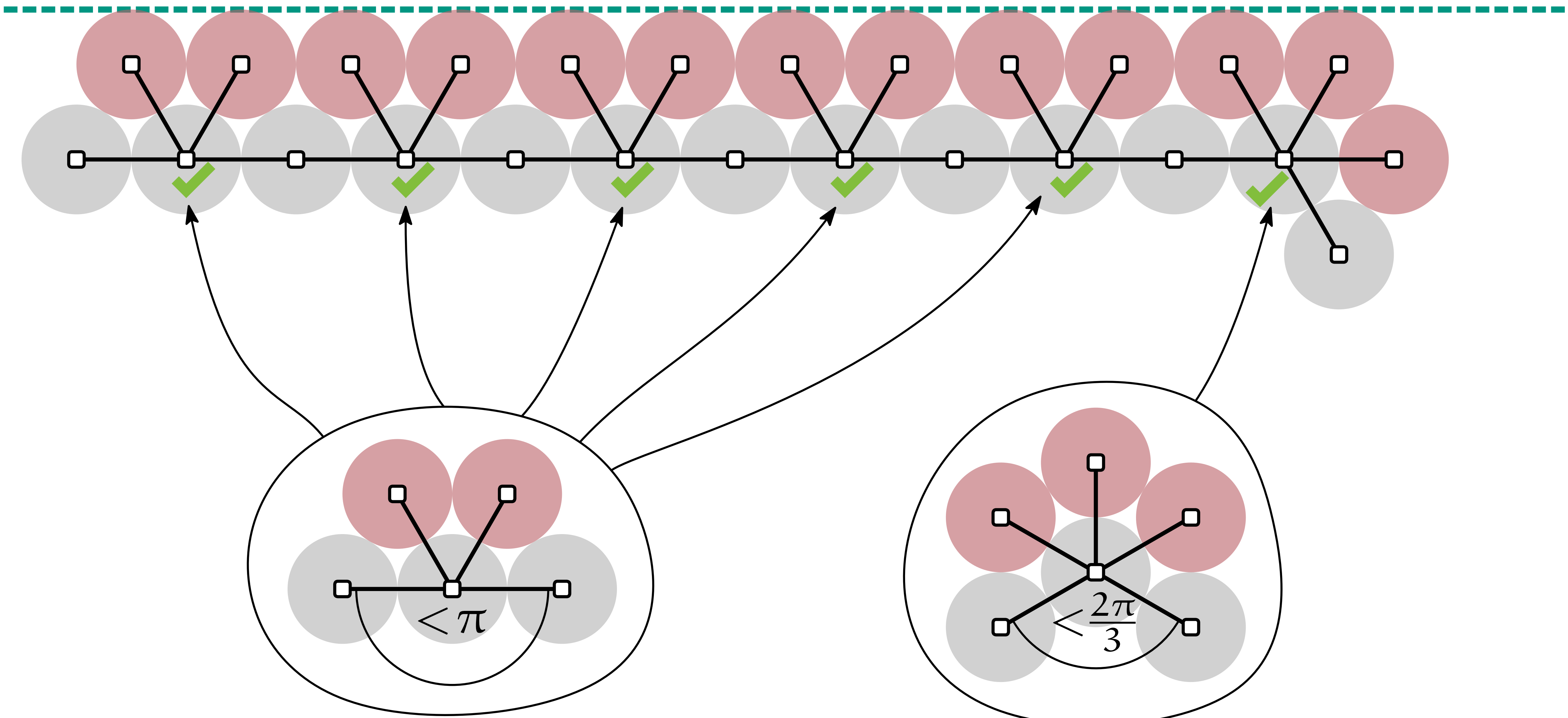
But non-connected disks need to be disjoint:



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

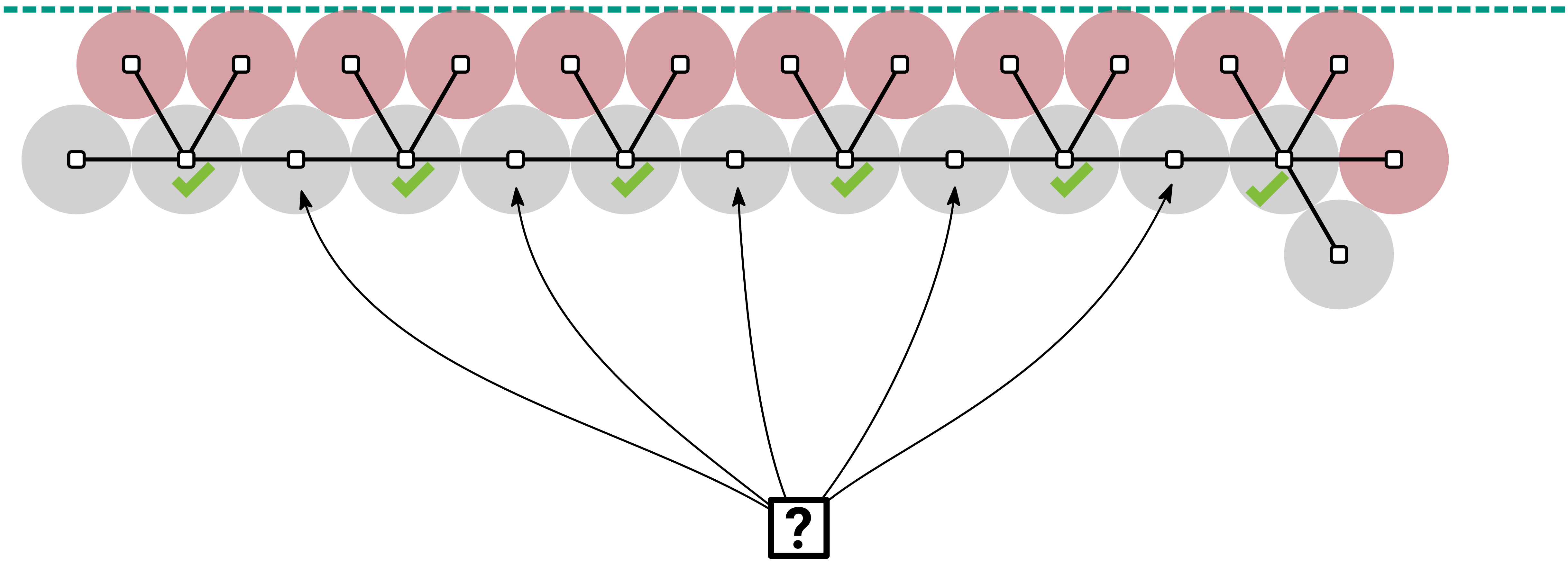
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

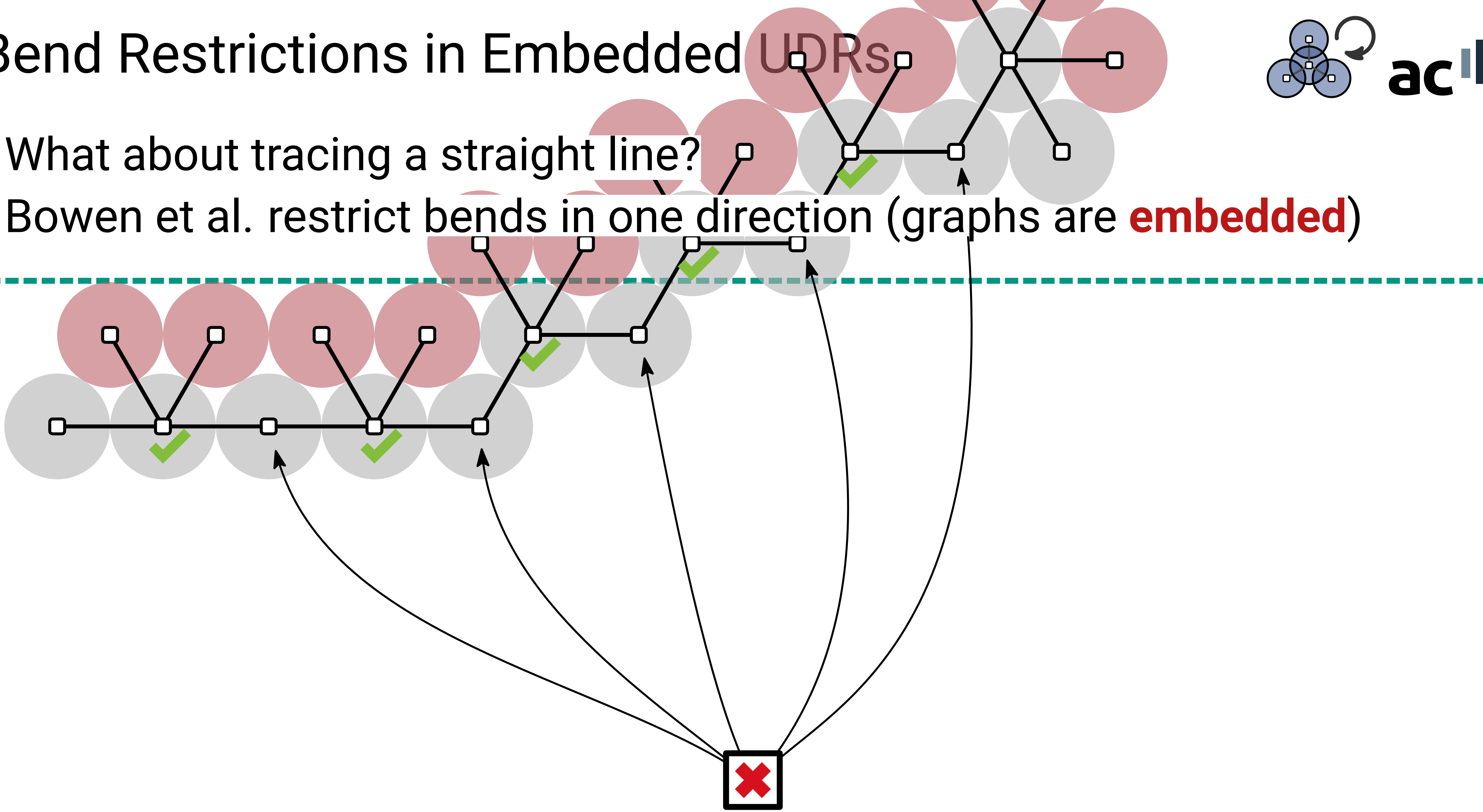
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)

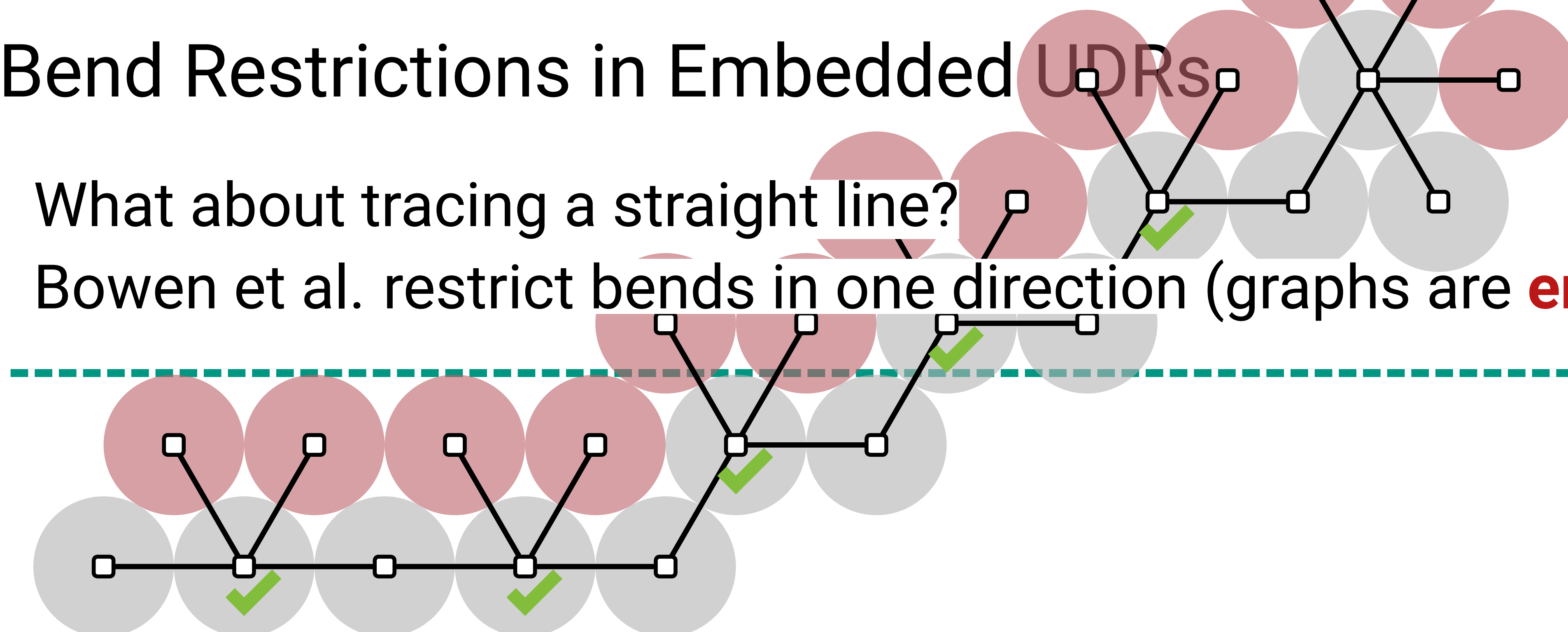


We need additional restricting neighbors

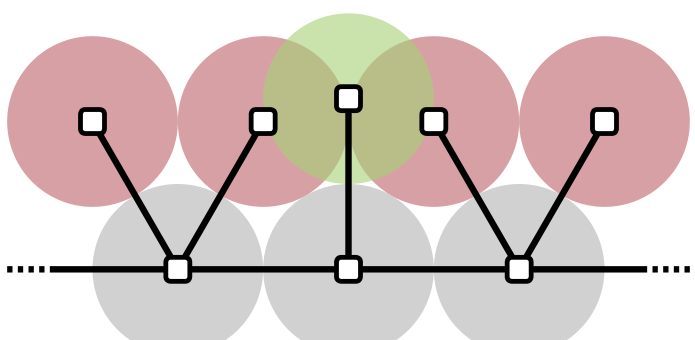
Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



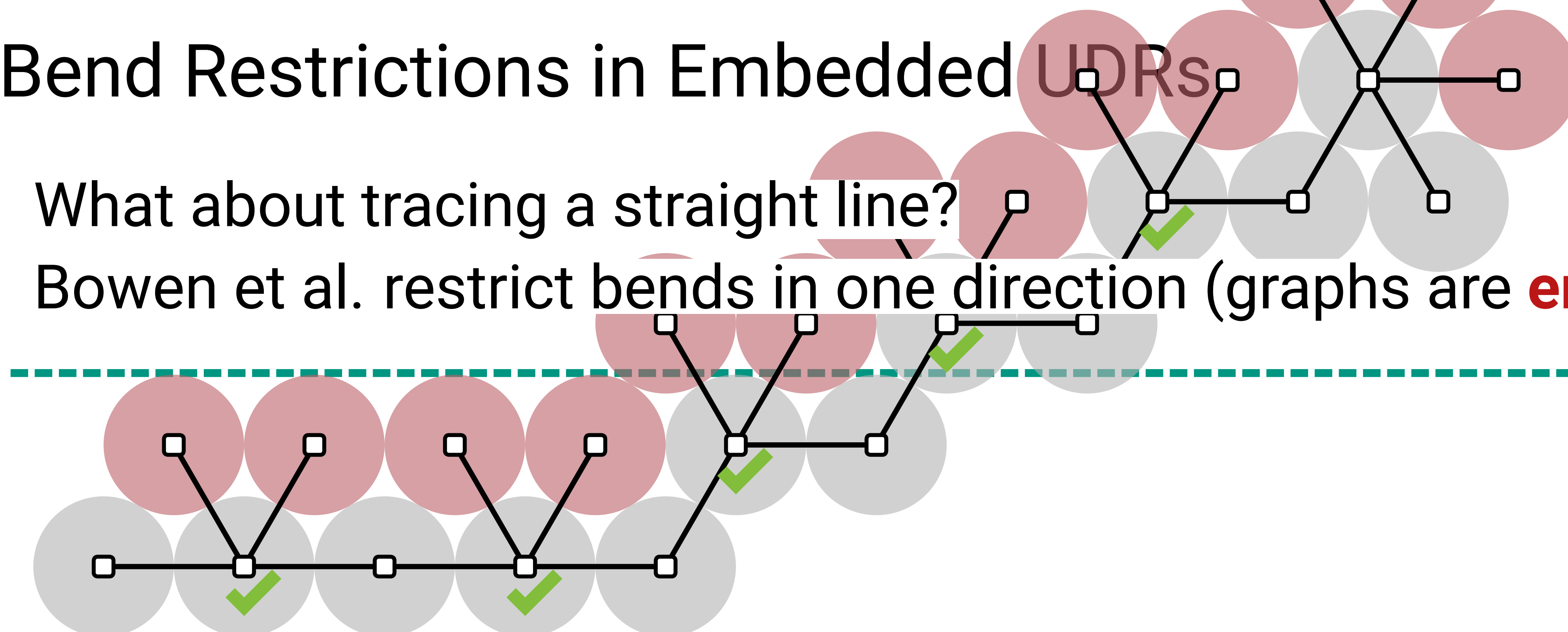
We can force nearly complete overlap with a **single disk** at these nodes



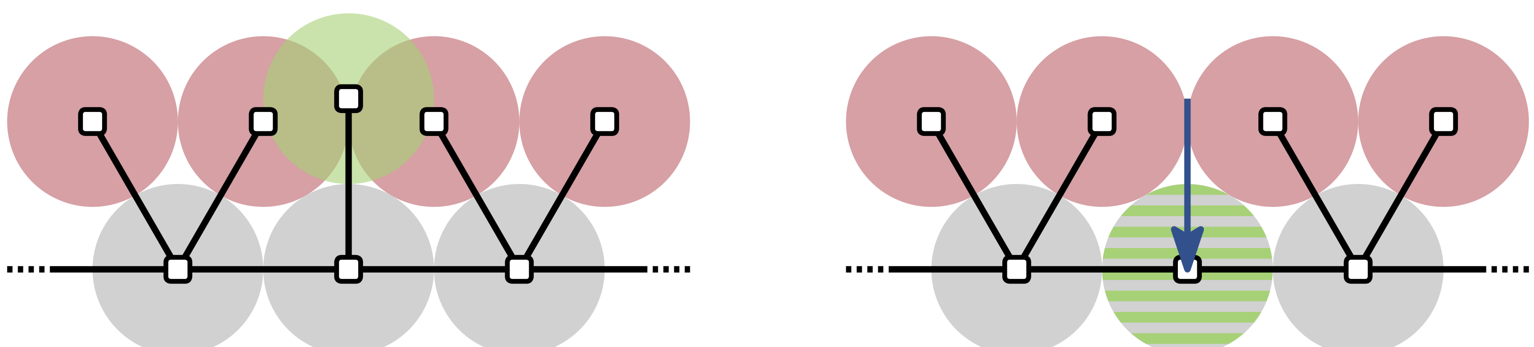
Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



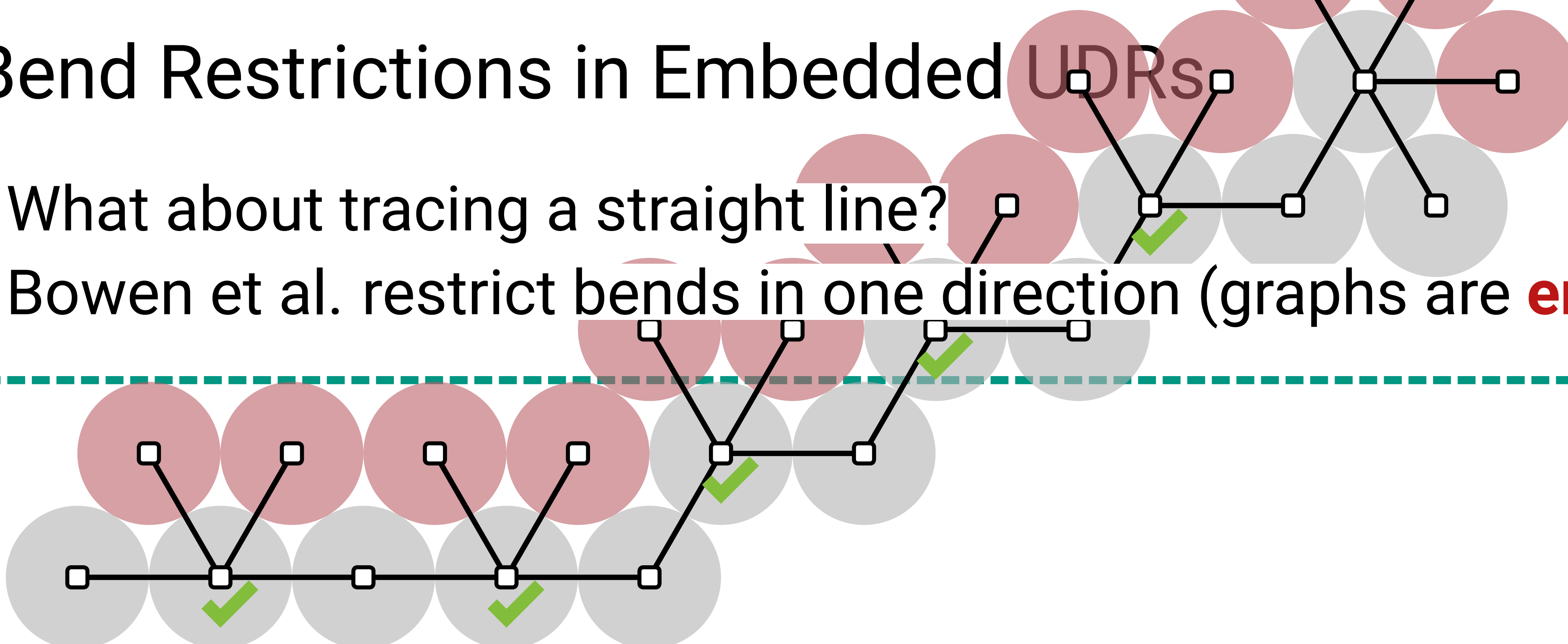
We can force nearly complete overlap with a **single disk** at these nodes



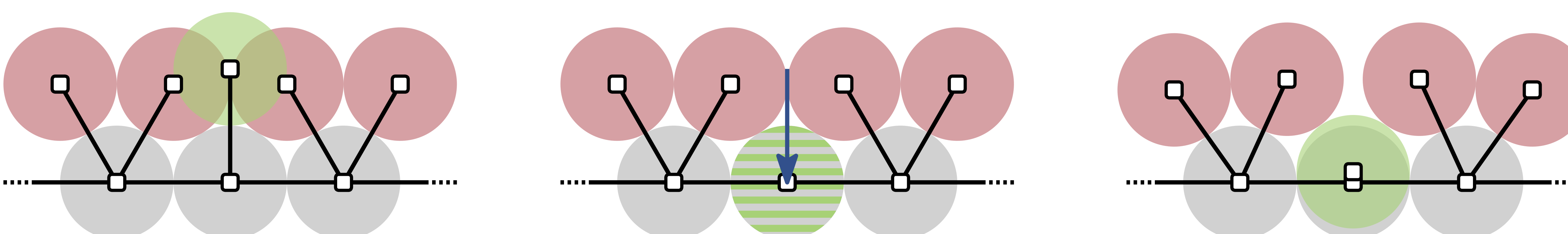
Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



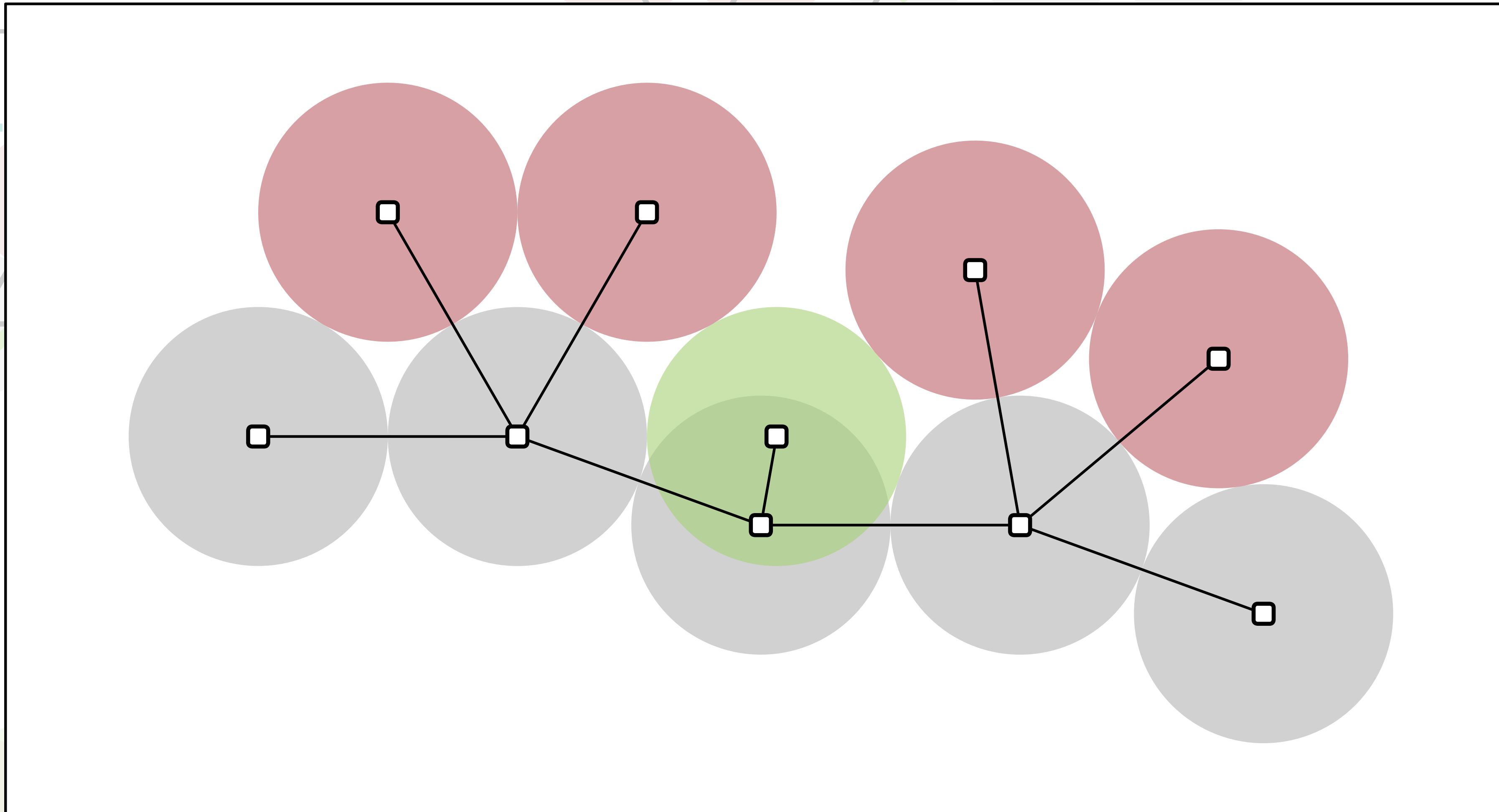
We can force nearly complete overlap with a **single disk** at these nodes



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen e ed)



We can fo nodes

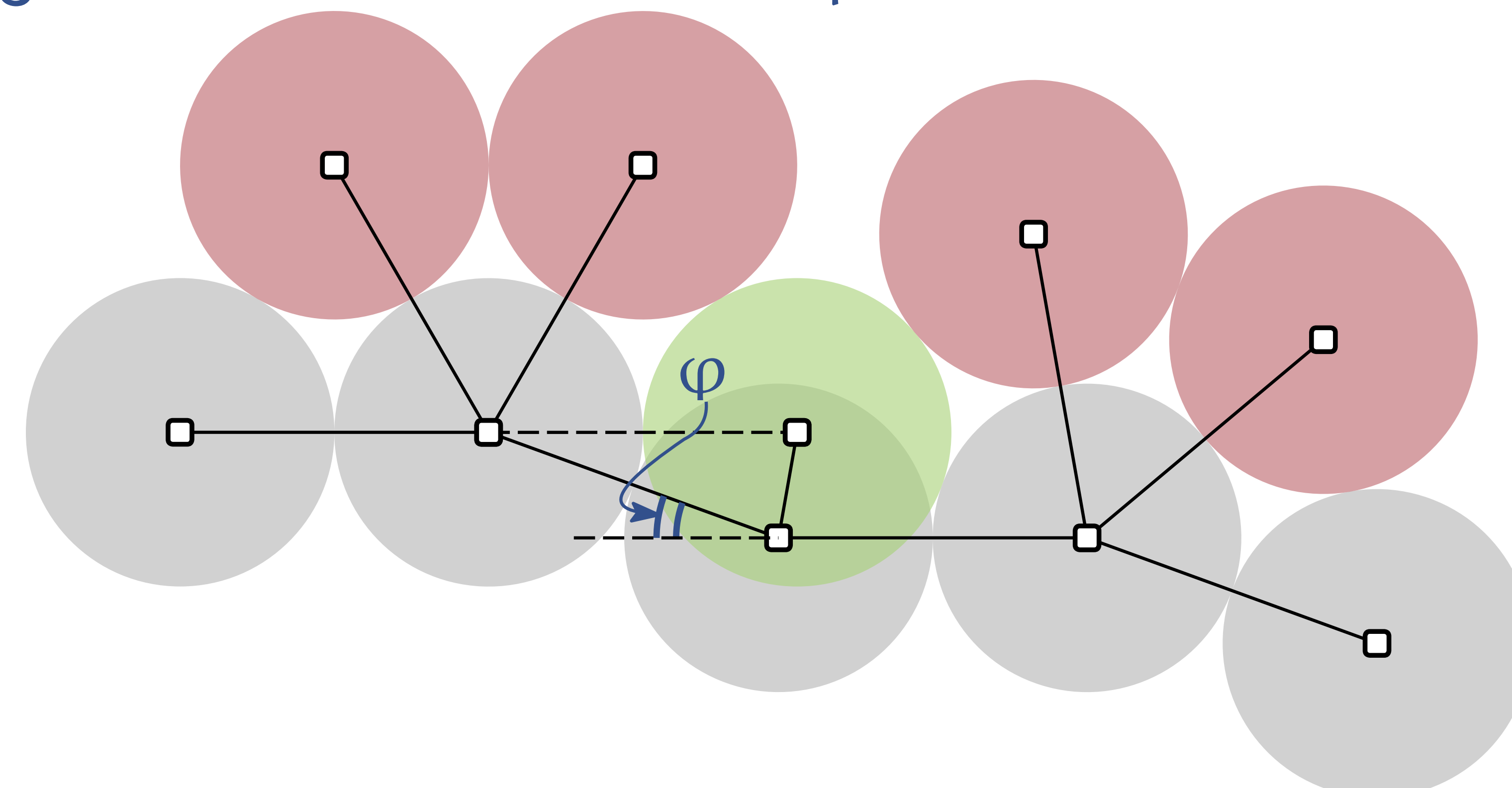
Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen e ... ed)

We can fo ... nodes

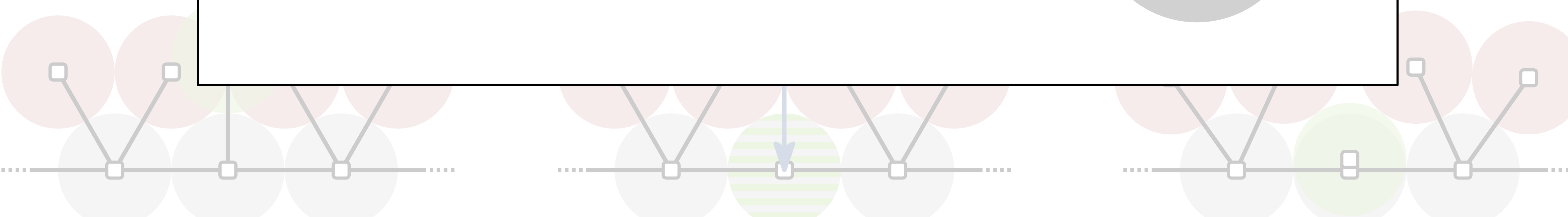
To gain an outward bend of φ...



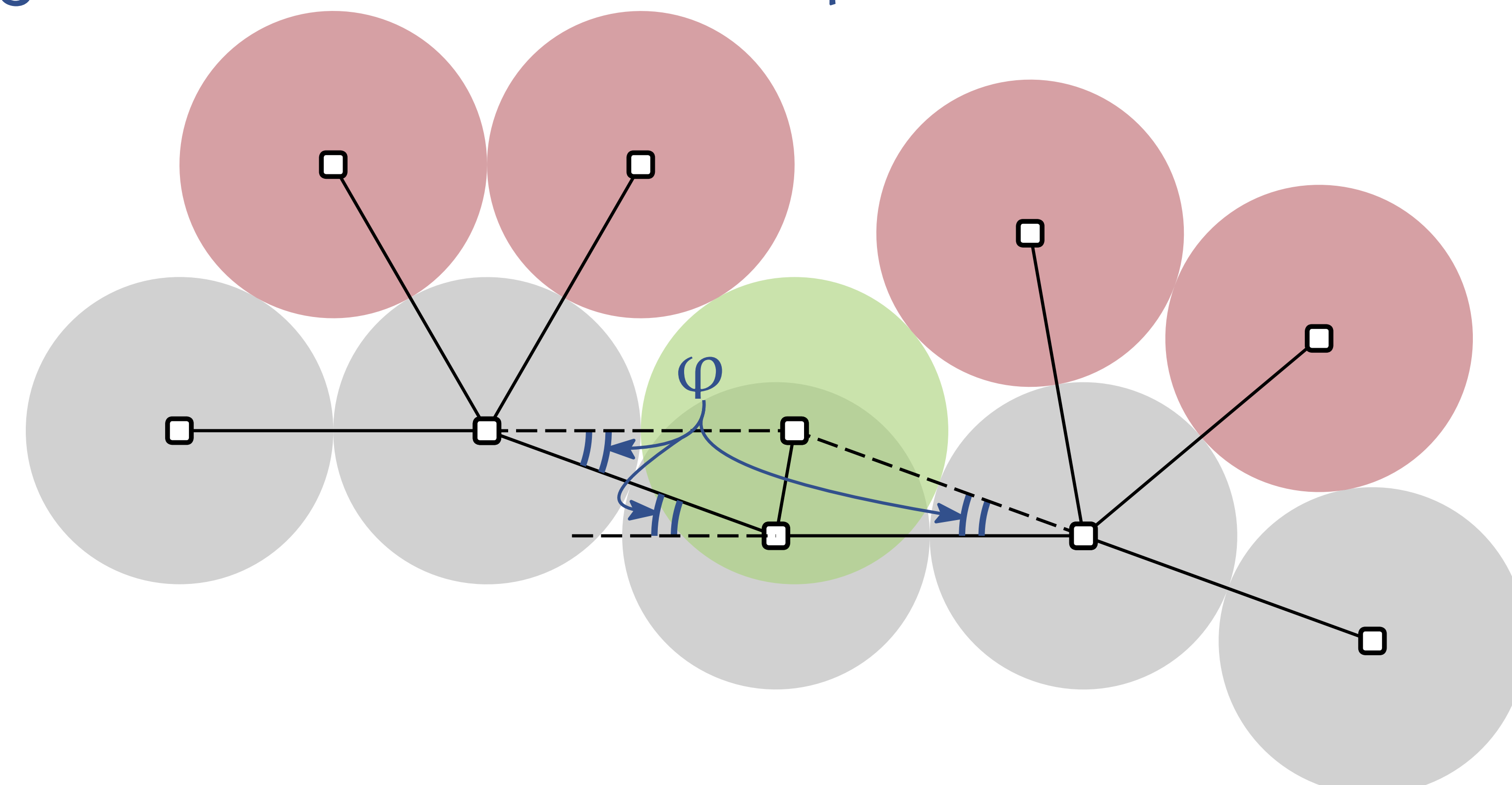
What about tracing a straight line?

Bowen e ... ed)

We can fo ... nodes



To gain an outward bend of φ...

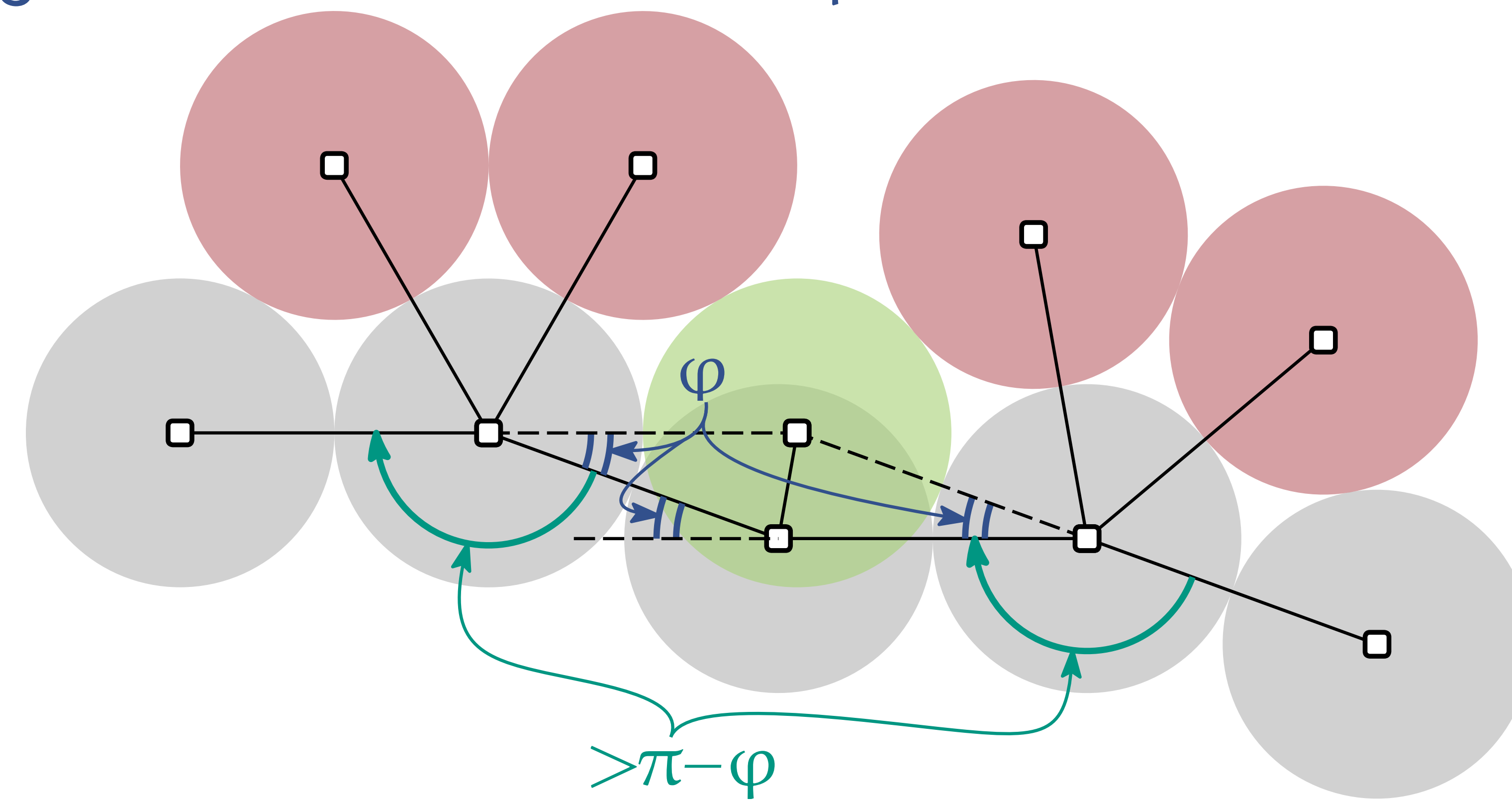


Bend Restrictions in Embedded UDRs

What about tracing a straight line?

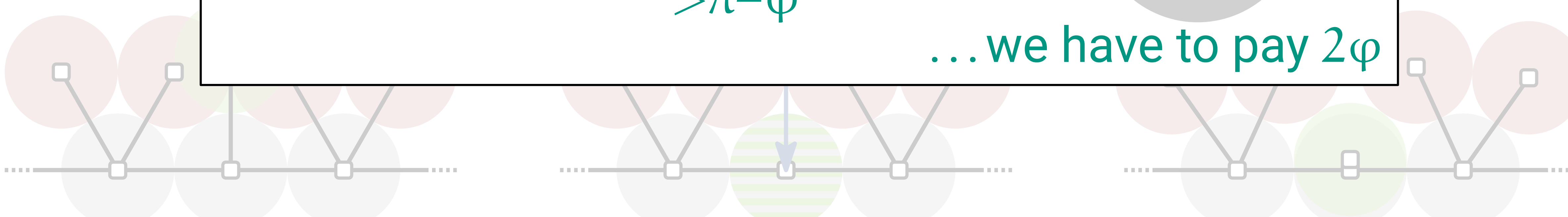
Bowen e… …ed)

To gain an outward bend of φ…



…we have to pay 2φ

We can fo… …nodes

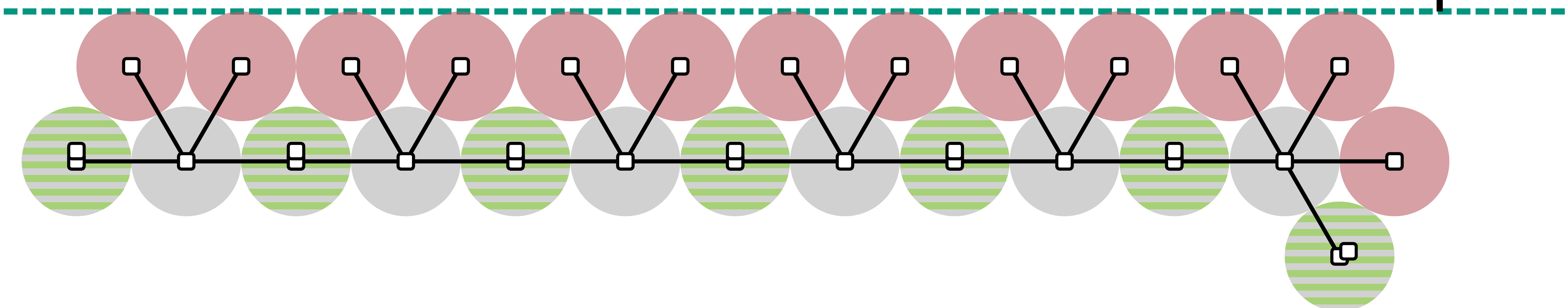


Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)

Outward bends are impossible

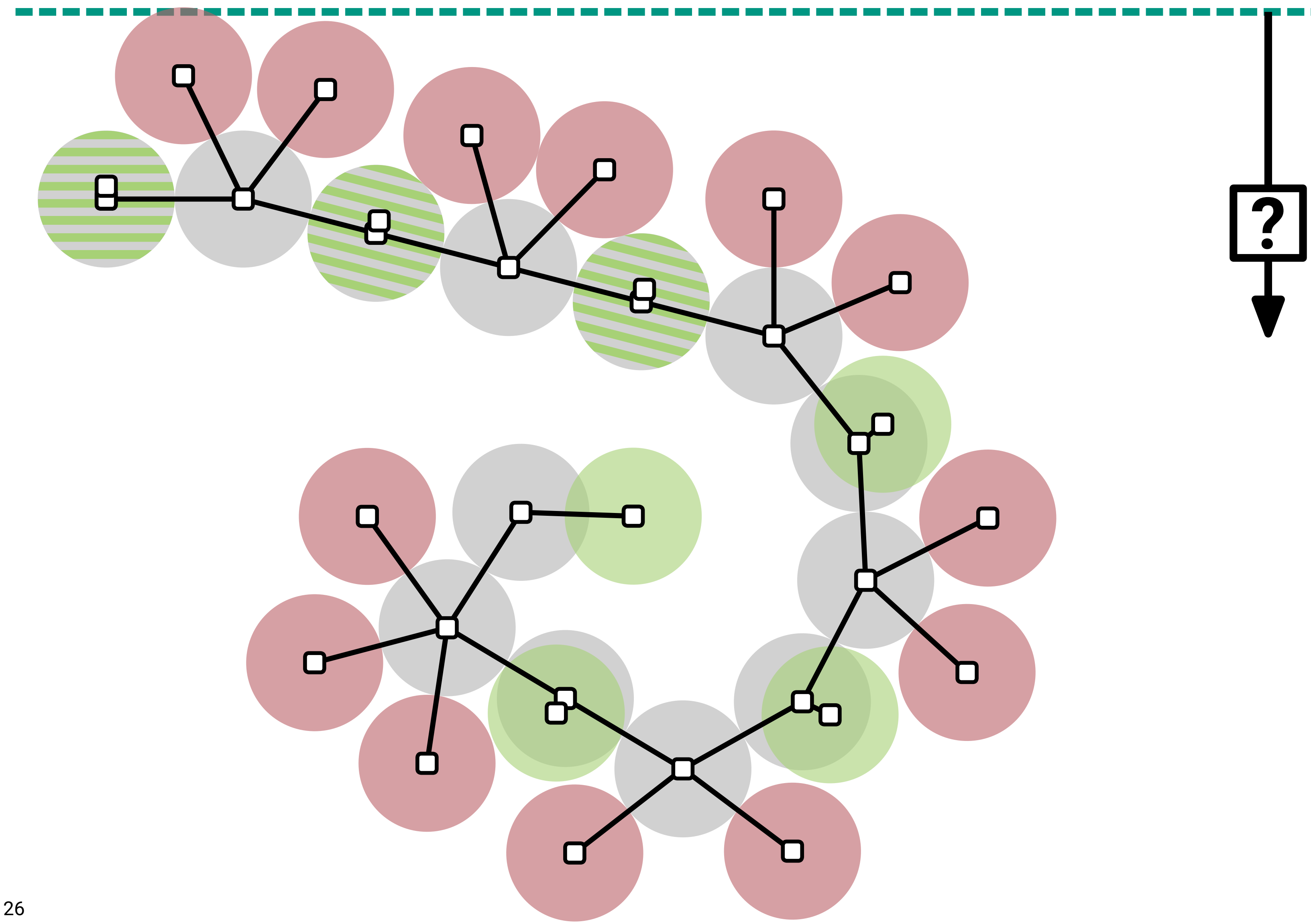


Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)

Outward bends are impossible

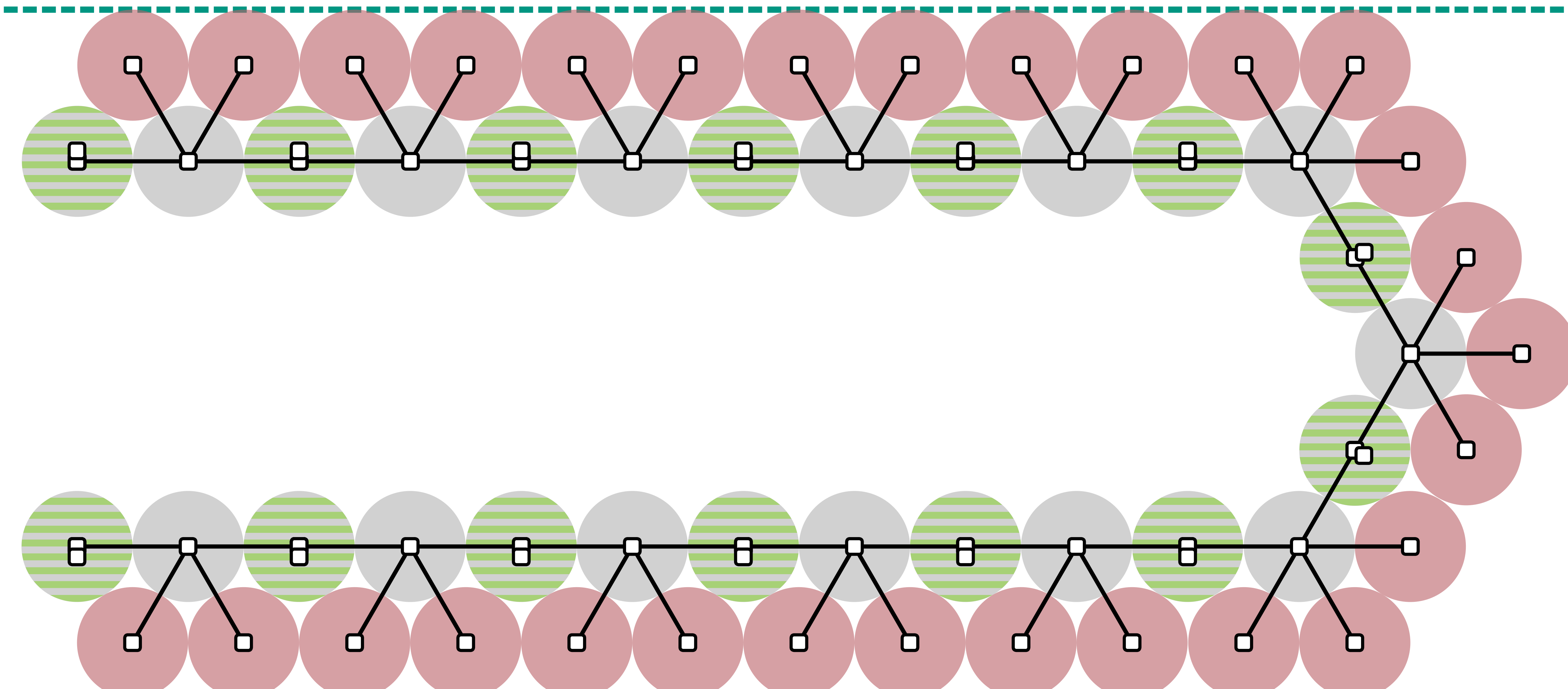


How do we treat inward flexibility?

Bend Restrictions in Embedded UDRs

What about tracing a straight line?

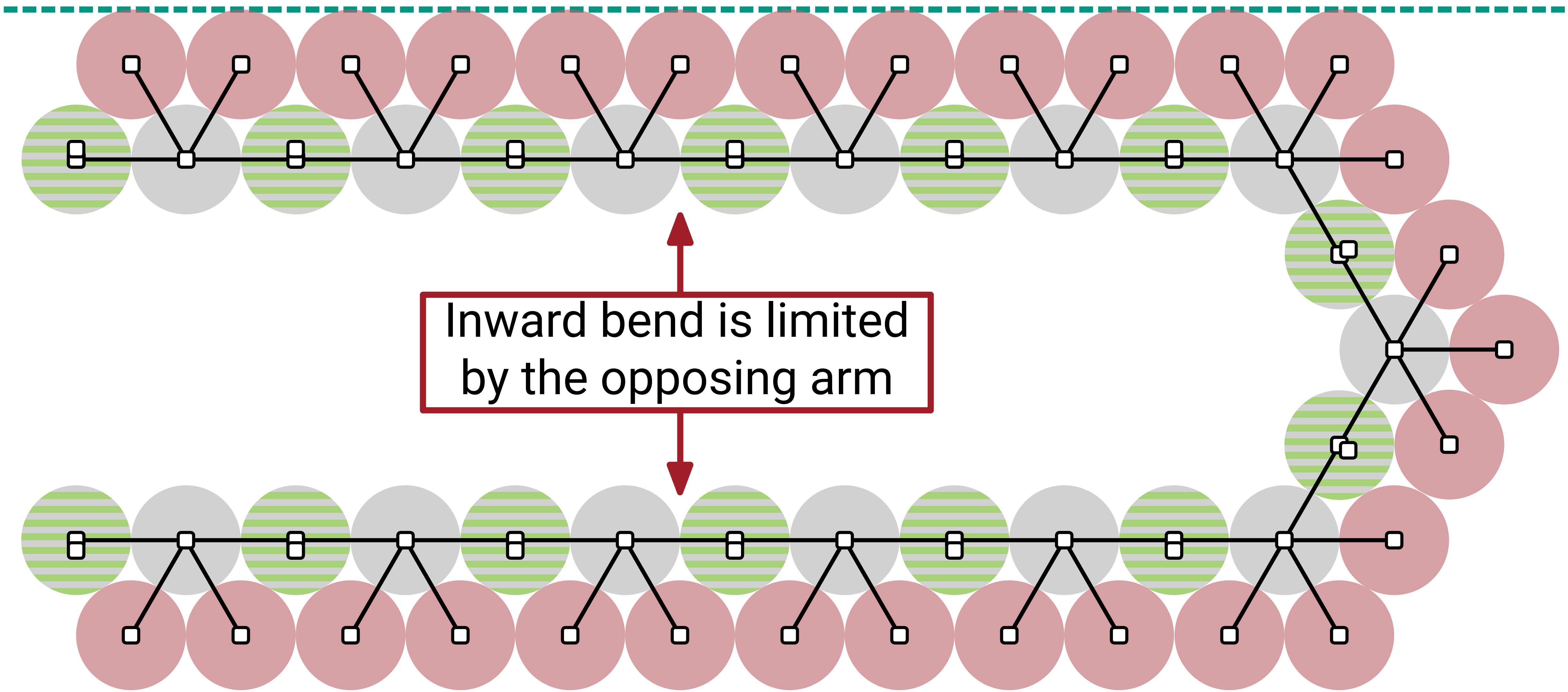
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

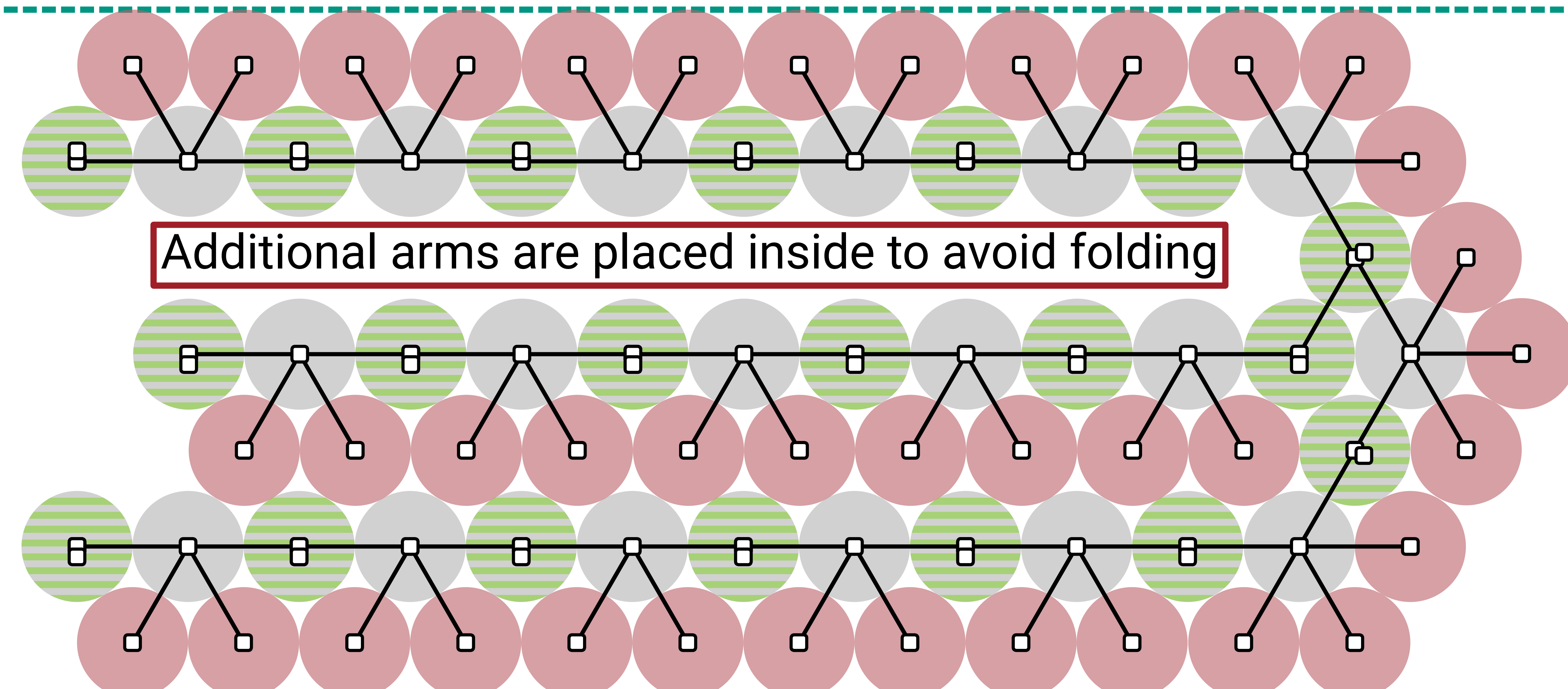
Bowen et al. restrict bends in one direction (graphs are **embedded**)



Bend Restrictions in Embedded UDRs

What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



Bend Restrictions in Embedded UDRs

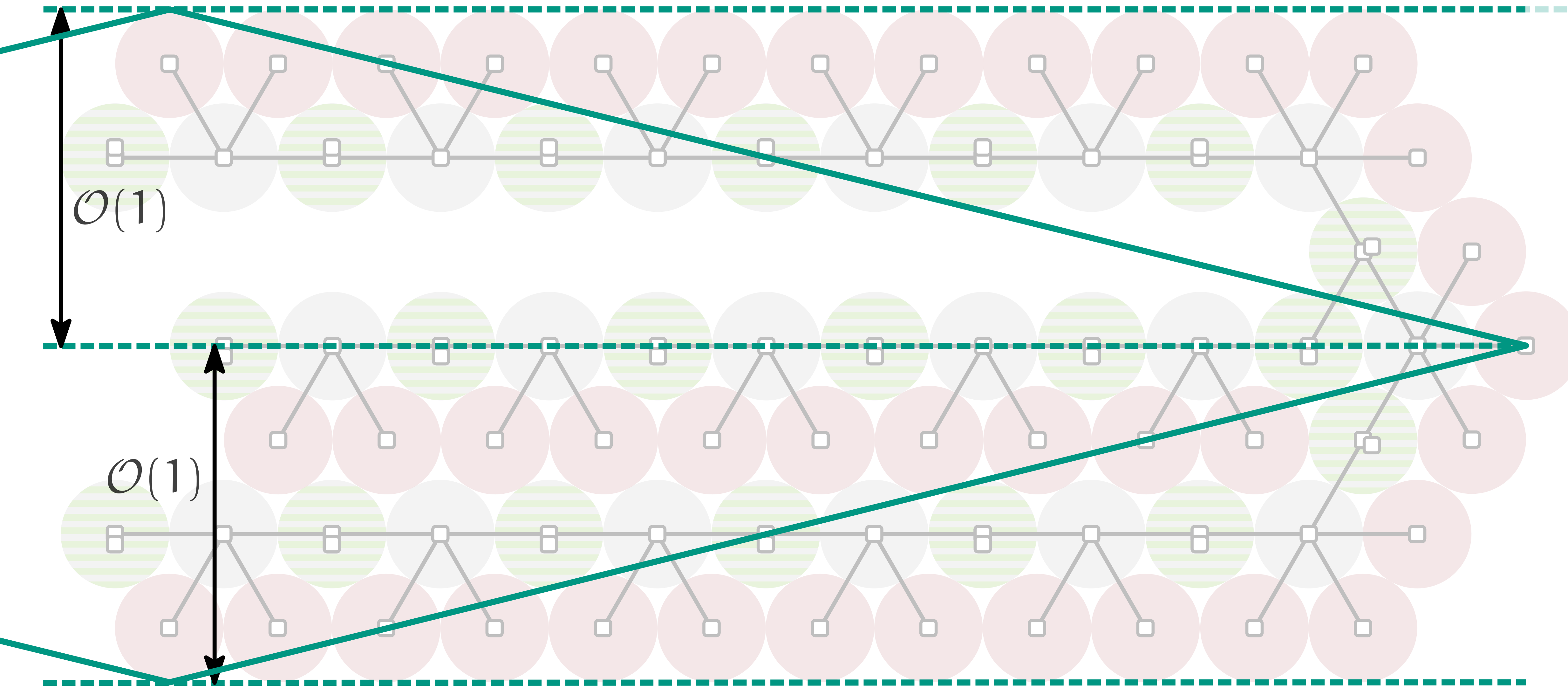
What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)

Bend Restrictions in Embedded UDRs

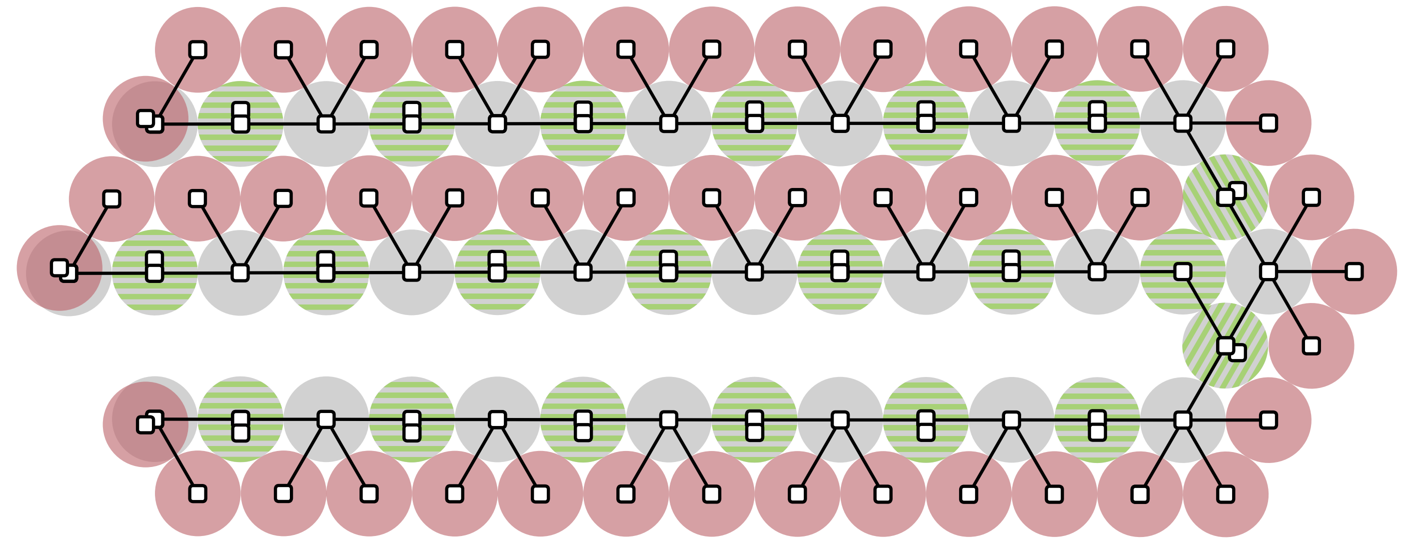
What about tracing a straight line?

Bowen et al. restrict bends in one direction (graphs are **embedded**)



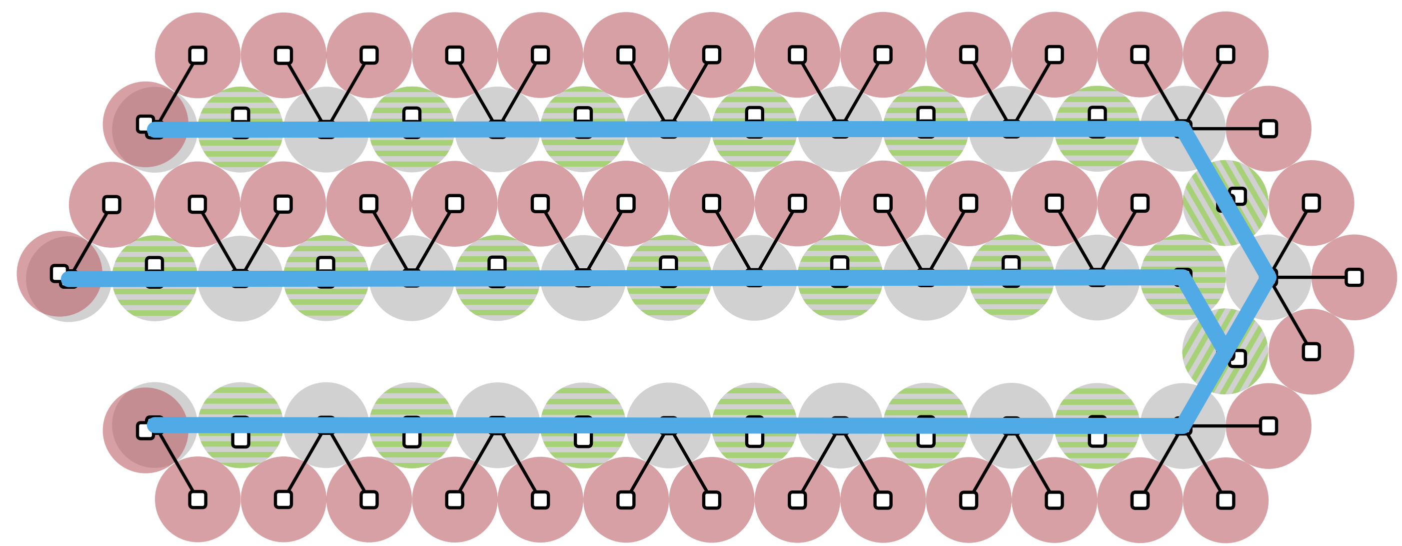
λ-Stable Hexagons Approximation

$\mathcal{O}(1)$-stable approximation of
a long thin rhombus



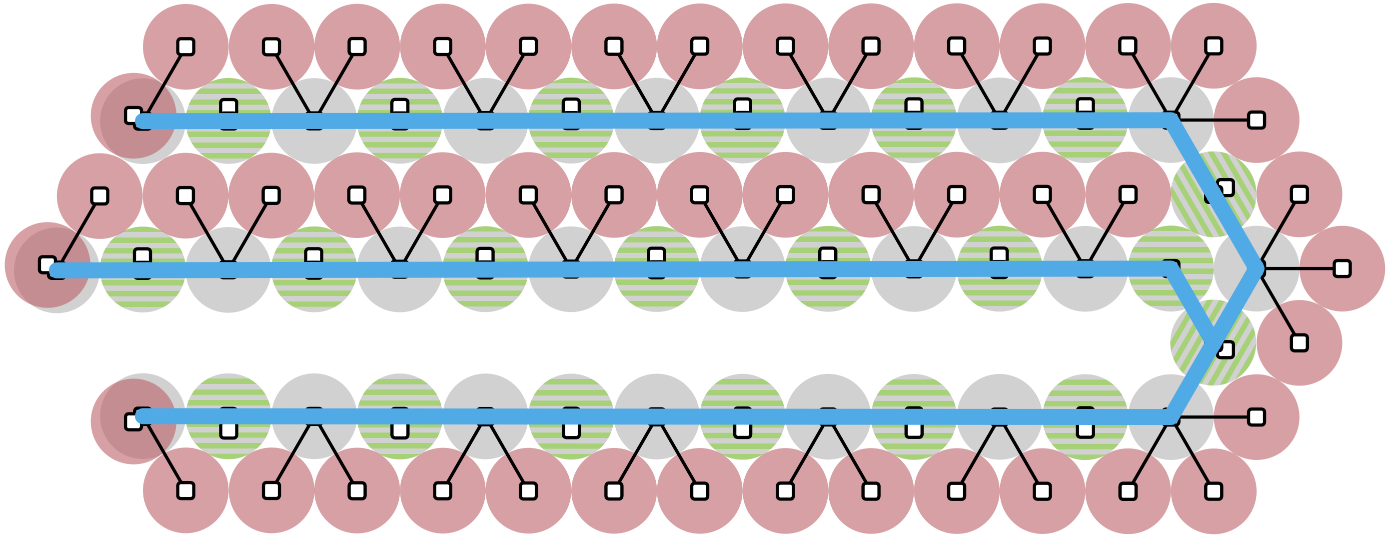
λ-Stable Hexagons Approximation

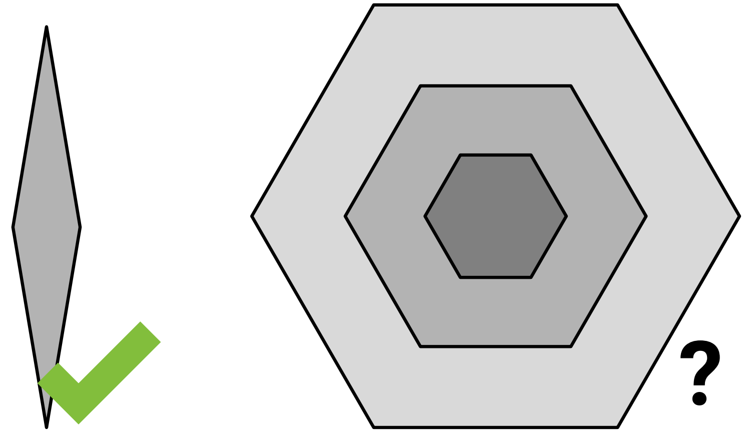
$\mathcal{O}(1)$-stable approximation of
a long thin rhombus



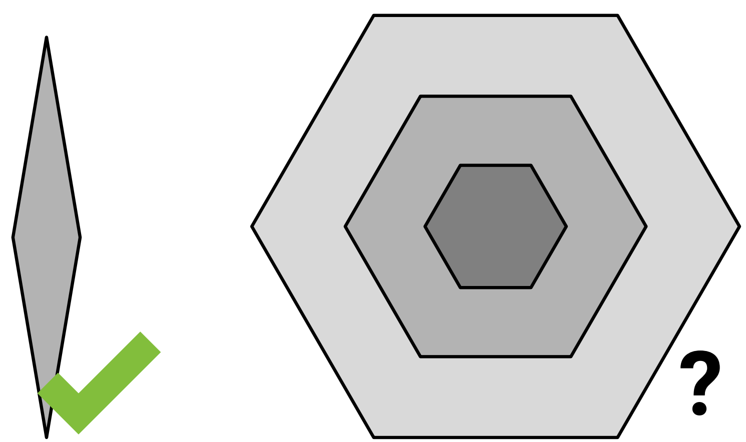
λ-Stable Hexagons Approximation

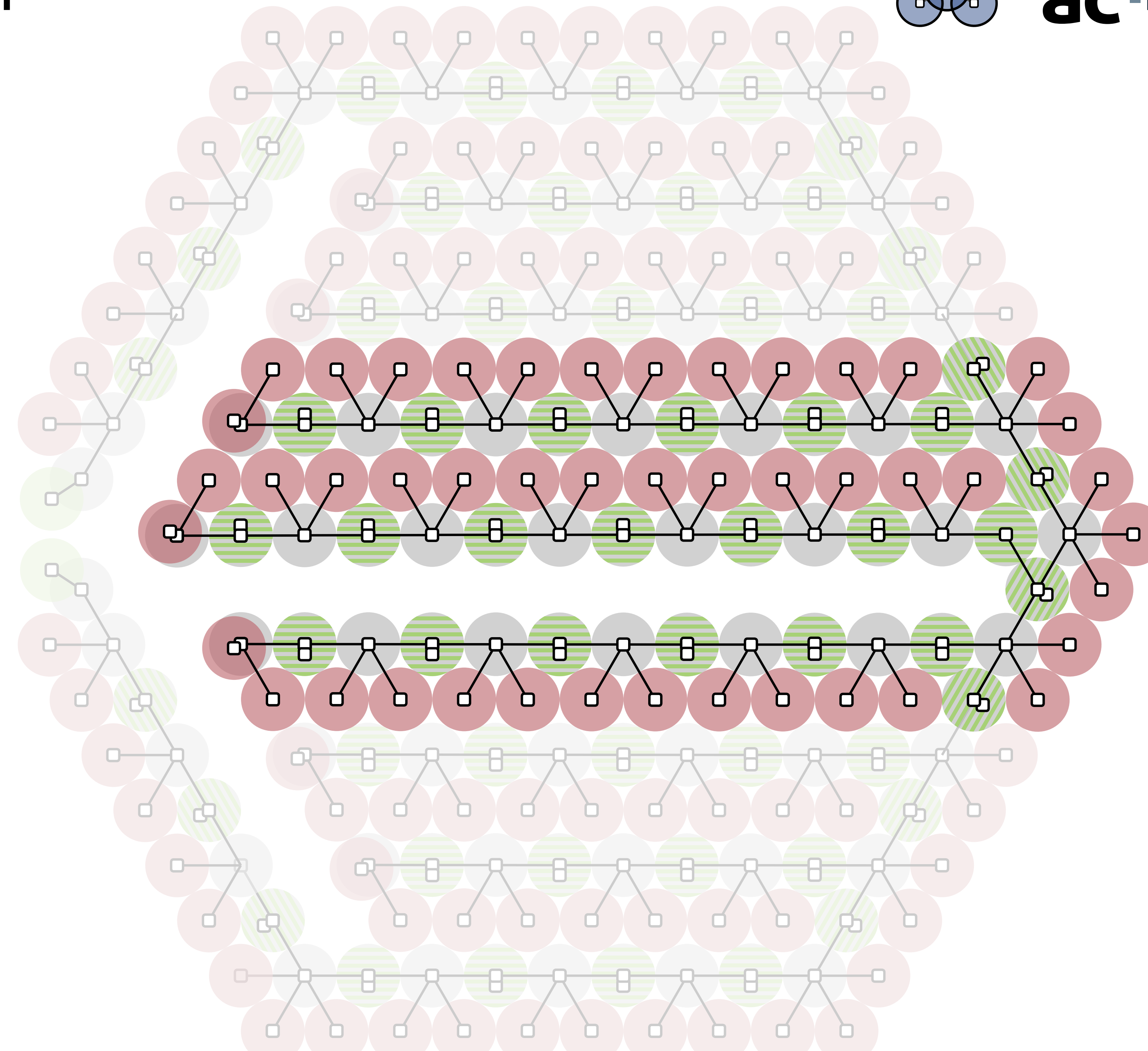
$\mathcal{O}(1)$-stable approximation of a long thin rhombus





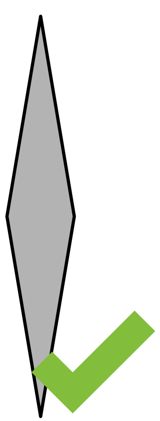
λ-Stable Hexagons Approximation

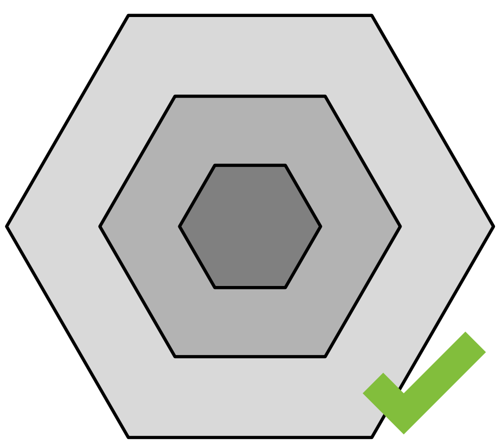


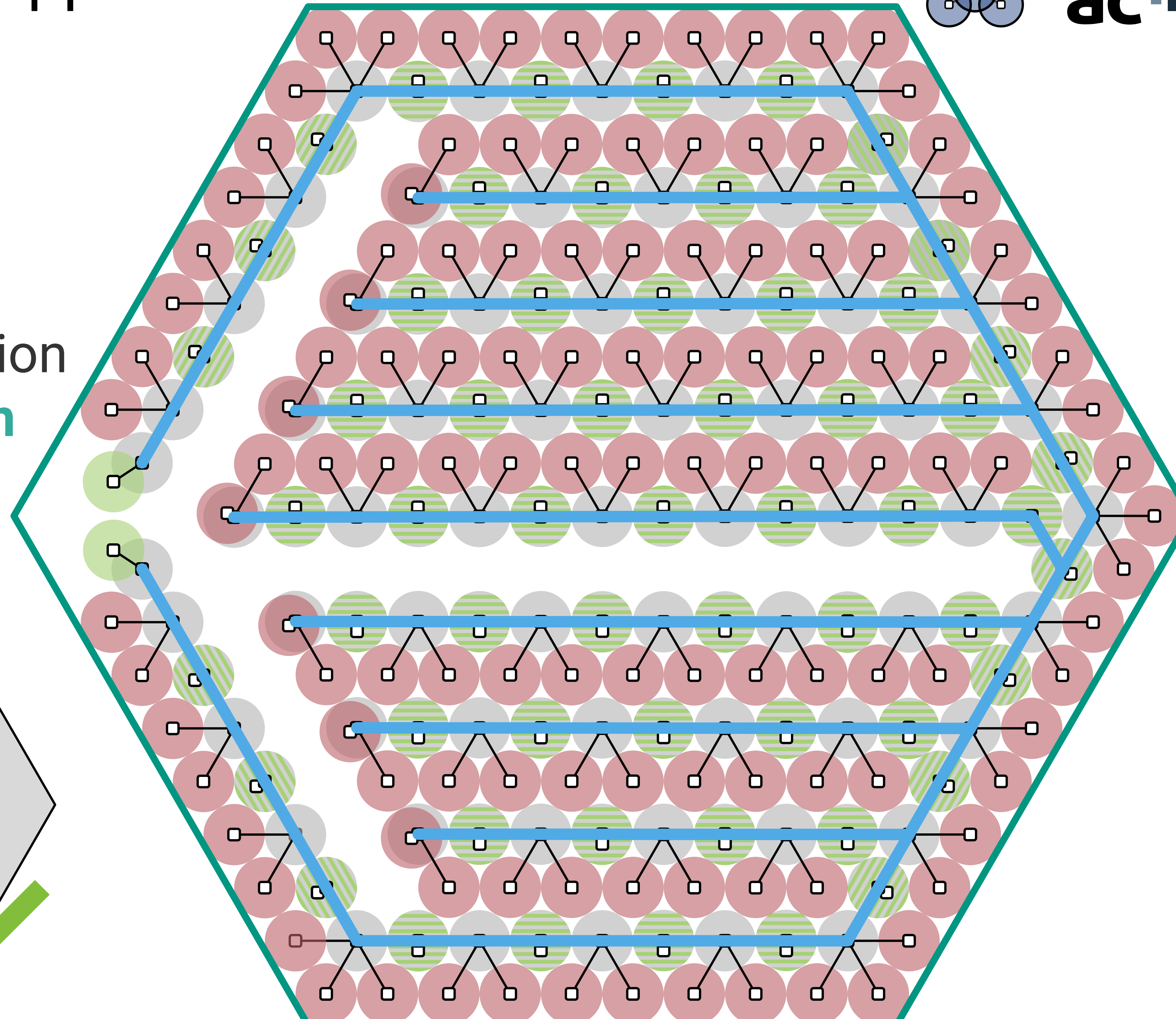


λ-Stable Hexagons Approximation

$\mathcal{O}(1)$-stable approximation
of a **scalable hexagon**







Hardness Results and Outerplanarity

Theorem 1. *Recognizing unit disk graphs is* NP*-hard for embedded trees.*

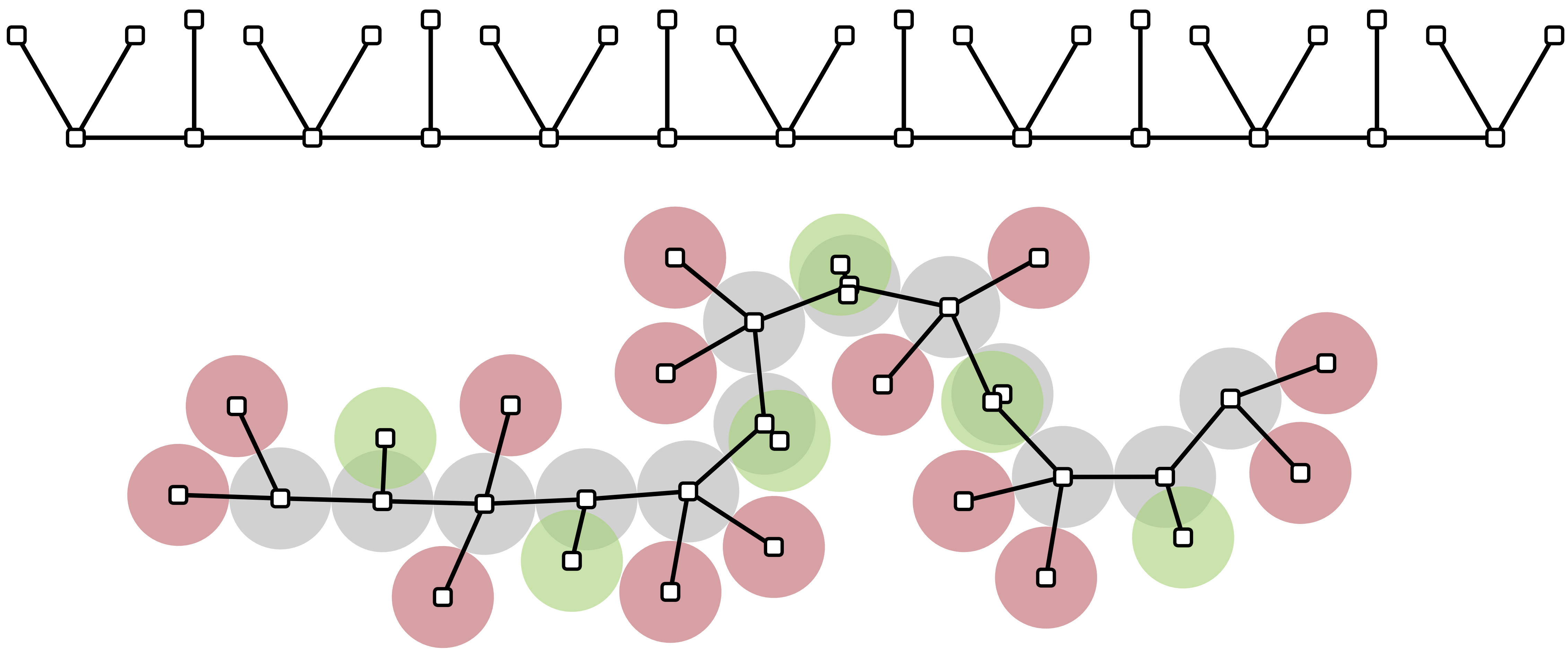
Hardness Results and Outerplanarity

Theorem 1. *Recognizing unit disk graphs is* NP-*hard for embedded trees.*

What about **non-embedded** graphs?

Theorem 1. *Recognizing unit disk graphs is* NP-*hard for embedded trees.*

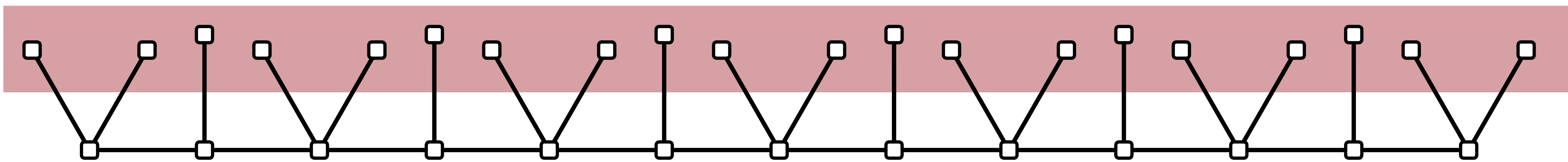
What about **non-embedded** graphs?

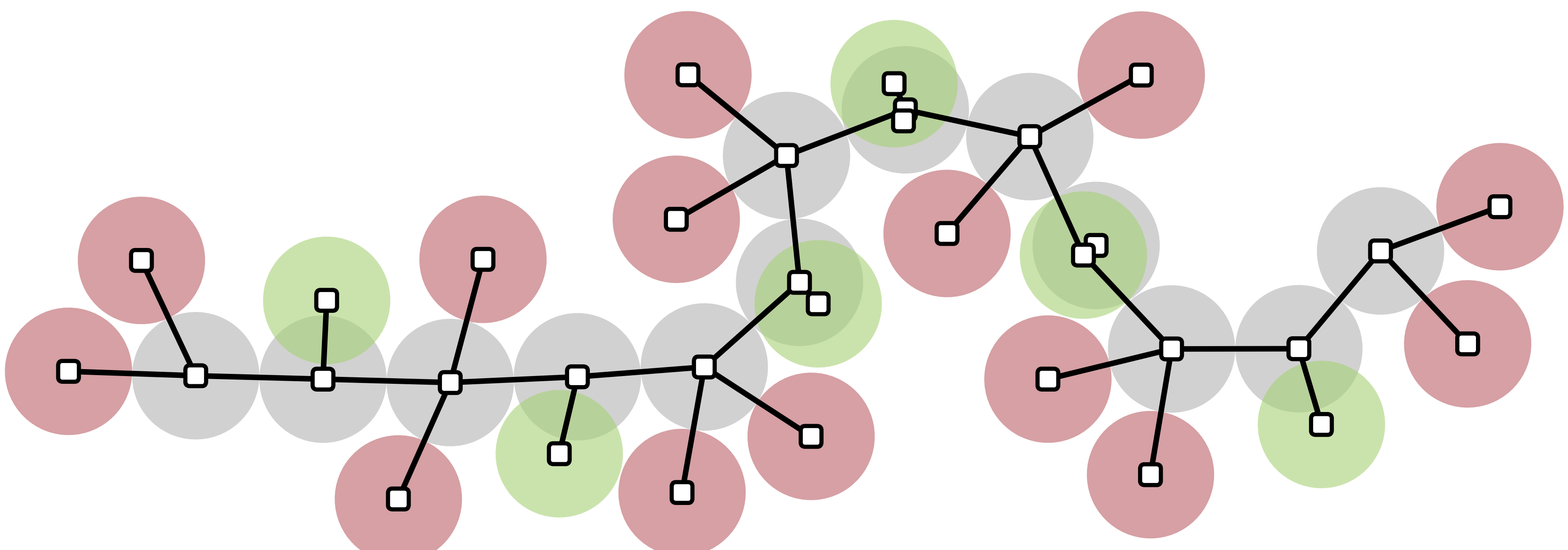


Theorem 1. *Recognizing unit disk graphs is* NP*-hard for embedded trees.*

What about **non-embedded** graphs?

These need to be forced on one side



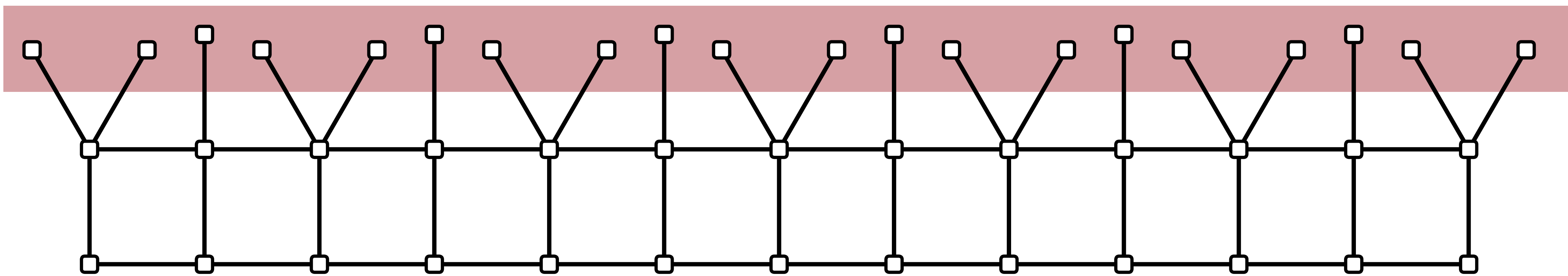


Hardness Results and Outerplanarity

Theorem 1. *Recognizing unit disk graphs is* NP*-hard for embedded trees.*

What about **non-embedded** graphs?

These need to be forced on one side

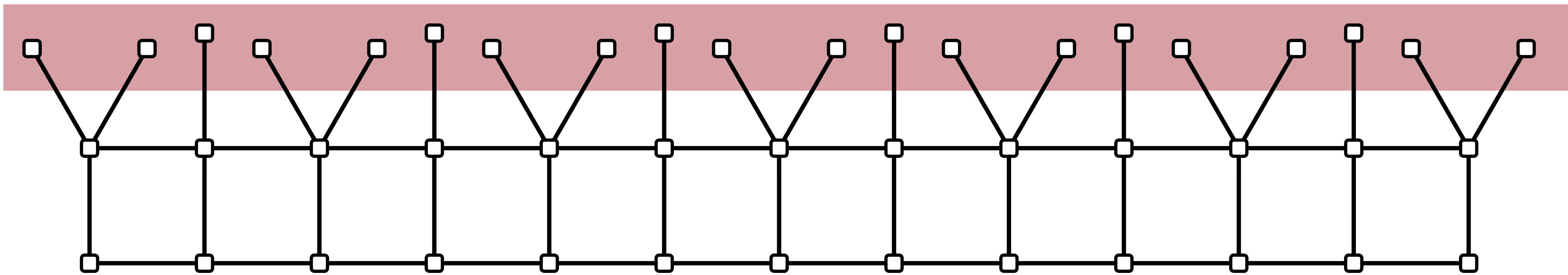


Hardness Results and Outerplanarity

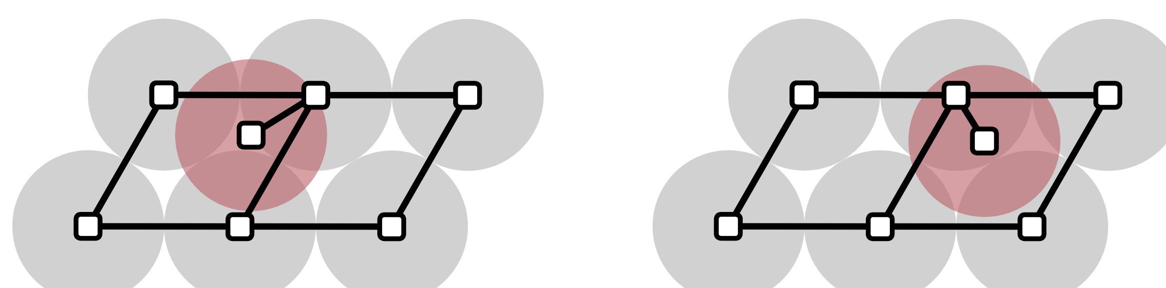
Theorem 1. *Recognizing unit disk graphs is* NP*-hard for embedded trees.*

What about **non-embedded** graphs?

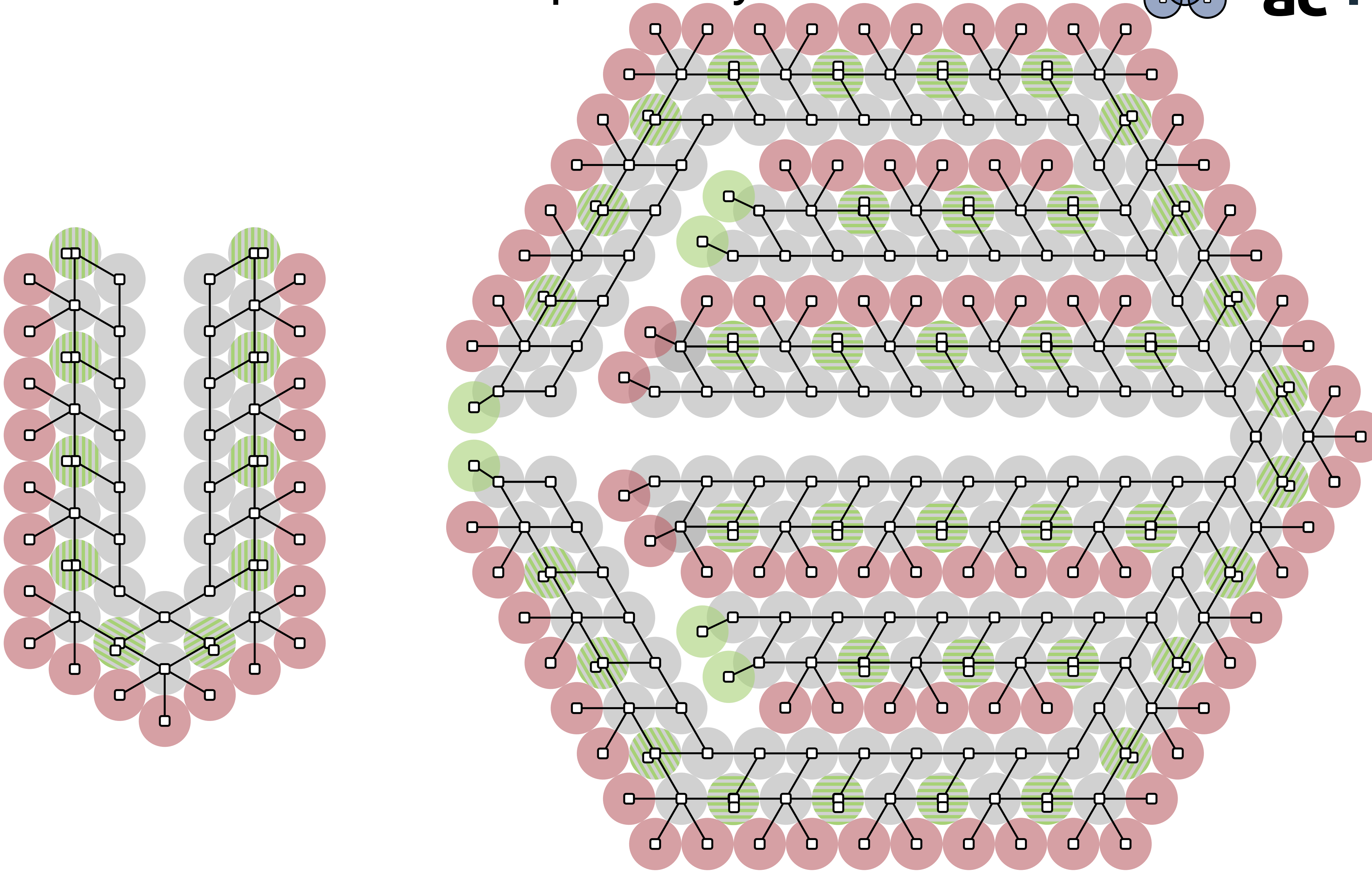
These need to be forced on one side



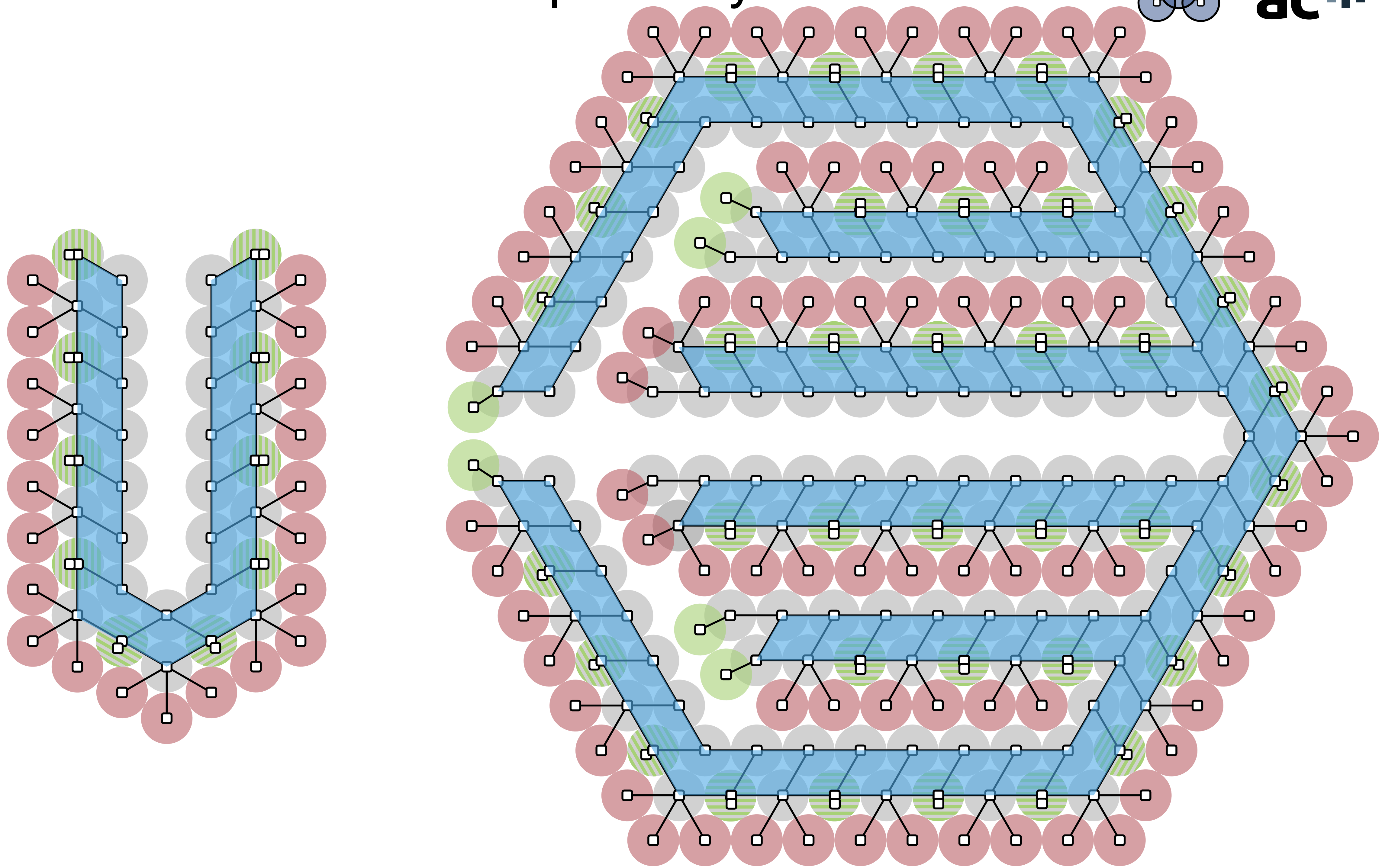
Outer neighbours can not be placed on the inside.



Hardness Results and Outerplanarity



Hardness Results and Outerplanarity

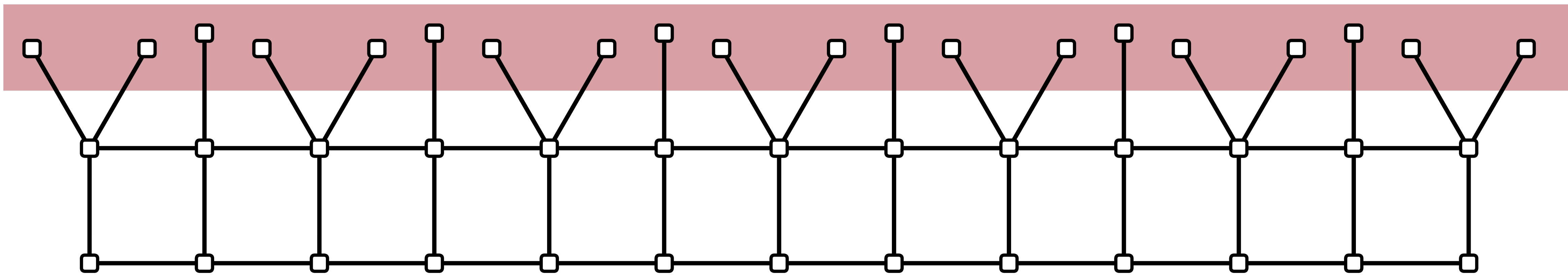


Hardness Results and Outerplanarity

Theorem 1. *Recognizing unit disk graphs is* NP*-hard for embedded trees.*

What about **non-embedded** graphs?

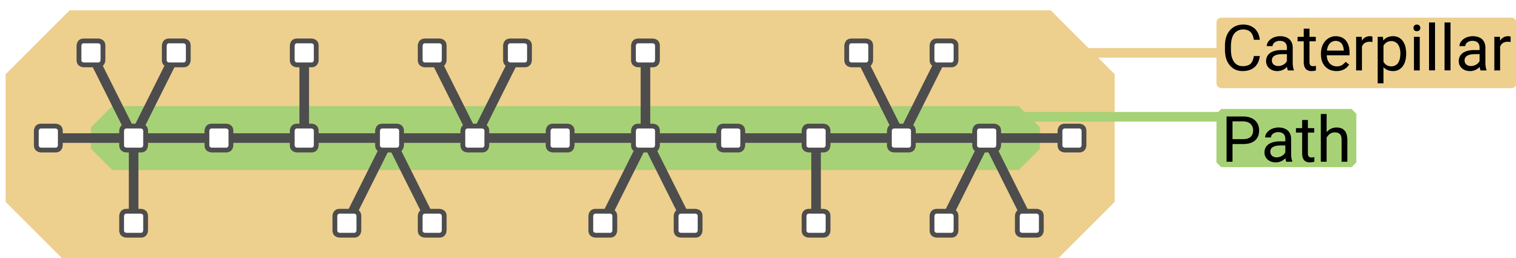
These need to be forced on one side



Theorem 2. *Recognizing unit disk graphs is* NP*-hard for non-embedded outerplanar graphs.*

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

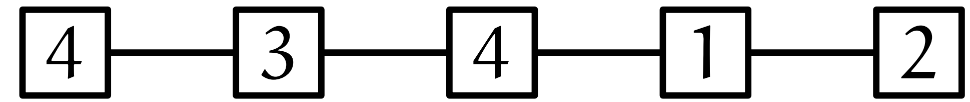


Caterpillar ≙ Path with leaves

Caterpillars admitting a UDR - Characterization

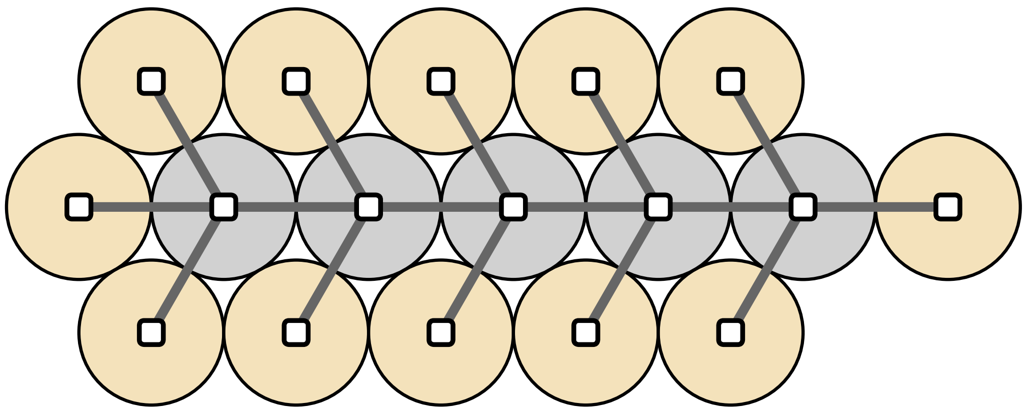
Caterpillars without embedding

A caterpillar C



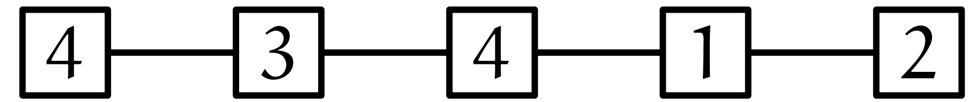
Caterpillars admitting a UDR - Characterization

Caterpillars without embedding



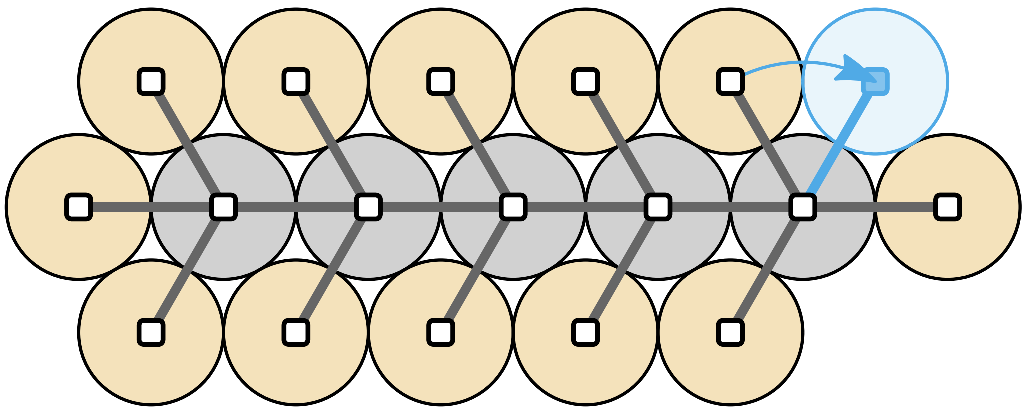
4-regular spines

A caterpillar $\mathcal{C}$



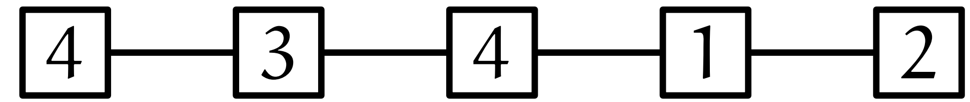
Caterpillars admitting a UDR - Characterization

Caterpillars without embedding



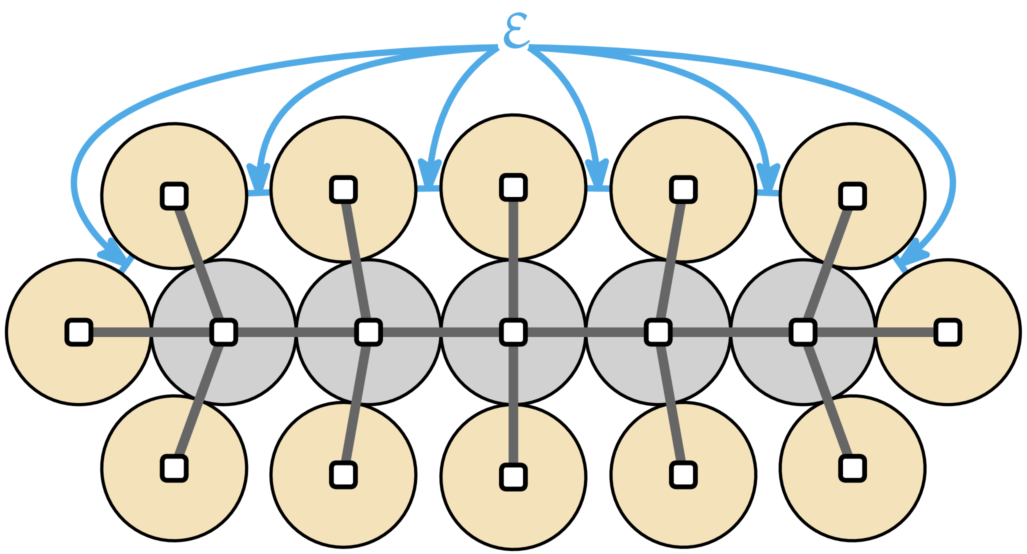
4-regular spines

A caterpillar C



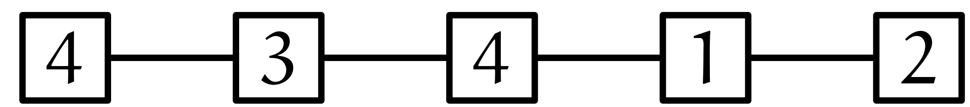
Caterpillars admitting a UDR - Characterization

Caterpillars without embedding



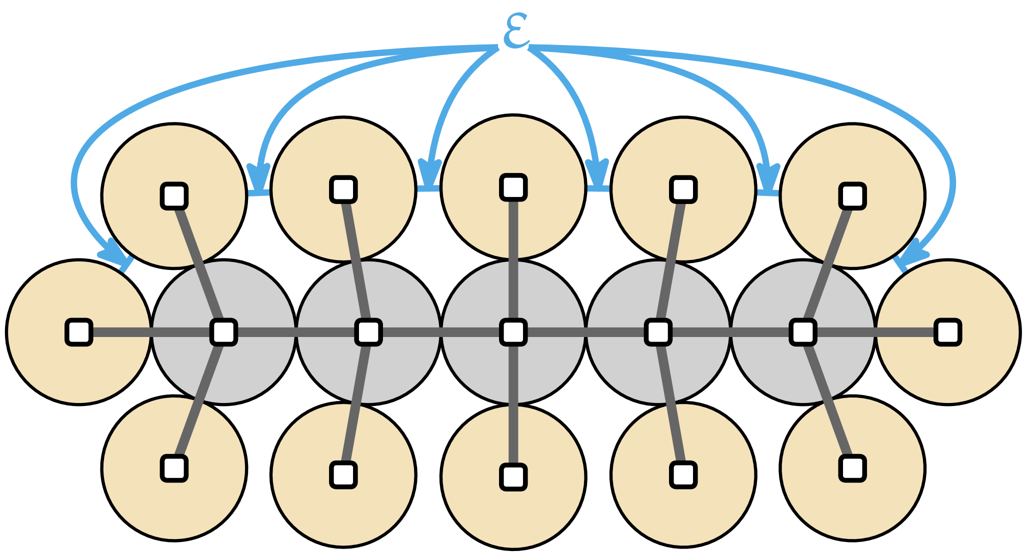
4-regular spines are **trivial**!

A caterpillar C



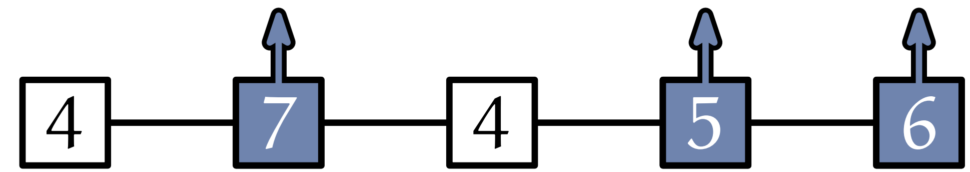
Caterpillars admitting a UDR - Characterization

Caterpillars without embedding



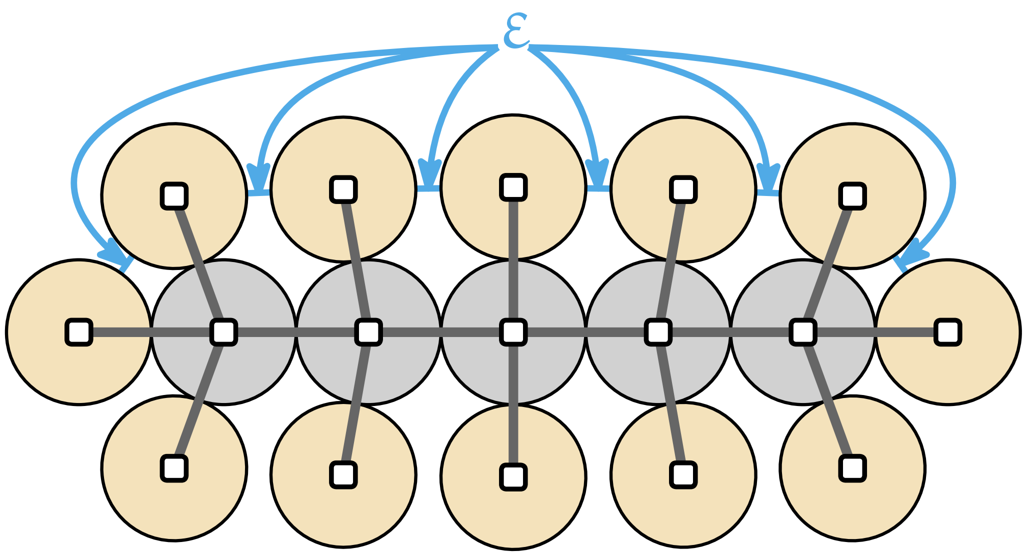
4-regular spines are **trivial**!

A caterpillar C

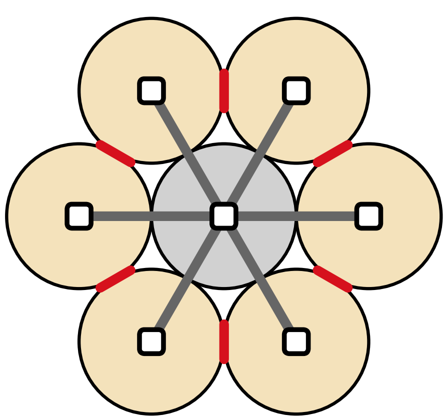


Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

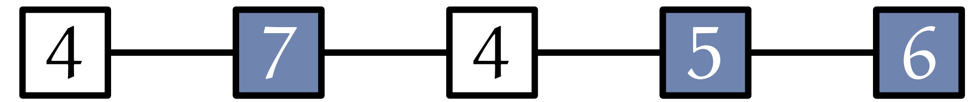


4-regular spines are **trivial**!



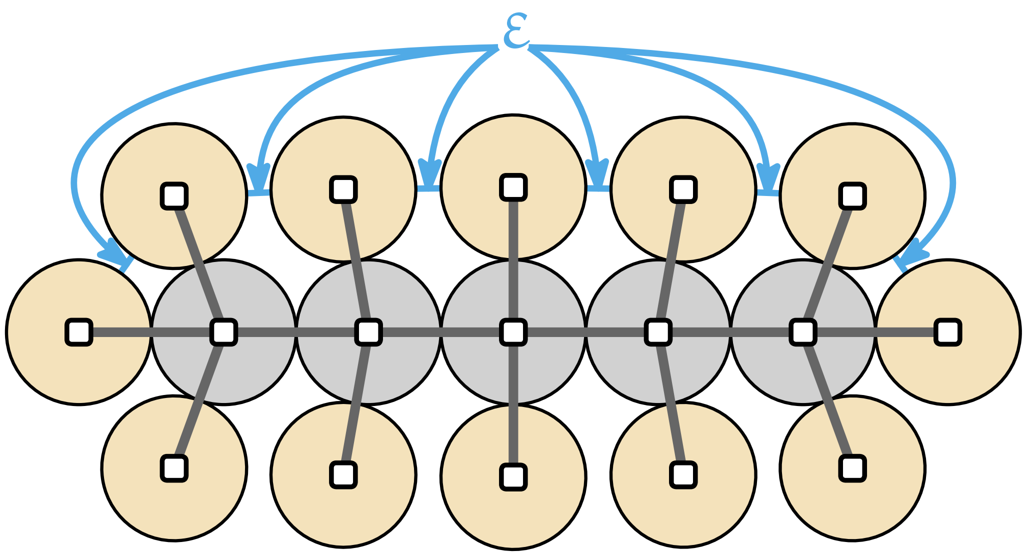
Degree 6 vertices are **impossible**!

A caterpillar C

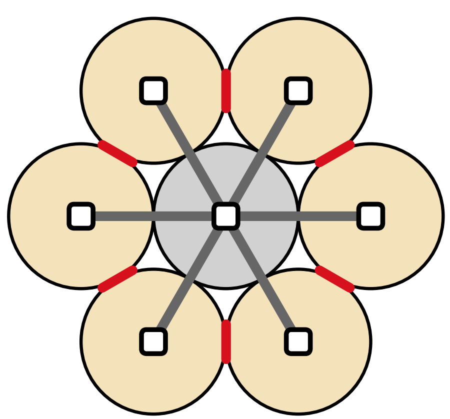


Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

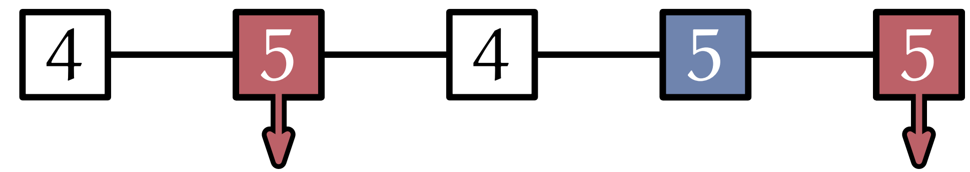


4-regular spines are **trivial**!



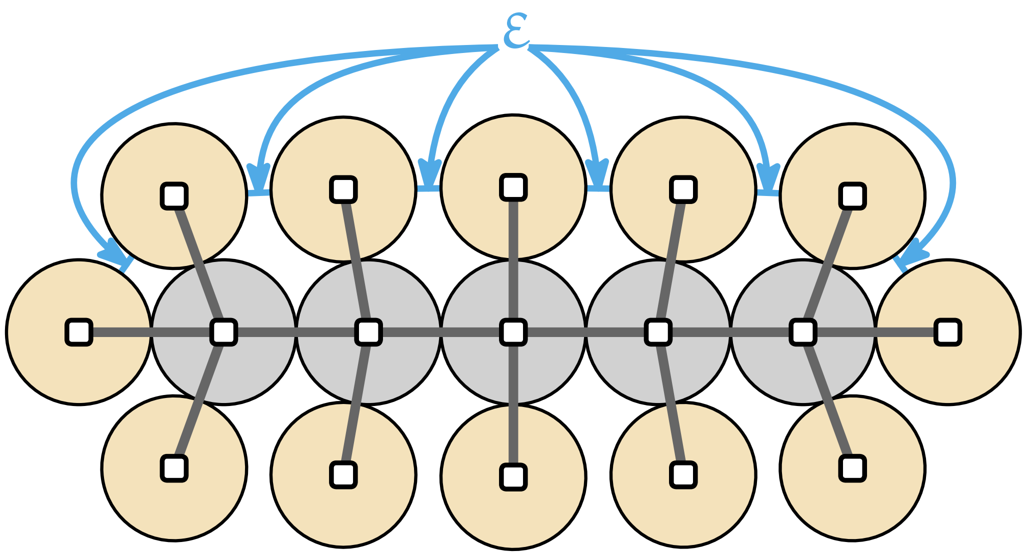
Degree 6 vertices are **impossible**!

A caterpillar C

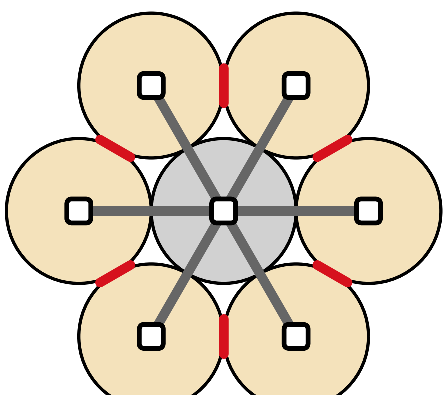


Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

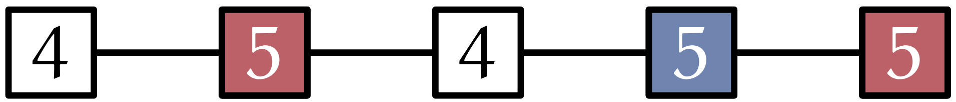


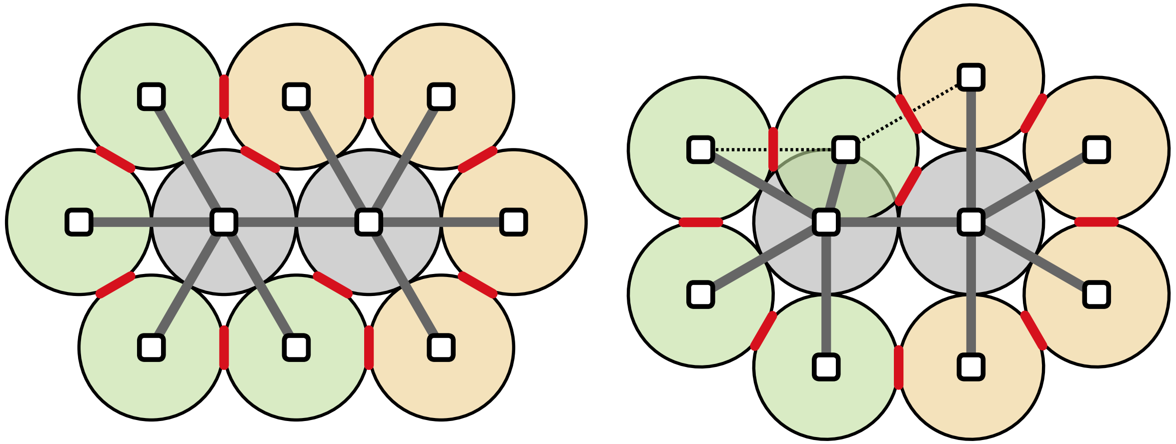
4-regular spines are **trivial**!



Degree 6 vertices are **impossible**!

A caterpillar C

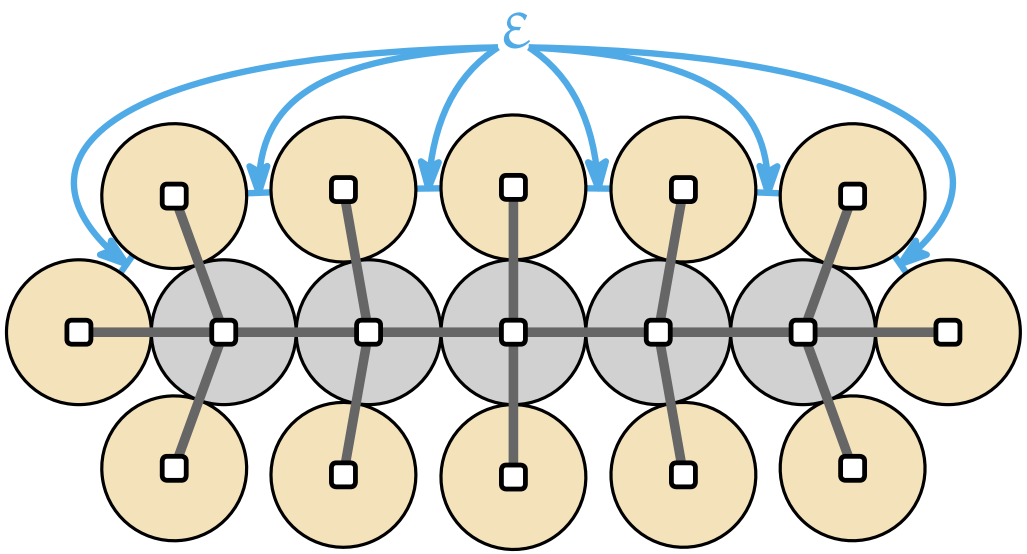




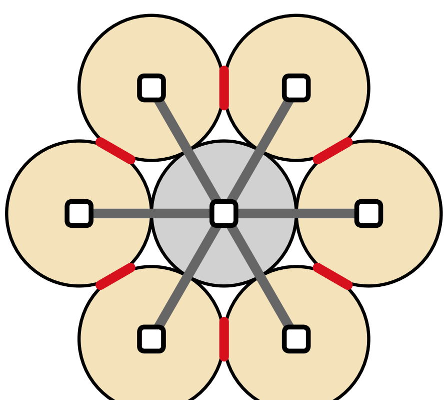
Consecutive degree 5 vertices are impossible!

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

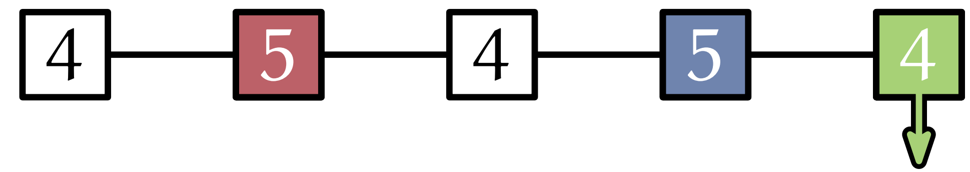


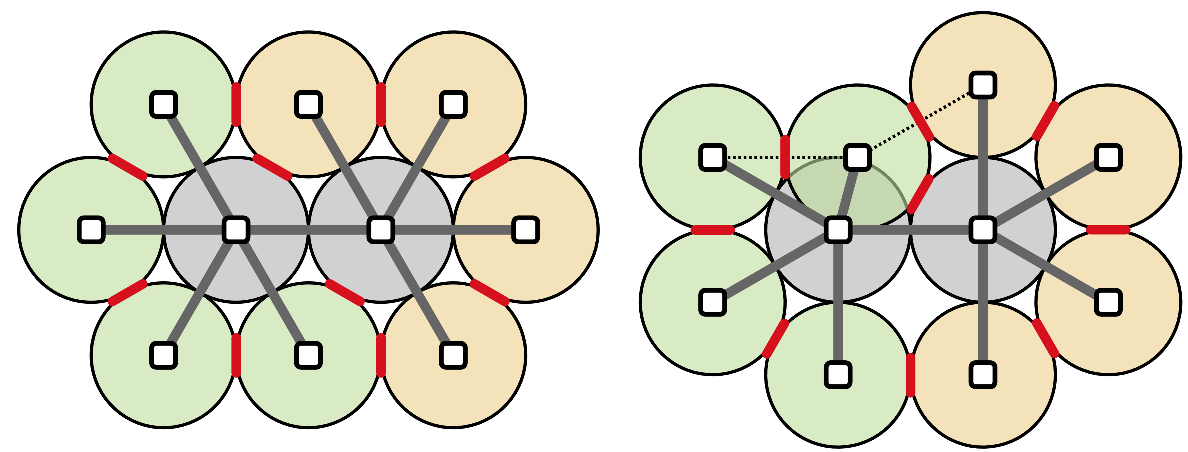
4-regular spines are **trivial**!



Degree 6 vertices are **impossible**!

A caterpillar C

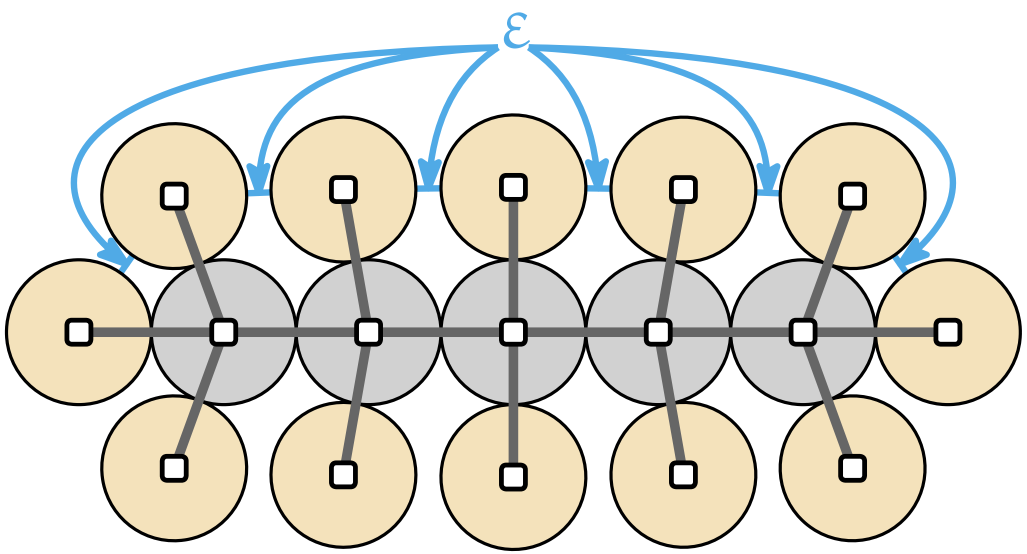




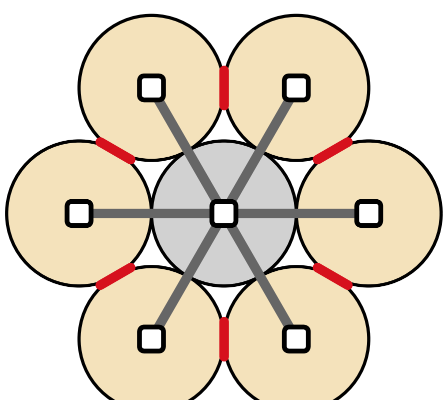
Consecutive degree 5
vertices are impossible!

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

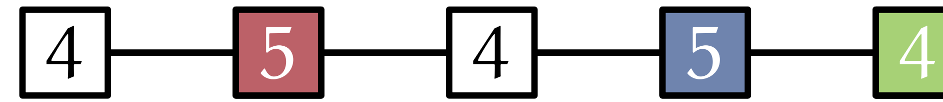


4-regular spines are **trivial**!

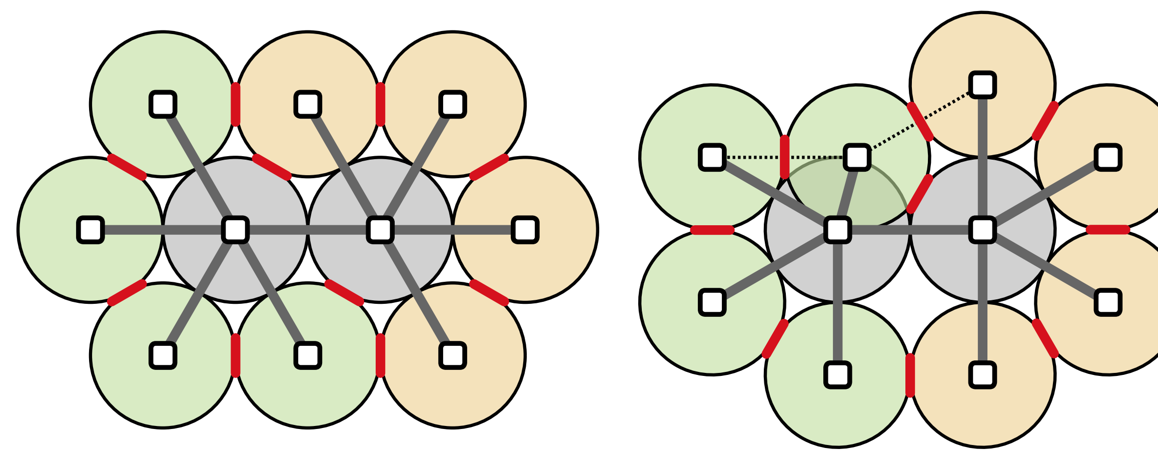


Degree 6 vertices are **impossible**!

A caterpillar C



Can we realize $C = \ldots 5454545 \ldots$?

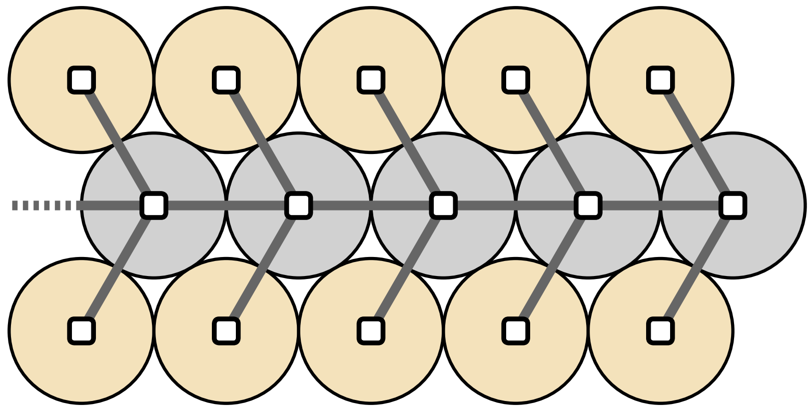


Consecutive degree 5
vertices are impossible!

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

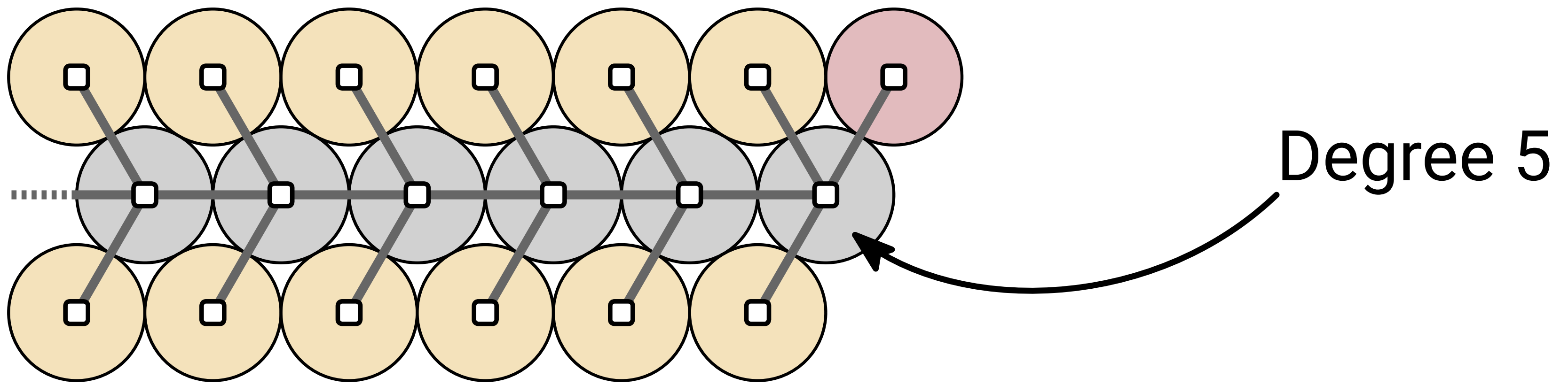
Backward position,
as long as degree is ≤ 4



Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

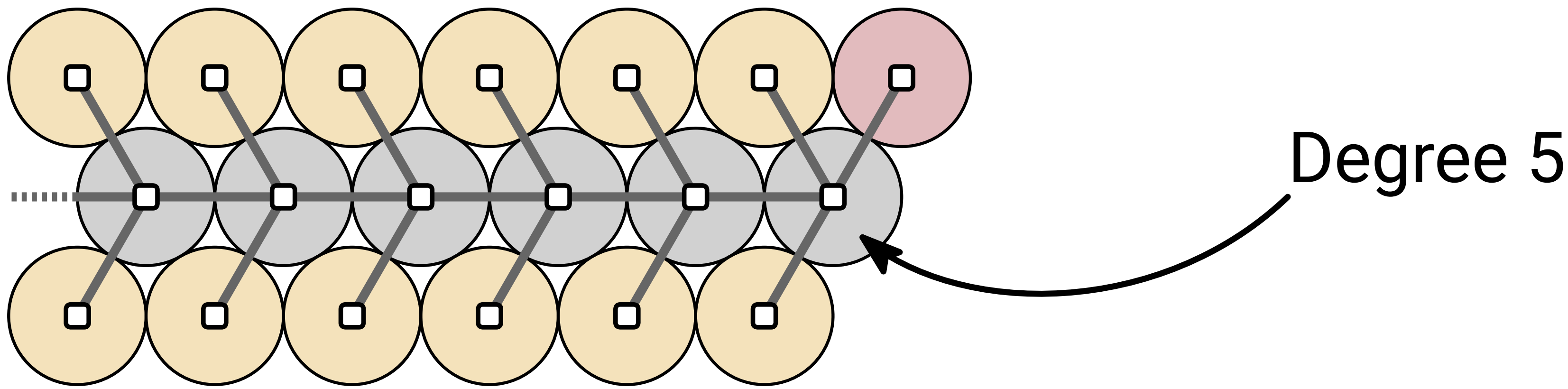
Backward position,
as long as degree is ≤ 4



Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4

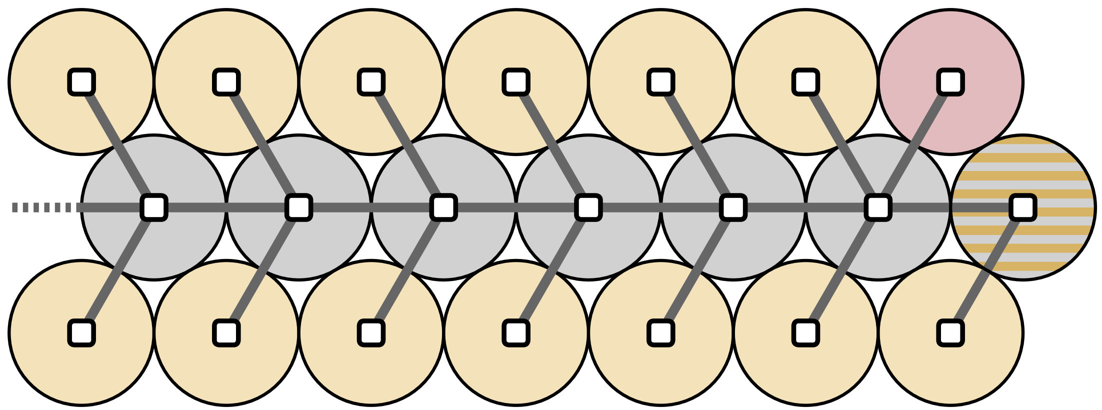


- If the next vertex has degree 5, we abort.

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4

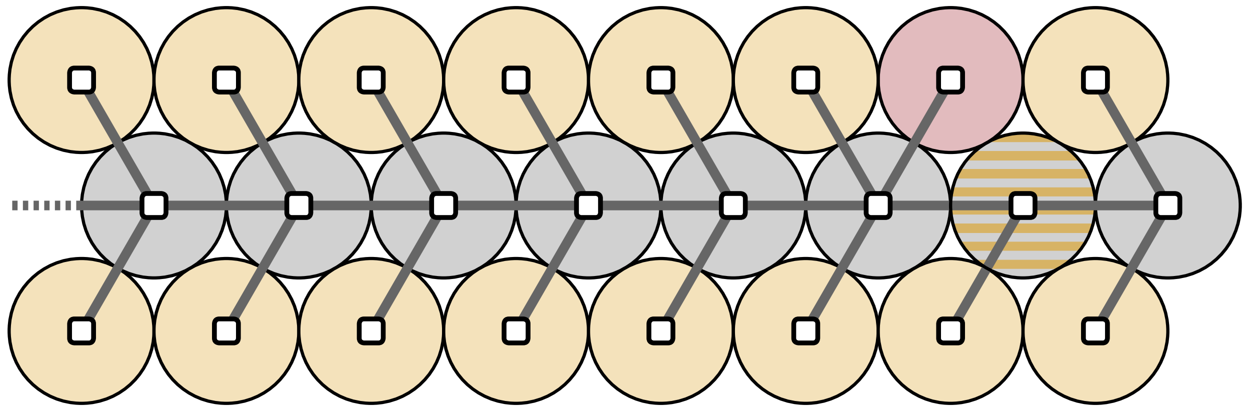


- If the next vertex has degree 5, we abort.
- Otherwise we can return to backward position.

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4

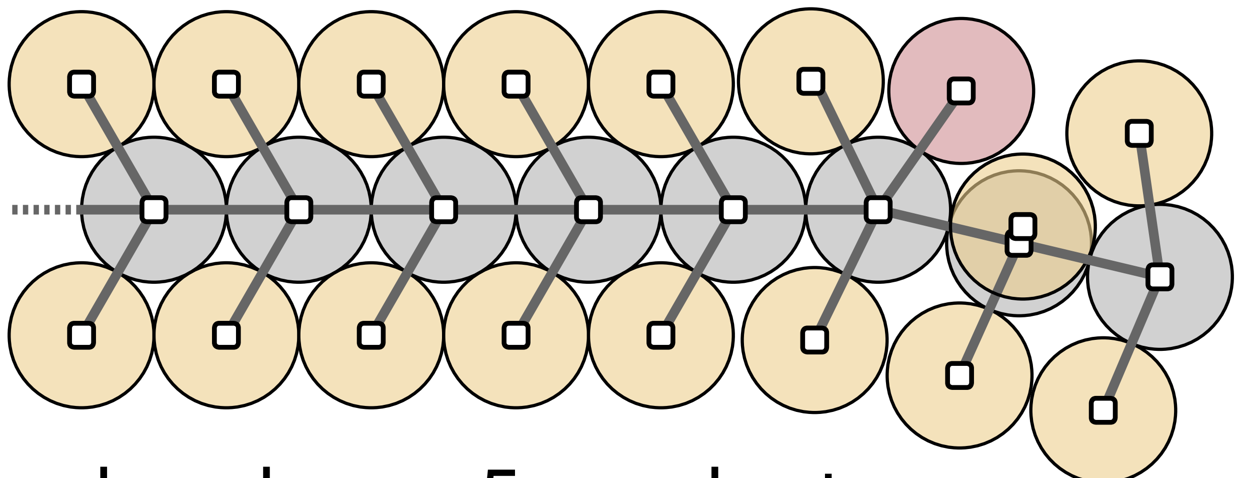


- If the next vertex has degree 5, we abort.
- Otherwise we can return to backward position.

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4

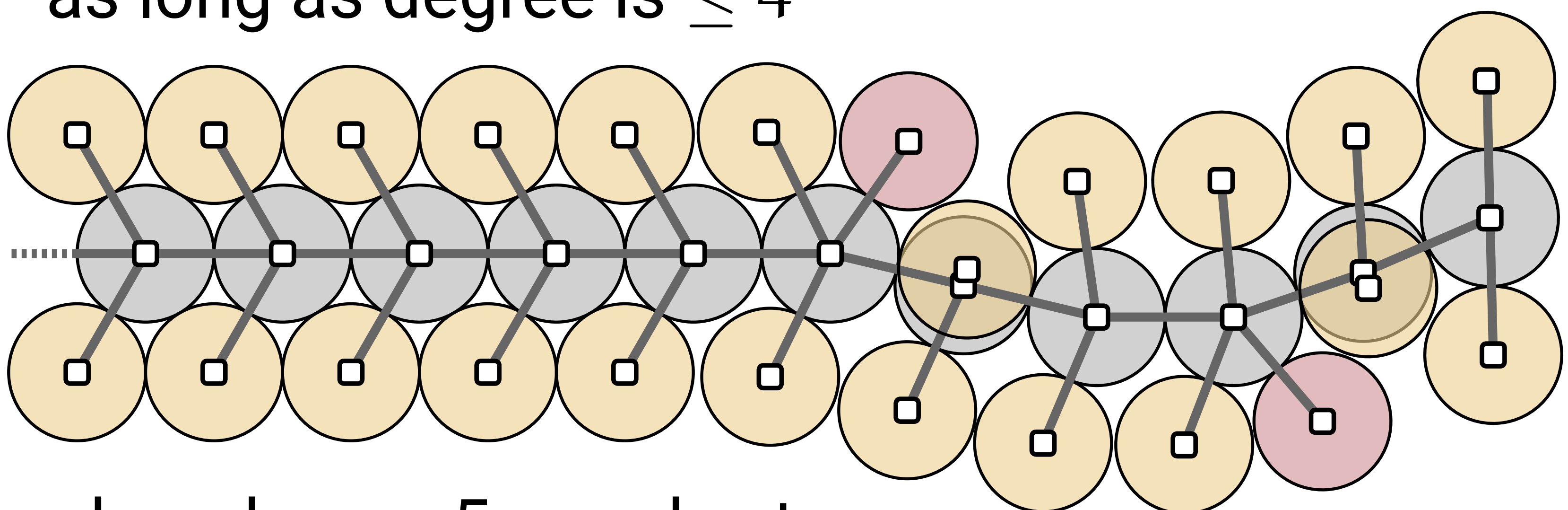


- If the next vertex has degree 5, we abort.
- Otherwise we can return to backward position.

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4

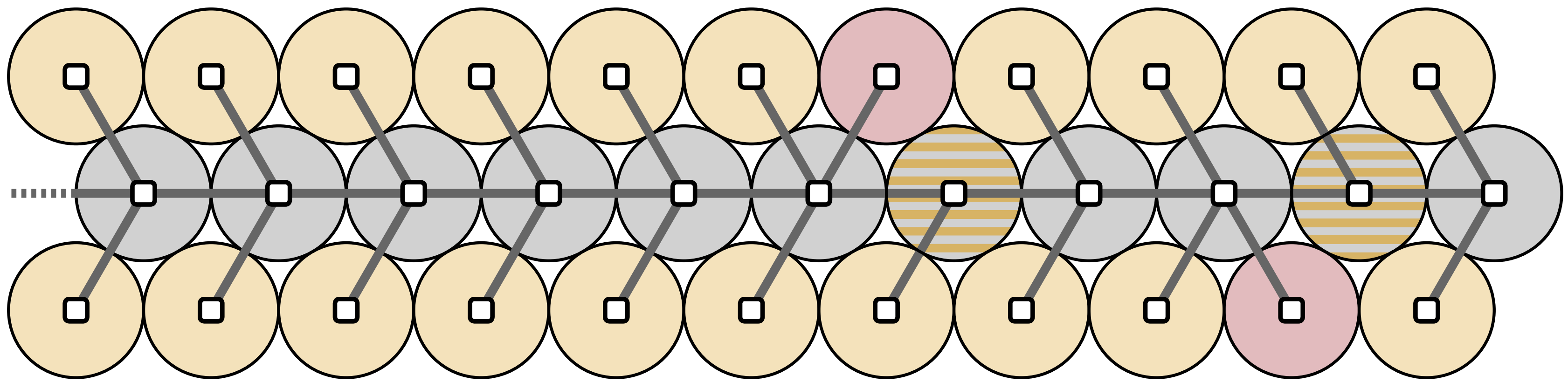


- If the next vertex has degree 5, we abort.
- Otherwise we can return to backward position.
- At the end, use available slack again to avoid wrong adjacencies

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4

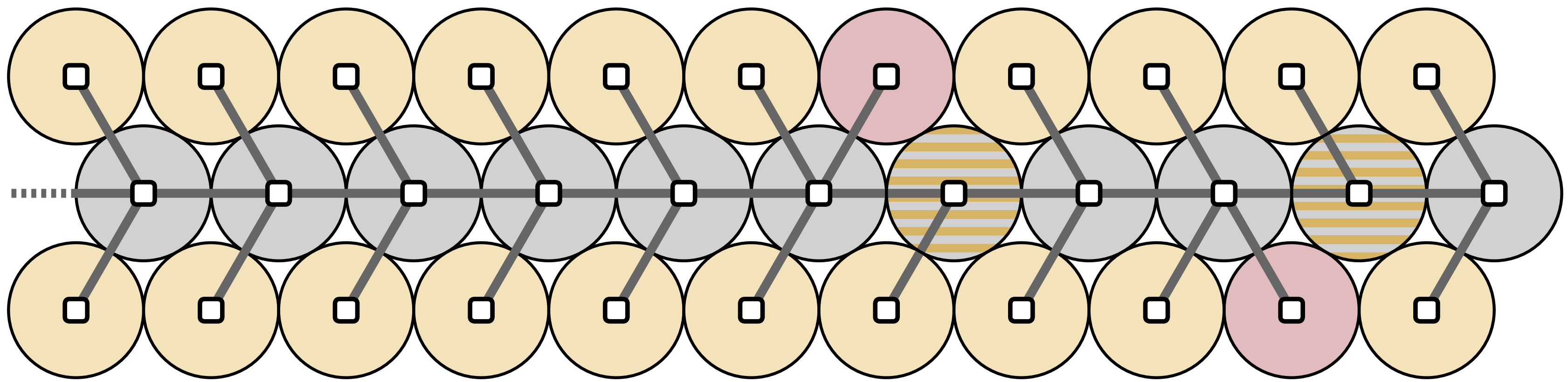


- If the next vertex has degree 5, we abort.
- Otherwise we can return to backward position.
- At the end, use available slack again to avoid wrong adjacencies

Caterpillars admitting a UDR - Characterization

Caterpillars without embedding

Backward position,
as long as degree is ≤ 4



- If the next vertex has degree 5, we abort.
- Otherwise we can return to backward position.
- At the end, use available slack again to avoid wrong adjacencies

Theorem 3. *Caterpillar graphs admit a UDR, if and only if they neither contain vertices of degree 6 or higher nor two adjacent vertices of degree 5.*

Conclusion

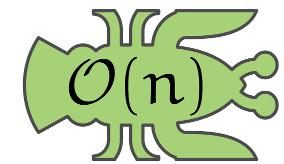
- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).

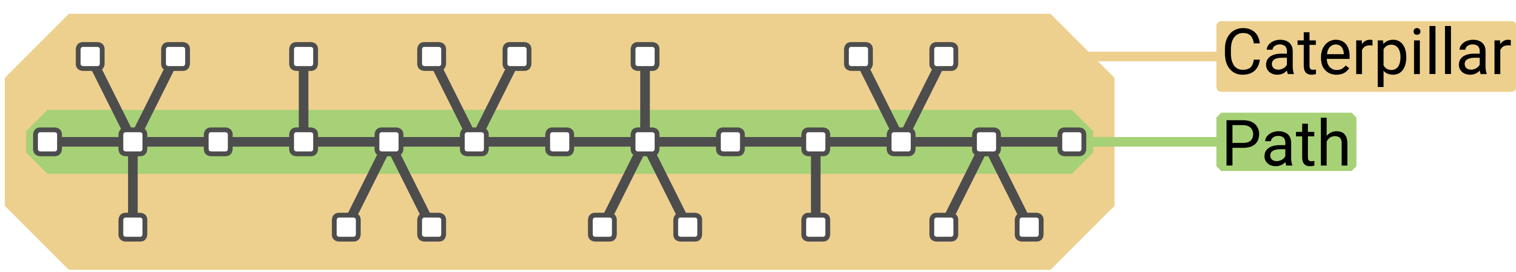
Conclusion

- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).
- Recognition can be done in **linear** time for **caterpillars** (non-embedded)

Conclusion

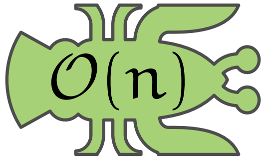
- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).
- Recognition can be done in **linear** time for **caterpillars** (non-embedded)
- **Computer-assisted** proof: **lobster graphs** on a **grid** can be represented as a x-monotone *weak* UDC and recognized in **linear time**

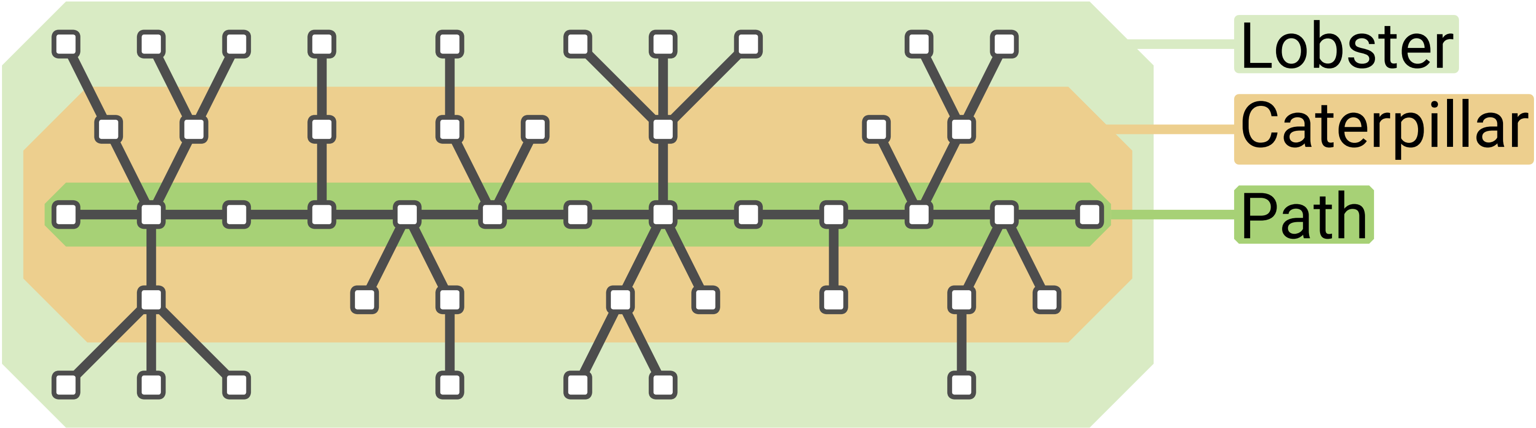




Conclusion

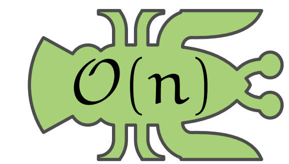
- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).
- Recognition can be done in **linear** time for **caterpillars** (non-embedded)
- **Computer-assisted** proof: **lobster graphs** on a **grid** can be represented as a x-monotone *weak* UDC and recognized in **linear time**

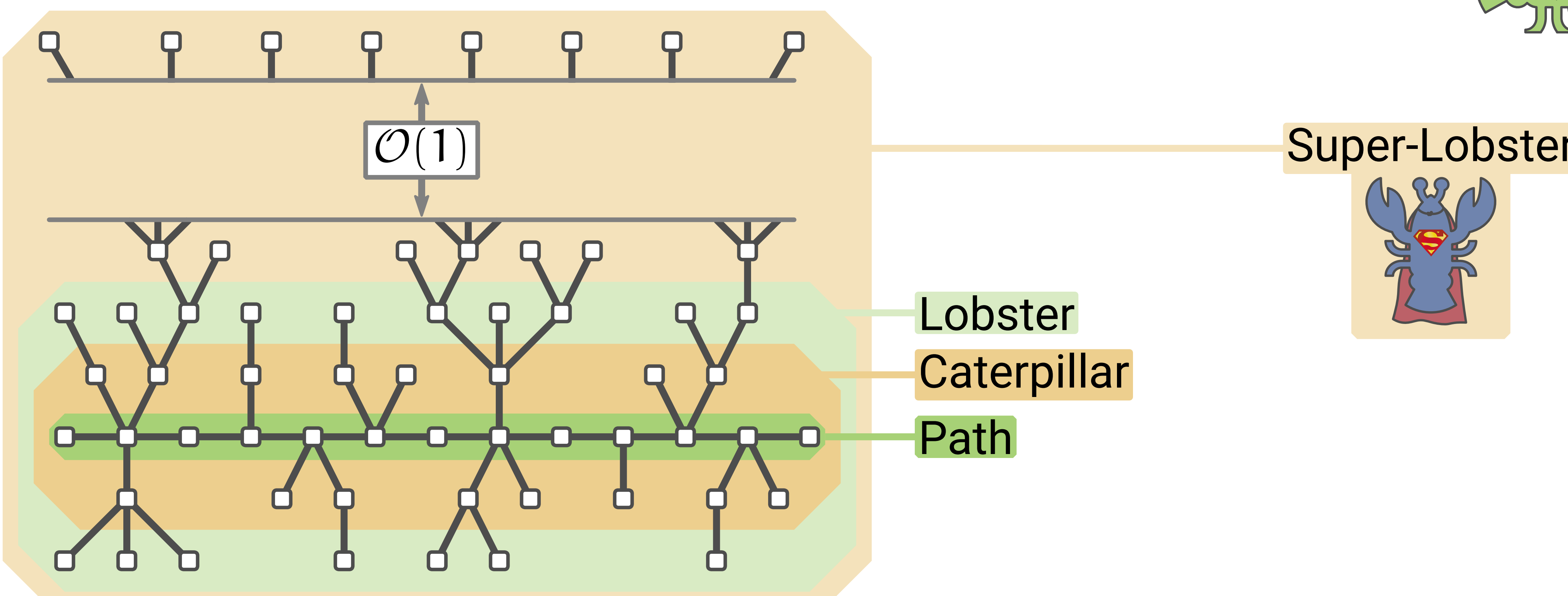




Conclusion

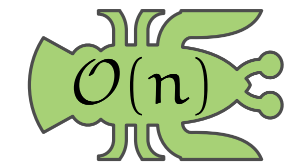
- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).
- Recognition can be done in **linear** time for **caterpillars** (non-embedded)
- **Computer-assisted** proof: **lobster graphs** on a **grid** can be represented as a x-monotone *weak* UDC and recognized in **linear time**

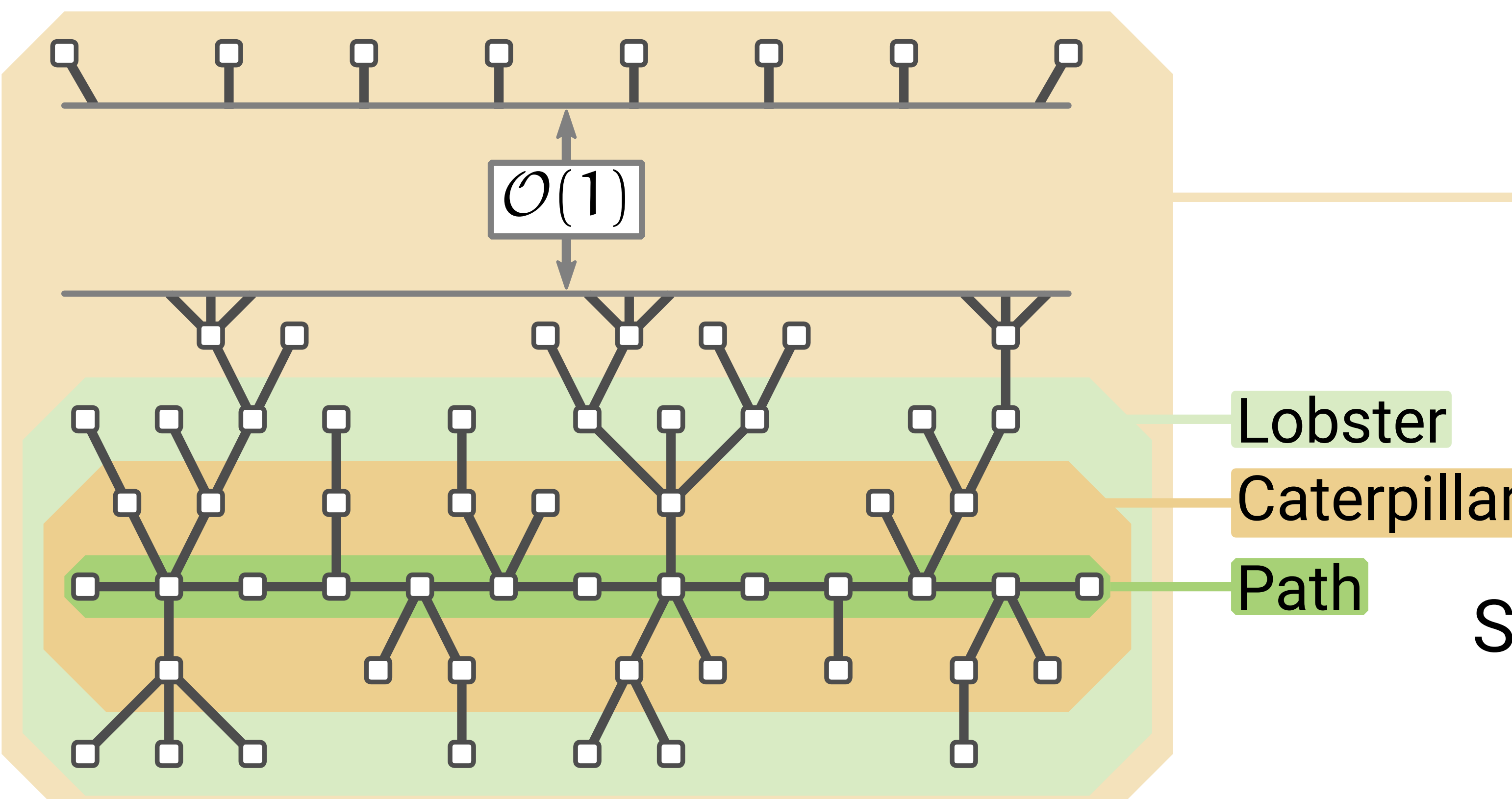




Conclusion

- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).
- Recognition can be done in **linear** time for **caterpillars** (non-embedded)
- **Computer-assisted** proof: **lobster graphs** on a **grid** can be represented as a x-monotone *weak* UDC and recognized in **linear time**

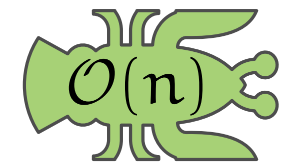




Weak UDCs for Super-Lobster in $\mathcal{O}(n)$ time?

Conclusion

- Recognizing unit disk graphs is NP-hard for **embedded trees** and **outerplanar** graphs (non-embedded).
- Recognition can be done in **linear** time for **caterpillars** (non-embedded)
- **Computer-assisted** proof: **lobster graphs** on a **grid** can be represented as a x-monotone *weak* UDC and recognized in **linear time**



Open questions:

	tree	caterpillar
contact graphs	**?**	$\mathcal{O}(n)$
↳ embedded	NP-hard	**?**
disk graphs	**?**	$\mathcal{O}(n)$
↳ embedded	NP-hard	**?**

Super-Lobster

Weak UDCs for Super-Lobster in $\mathcal{O}(n)$ time?