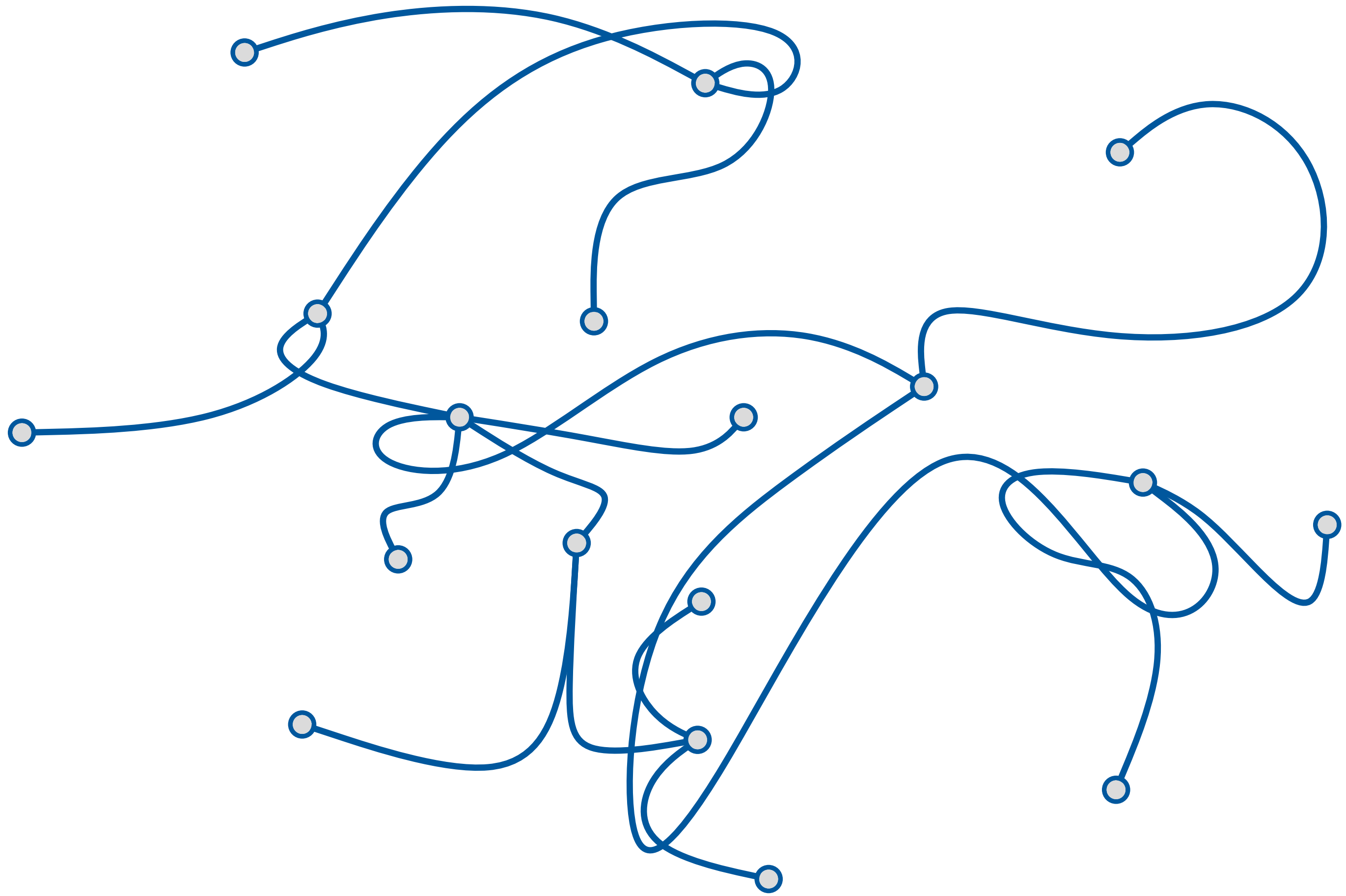
Drawing Trees with Perfect Angular Resolution and Polynomial Area

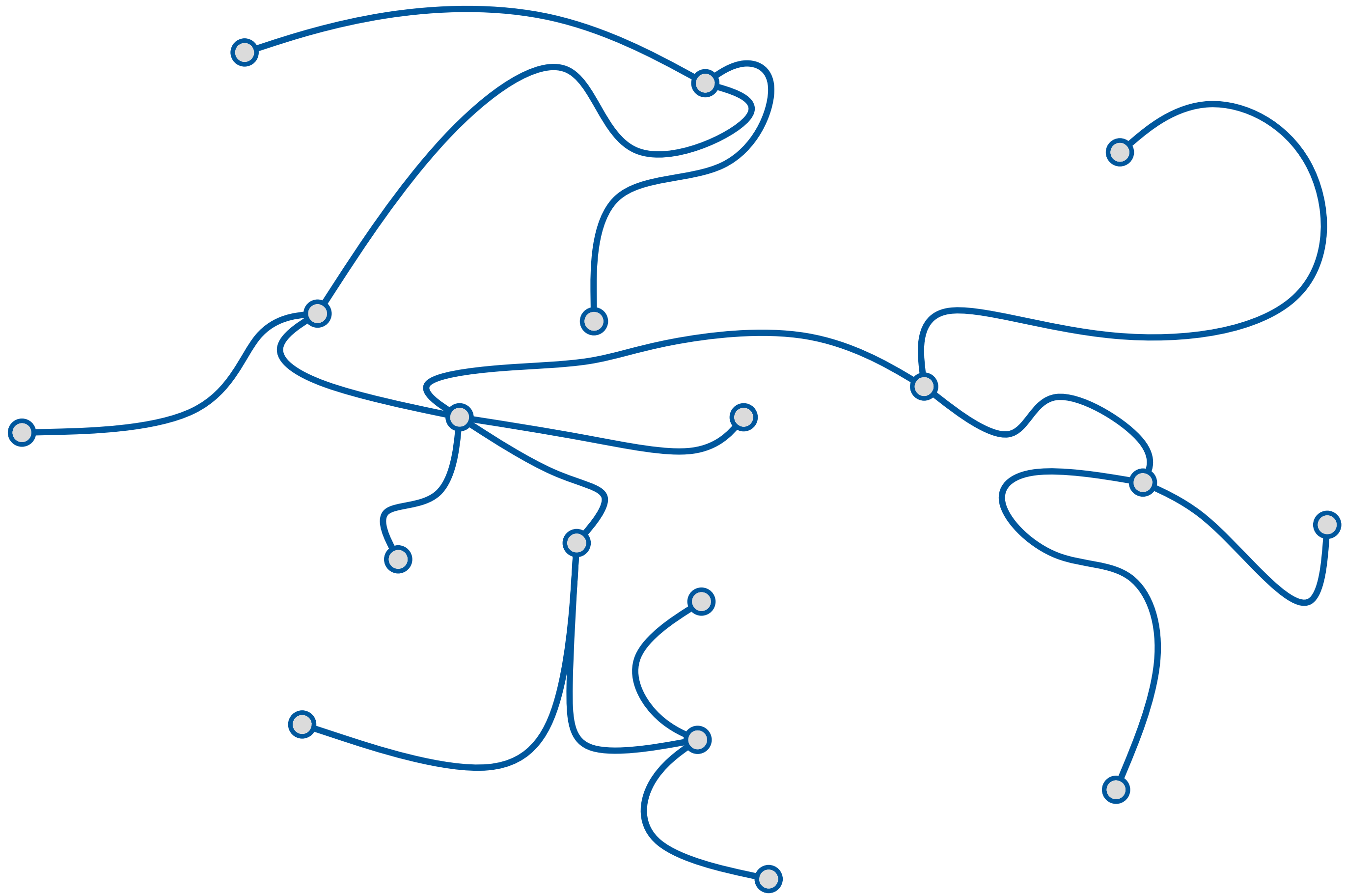
Christian A. Duncan · Quinnipiac University
David Eppstein · University of California Irvine
Michael T. Goodrich · University of California Irvine
Stephen G. Kobourov · University of Arizona Tucson
Martin Nöllenburg · KIT

Vorlesung *Algorithmen zur Visualisierung von Graphen* · 05.02.2013

What makes a good tree drawing?

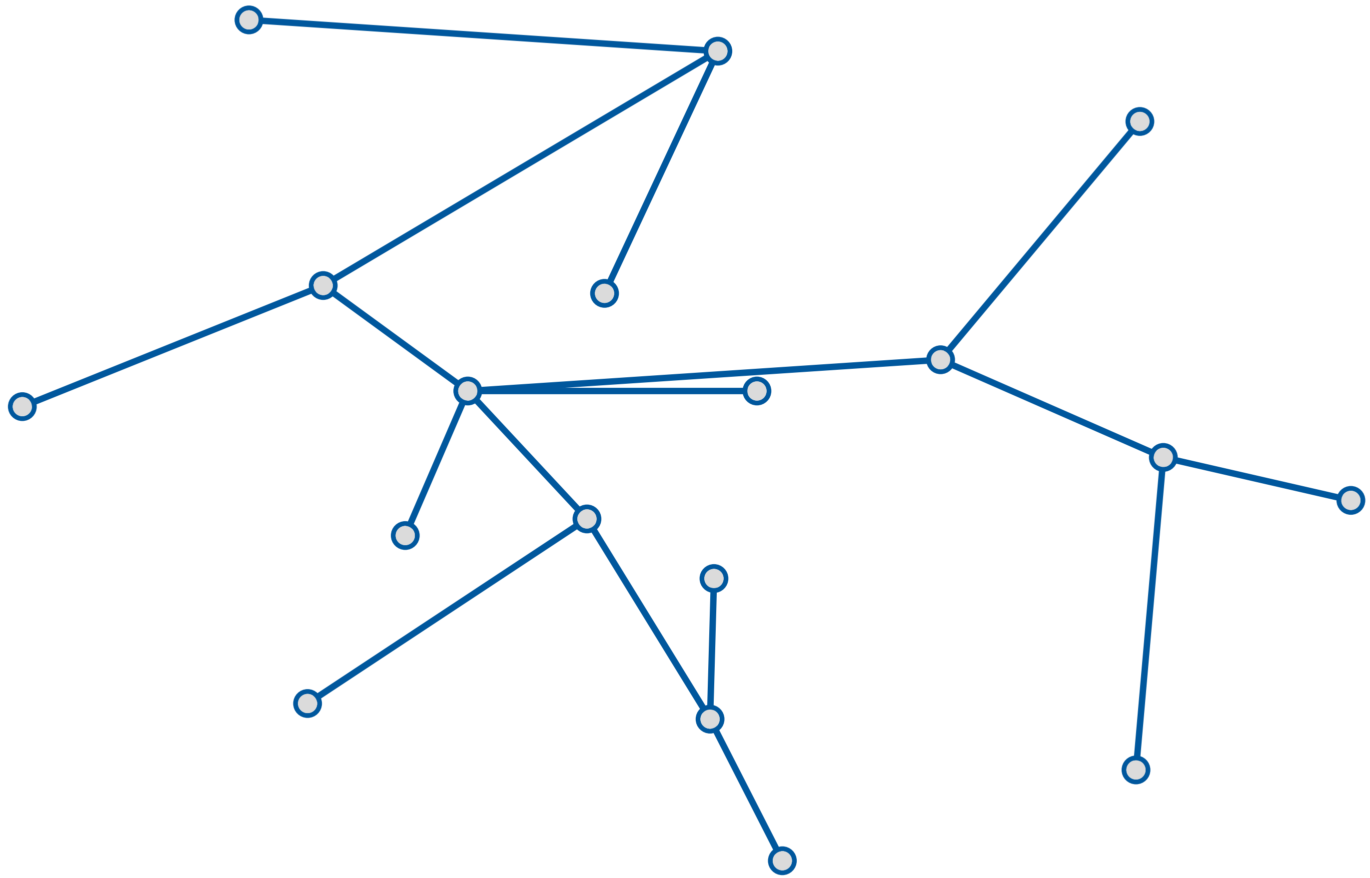


What makes a good tree drawing?



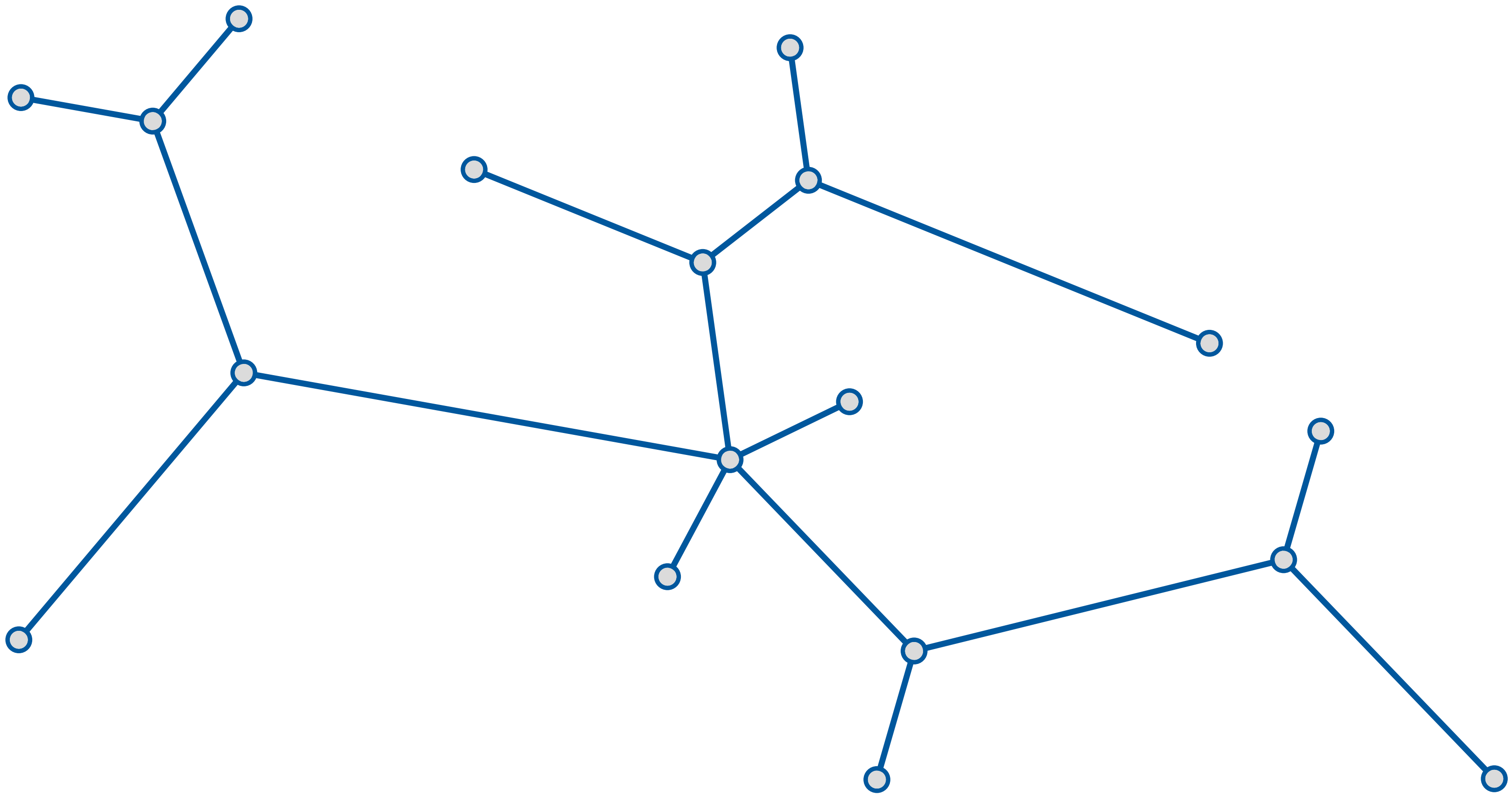
- crossing-free edges

What makes a good tree drawing?



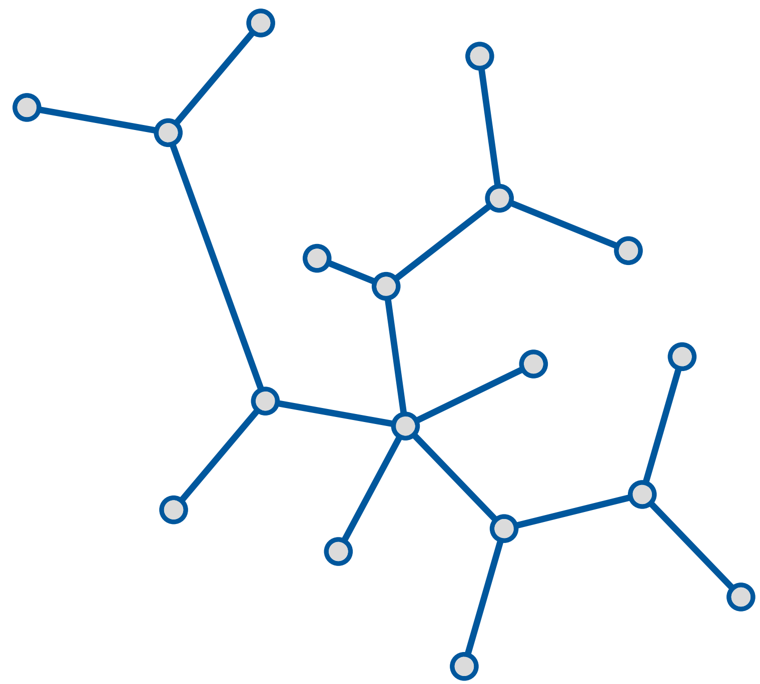
- crossing-free edges
- easy-to-follow edges

What makes a good tree drawing?



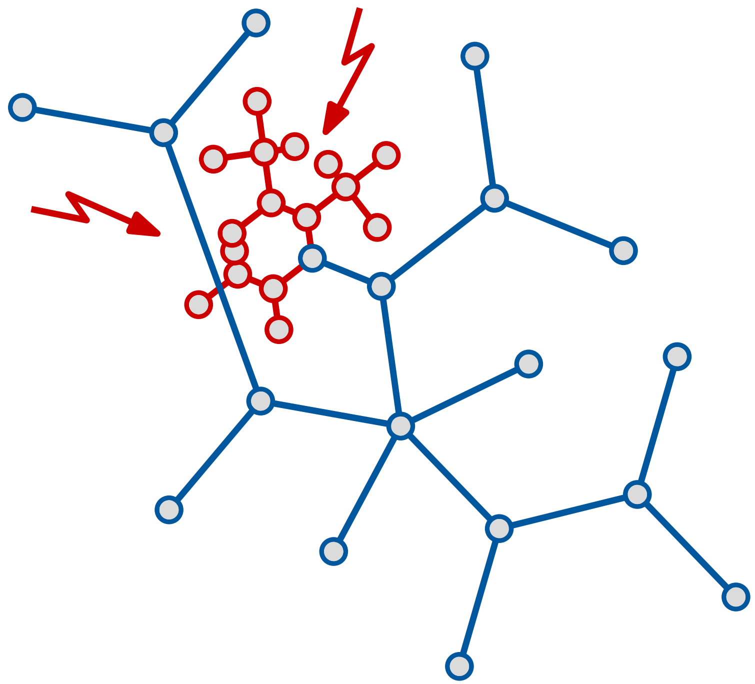
- crossing-free edges
- easy-to-follow edges
- good angular resolution

What makes a good tree drawing?



- crossing-free edges
- easy-to-follow edges
- good angular resolution
- small space consumption

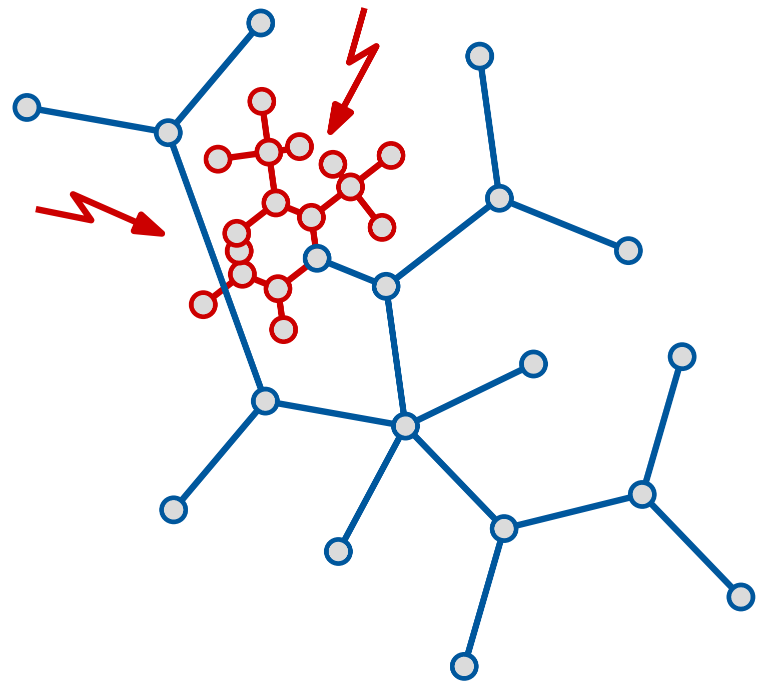
What makes a good tree drawing?



Can we achieve all these goals simultaneously?

- crossing-free edges
- easy-to-follow edges
- good angular resolution
- small space consumption

What makes a good tree drawing?



Can we achieve all these goals simultaneously?

It depends...

- crossing-free edges
- easy-to-follow edges
- good angular resolution
- small space consumption

Our results

Any tree has a drawing with

- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	✗

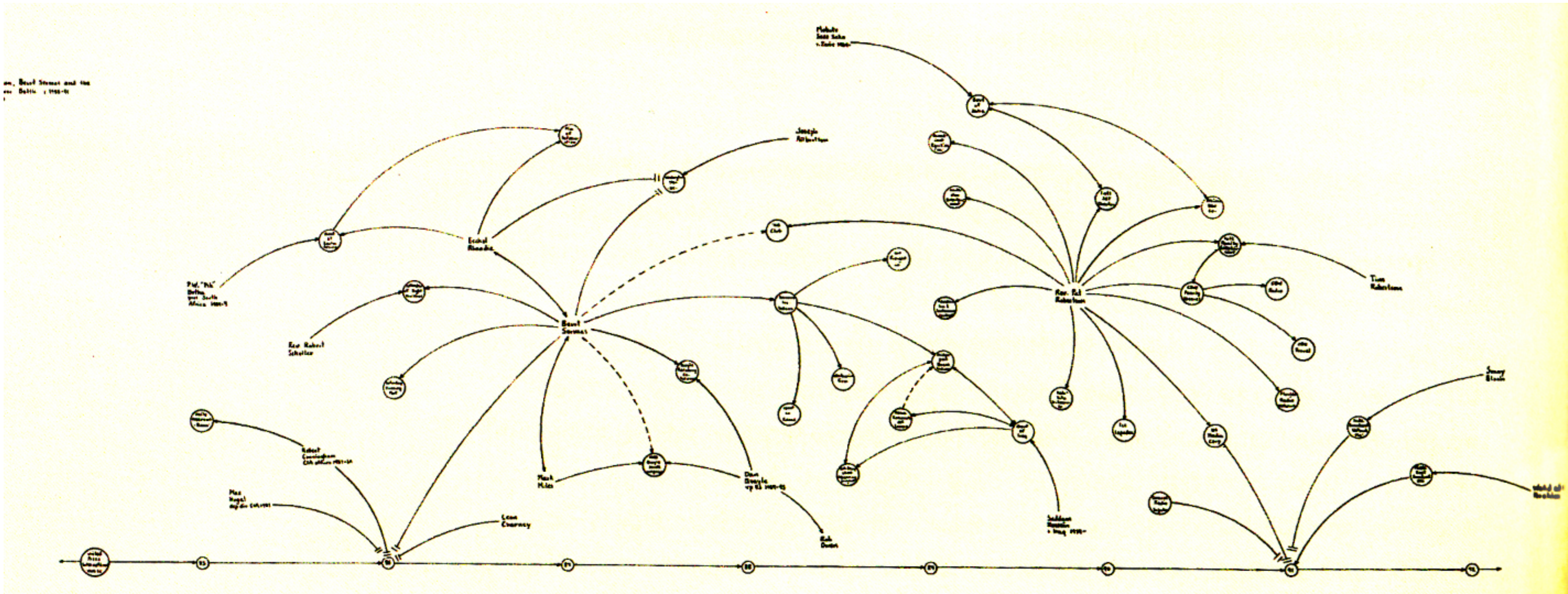
Our results

Any tree has a drawing with

- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	✗
?		? ? ✓ ? ? ? ?

Inspiration from fine arts

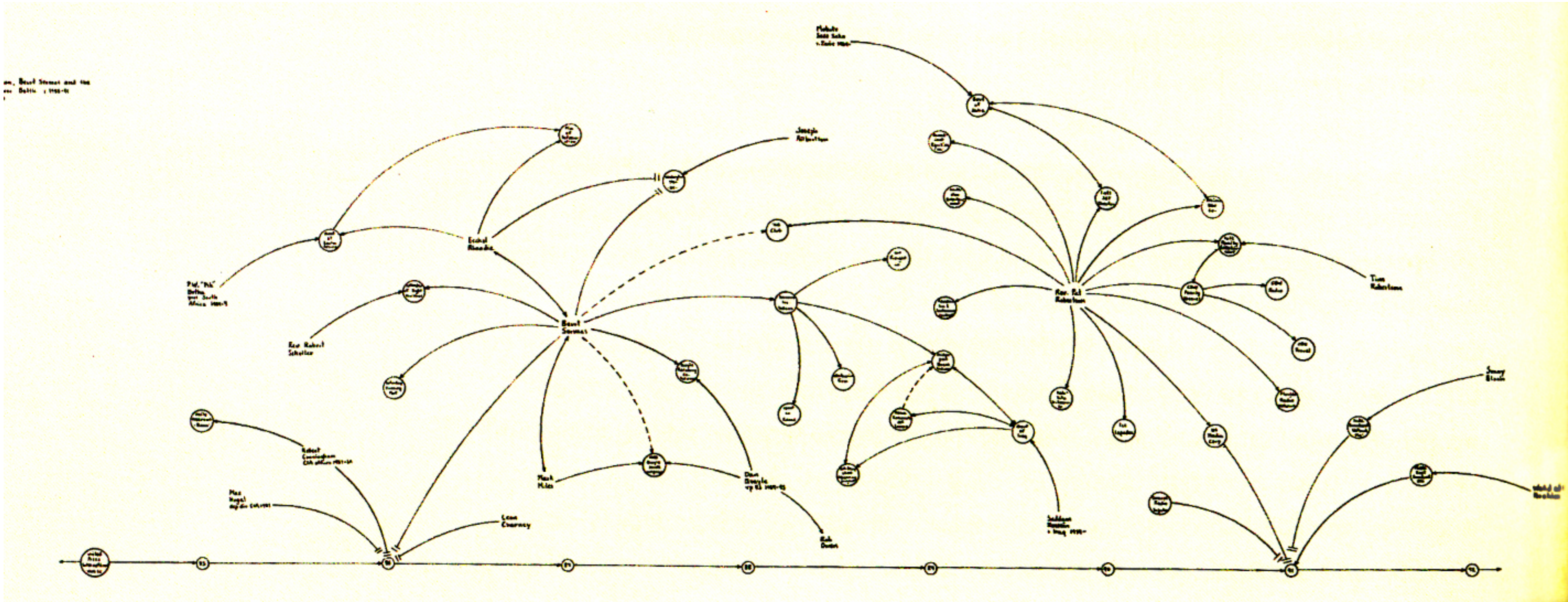


Mark Lombardi “Pat Robertson, Beurt Servaas, and the UPI Takeover Battle, ca. 1985–91”

Edges in Mark Lombardi’s network drawings

- easy-to-follow circular arcs
- generalization of straight-line edges

Inspiration from fine arts



Mark Lombardi "Pat Robertson, Beurt Servaas, and the UPI Takeover Battle, ca. 1985–91"

Edges in Mark Lombardi's network drawings

- easy-to-follow circular arcs
- generalization of straight-line edges

Draw edges as circular arcs!

Our results

Any tree has a drawing with

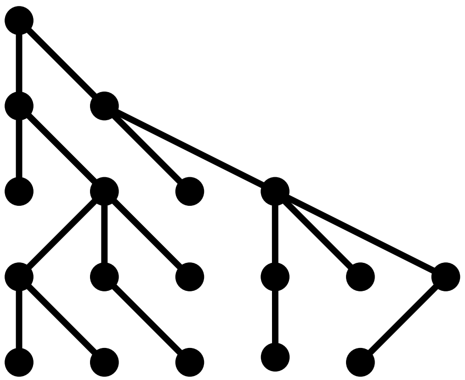
- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	✗
Lombardi edges	✓	✓

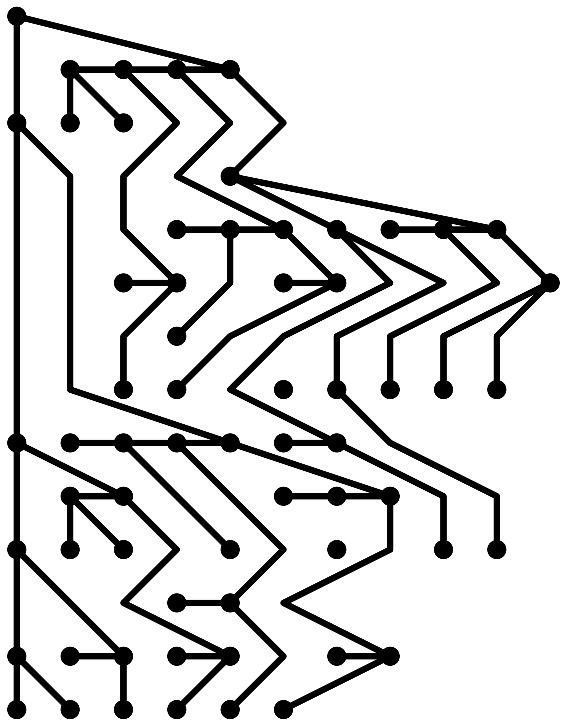
Related work

Straight-line drawings of trees

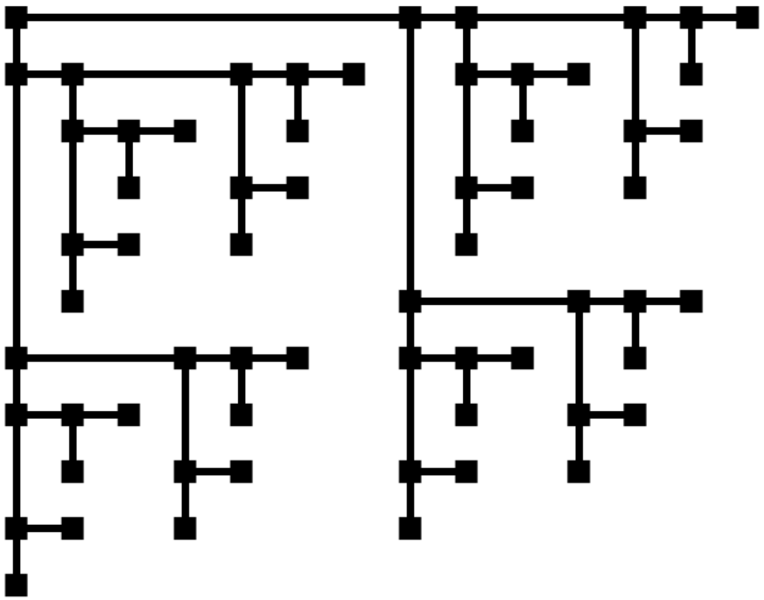
- old topic in graph drawing [Wetherell & Shannon, 1979]
- optimize area but not angular resolution



[Wetherell & Shannon, 1979]



[Garg, Goodrich, Tamassia, 1994]



[Garg & Rusu, 2004]

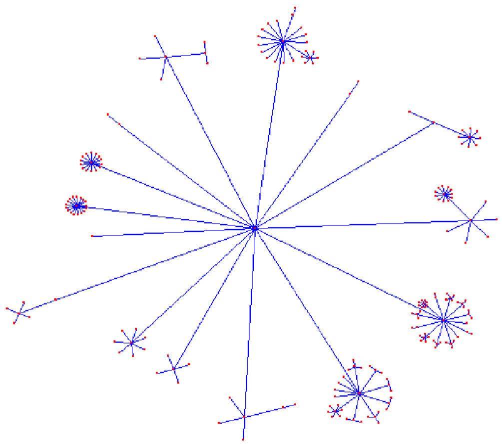
Related work

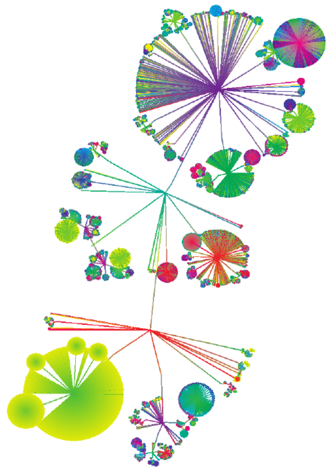
Straight-line drawings of trees

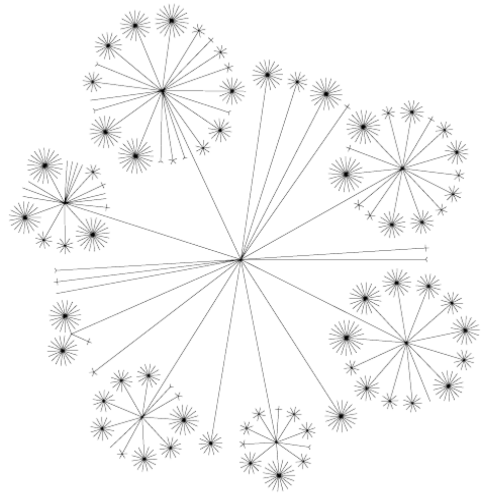
- old topic in graph drawing [Wetherell & Shannon, 1979]
- optimize area but not angular resolution

Angular-resolution-aware tree drawings

- circular drawings for rooted trees [Melançon & Herman, 1998]
- bubble drawings for ordered trees [Grivet et al., 2004]
- balloon drawings for (un)ordered trees [Lin & Yen, 2007]
- closer to our goal but no guarantee of *all* constraints







Related work

Straight-line drawings of trees

- old topic in graph drawing [Wetherell & Shannon, 1979]
- optimize area but not angular resolution

Angular-resolution-aware tree drawings

- circular drawings for rooted trees [Melançon & Herman, 1998]
- bubble drawings for ordered trees [Grivet et al., 2004]
- balloon drawings for (un)ordered trees [Lin & Yen, 2007]
- closer to our goal but no guarantee of *all* constraints

Drawing graphs with smooth curves

- circular arcs for planar graphs with bounded angular resolution [Cheng et al., 2001]
- force-directed curvilinear drawings to improve angular resolution [Finkel & Tamassia, 2004]

Overview

Any tree has a drawing with

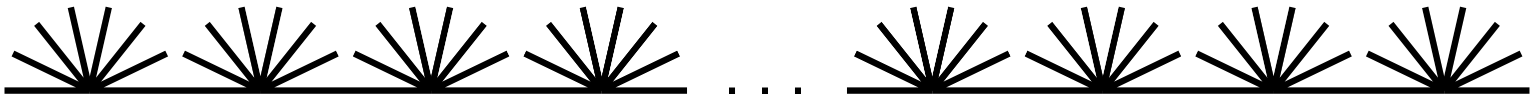
- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	1 ✗
Lombardi edges	✓	2 ✓

Ordered trees and straight-line edges

Fibonacci caterpillars

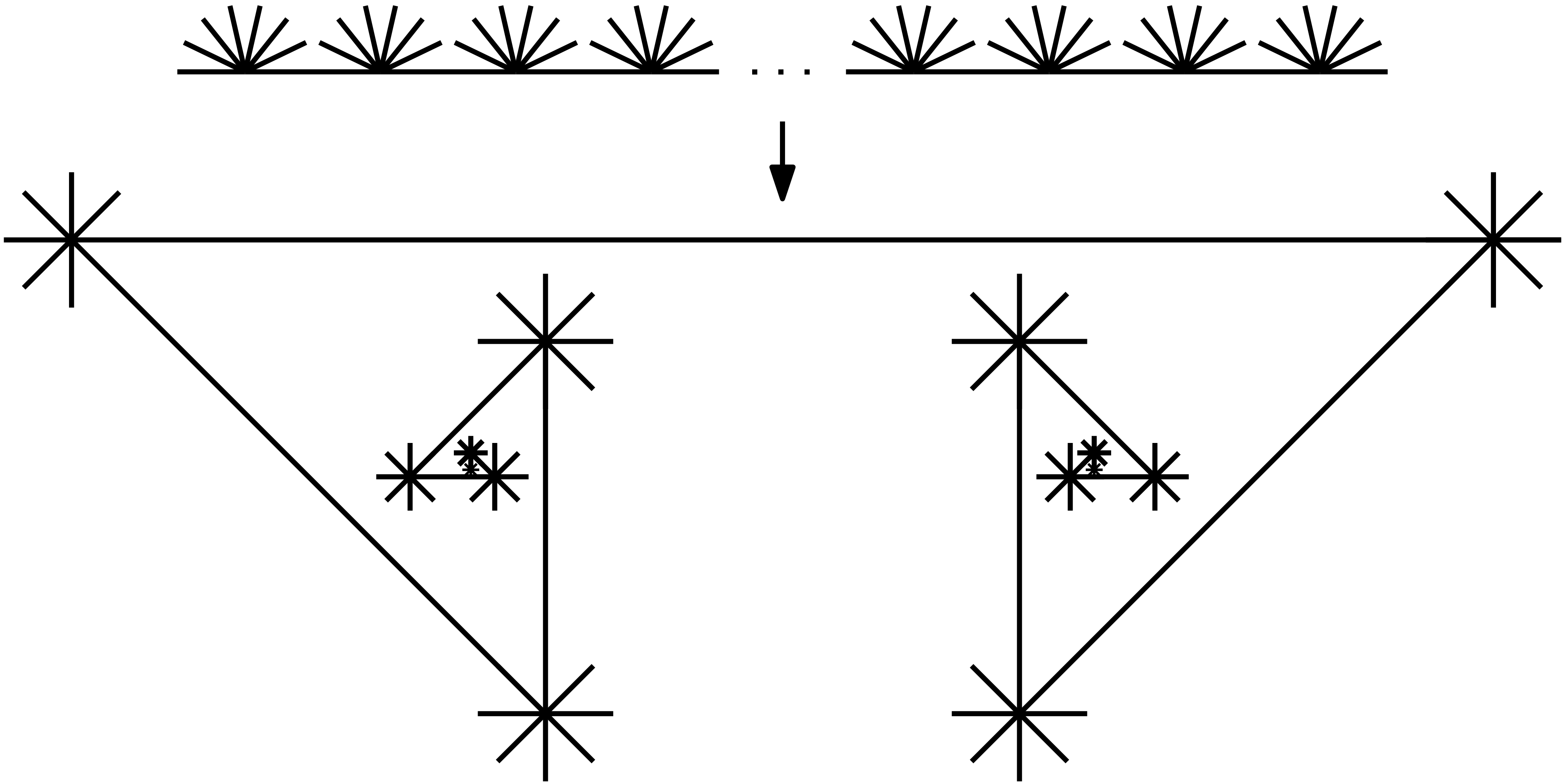
There is an infinite family of trees that require exponential area in any straight-line drawing with perfect angular resolution.



Ordered trees and straight-line edges

Fibonacci caterpillars

There is an infinite family of trees that require exponential area in any straight-line drawing with perfect angular resolution.



Overview

Any tree has a drawing with

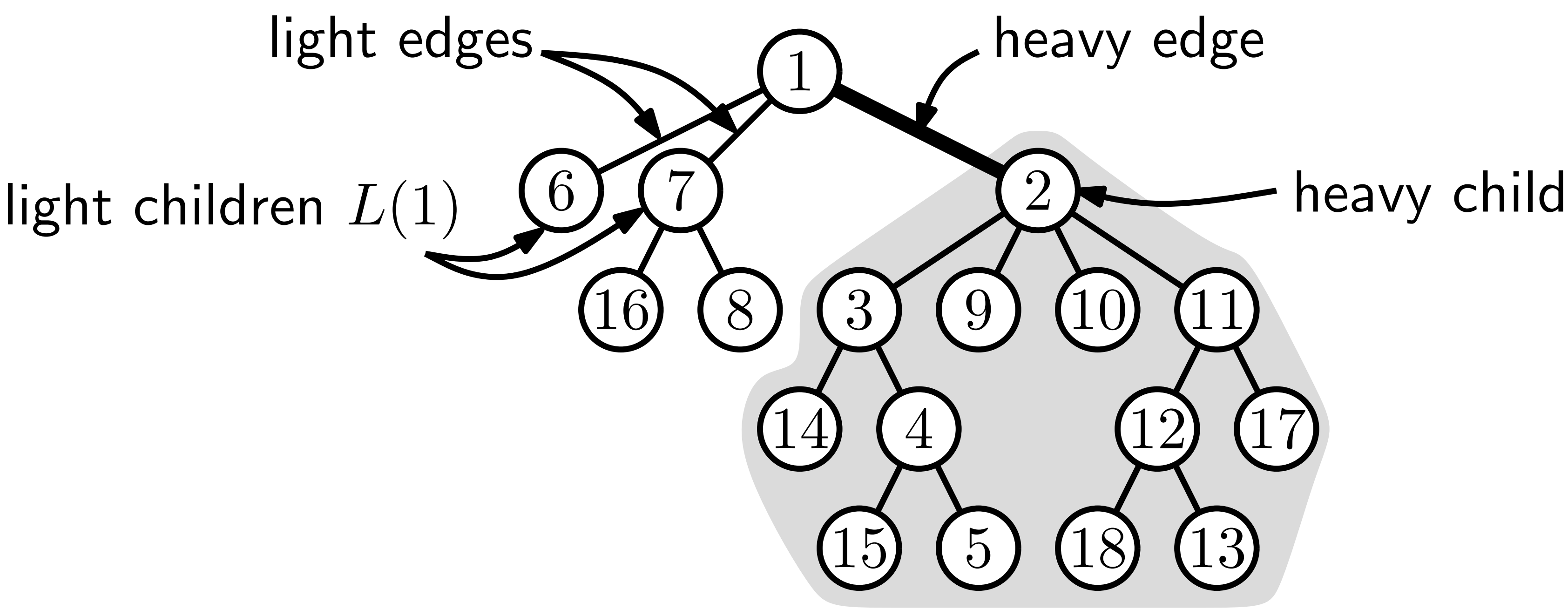
- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	1 ✗
Lombardi edges	✓	2 ✓

Heavy path decomposition (I)

Definition

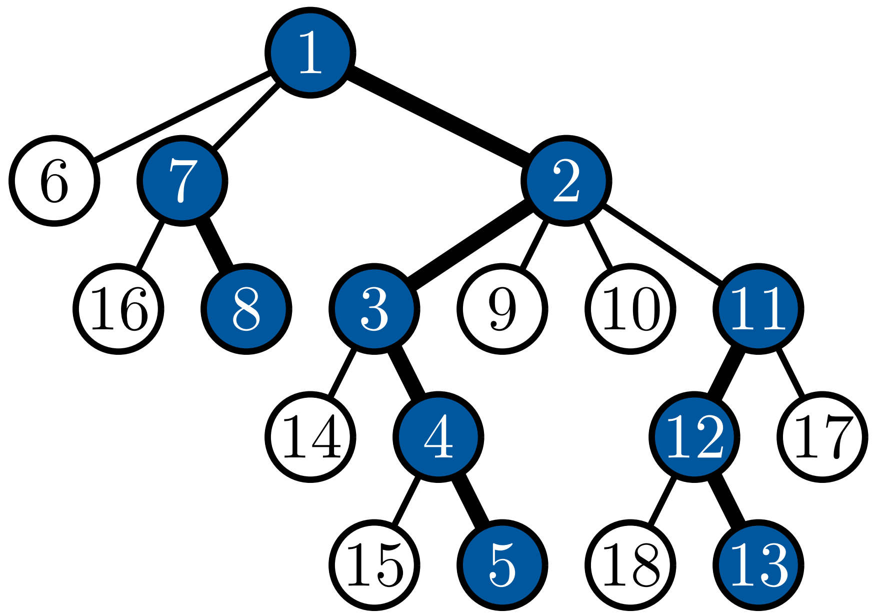
- choose arbitrary root r of tree T
- u is the **heavy child** of v if subtree T_u is largest among children of v
- all other children of v are **light children** in a set $L(v)$
- edge (u, v) from heavy child u to parent v is a **heavy edge**
- edges from light children to parent v are **light edges**



Heavy path decomposition (I)

Definition

- choose arbitrary root r of tree T
- u is the **heavy child** of v if subtree T_u is largest among children of v
- all other children of v are **light children** in a set $L(v)$
- edge (u, v) from heavy child u to parent v is a **heavy edge**
- edges from light children to parent v are **light edges**



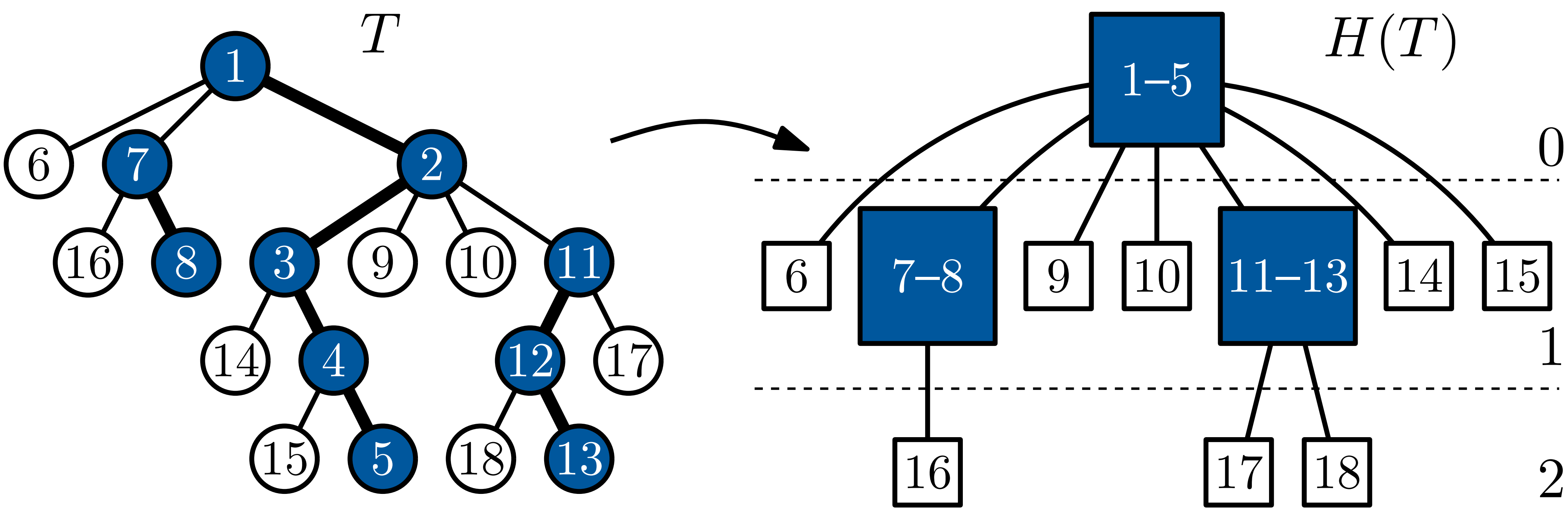
Heavy path decomposition (II)

Definition

The set of heavy edges induces the **heavy path decomposition** (HPD) of T into heavy paths and singleton nodes. [Harel & Tarjan, 1984]

Property

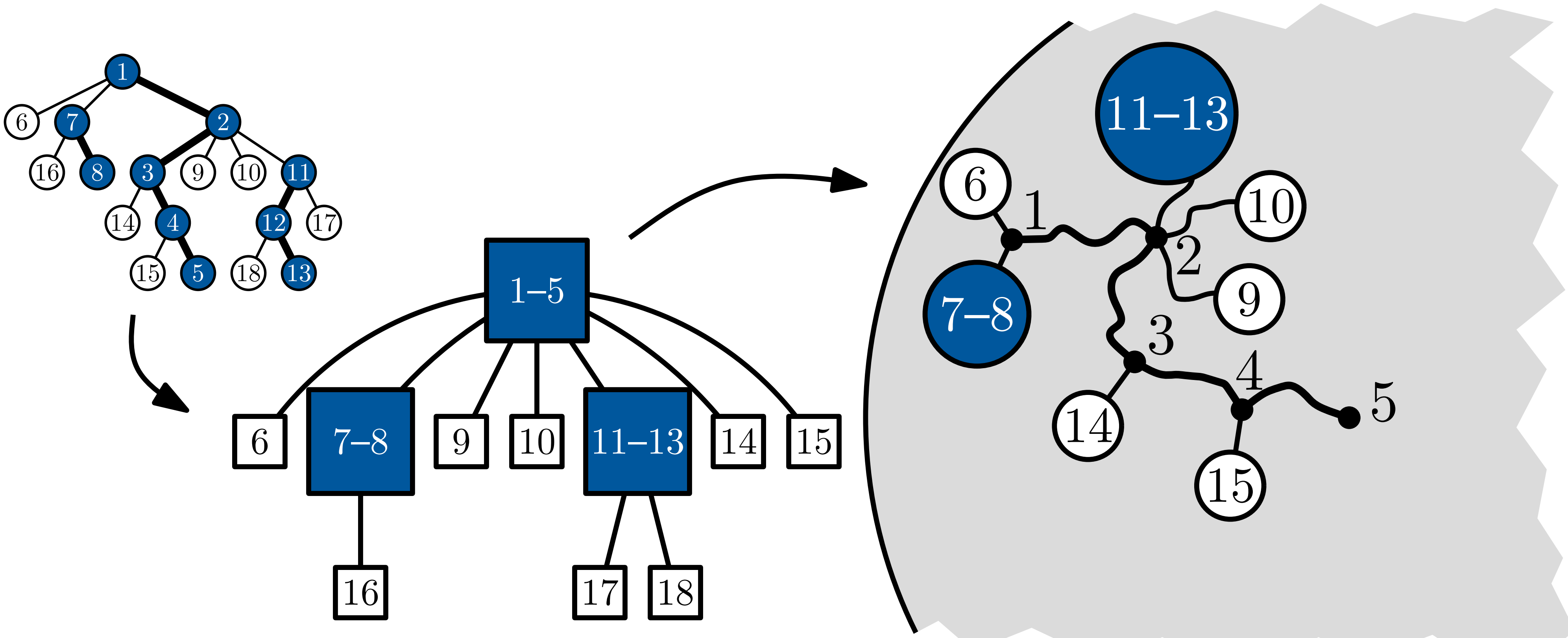
The HPD induces a decomposition tree $H(T)$ of height $h \leq \log_2 n$.



Drawing ordered trees

Sketch of the algorithm

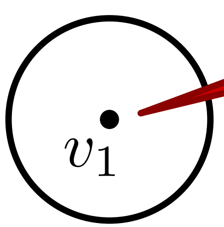
- draw each heavy path P of T within a disk of polynomial area
- each heavy path at level $j \geq 1$ of $H(T)$ is a light child of a heavy path at level $j-1$
- given drawings of all heavy paths at level j recursively construct drawings of all heavy paths at level $j-1$



Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} $(c \in \mathbb{N})$
- disks D_i containing v_i and all subtrees of light children of v_i

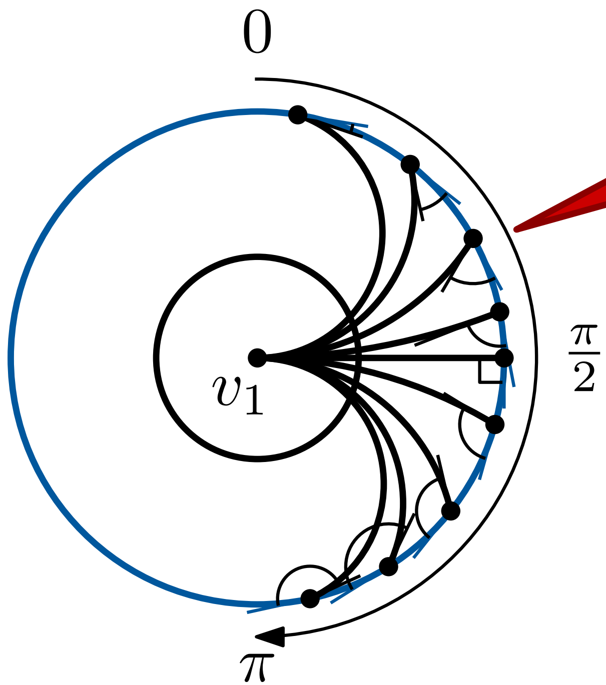


place leaf v_1 and disk D_1 in the center

Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} $(c \in \mathbb{N})$
- disks D_i containing v_i and all subtrees of light children of v_i

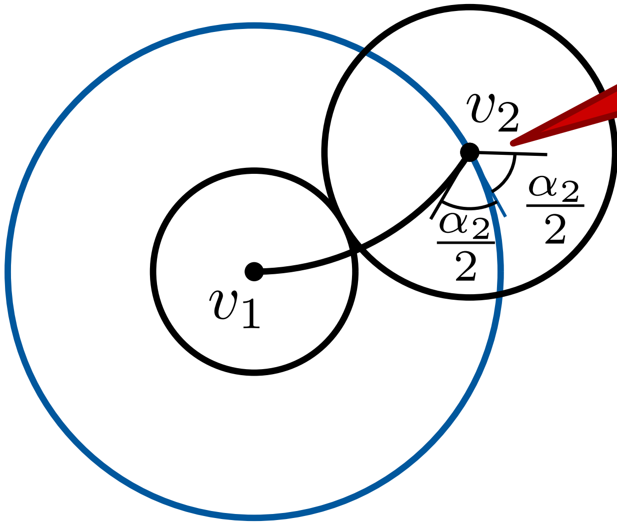


place v_2 (and D_2) on a concentric circle based on $\frac{\alpha_2}{2} \in [0, \pi]$

Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} $(c \in \mathbb{N})$
- disks D_i containing v_i and all subtrees of light children of v_i

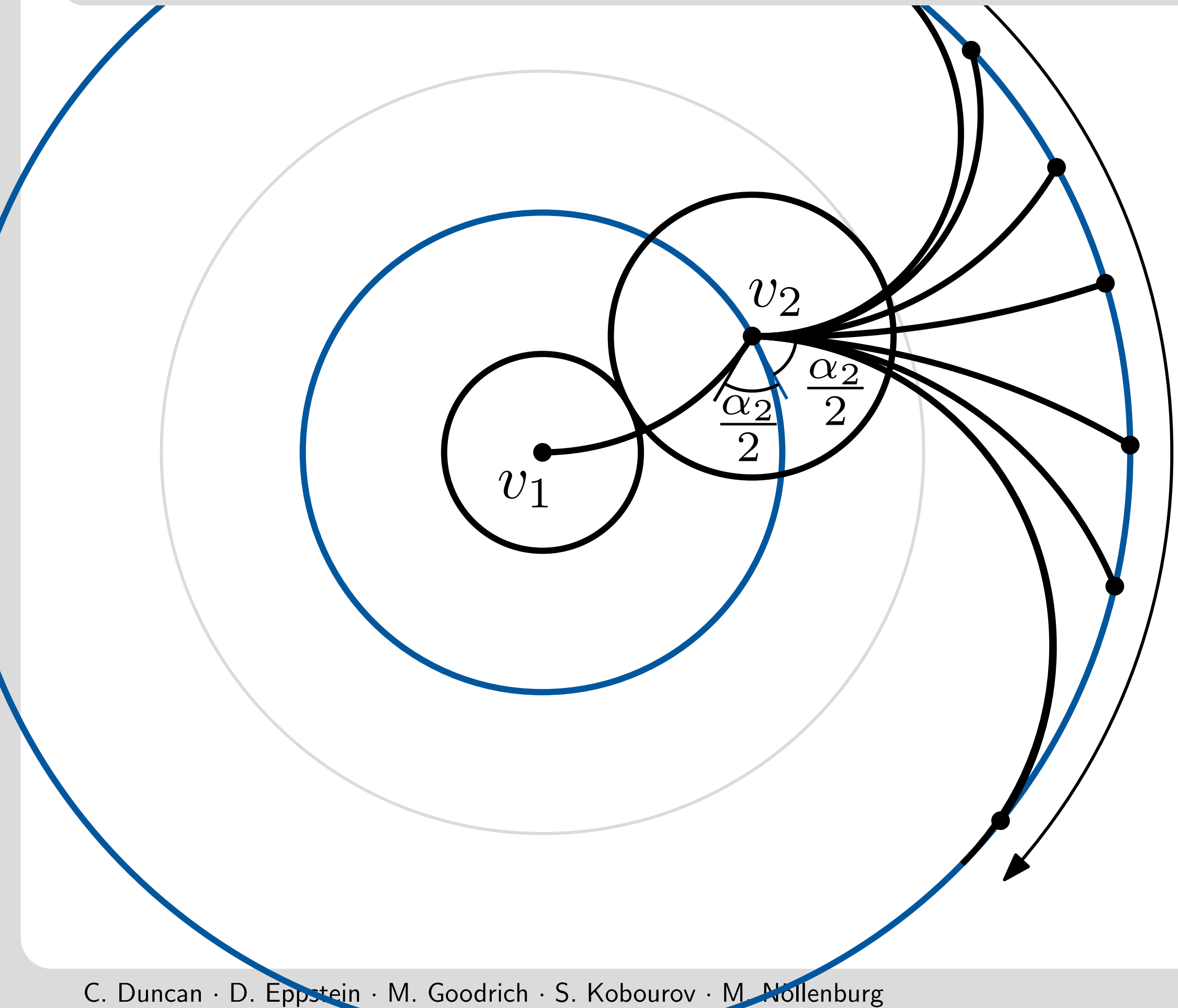


place v_2 (and D_2) on a concentric circle based on $\frac{\alpha_2}{2} \in [0, \pi]$

Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} $(c \in \mathbb{N})$
- disks D_i containing v_i and all subtrees of light children of v_i

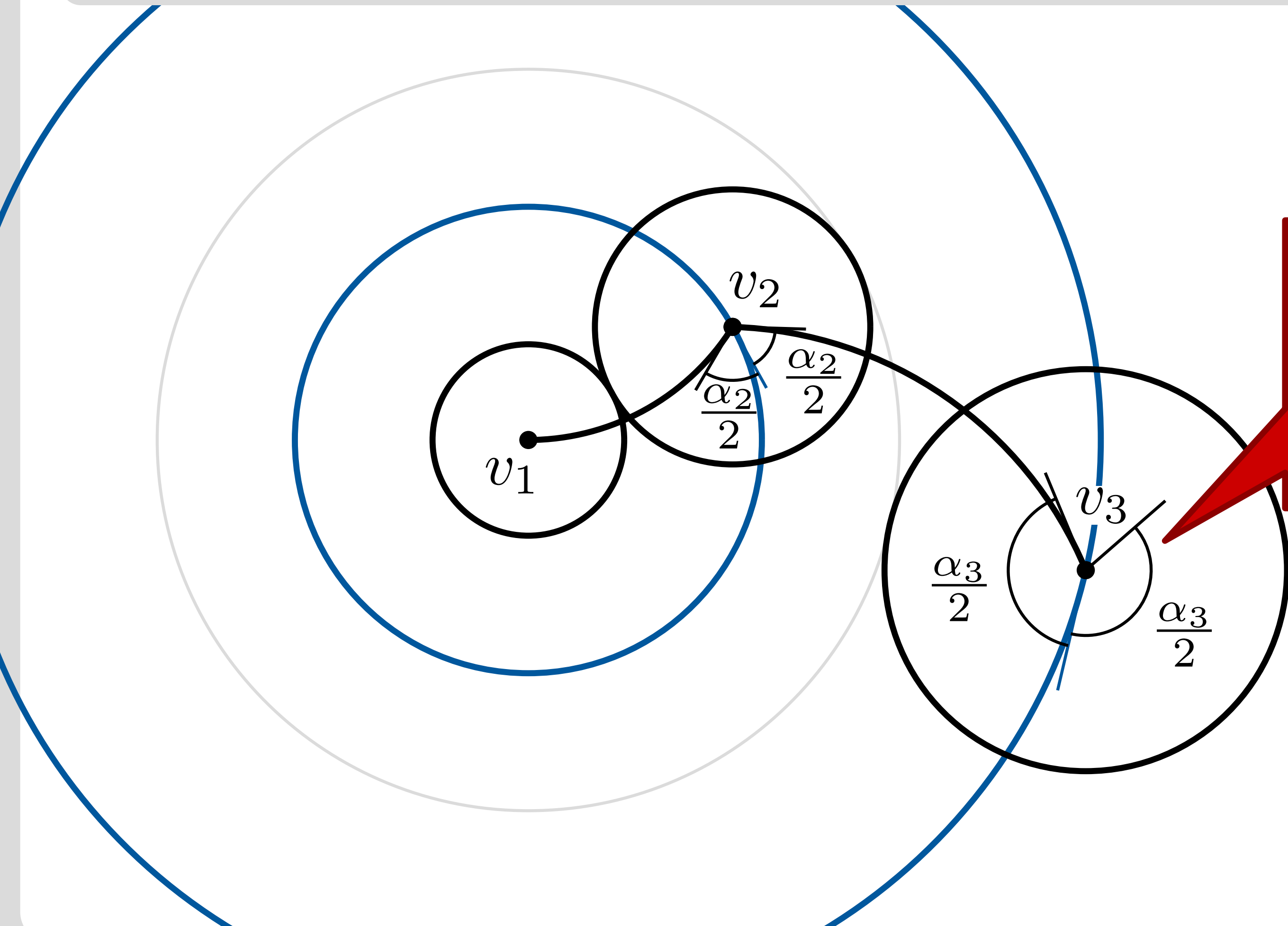


place v_3 (and D_3) on a concentric circle based on $\frac{\alpha_3}{2} \in [0, \pi]$

Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} $(c \in \mathbb{N})$
- disks D_i containing v_i and all subtrees of light children of v_i

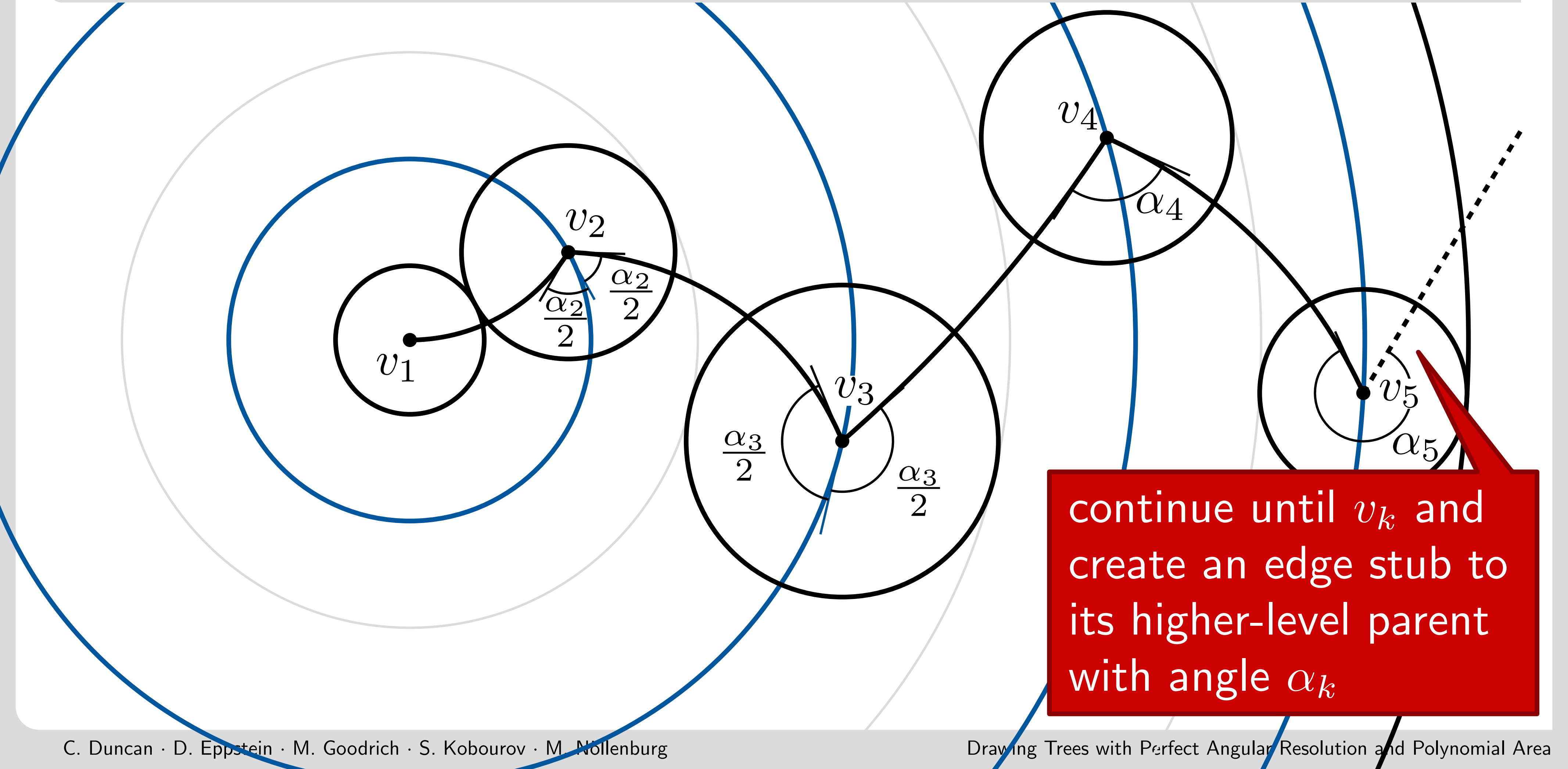


place v_3 (and D_3) on a concentric circle based on $\frac{\alpha_3}{2} \in [0, \pi]$

Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} ($c \in \mathbb{N}$)
- disks D_i containing v_i and all subtrees of light children of v_i

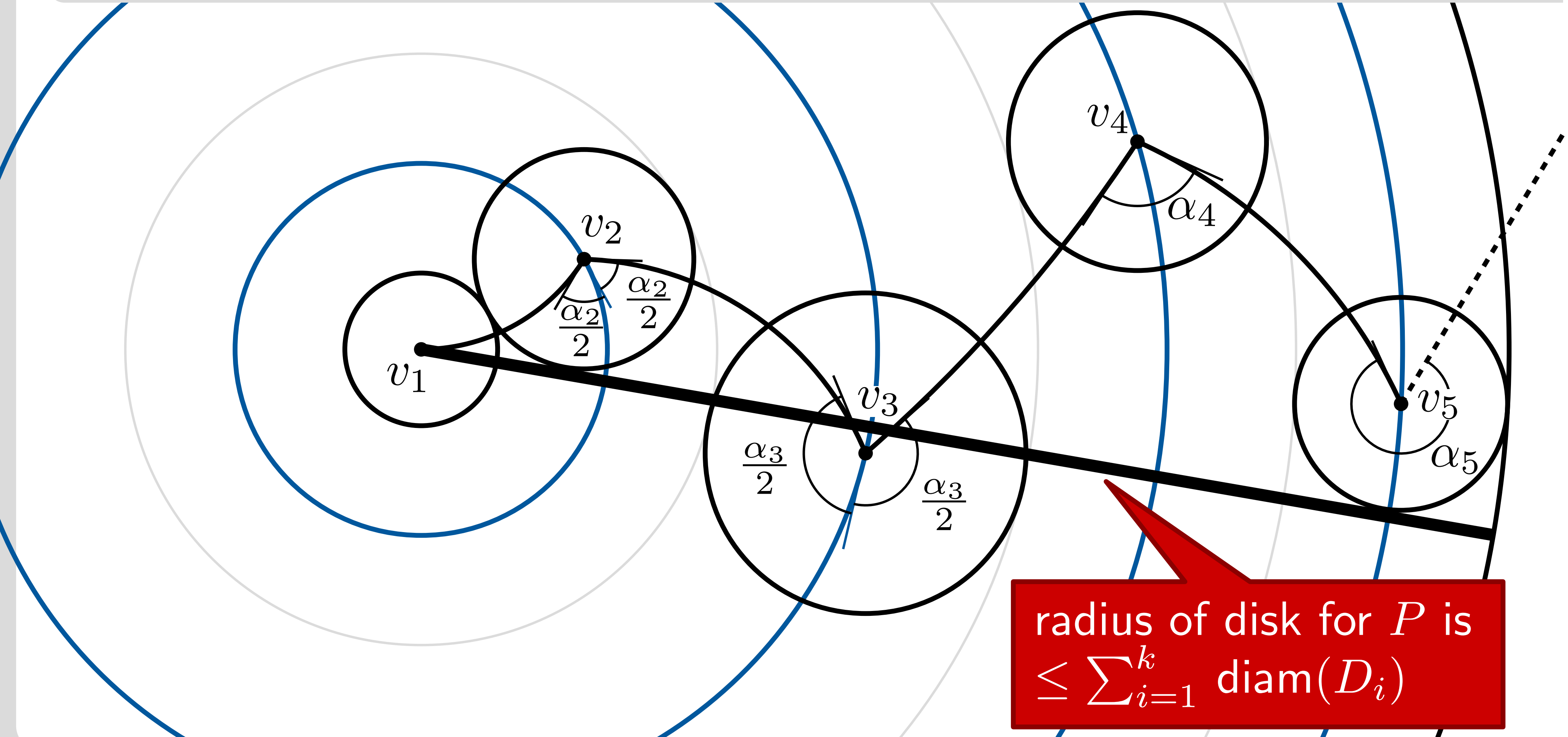


continue until v_k and create an edge stub to its higher-level parent with angle α_k

Step 1: Drawing heavy paths

Input

- heavy path $P = (v_1, v_2, \ldots, v_k)$ rooted at v_k
- required angles $\alpha_i = c \cdot \frac{2\pi}{\deg(v_i)}$ between $v_{i-1}v_i$ and v_iv_{i+1} $(c \in \mathbb{N})$
- disks D_i containing v_i and all subtrees of light children of v_i



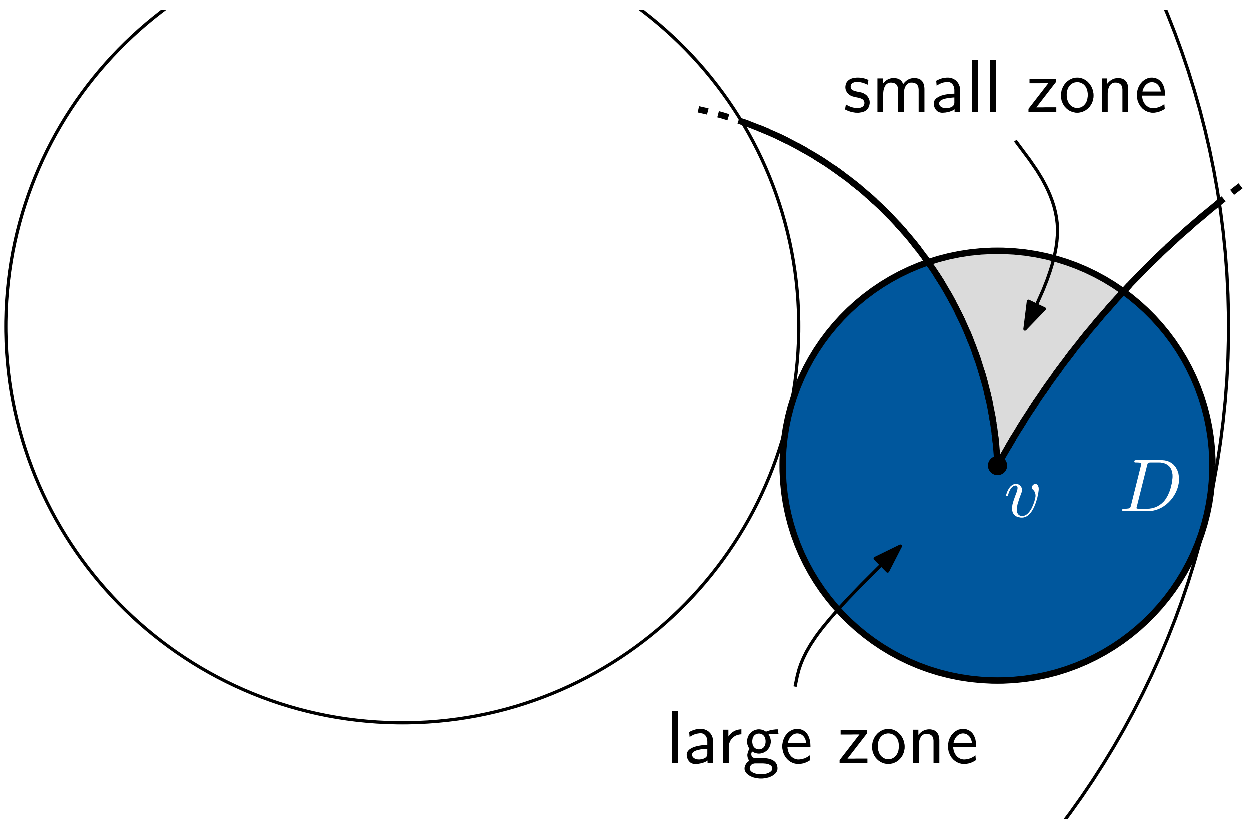
radius of disk for P is $\leq \sum_{i=1}^{k} \text{diam}(D_i)$

Step 2: Drawing light children

Distinguish two cases

Given a node v of a heavy path draw its light children in the given order within disk D. The two heavy edges incident to v divide D into

- **small zone** with opening angle $\leq \pi$
- **large zone** with opening angle $> \pi$

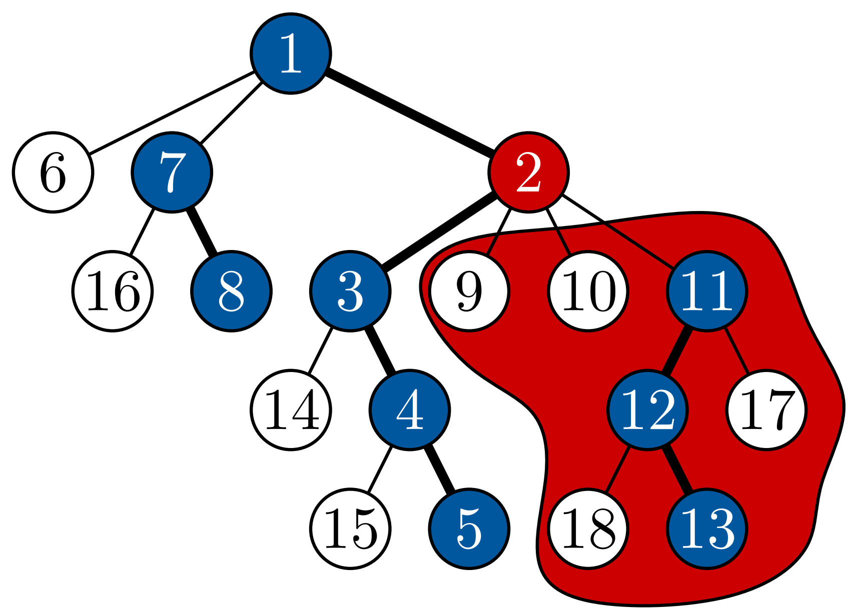


Choose the right disk radii

Definition

For heavy path node v at level j of HPD $H(T)$ define disk radius

$$r_v = 4^{h-j}(1 + \sum_{u \in L(v)} |T_u|).$$



Choose the right disk radii

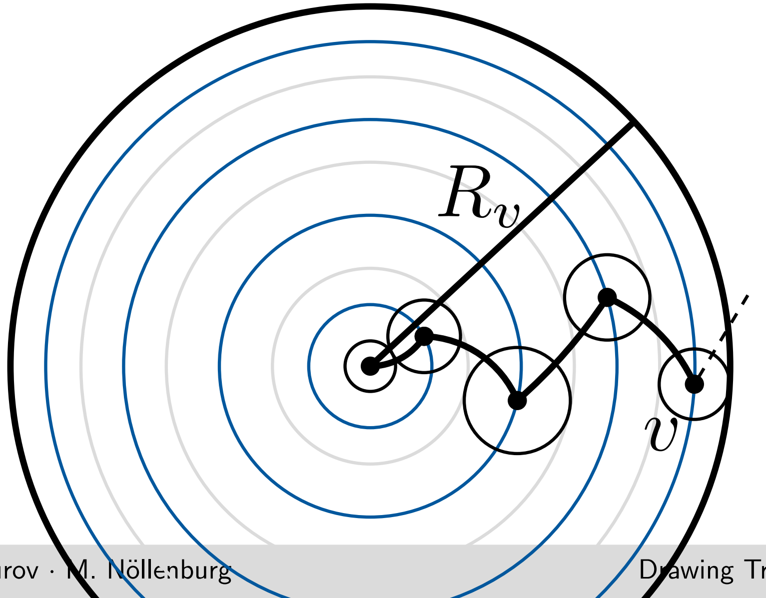
Definition

For heavy path node v at level j of HPD $H(T)$ define disk radius

$$r_v = 4^{h-j}(1 + \sum_{u \in L(v)} |T_u|).$$

Induction hypothesis

A heavy path P with root v at level j of $H(T)$ is contained in a disk of radius $R_v = 2 \cdot 4^{h-j}|T_v|$.



Choose the right disk radii

Definition

For heavy path node v at level j of HPD $H(T)$ define disk radius

$$r_v = 4^{h-j}(1 + \sum_{u \in L(v)} |T_u|).$$

Induction hypothesis

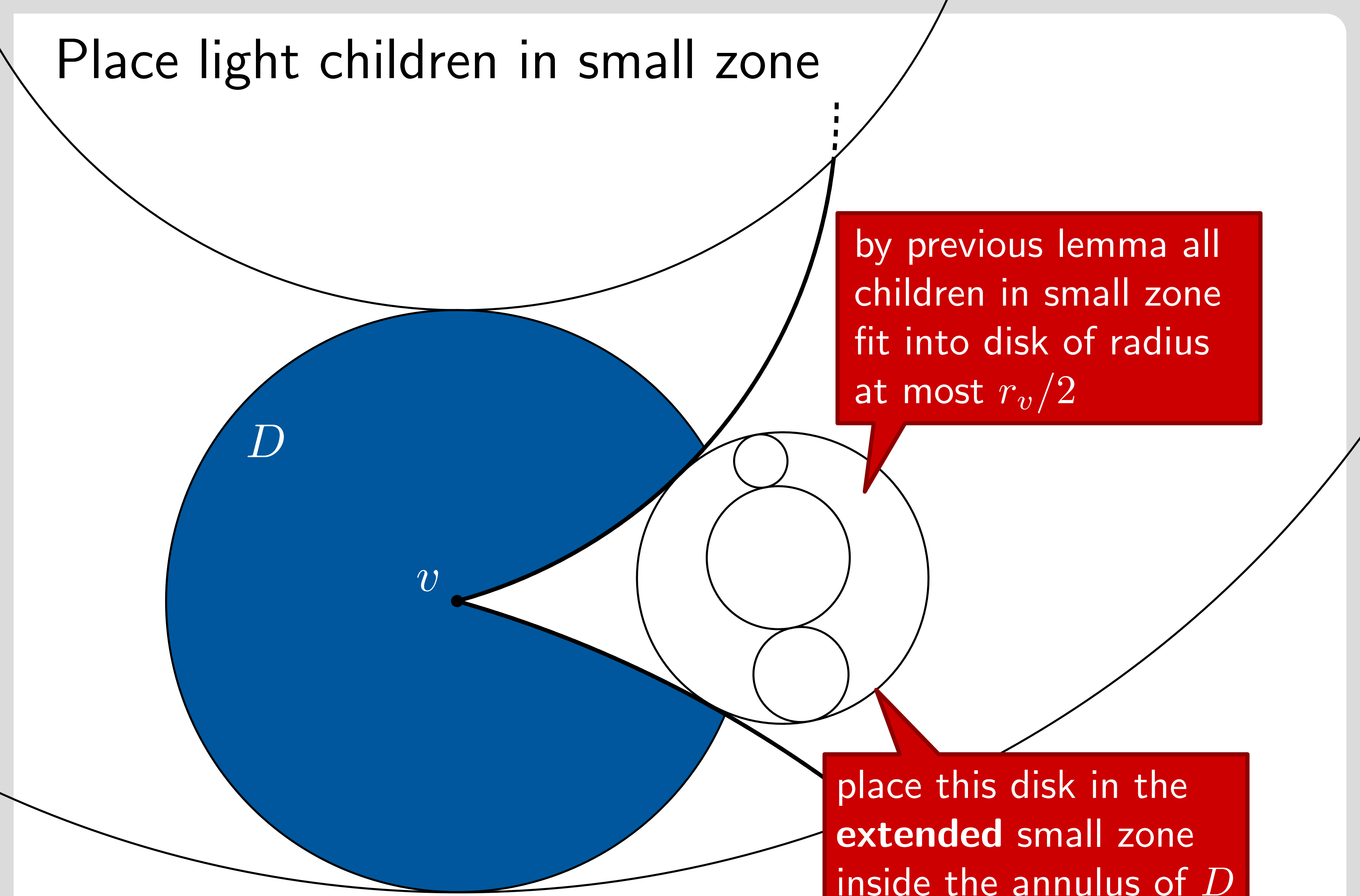
A heavy path P with root v at level j of $H(T)$ is contained in a disk of radius $R_v = 2 \cdot 4^{h-j}|T_v|$.

Lemma

Every light child of a heavy path node v at level j is the root of a heavy path at level $j+1$. Hence

$$\sum_{u \in L(v)} R_u = \sum_{u \in L(v)} (2 \cdot 4^{h-j-1}|T_u|) = \frac{1}{2} \cdot 4^{h-j} \sum_{u \in L(v)} |T_u| \leq \frac{r_v}{2}.$$

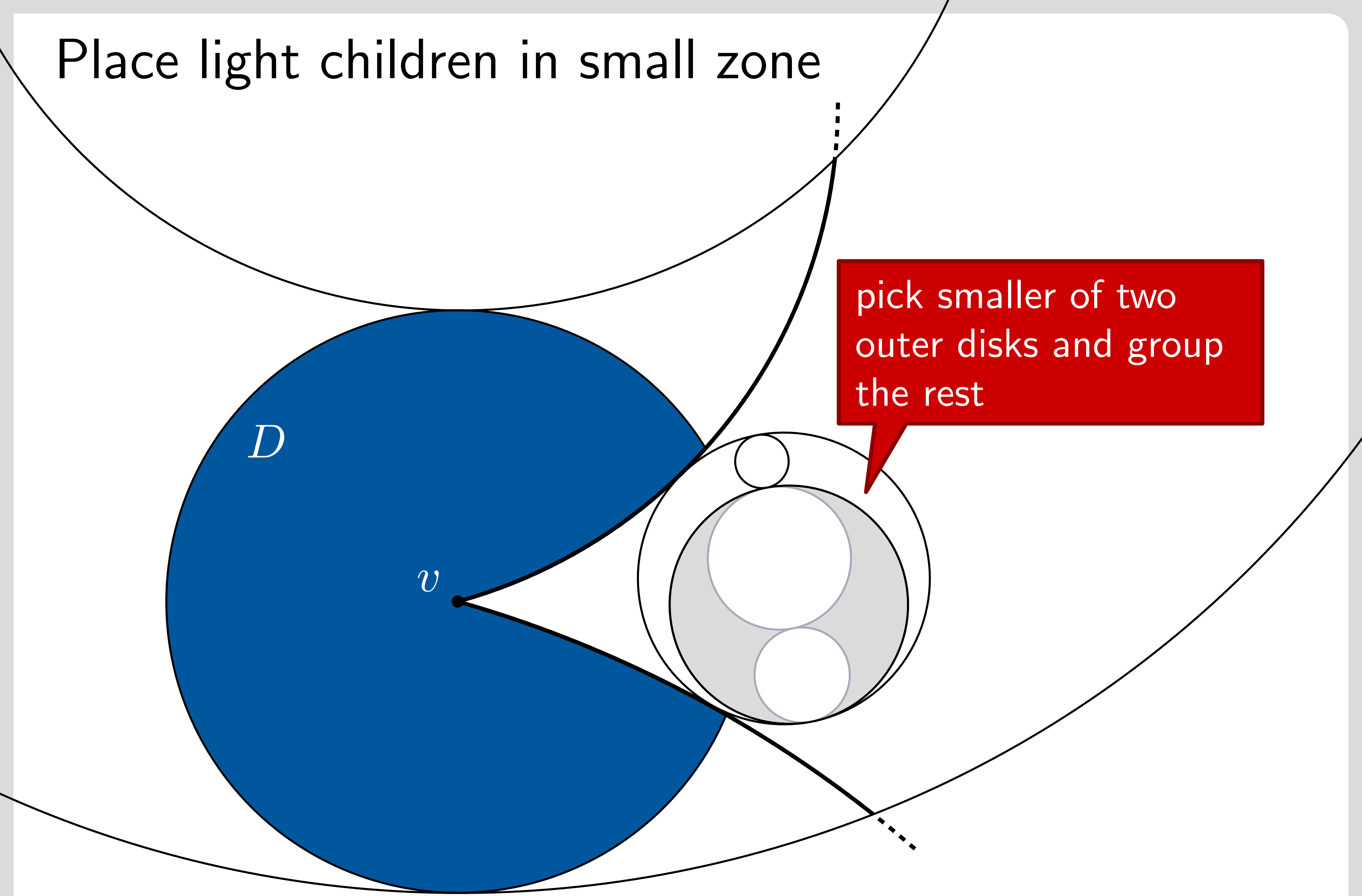
Place light children in small zone



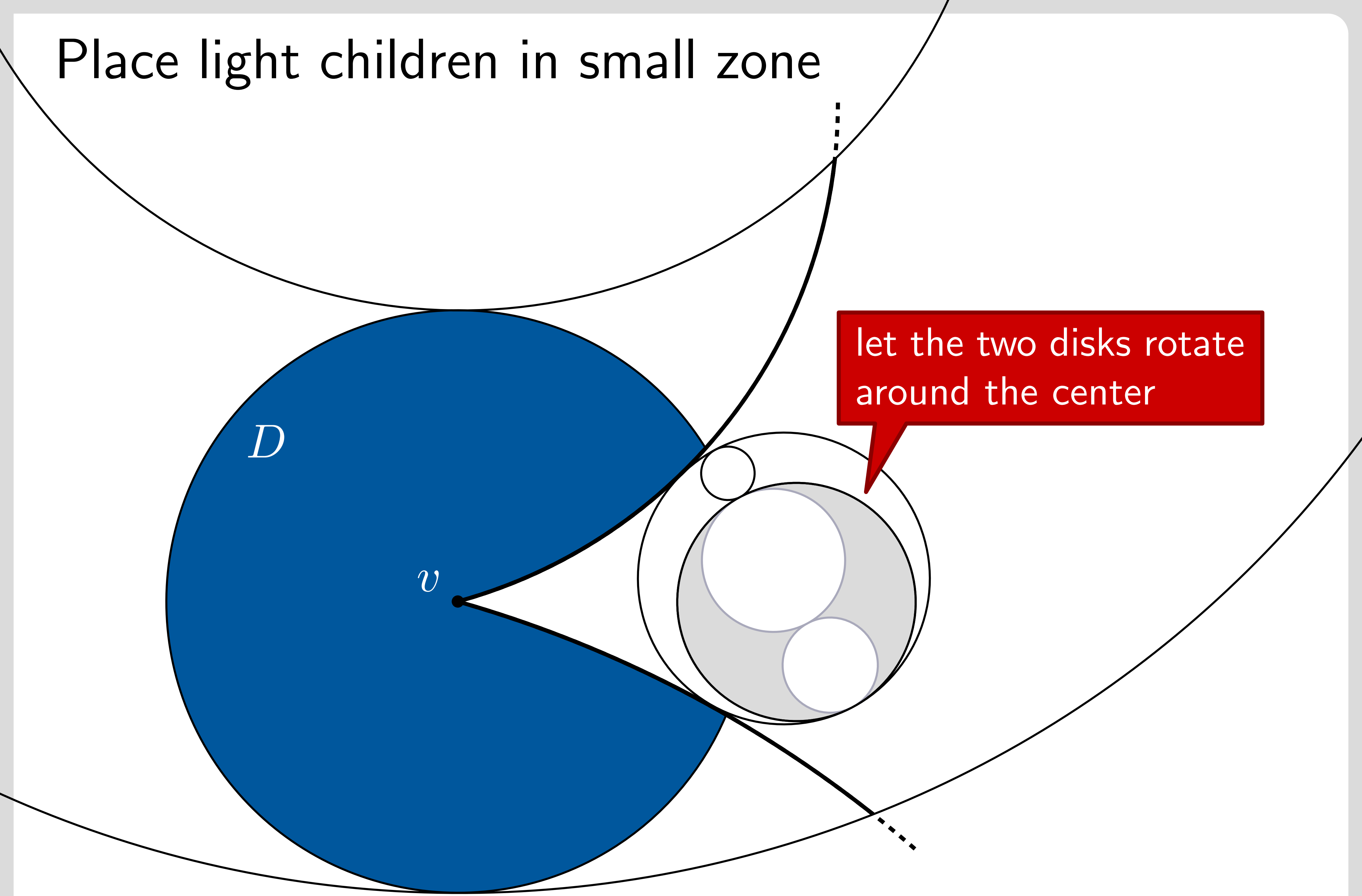
by previous lemma all children in small zone fit into disk of radius at most $r_v/2$

place this disk in the **extended** small zone inside the annulus of D

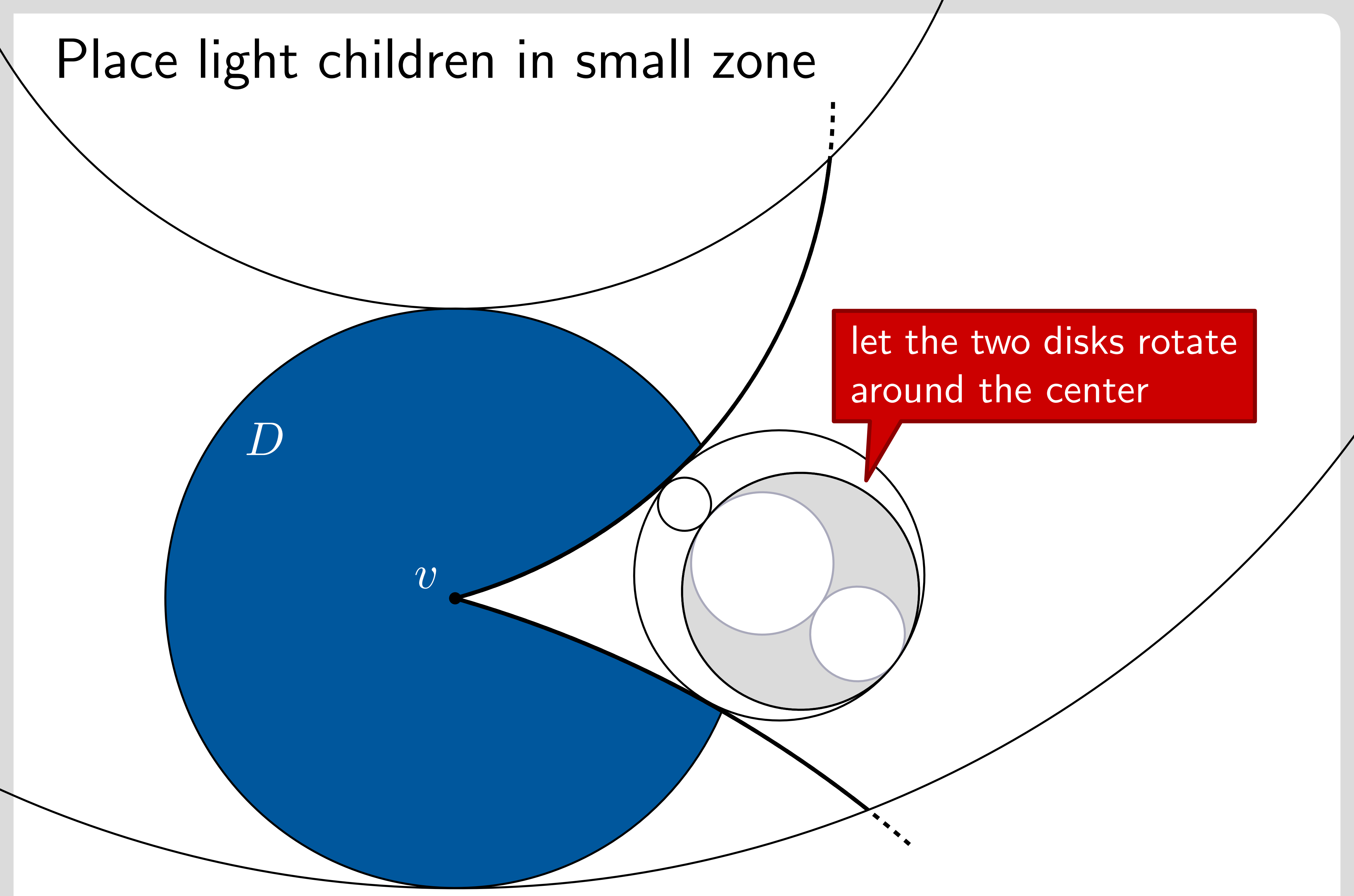
Place light children in small zone



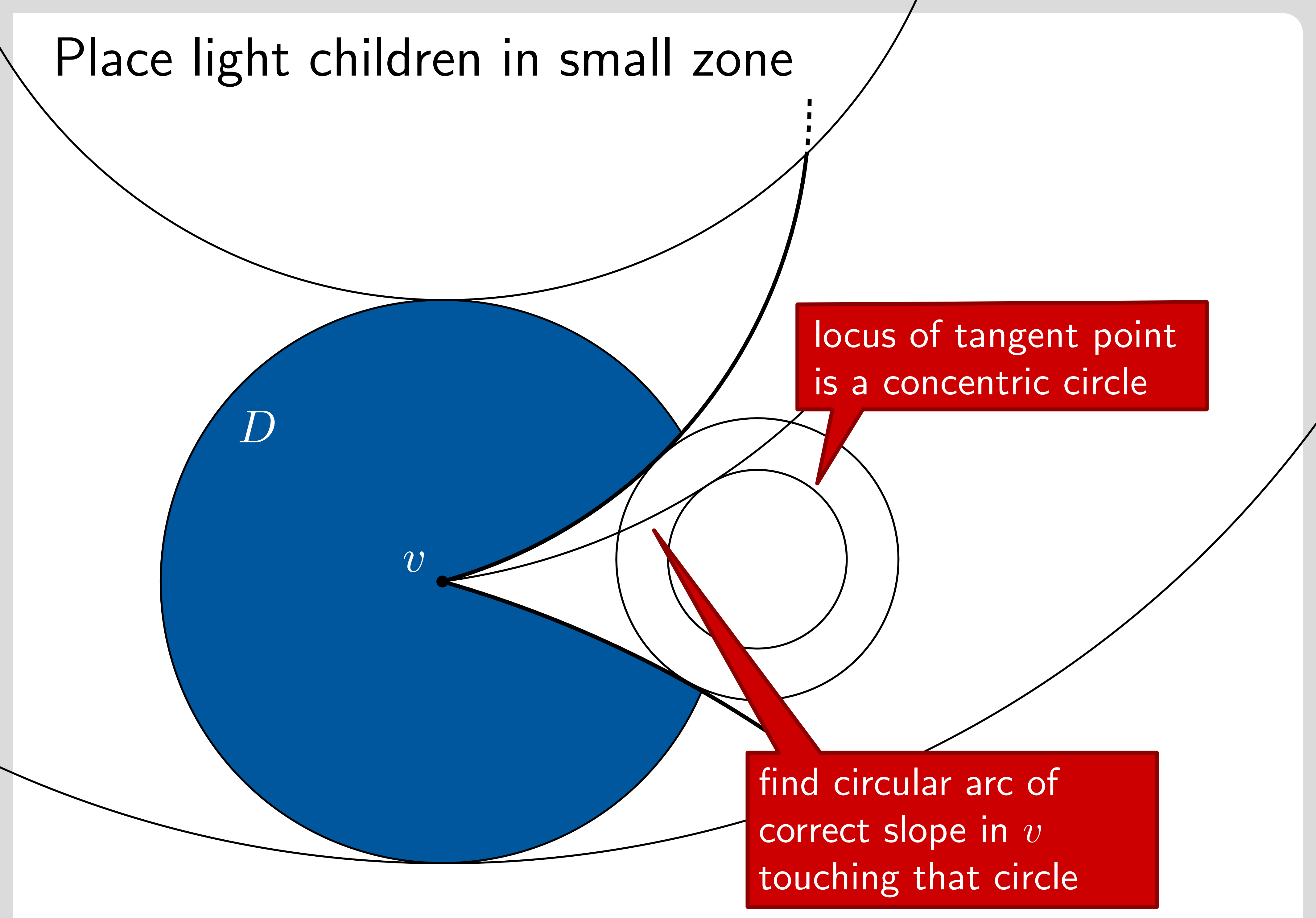
Place light children in small zone



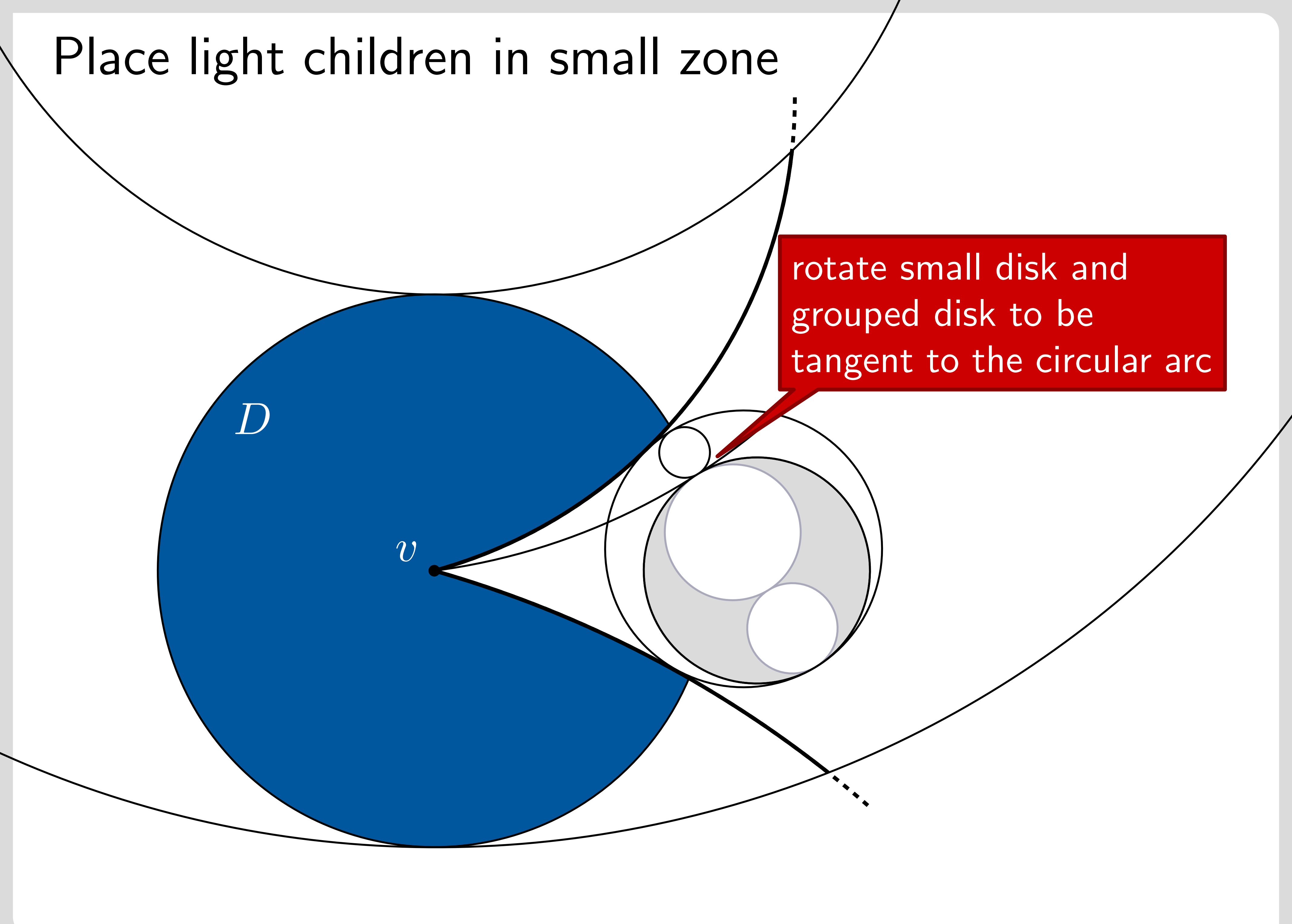
Place light children in small zone



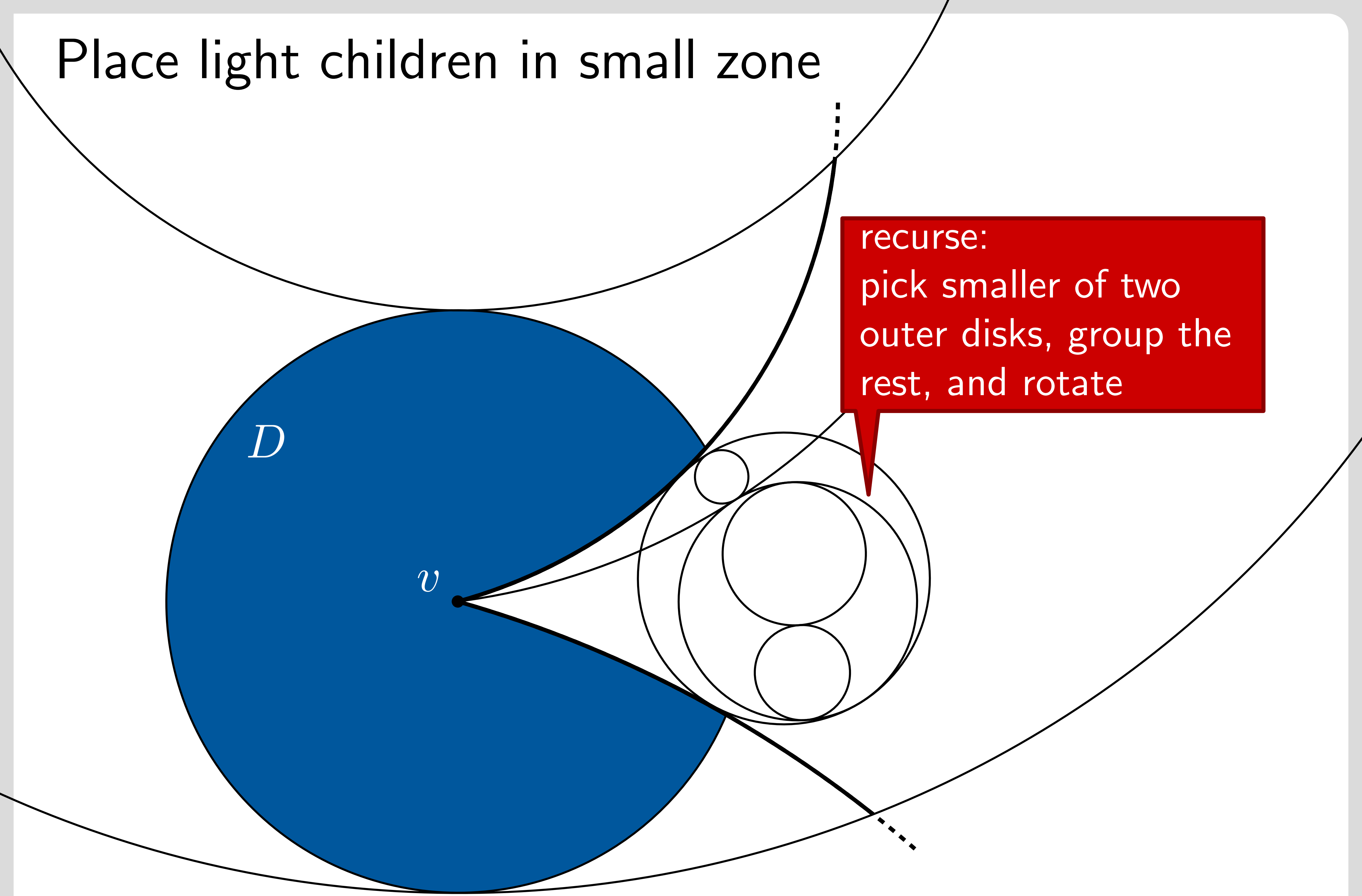
Place light children in small zone



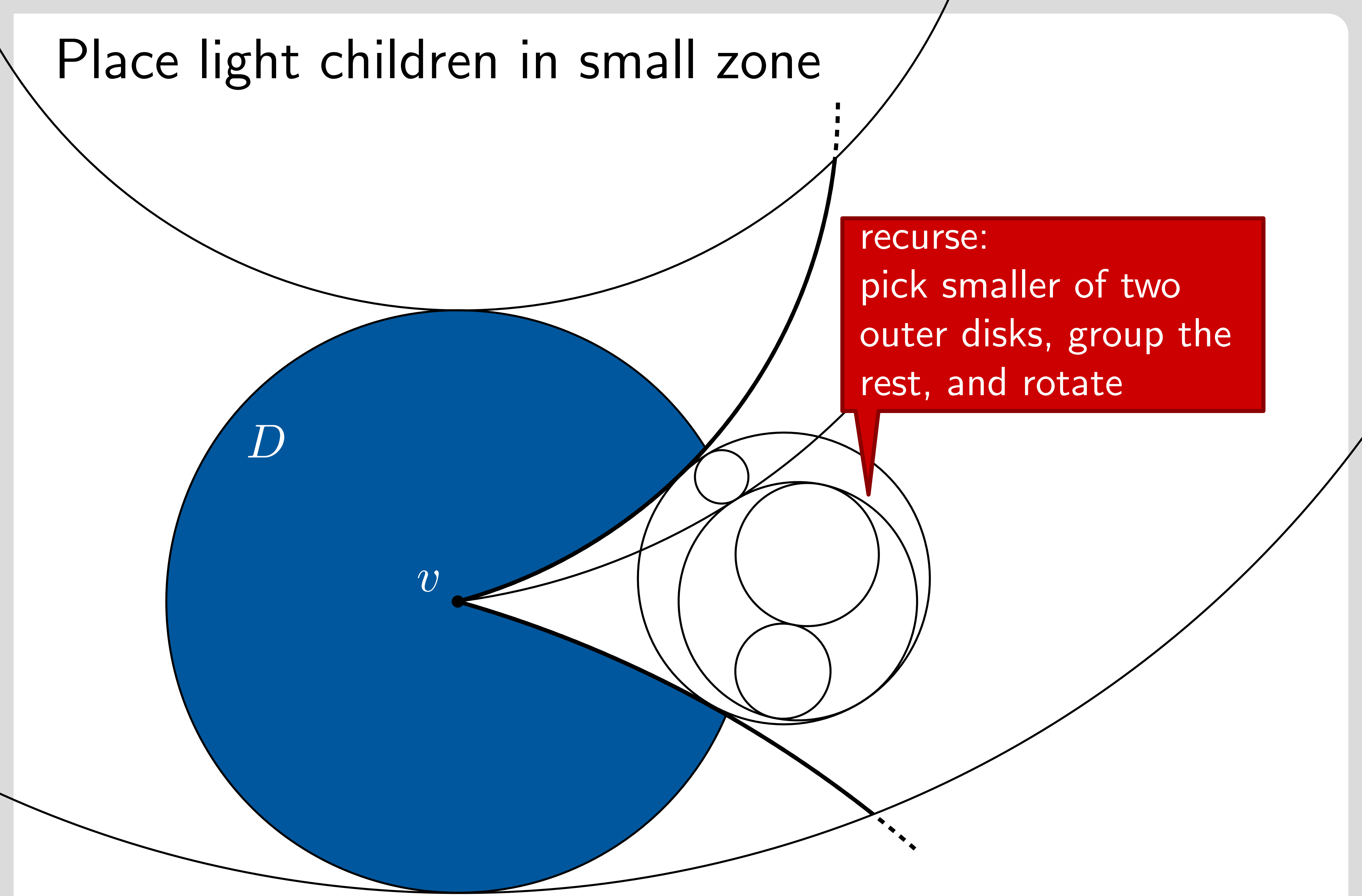
Place light children in small zone



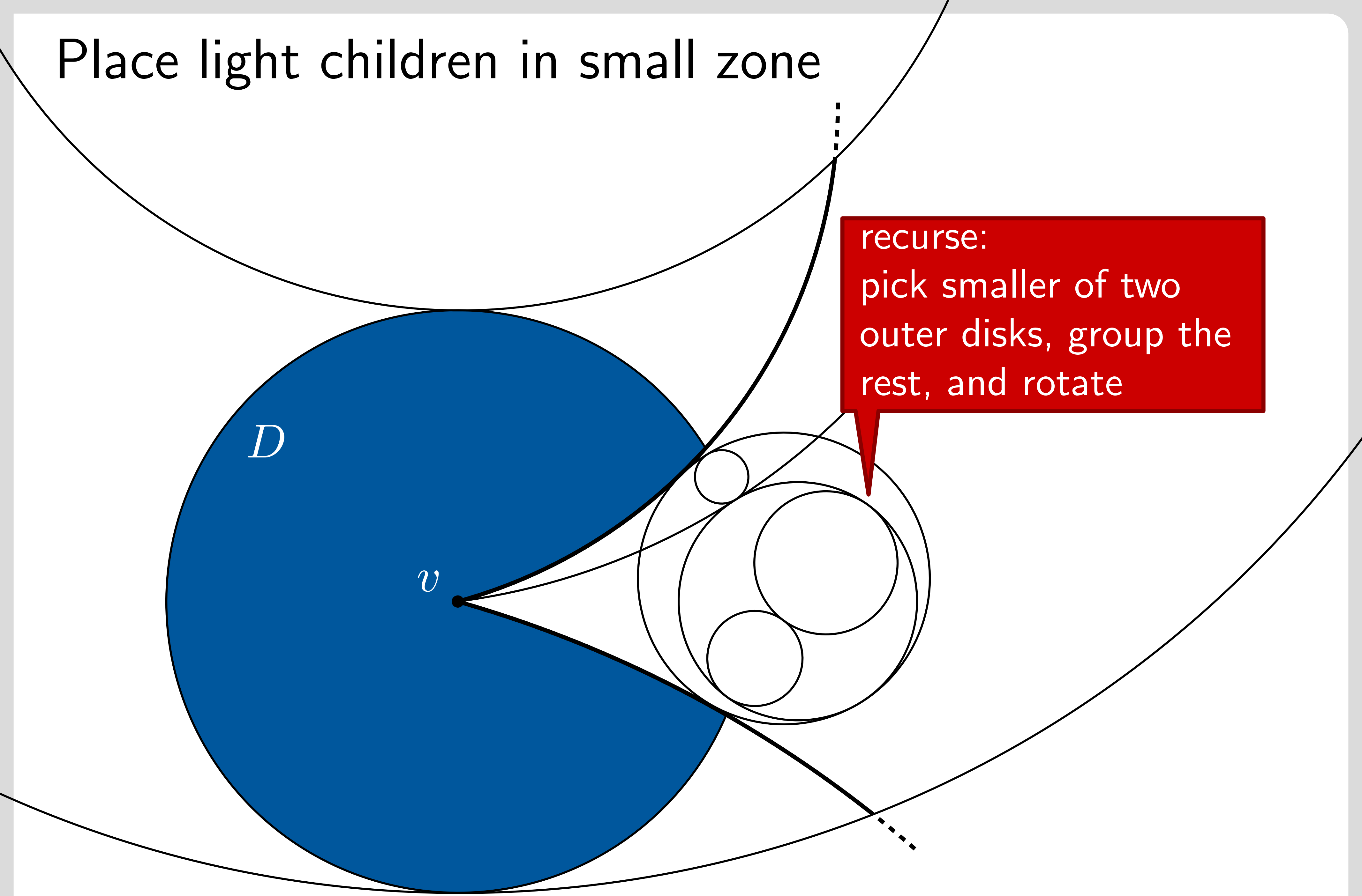
Place light children in small zone



Place light children in small zone

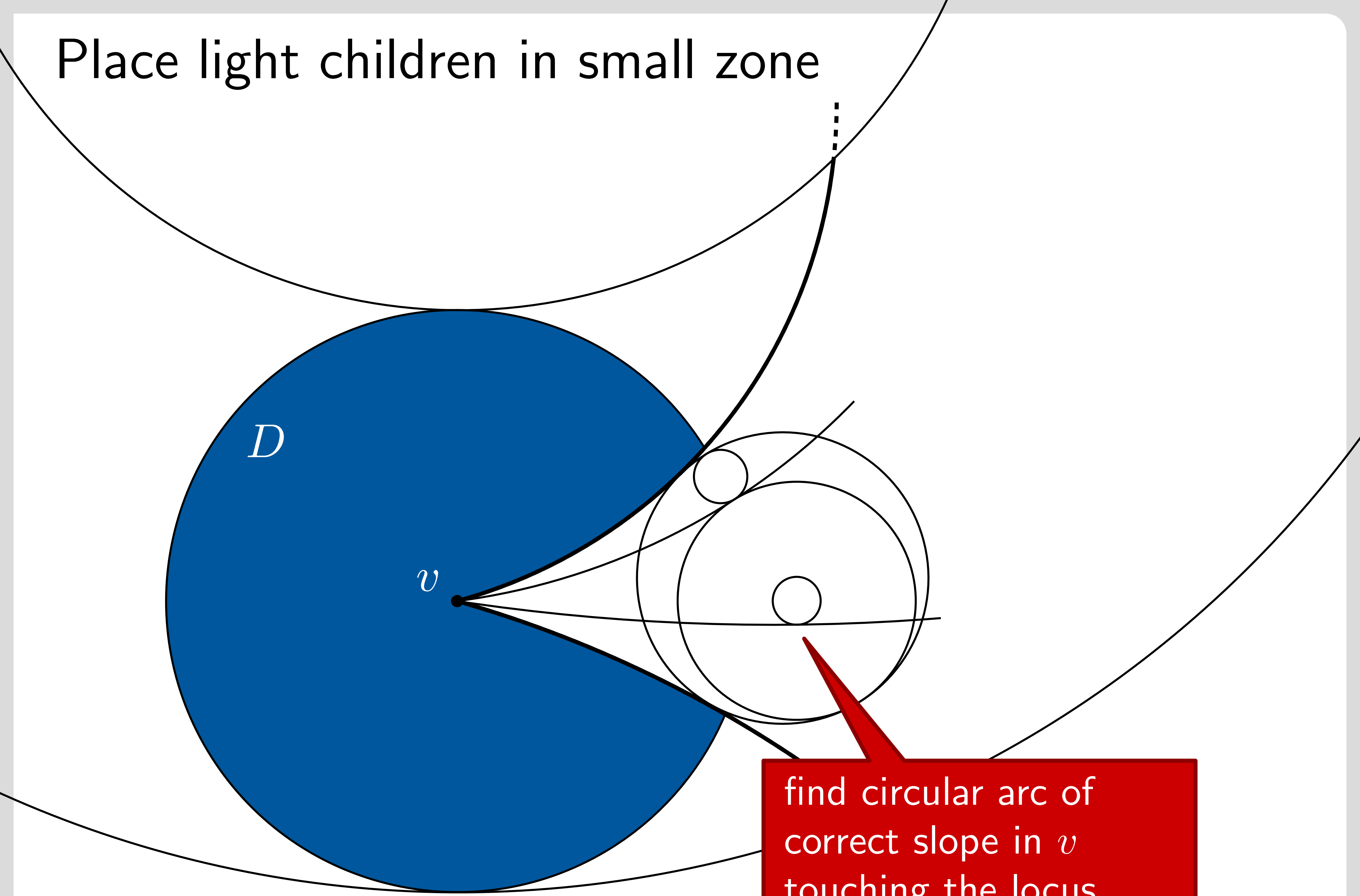


Place light children in small zone



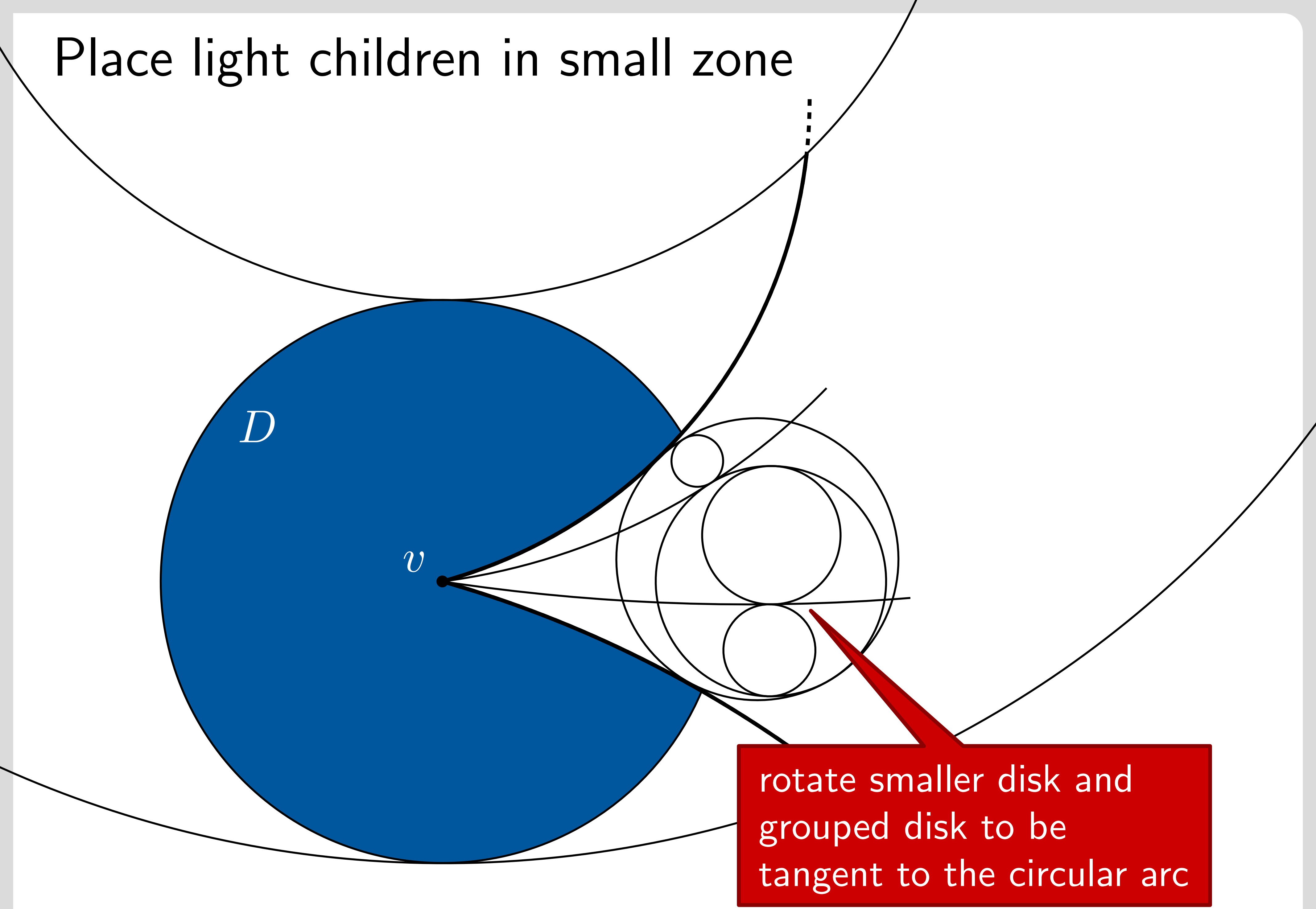
recurse:
pick smaller of two outer disks, group the rest, and rotate

Place light children in small zone

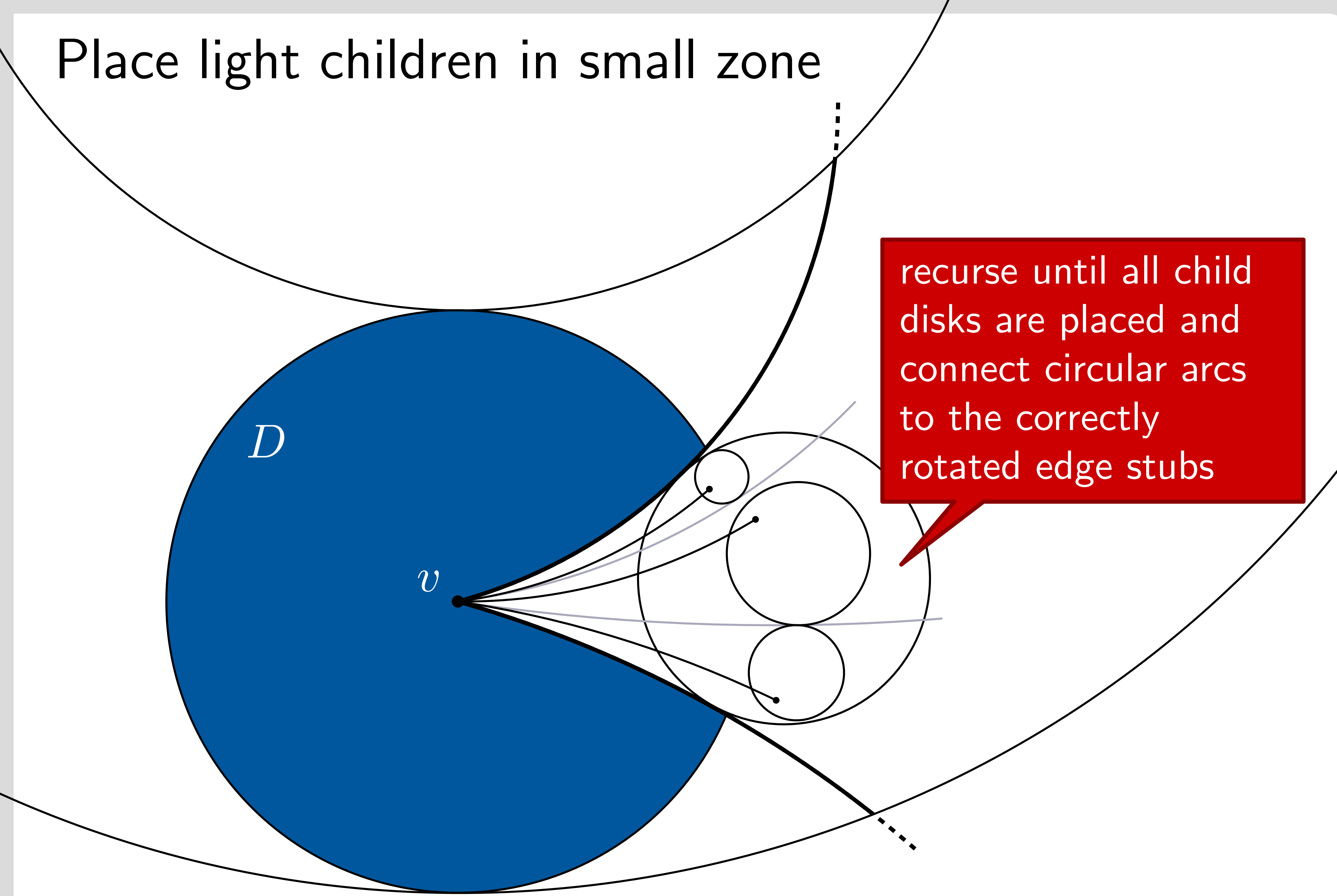


find circular arc of correct slope in v touching the locus circle of tangent point

Place light children in small zone

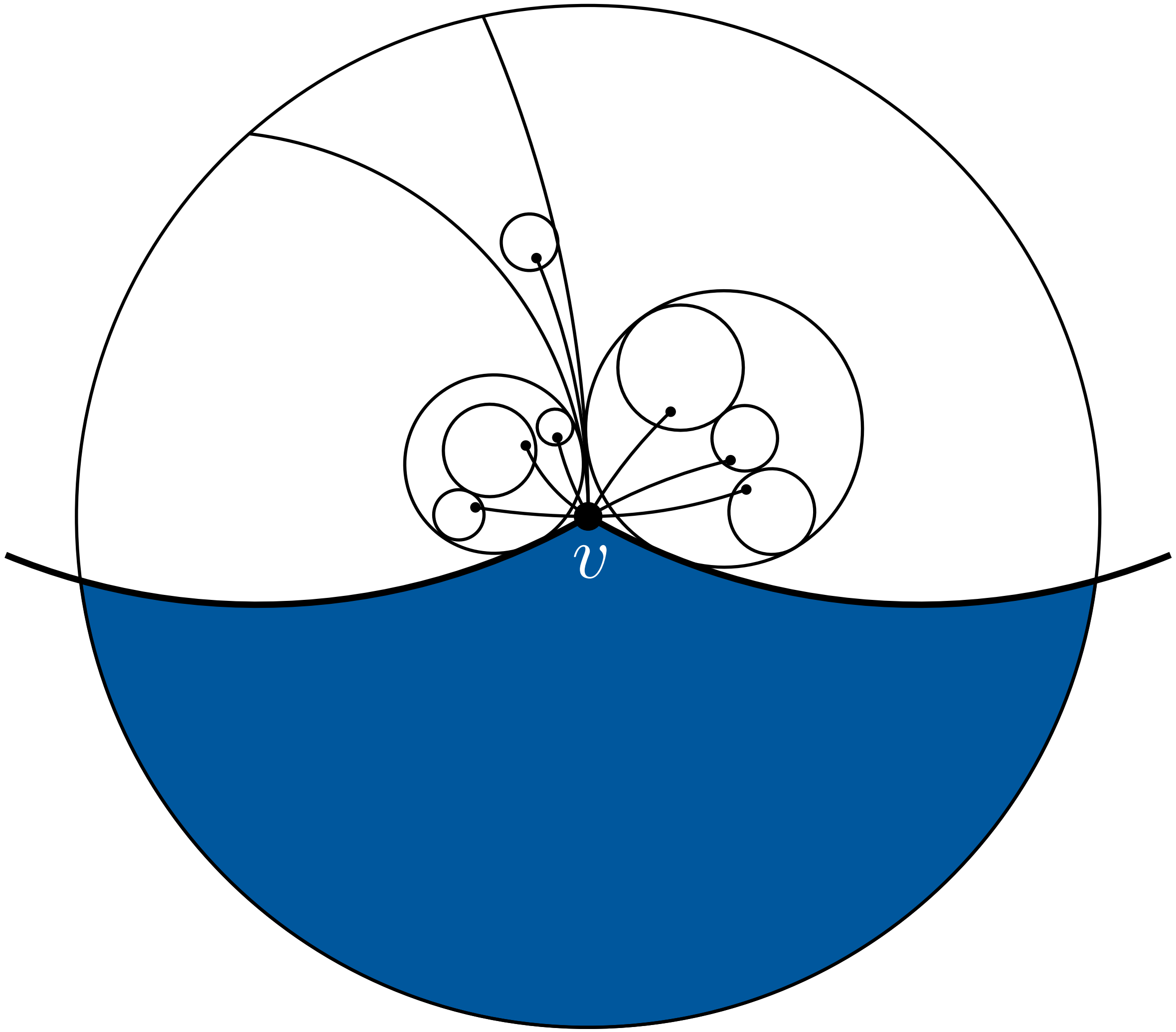


Place light children in small zone



recurse until all child disks are placed and connect circular arcs to the correctly rotated edge stubs

Place light children in large zone



placement similar to the small zone:
split large zone into two small parts and apply same method

Main result

Theorem

Given an ordered tree T with n nodes our algorithm finds a drawing of T with crossing-free Lombardi edges and perfect angular resolution within a disk of radius $2n^3$.

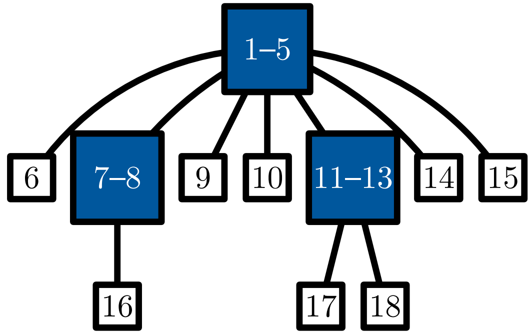
Main result

Theorem

Given an ordered tree T with n nodes our algorithm finds a drawing of T with crossing-free Lombardi edges and perfect angular resolution within a disk of radius $2n^3$.

Proof:

- inductive construction places every heavy path P at level j with root node v within a disk of radius $R = 2 \cdot 4^{h-j}|T_v|$
- heavy path at level 0 corresponds to full tree T
- height of the HPD is $h \leq \log_2 n$
- radius of disk containing T is $R = 2 \cdot 4^{\log_2 n} n = 2 \cdot n^3$



$\square$

Summary

Any tree has a drawing with

- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	✗
Lombardi edges	✓	✓

Summary

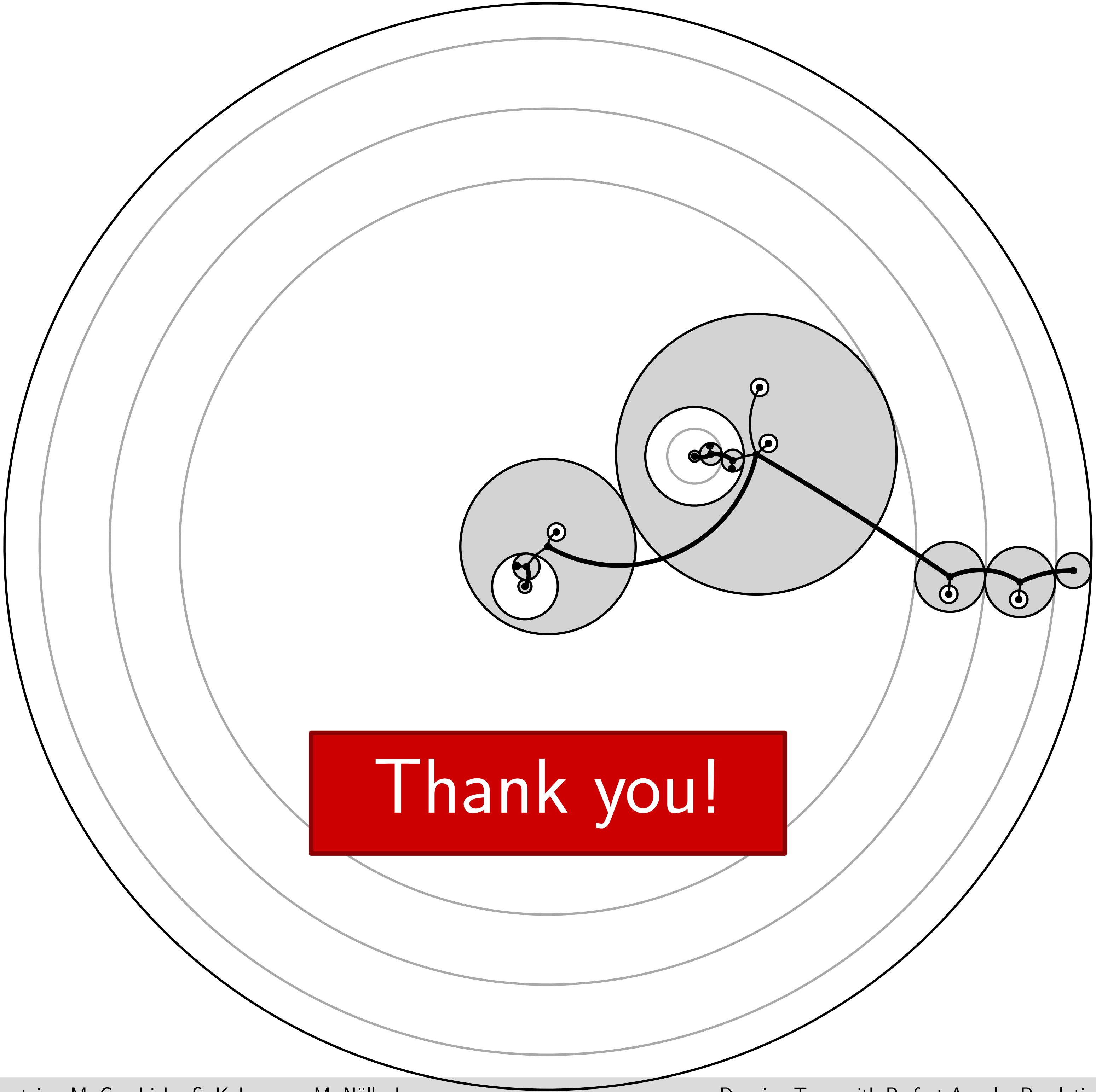
Any tree has a drawing with

- crossing-free edges
- perfect angular resolution
- polynomial area

	unordered trees	ordered trees
straight edges	✓	✗
Lombardi edges	✓	✓

Open questions

- area bound unlikely to be tight
- find simpler algorithms for special classes of trees (e.g. bounded degree)



Thank you!