Fixed-Priority Schedulability of Sporadic Tasks on Uniprocessors is NP-hard

Pontus Ekberg & Wang Yi

Uppsala University

RTSS 2017

Overview

	Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
Arbitrary utilization			
Utilization bounded by a constant c			

Overview

	Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
Arbitrary utilization	Pseudo-poly. time algorithm *	Pseudo-poly. time algorithm *	
Utilization bounded by a constant c	Pseudo-poly. time algorithm *	Pseudo-poly. time algorithm *	

(*) Joseph and Pandya, 1986

Overview

	Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
Arbitrary utilization	Pseudo-poly. time algorithm *	Pseudo-poly. time algorithm *	
Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities †	Pseudo-poly. time algorithm *	

(∗) Joseph and Pandya, 1986

(†) Liu and Layland, 1973

Overview

	Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
Arbitrary utilization	Pseudo-poly. time algorithm *	Pseudo-poly. time algorithm *	Exponential time algorithm ‡
Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities †	Pseudo-poly. time algorithm *	Exponential time algorithm ‡

(∗) Joseph and Pandya, 1986

(†) Liu and Layland, 1973

(‡) Lehoczky, 1990

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Pseudo-poly. time algorithm	Pseudo-poly. time algorithm	Exponential time algorithm
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Pseudo-poly. time algorithm	Exponential time algorithm
EDF	Arbitrary utilization			
	Utilization bounded by a constant c			

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Pseudo-poly. time algorithm	Pseudo-poly. time algorithm	Exponential time algorithm
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Pseudo-poly. time algorithm	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Pseudo-poly. time algorithm	Pseudo-poly. time algorithm	Exponential time algorithm
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Pseudo-poly. time algorithm	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Pseudo-poly. time algorithm	Pseudo-poly. time algorithm	Exponential time algorithm
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Pseudo-poly. time algorithm	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-hard	Pseudo-poly. time algorithm	Exponential time algorithm
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Pseudo-poly. time algorithm	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-hard	Pseudo-poly. time algorithm	Exponential time algorithm
FP	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Pseudo-poly. time algorithm	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
EDF	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-hard	Pseudo-poly. time algorithm	Exponential time algorithm
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-hard for $0 < c < 1$	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-hard	Pseudo-poly. time algorithm	Exponential time algorithm
FP	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-hard for $0 < c < 1$	Exponential time algorithm
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
EDF	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-hard	Weakly NP-hard	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-hard for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Overview

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-complete	Weakly NP-complete	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-complete for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

→

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

⇝

EDF-schedulability

- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

⇝

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

⇝

EDF-schedulability

- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

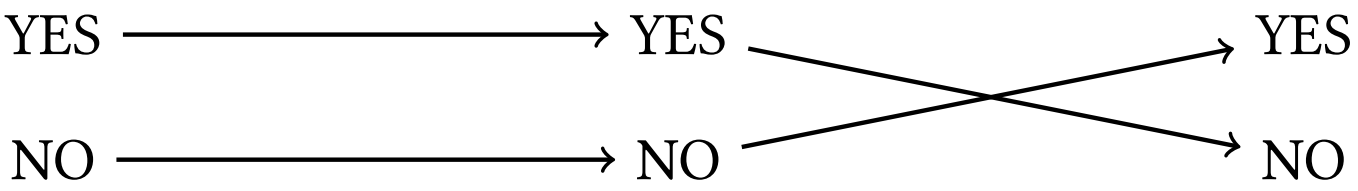
⇝

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

⇝

EDF-schedulability

- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

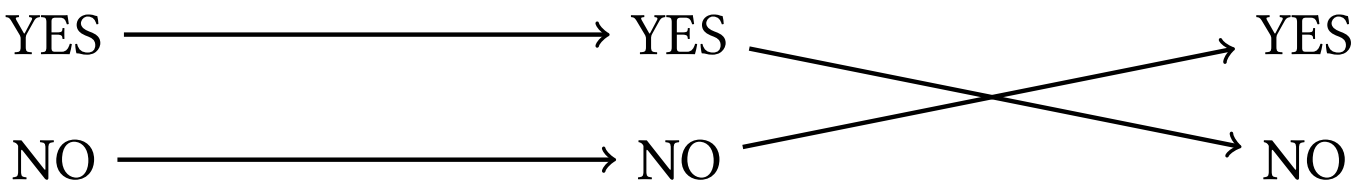
⇝

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



coNP-hard

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

⇝

EDF-schedulability

- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

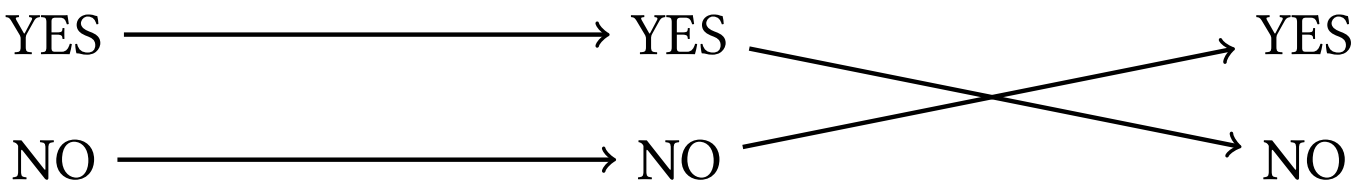
⇝

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



coNP-hard

coNP-hard

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

⇝

EDF-schedulability

- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

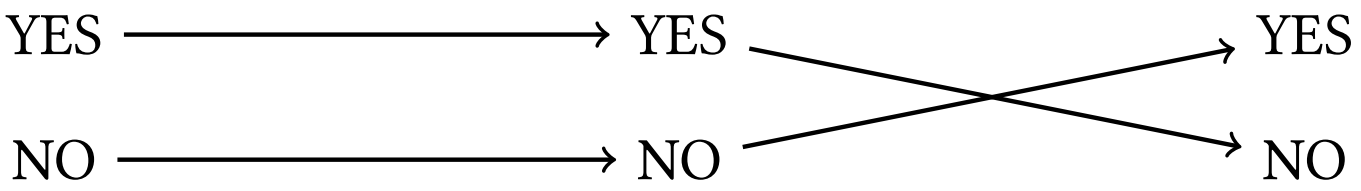
⇝

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



coNP-hard | coNP-hard | NP-hard

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

EDF-schedulability

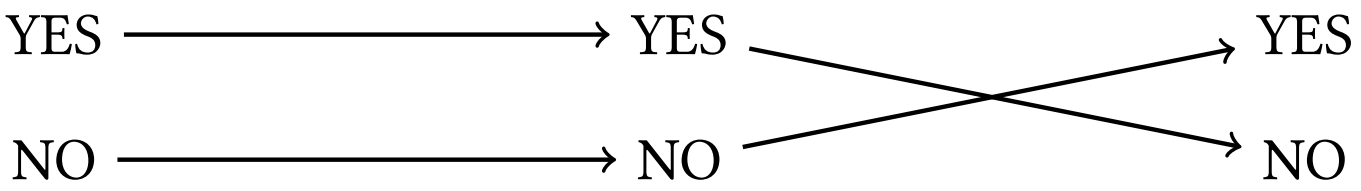
- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



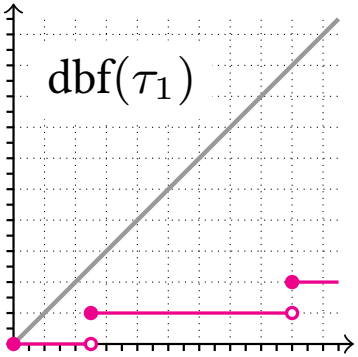
coNP-hard

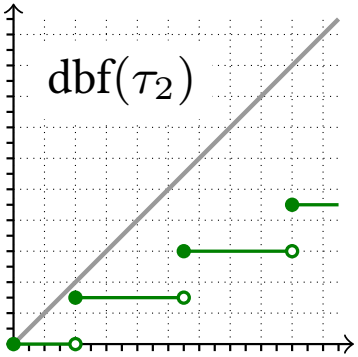
coNP-hard

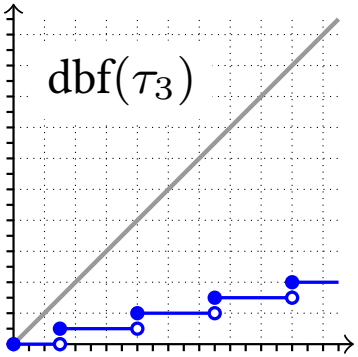
NP-hard

EDF-schedulability $\rightsquigarrow$ FP-schedulability

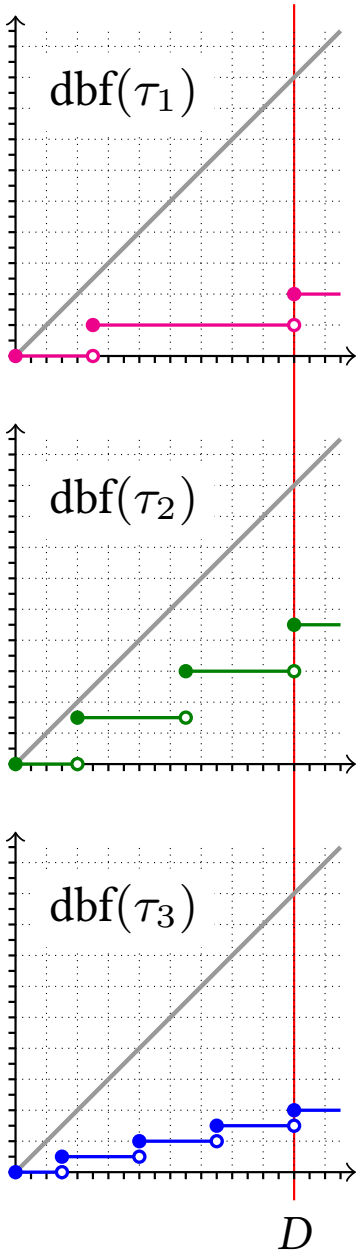
EDF-schedulability $\rightsquigarrow$ FP-schedulability



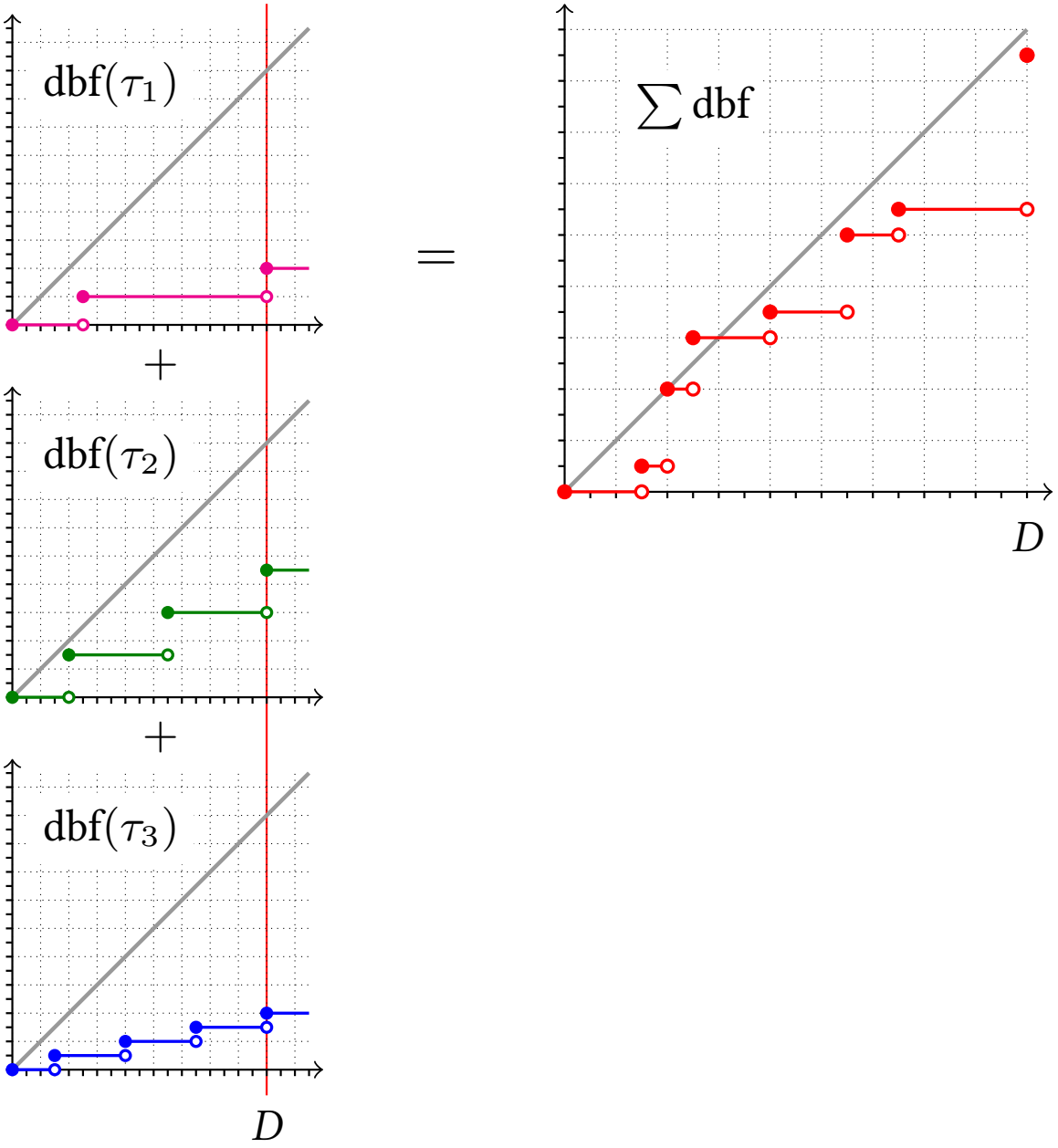




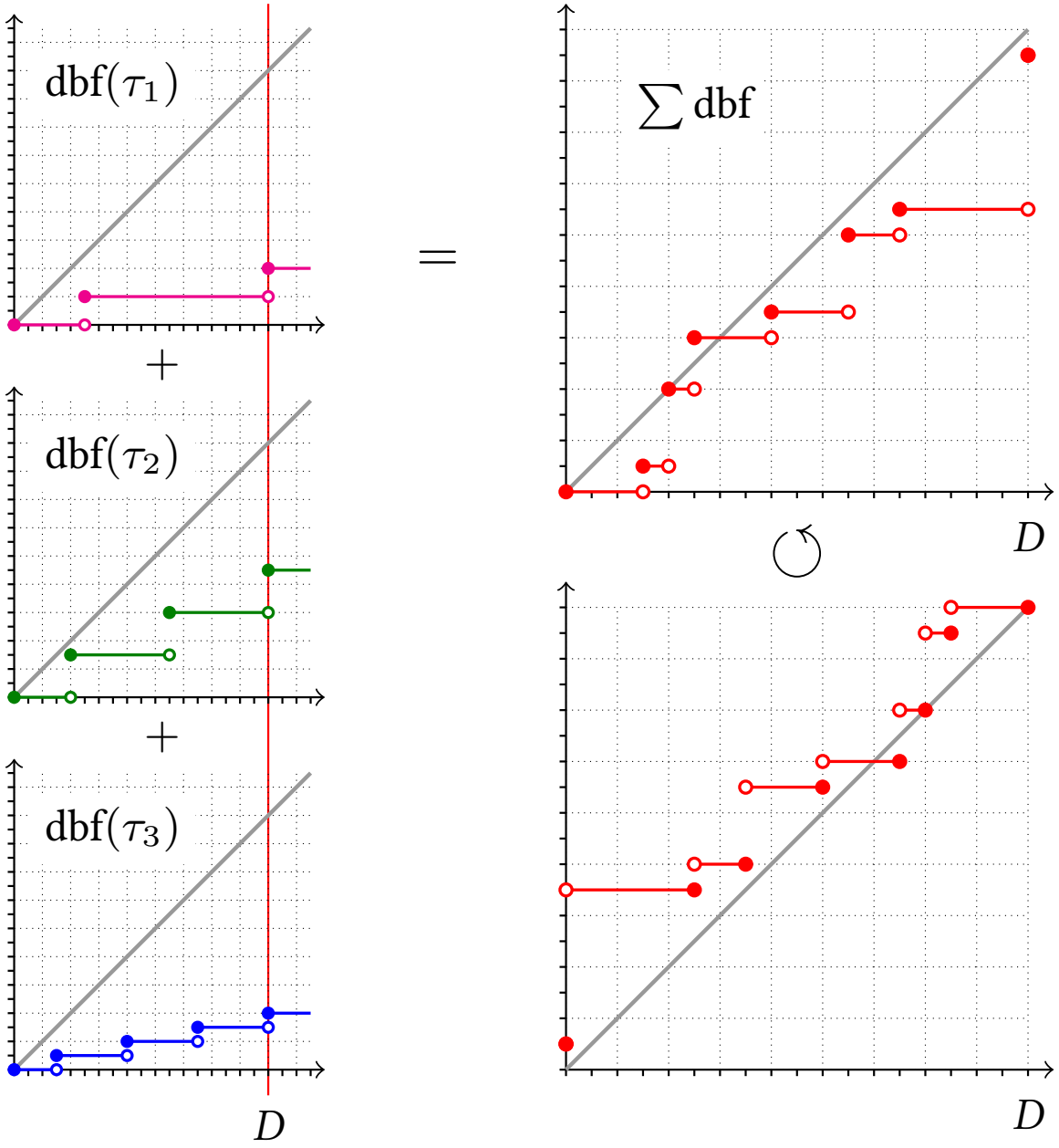
EDF-schedulability ⇝ FP-schedulability



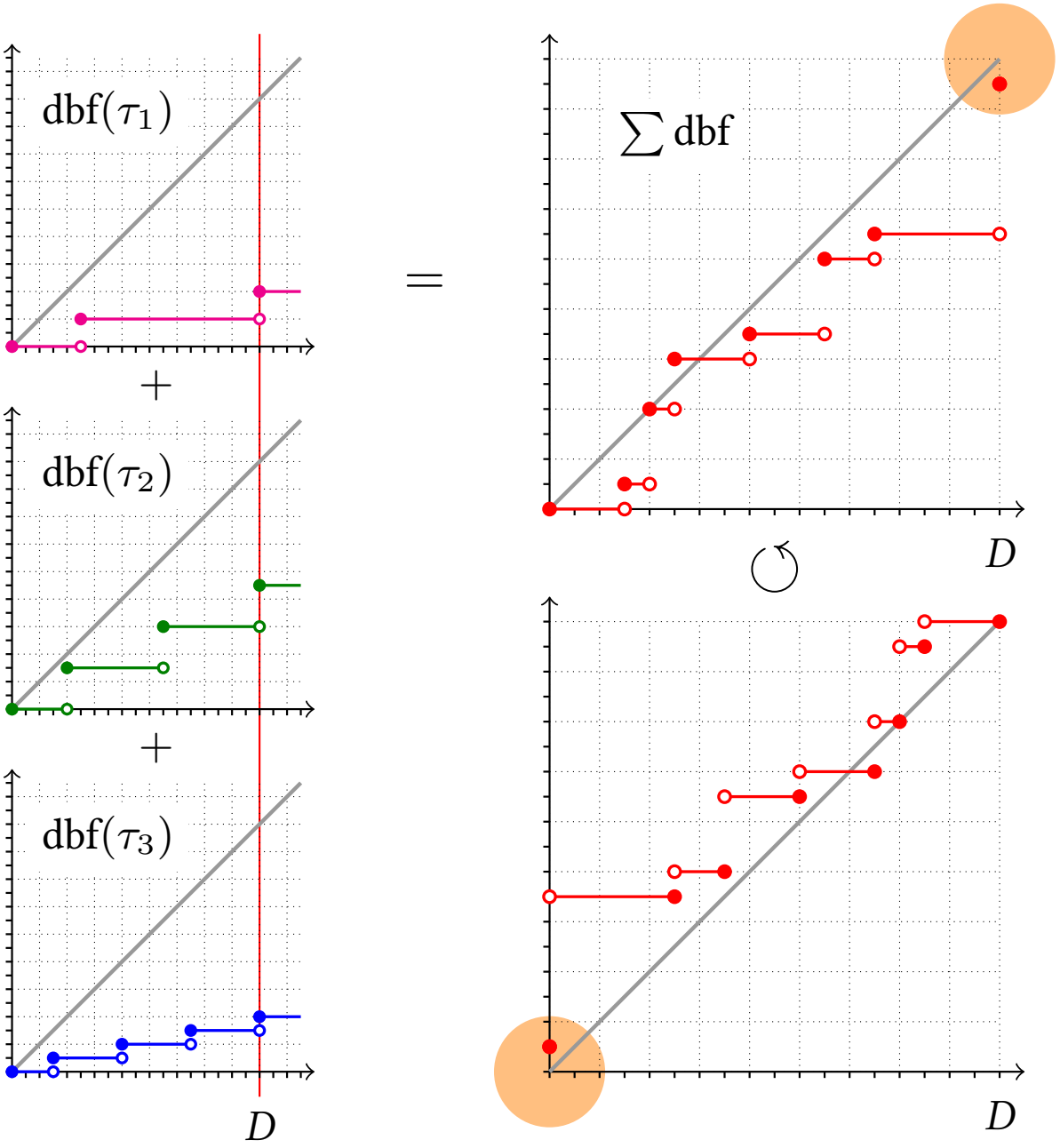
EDF-schedulability ⇝ FP-schedulability



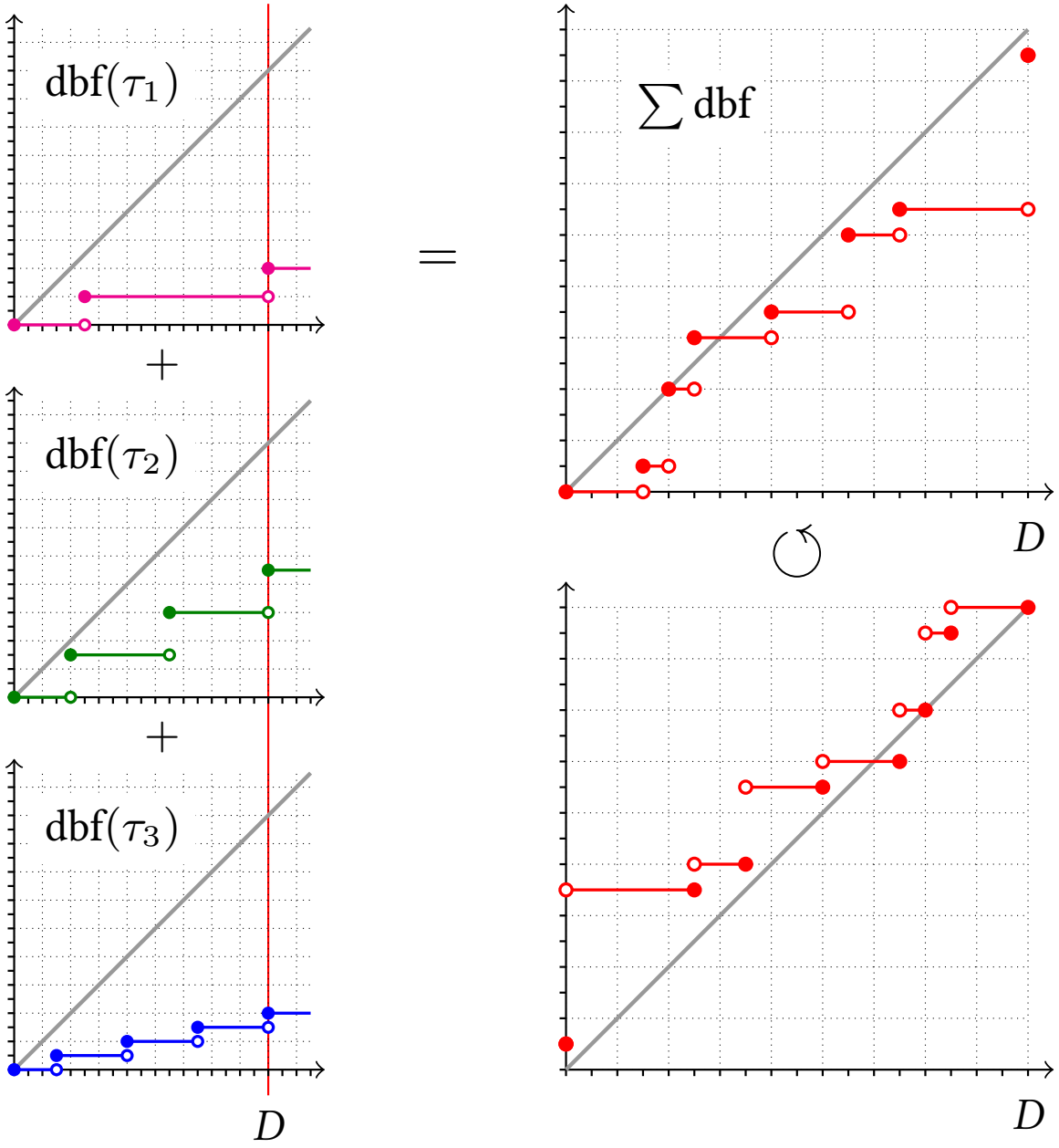
EDF-schedulability ⇝ FP-schedulability



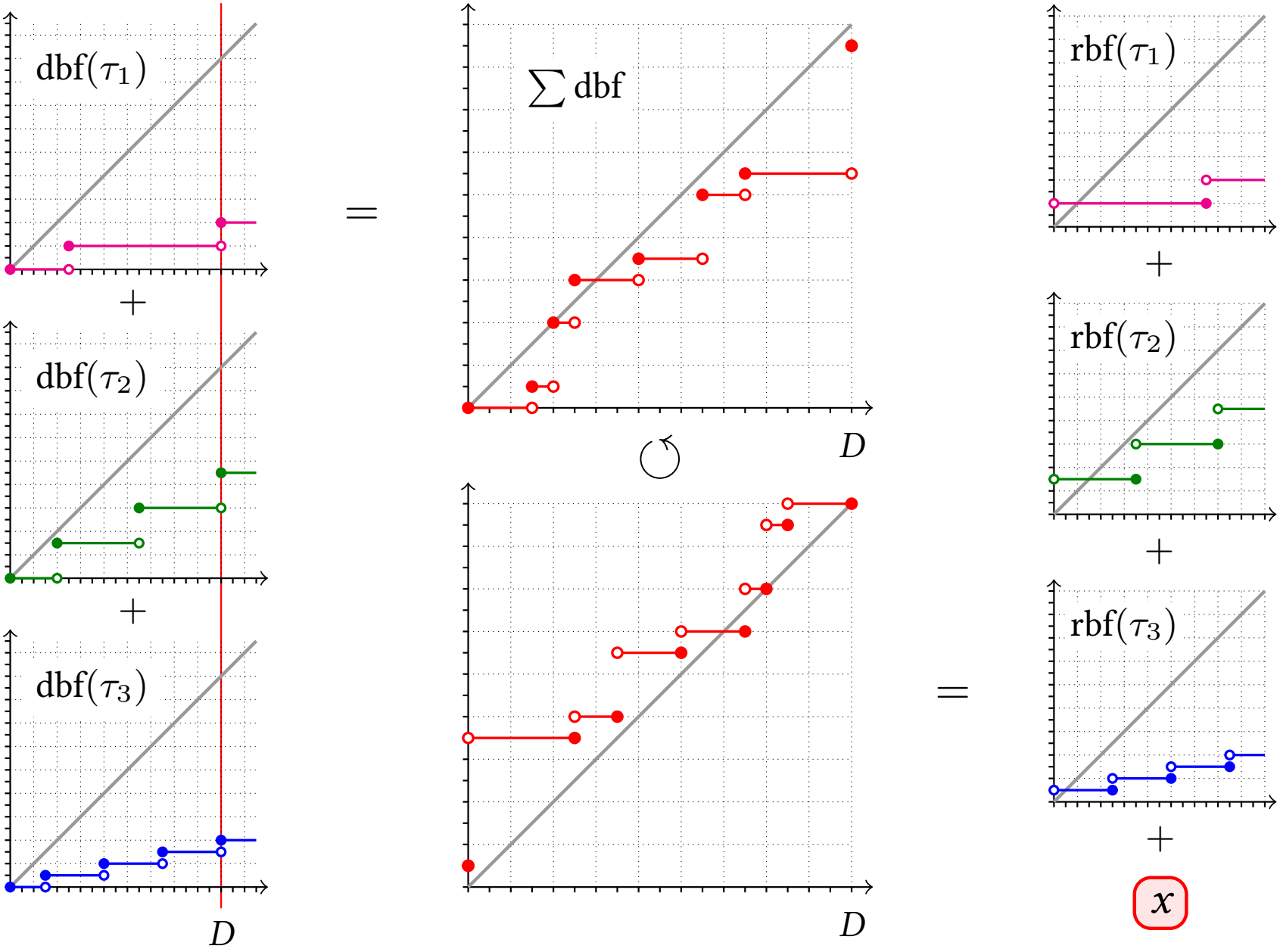
EDF-schedulability ⇝ FP-schedulability



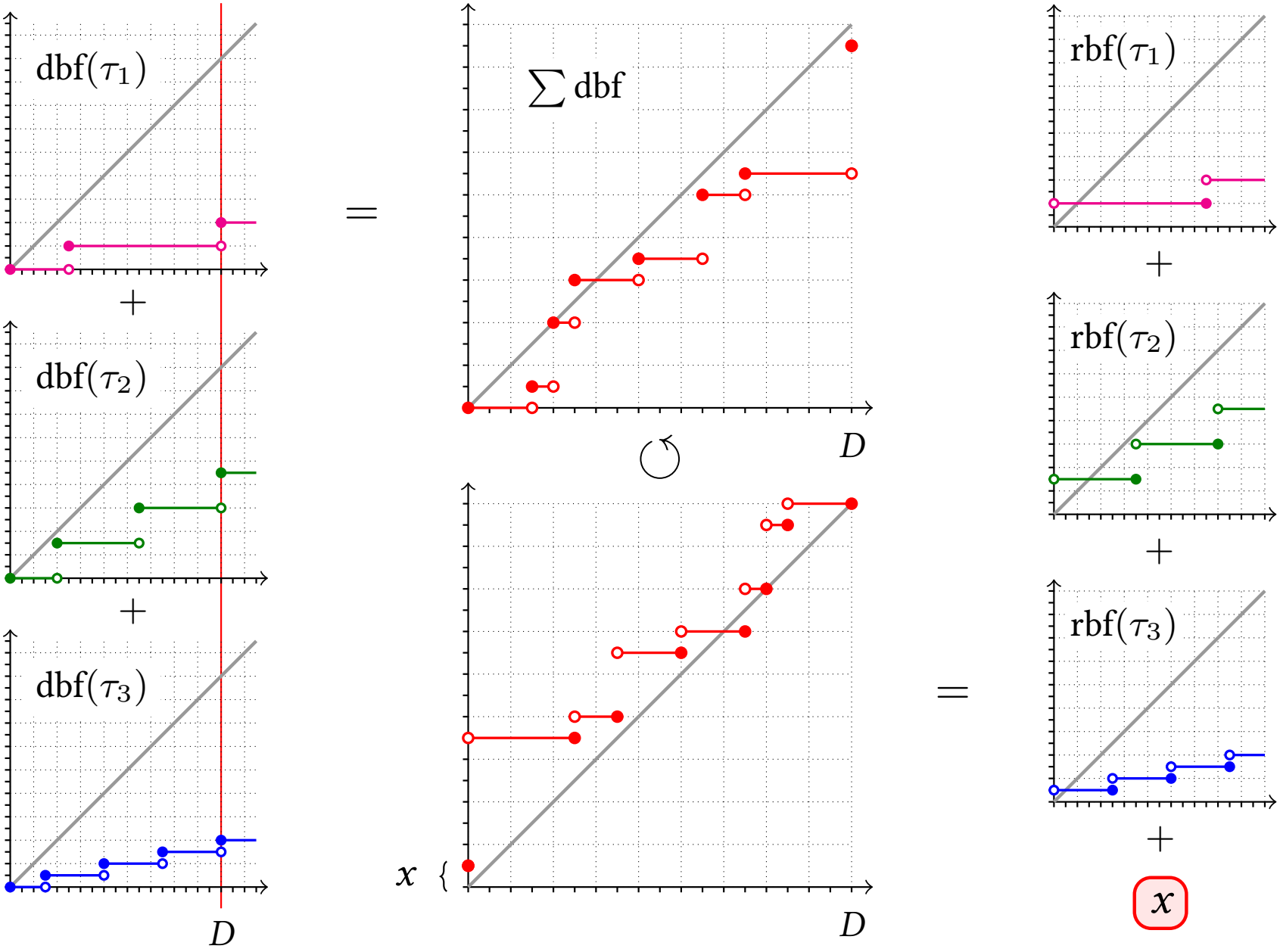
EDF-schedulability ⇝ FP-schedulability



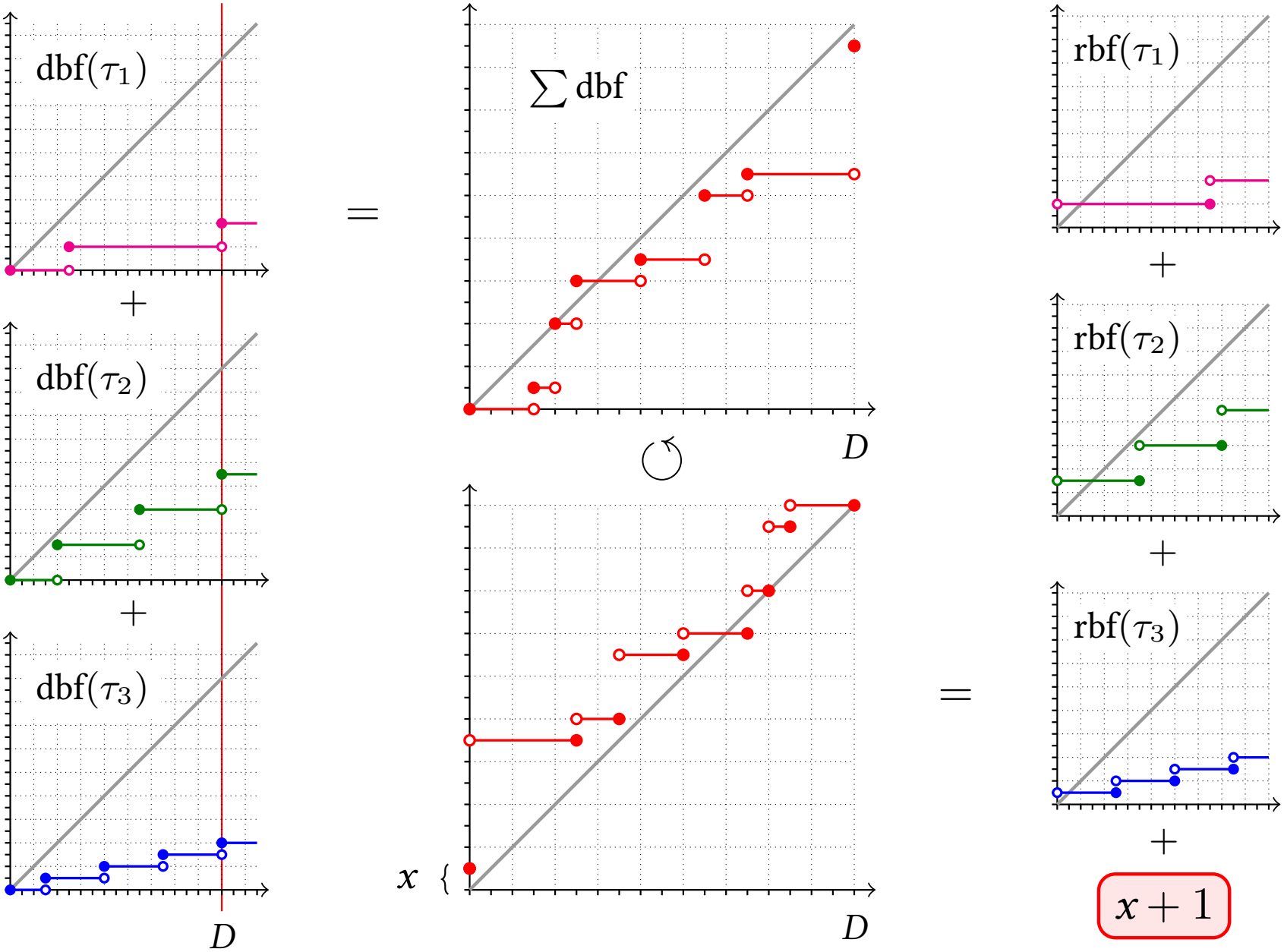
EDF-schedulability ⇝ FP-schedulability



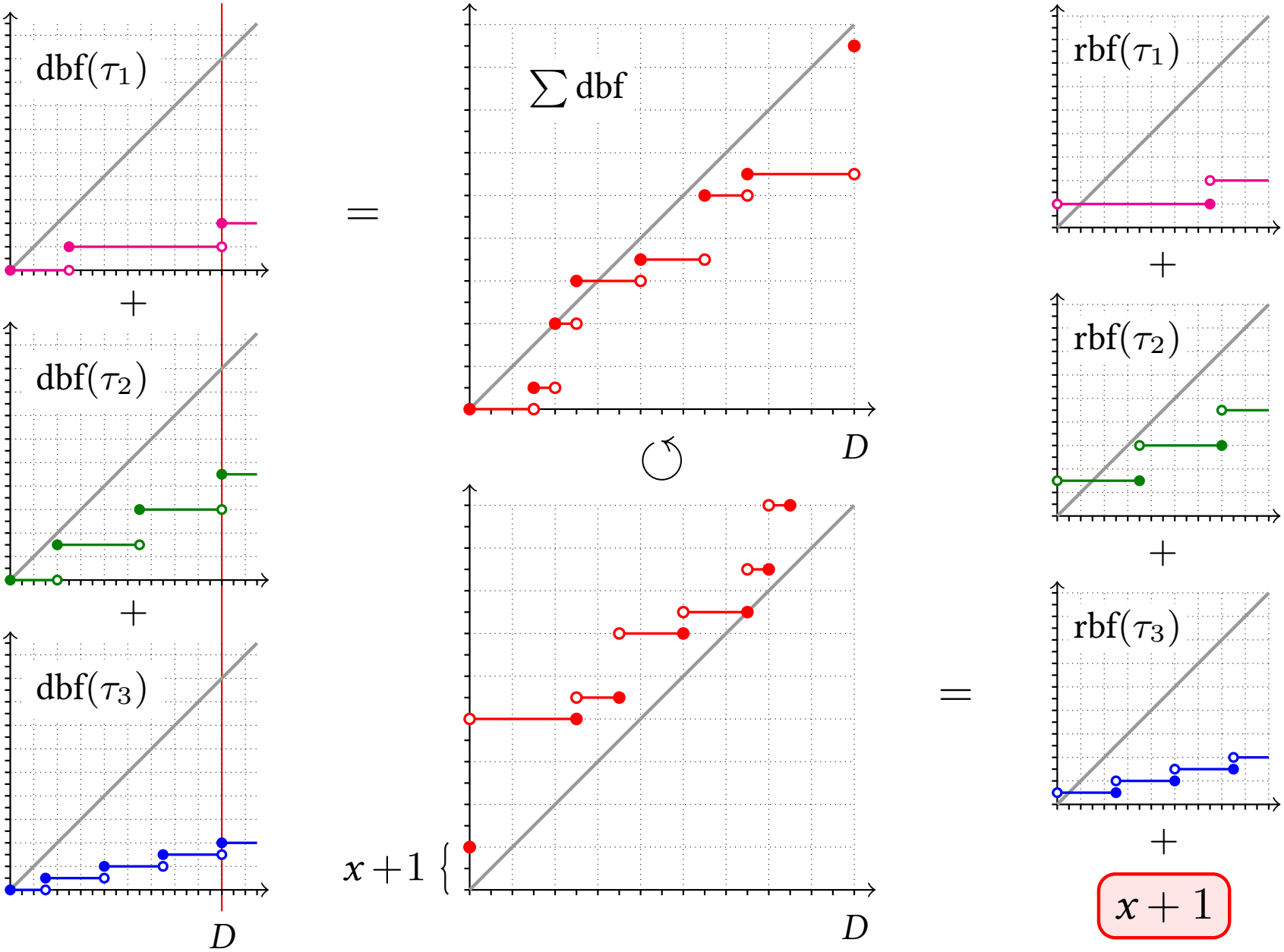
EDF-schedulability ⇝ FP-schedulability



EDF-schedulability ⇝ FP-schedulability



EDF-schedulability ⇝ FP-schedulability



A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

EDF-schedulability

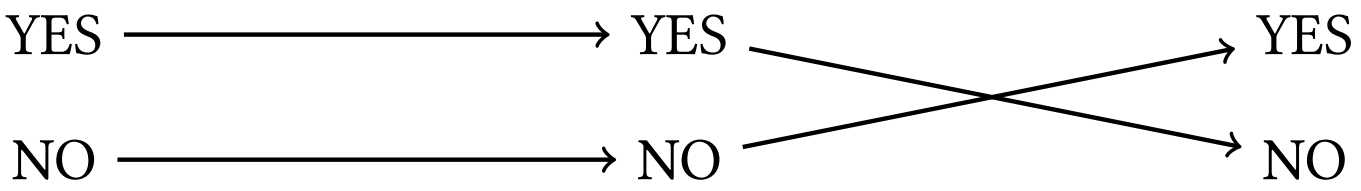
- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



coNP-hard

coNP-hard

NP-hard

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

EDF-schedulability

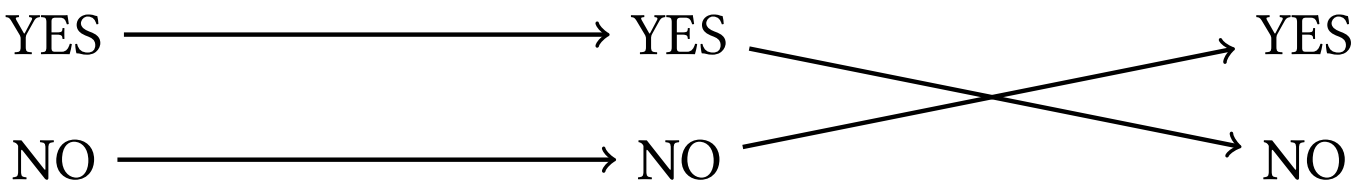
- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



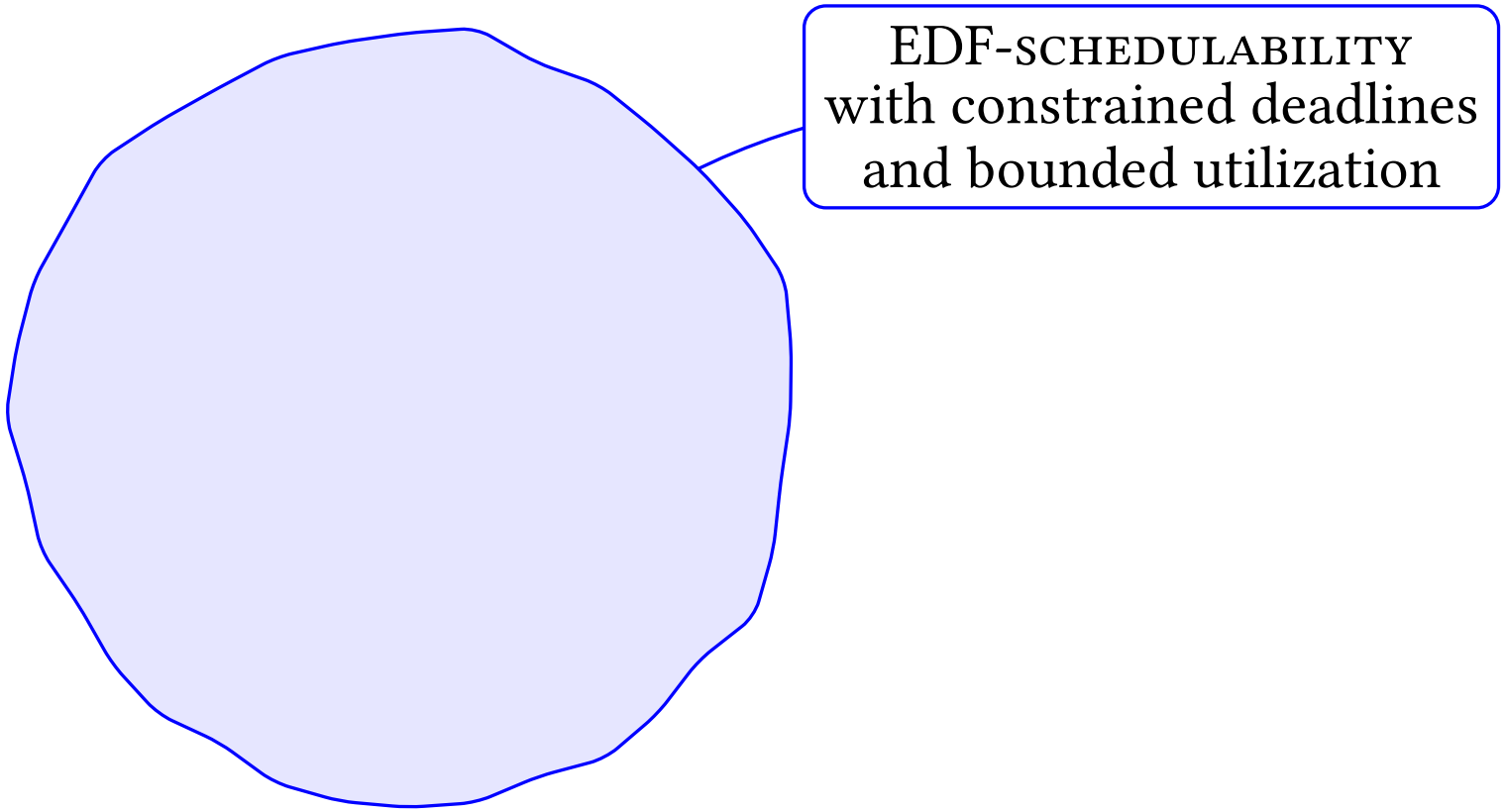
coNP-hard

coNP-hard

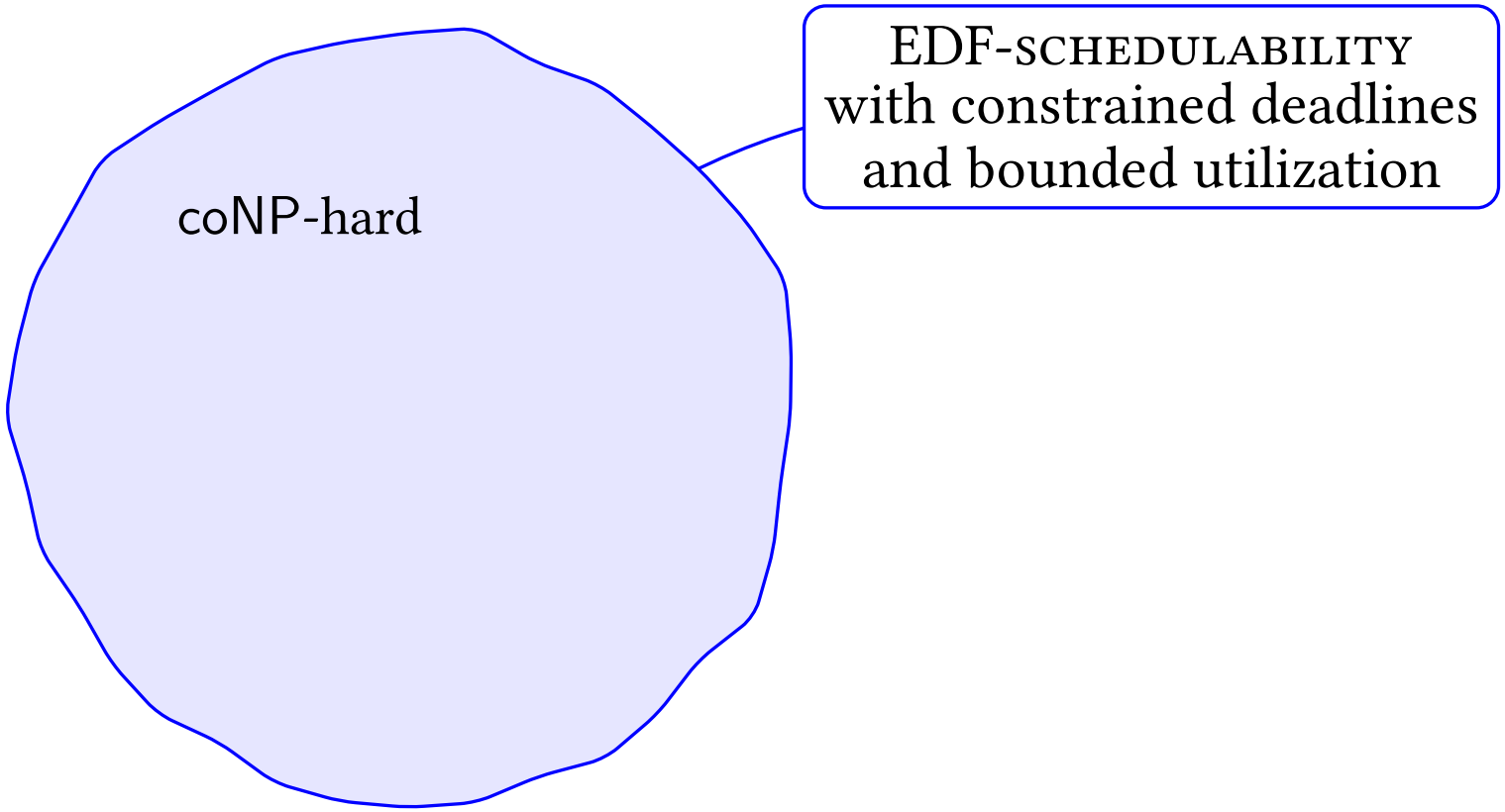
NP-hard

Reducing EDF-schedulability to a Special Case

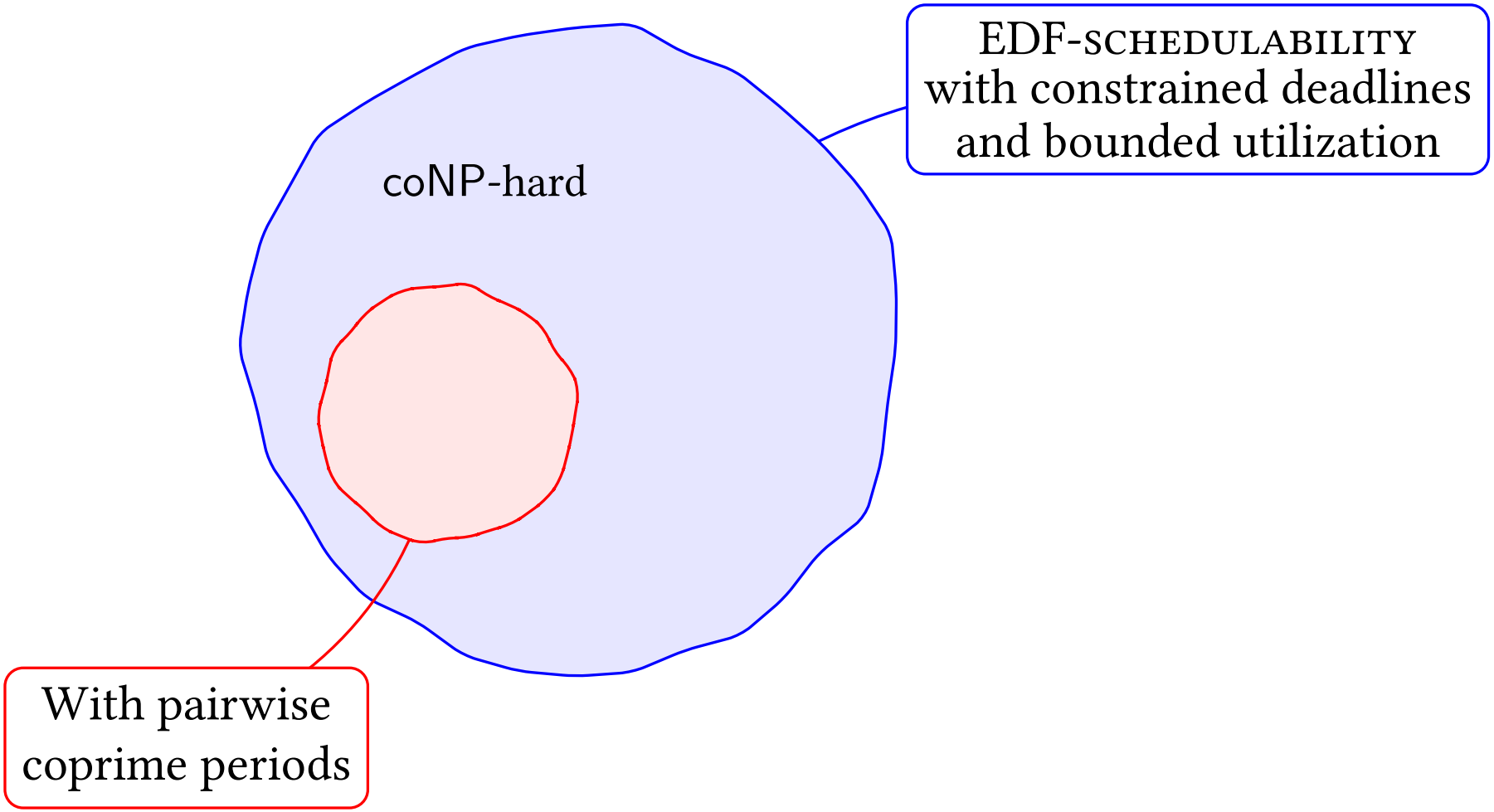
Reducing EDF-schedulability to a Special Case



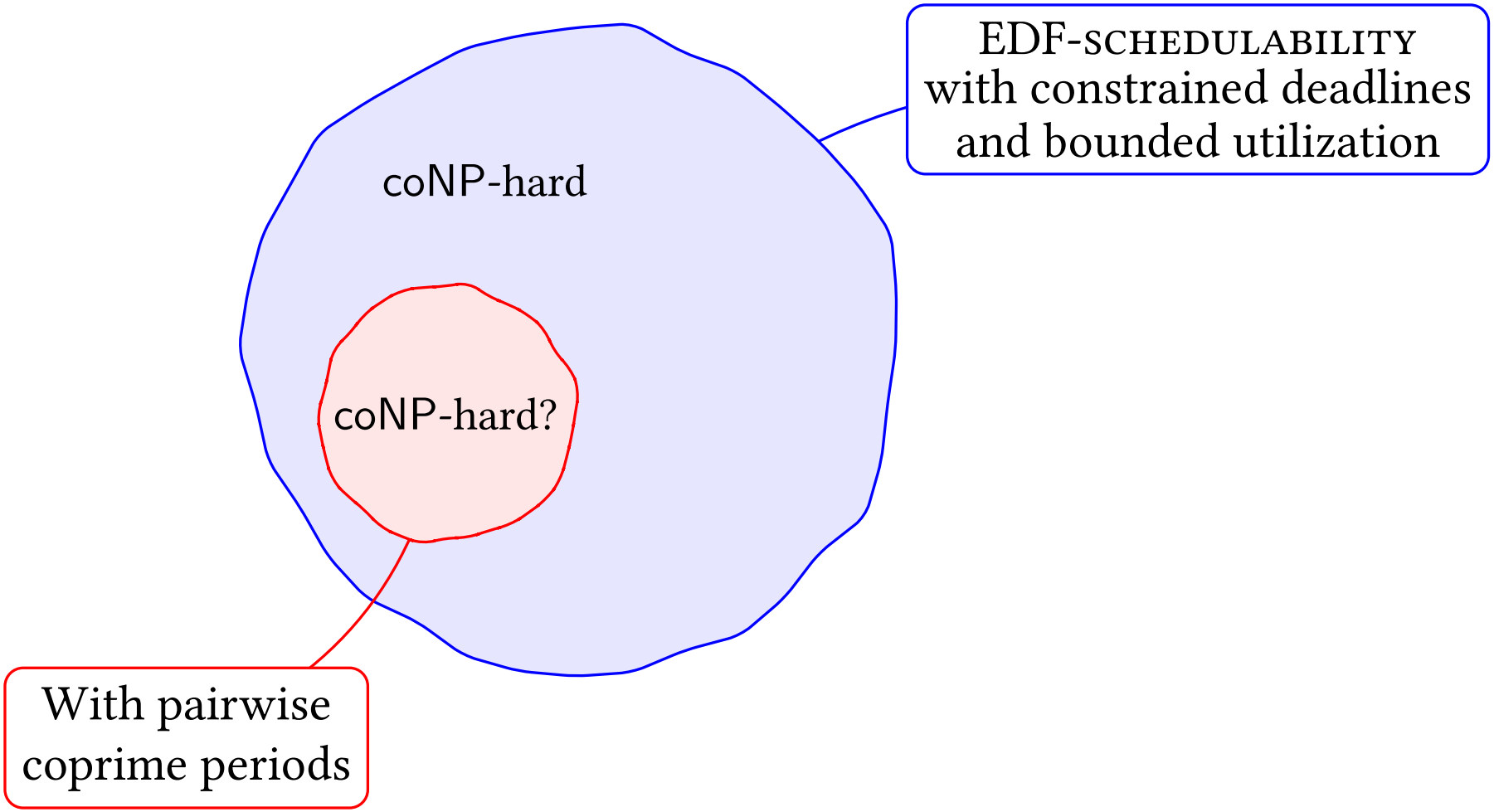
Reducing EDF-schedulability to a Special Case



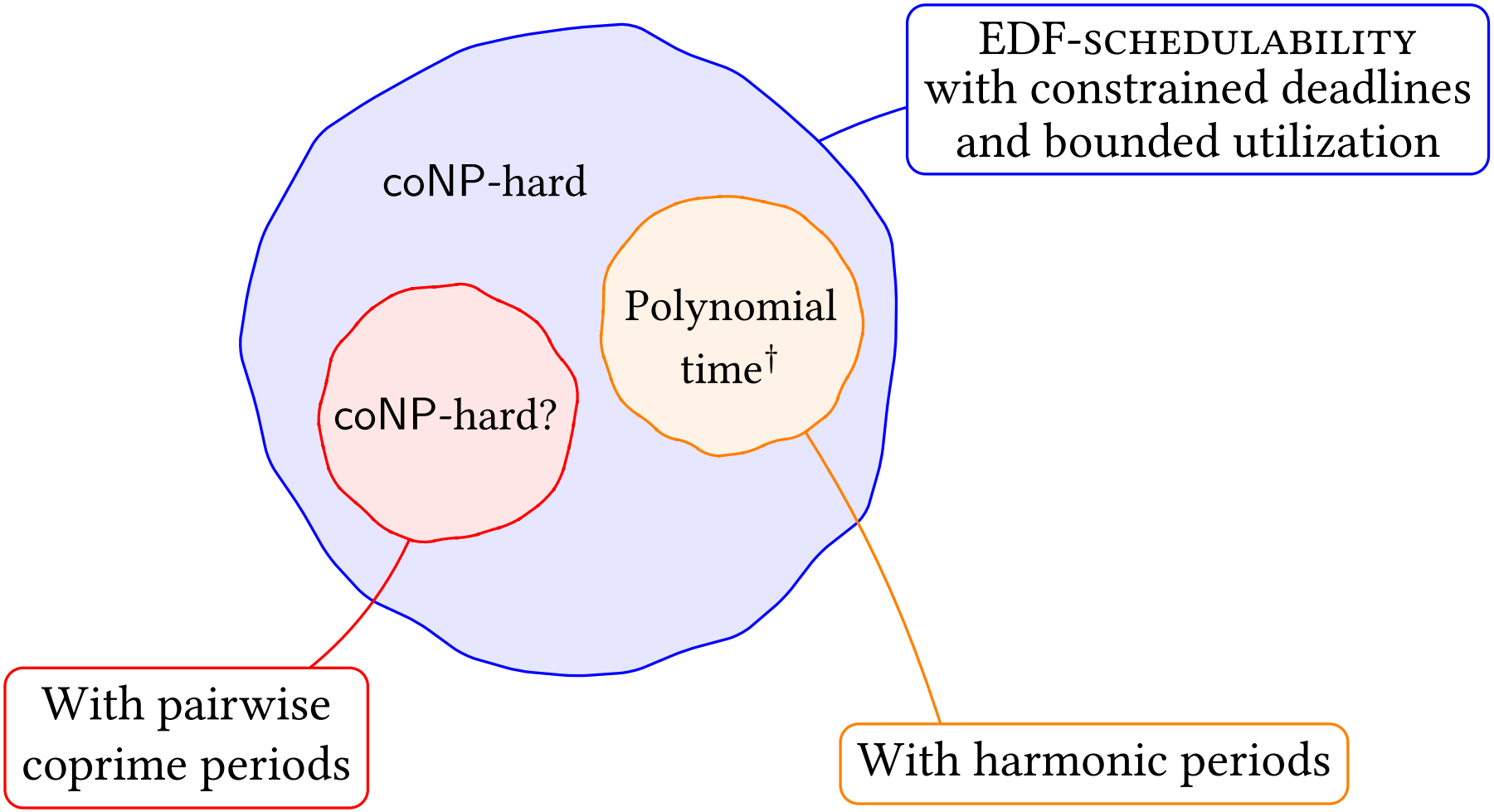
Reducing EDF-schedulability to a Special Case



Reducing EDF-schedulability to a Special Case

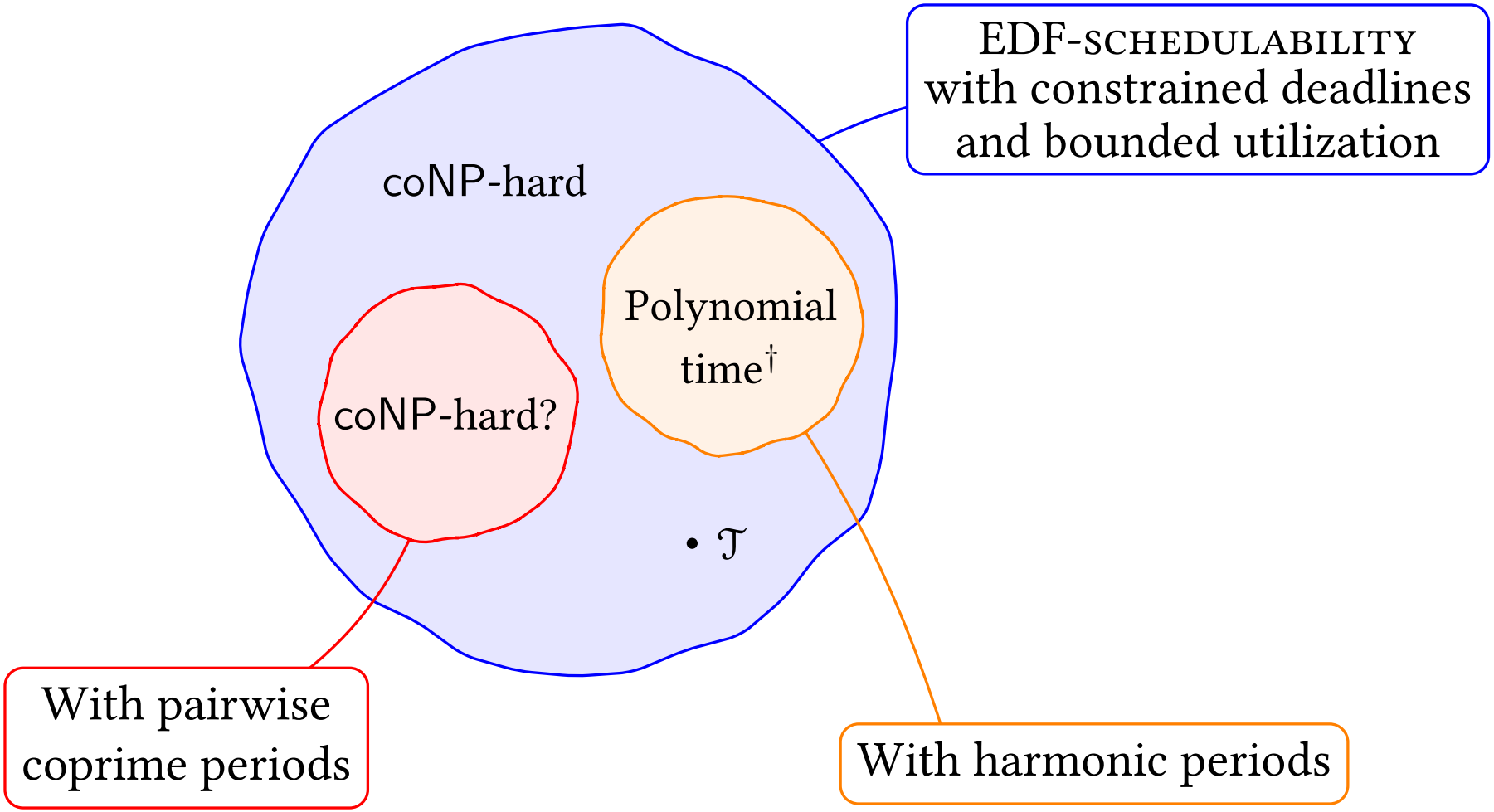


Reducing EDF-schedulability to a Special Case



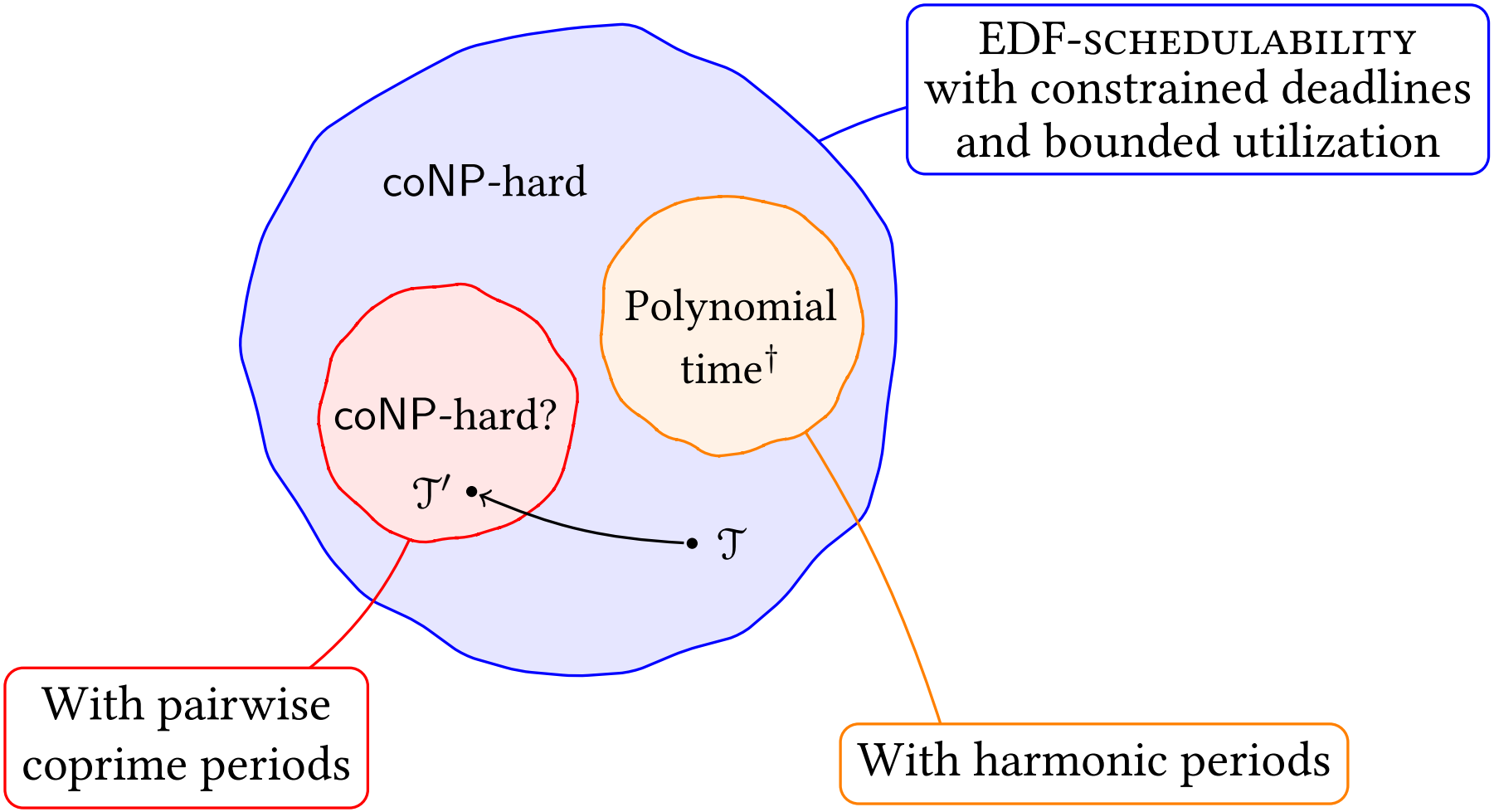
† Bonifaci et al., 2013

Reducing EDF-schedulability to a Special Case



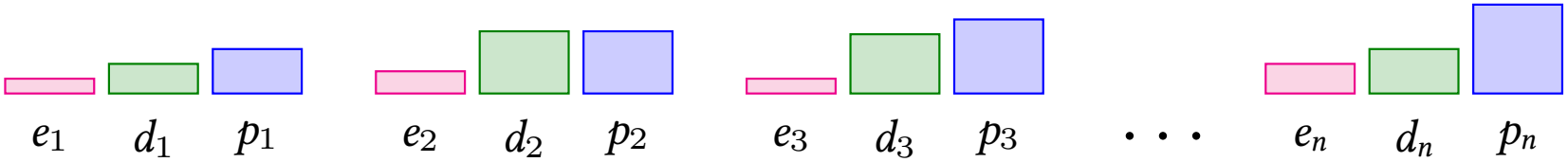
† Bonifaci et al., 2013

Reducing EDF-schedulability to a Special Case



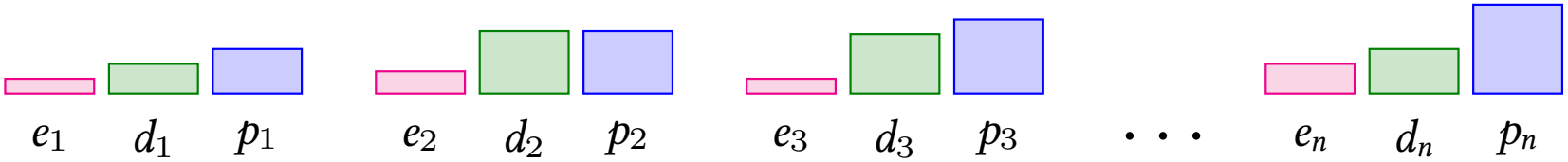
† Bonifaci et al., 2013

Outline of the Reduction



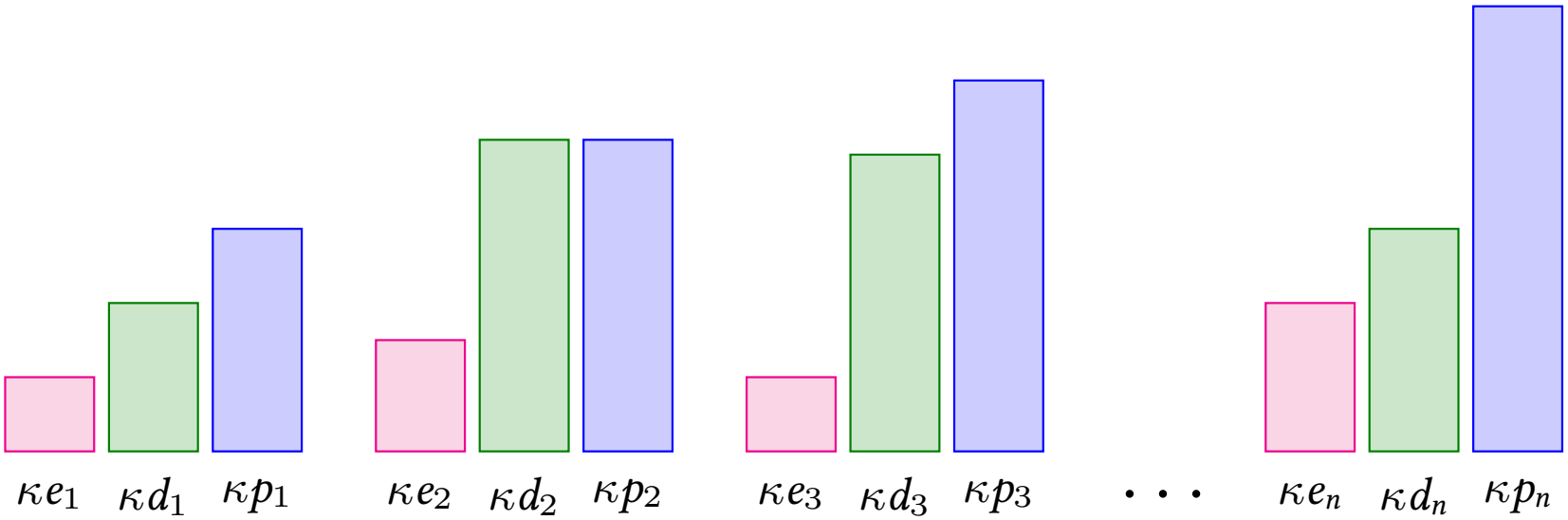
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.



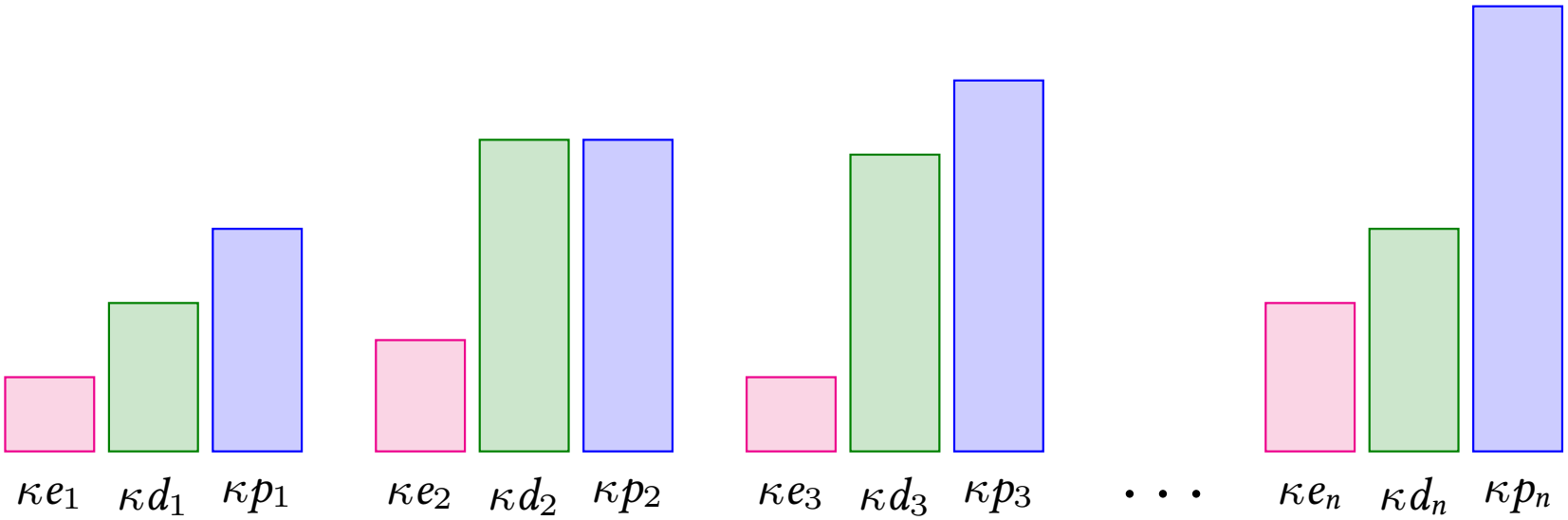
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.



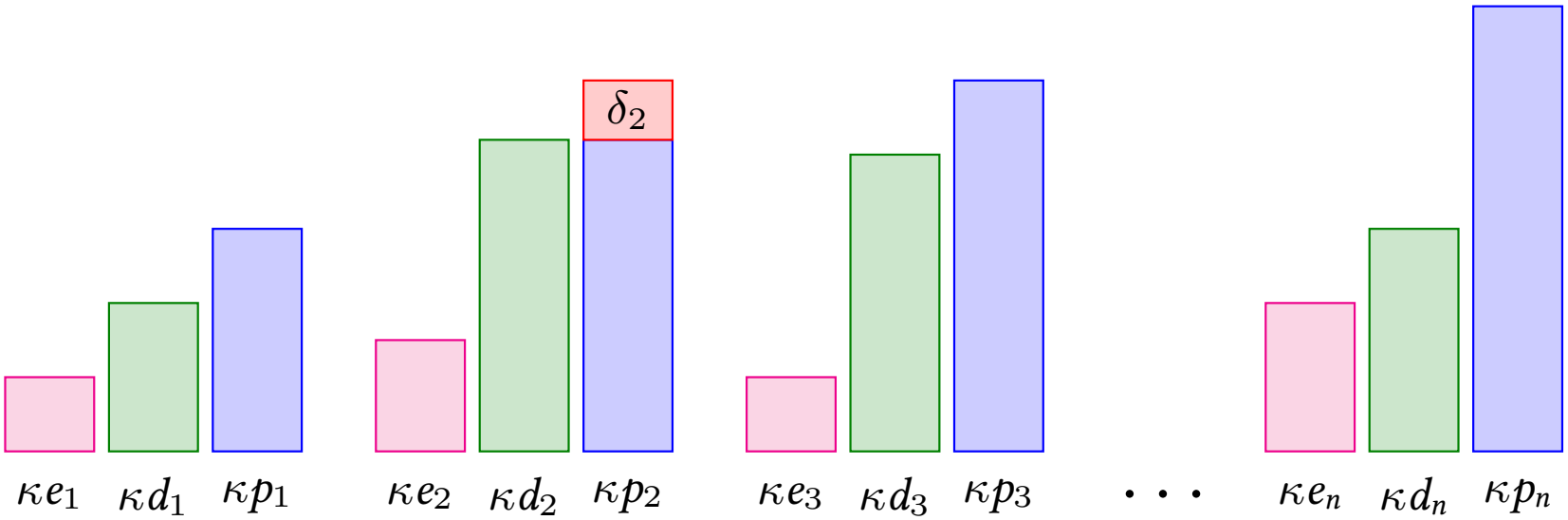
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



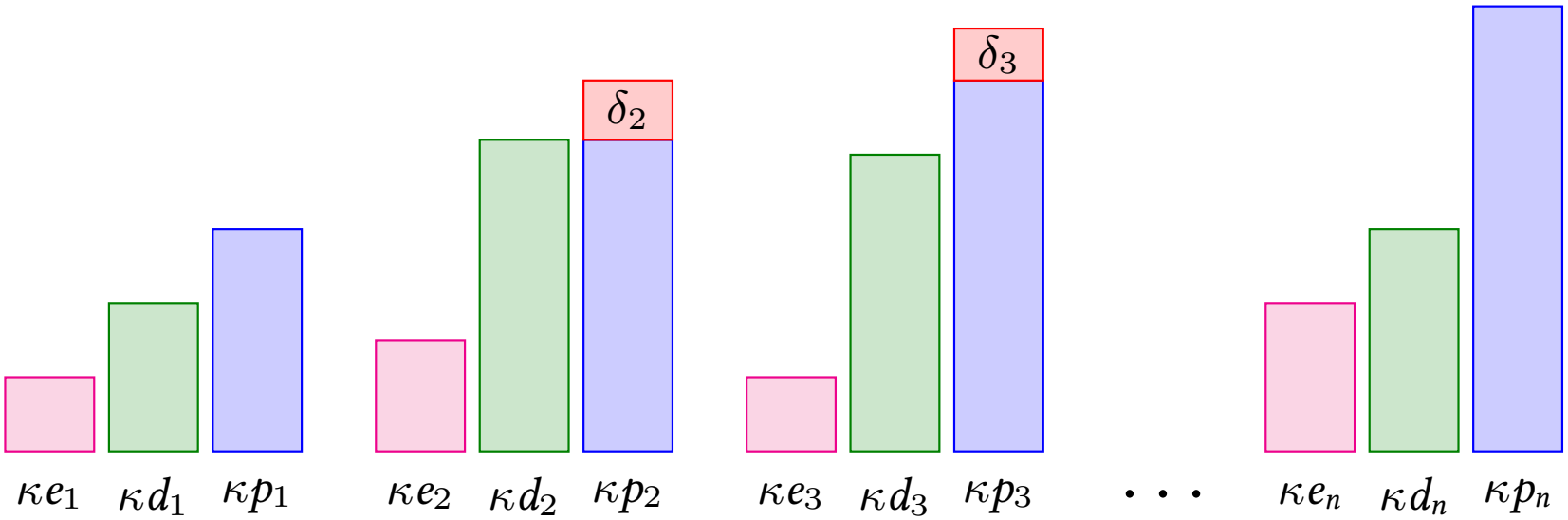
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



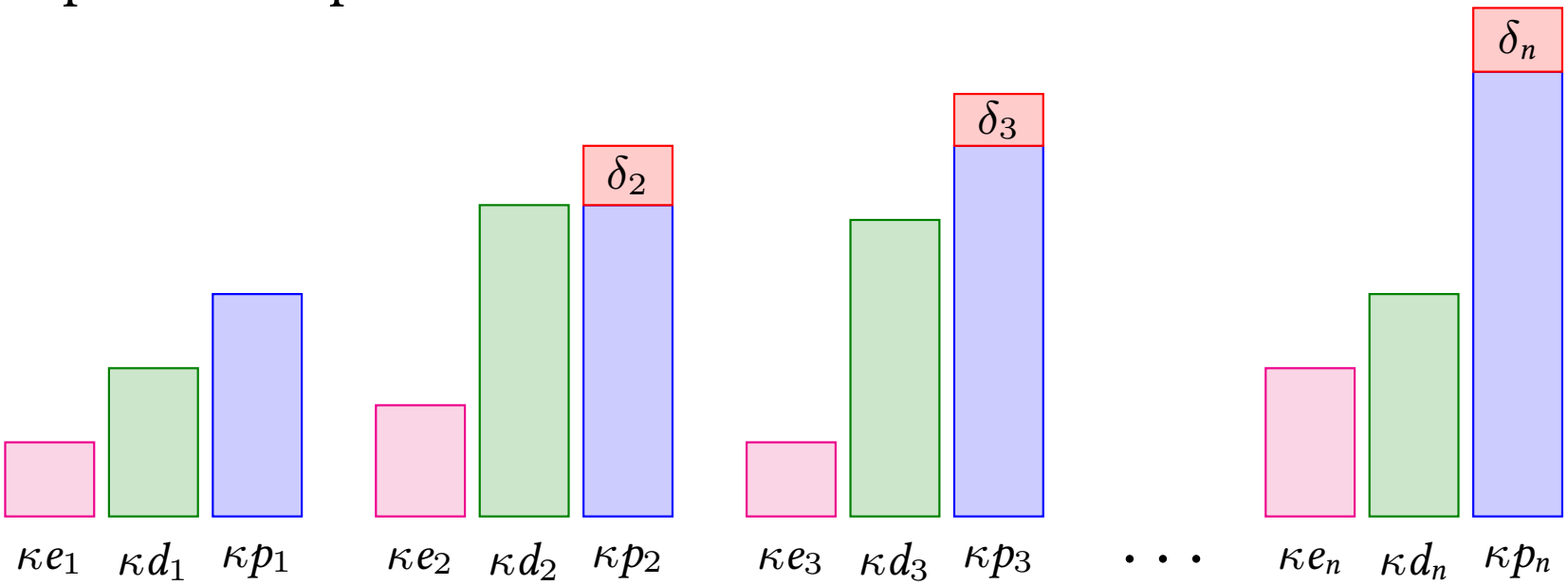
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



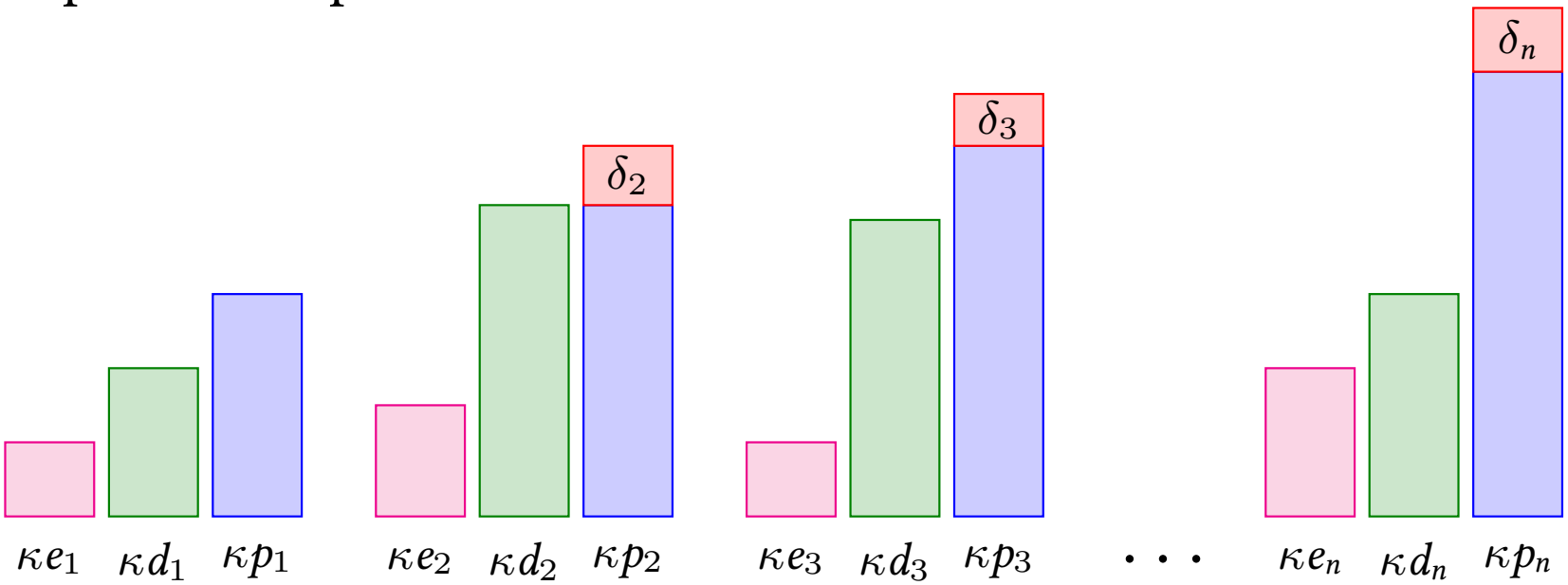
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



Outline of the Reduction

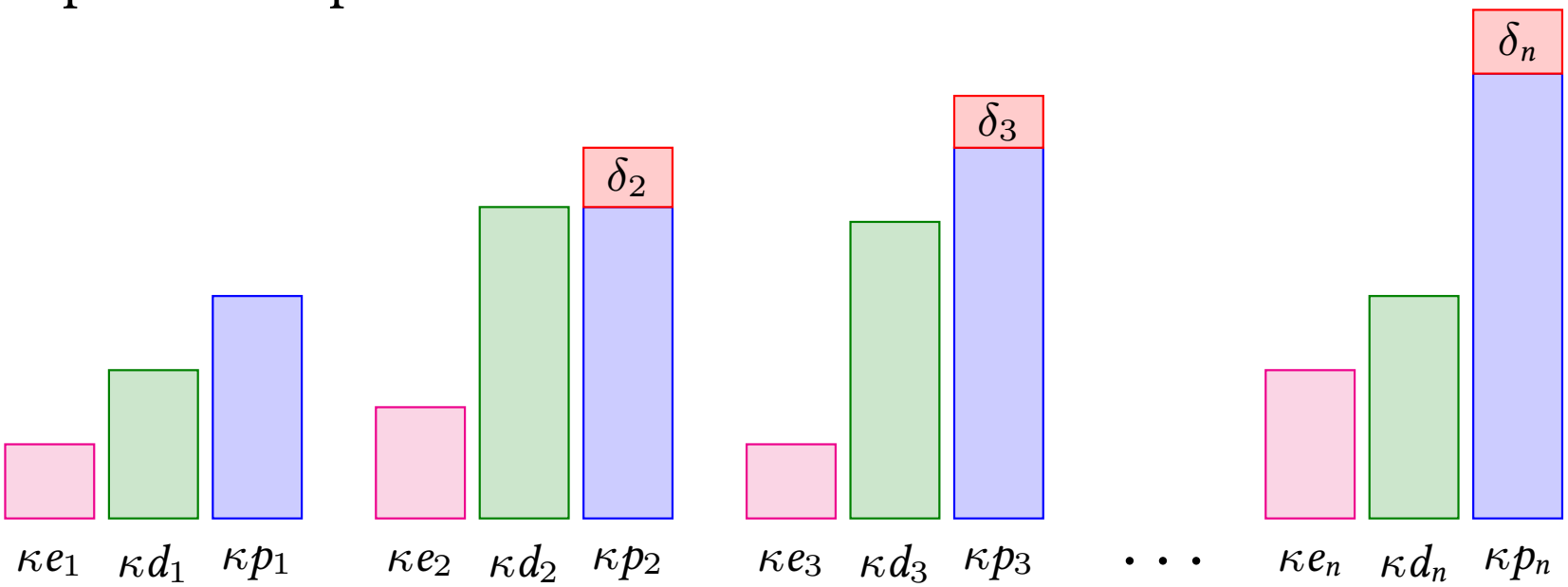
1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



- The δ_i can be found in polynomial time.

Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



- The δ_i can be found in polynomial time.
- The δ_i are so small relative to κ that schedulability is *unaffected*.

Some Number Theory

The Jacobsthal function

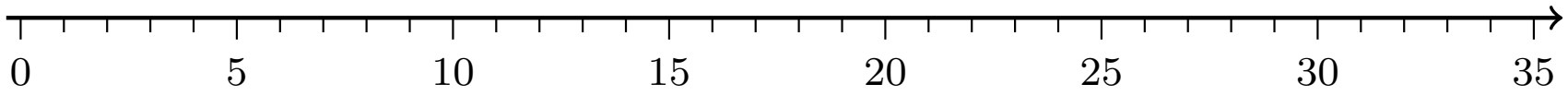
The Jacobsthal function $g(n)$ gives the largest gap between numbers that are coprime to n.

Some Number Theory

The Jacobsthal function

The Jacobsthal function $g(n)$ gives the largest gap between numbers that are coprime to n.

Coprime to 100?

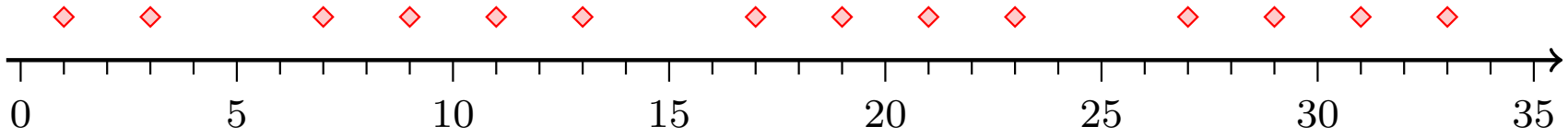


Some Number Theory

The Jacobsthal function

The Jacobsthal function $g(n)$ gives the largest gap between numbers that are coprime to n.

Coprime to 100?

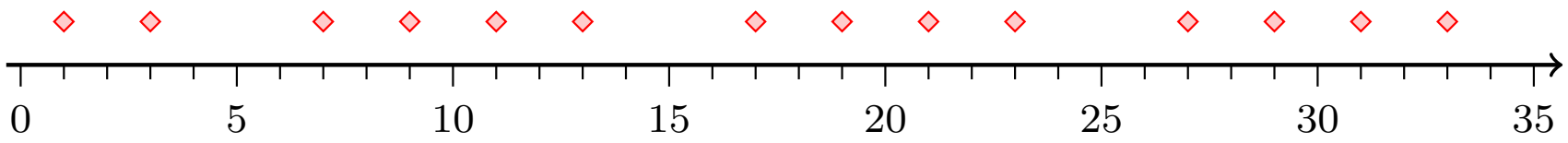


Some Number Theory

The Jacobsthal function

The Jacobsthal function $g(n)$ gives the largest gap between numbers that are coprime to n.

Coprime to 100?



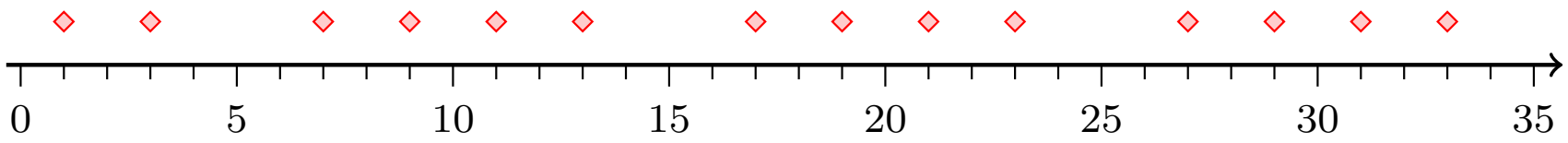
$$g(100) = 4$$

Some Number Theory

The Jacobsthal function

The Jacobsthal function $g(n)$ gives the largest gap between numbers that are coprime to n.

Coprime to 100?



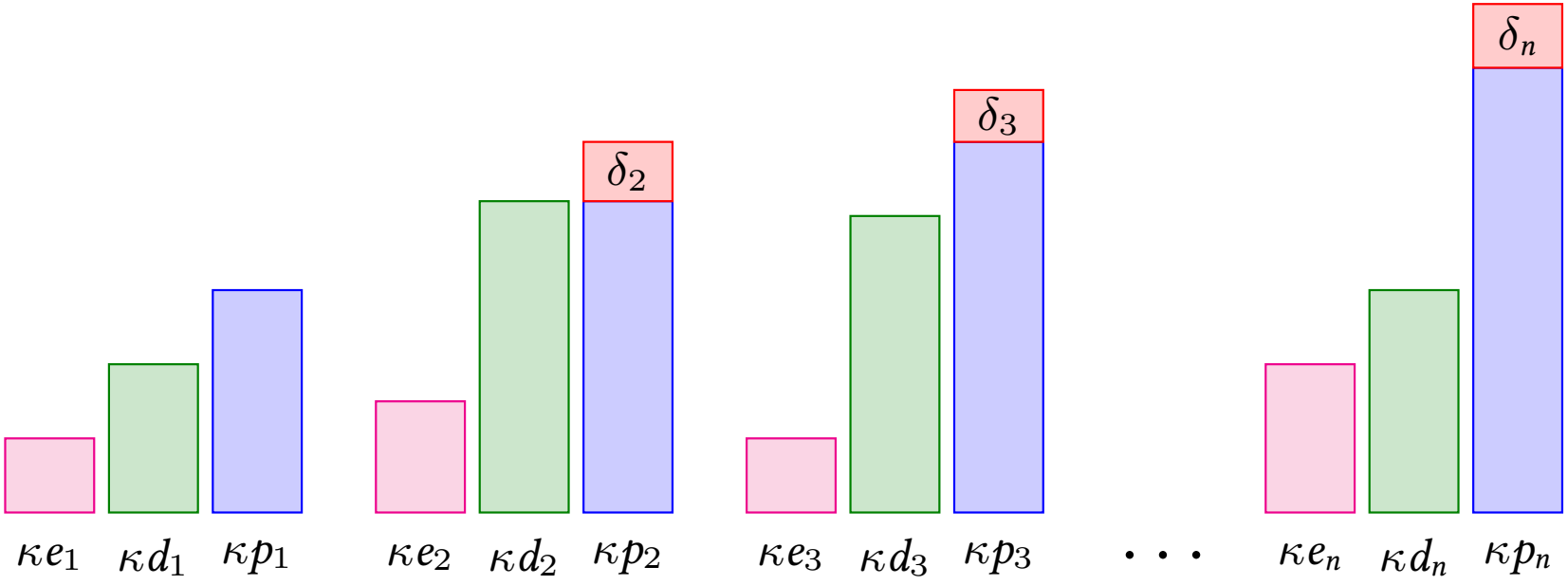
$$g(100) = 4$$

$$g(n) \in \mathcal{O}(\log^2 n)$$

Iwaniec, 1978

Outline of the Reduction

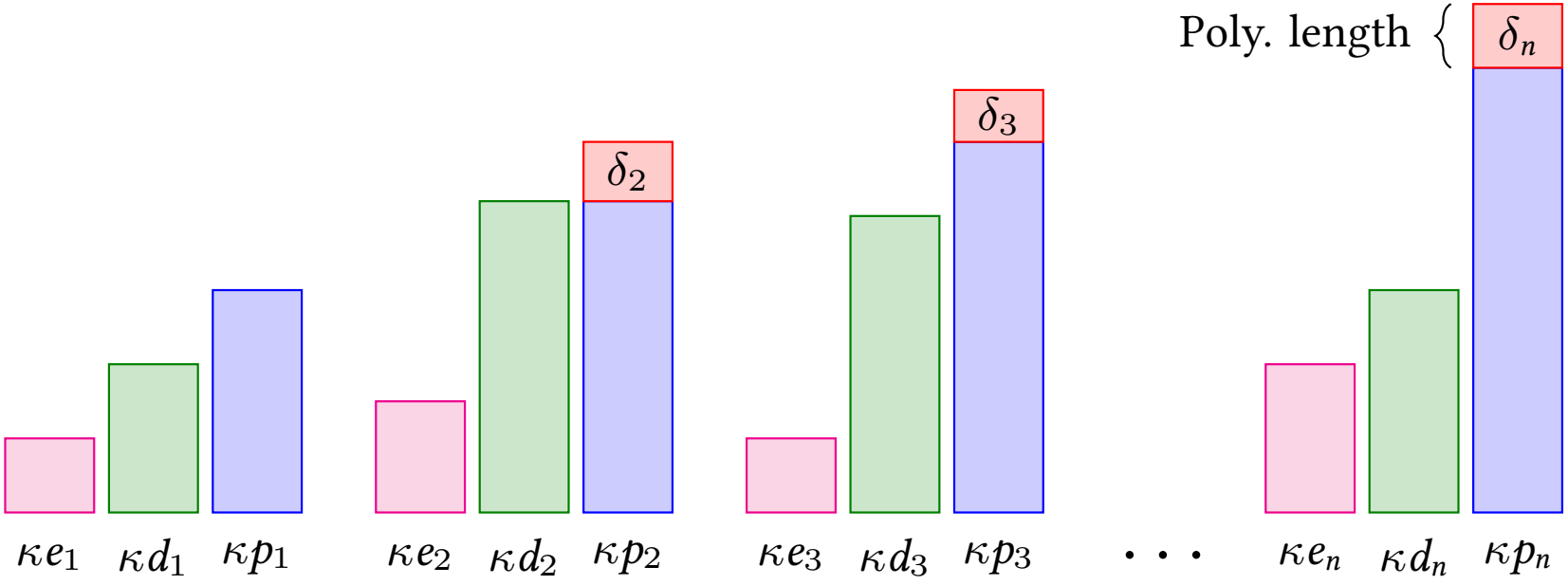
1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



- The δ_i can be found in polynomial time.
- The δ_i are so small relative to κ that schedulability is *unaffected*.

Outline of the Reduction

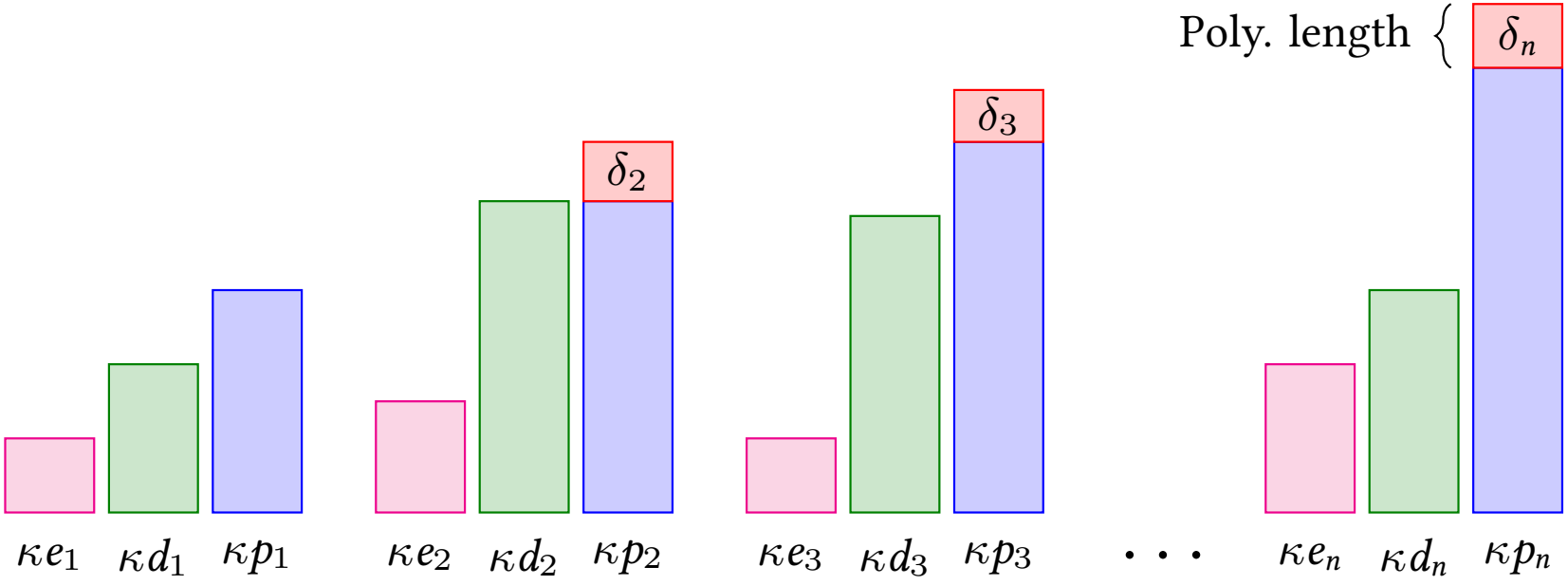
1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



- The δ_i can be found in polynomial time.
- The δ_i are so small relative to κ that schedulability is *unaffected*.

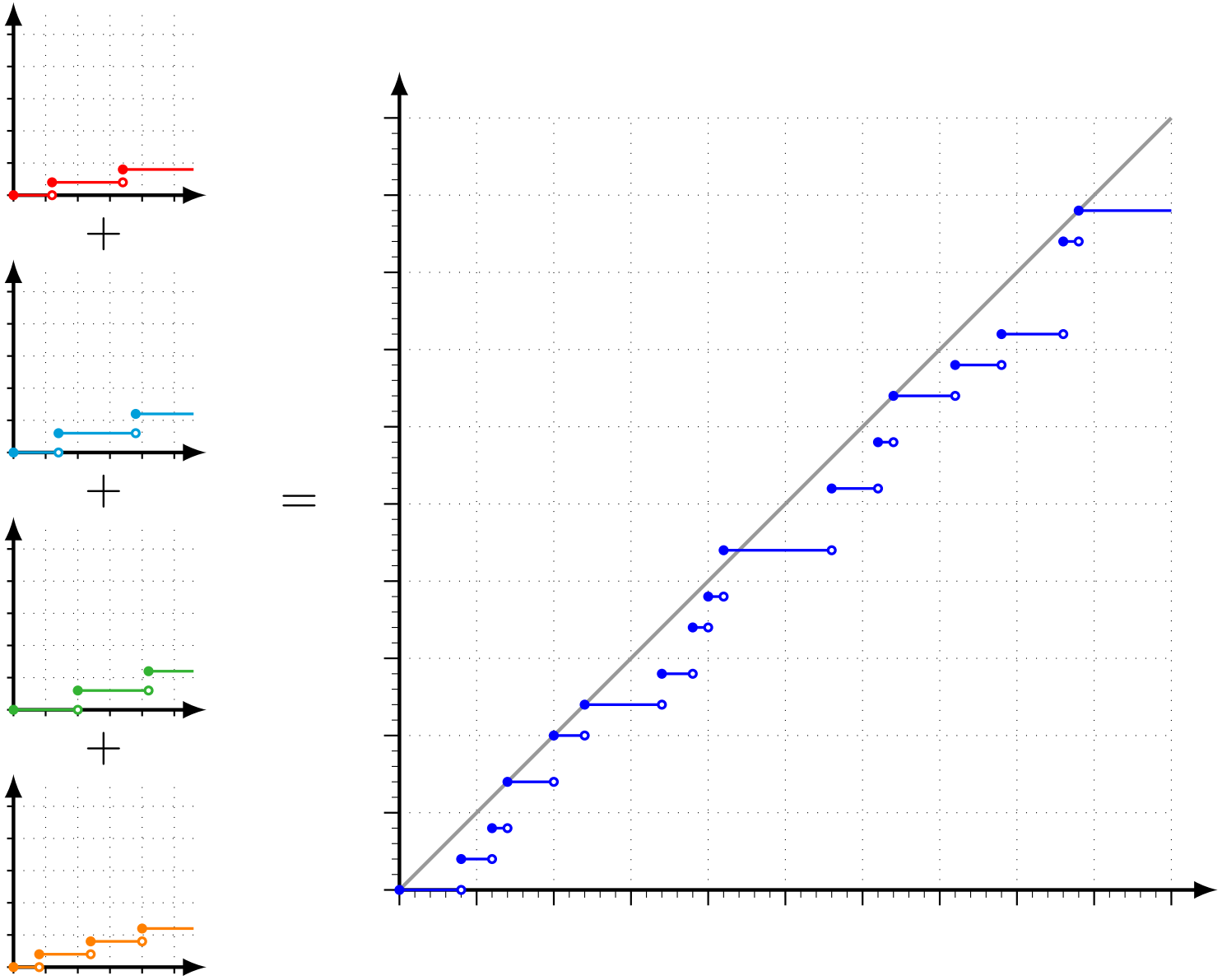
Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.

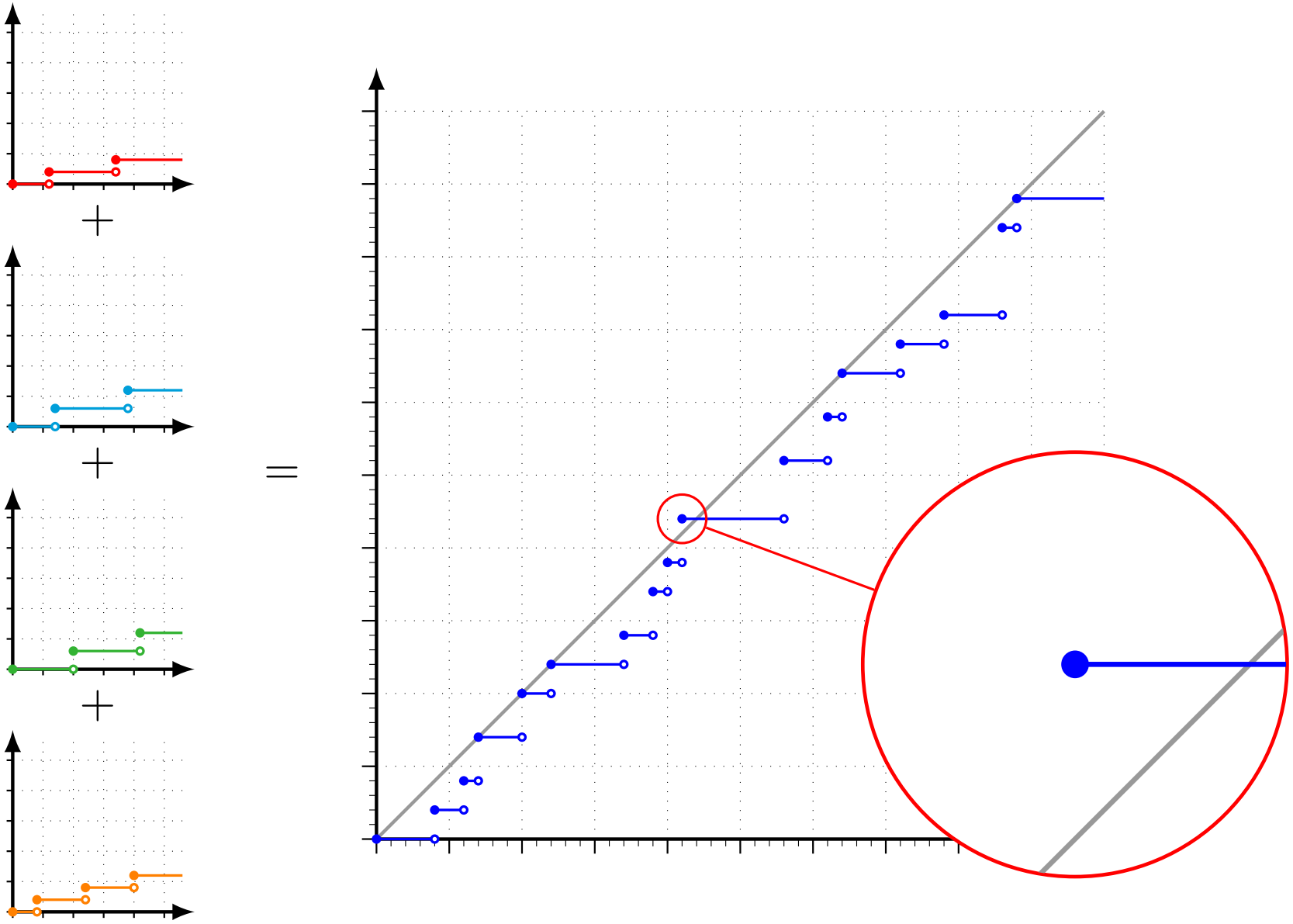


- The δ_i can be found in polynomial time. ✓
- The δ_i are so small relative to κ that schedulability is *unaffected*.

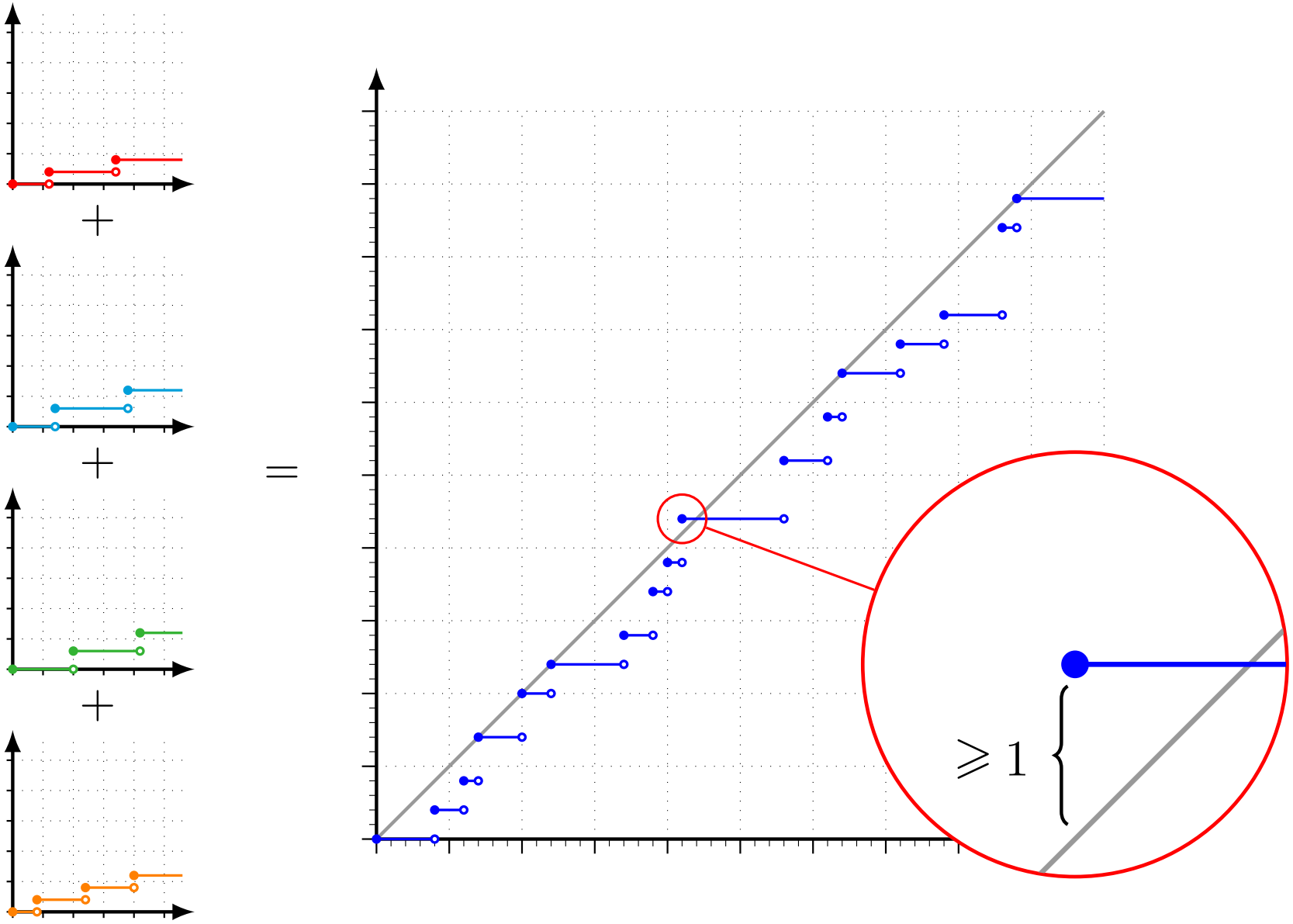
Schedulability unaffected



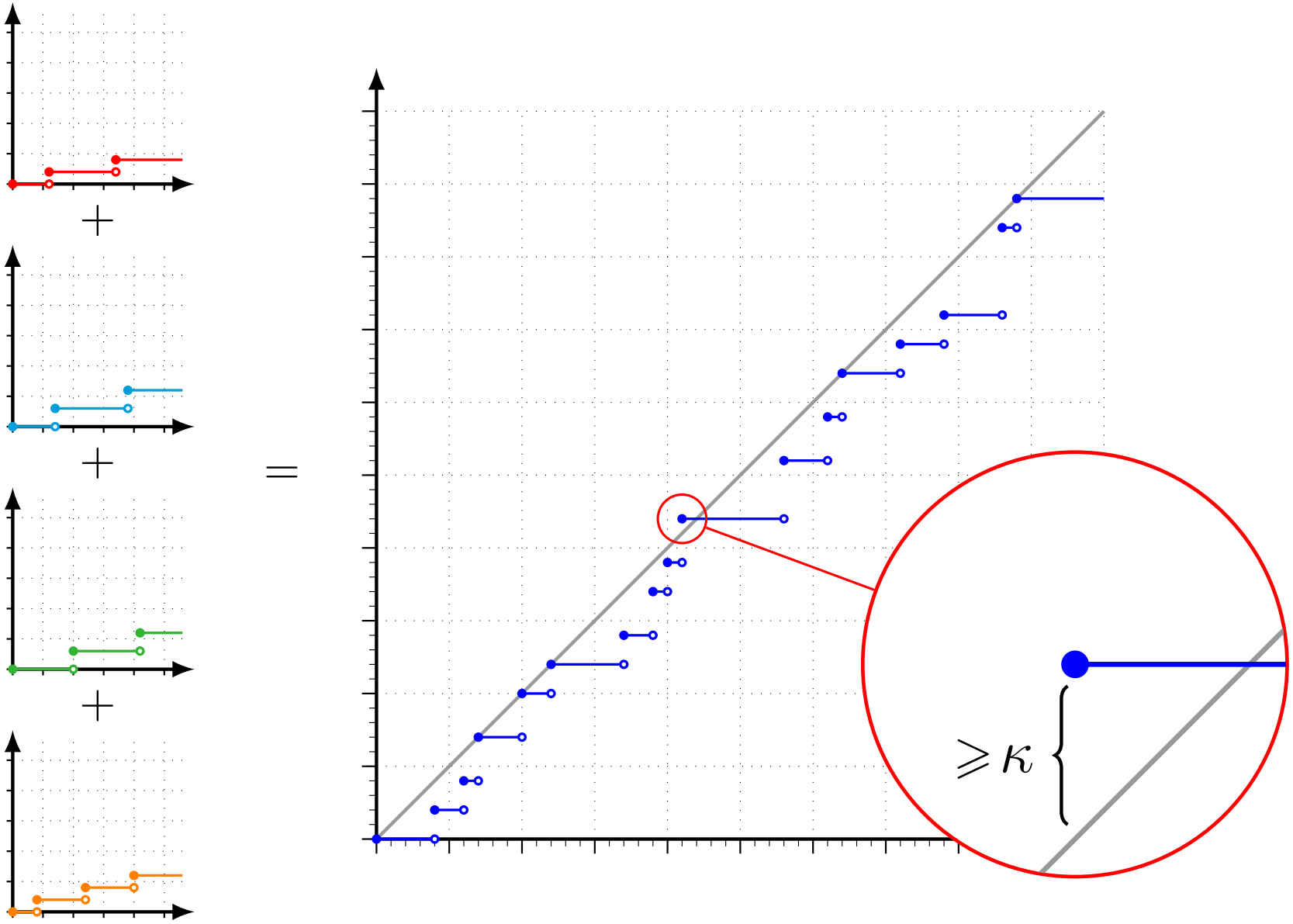
Schedulability unaffected



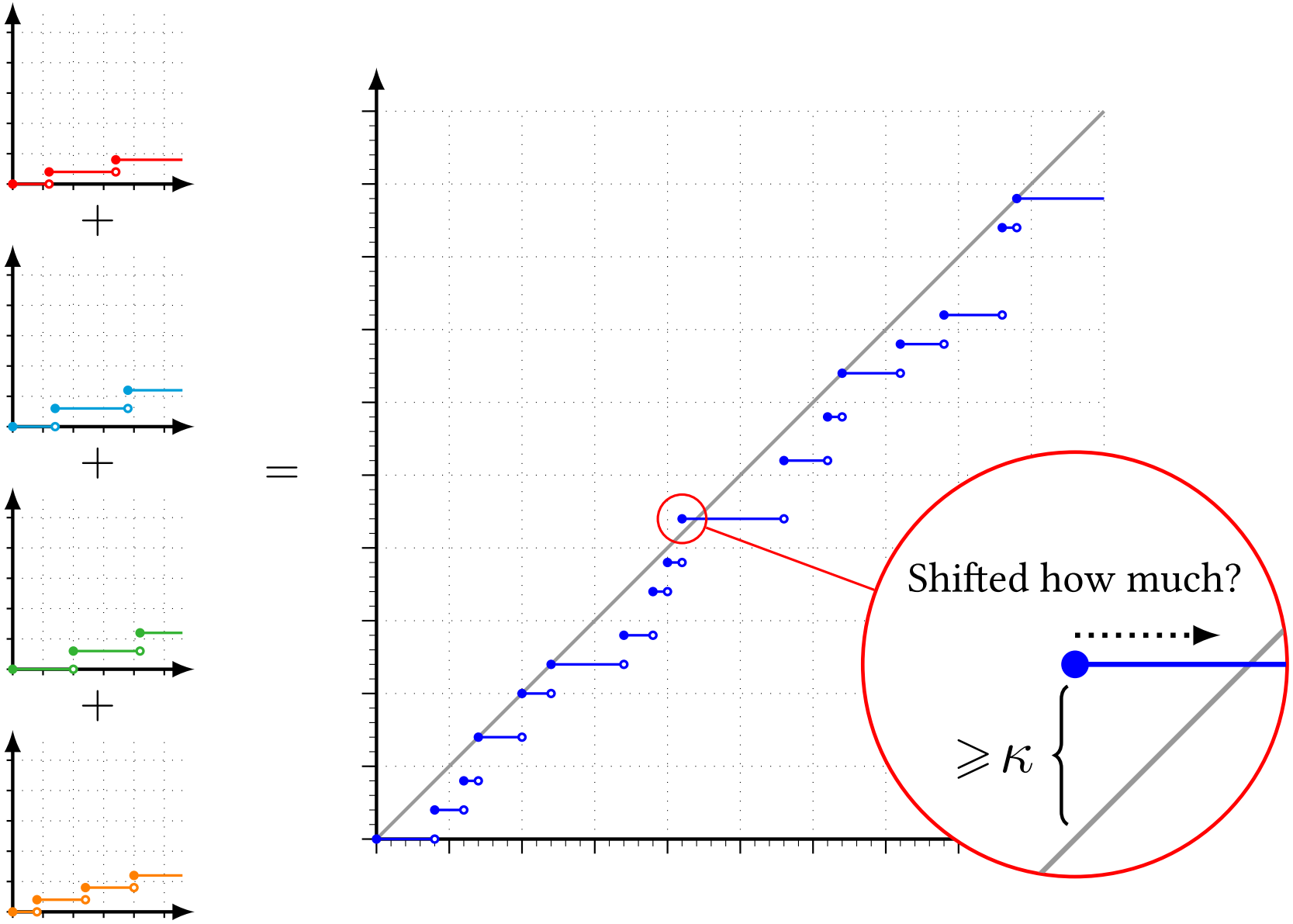
Schedulability unaffected



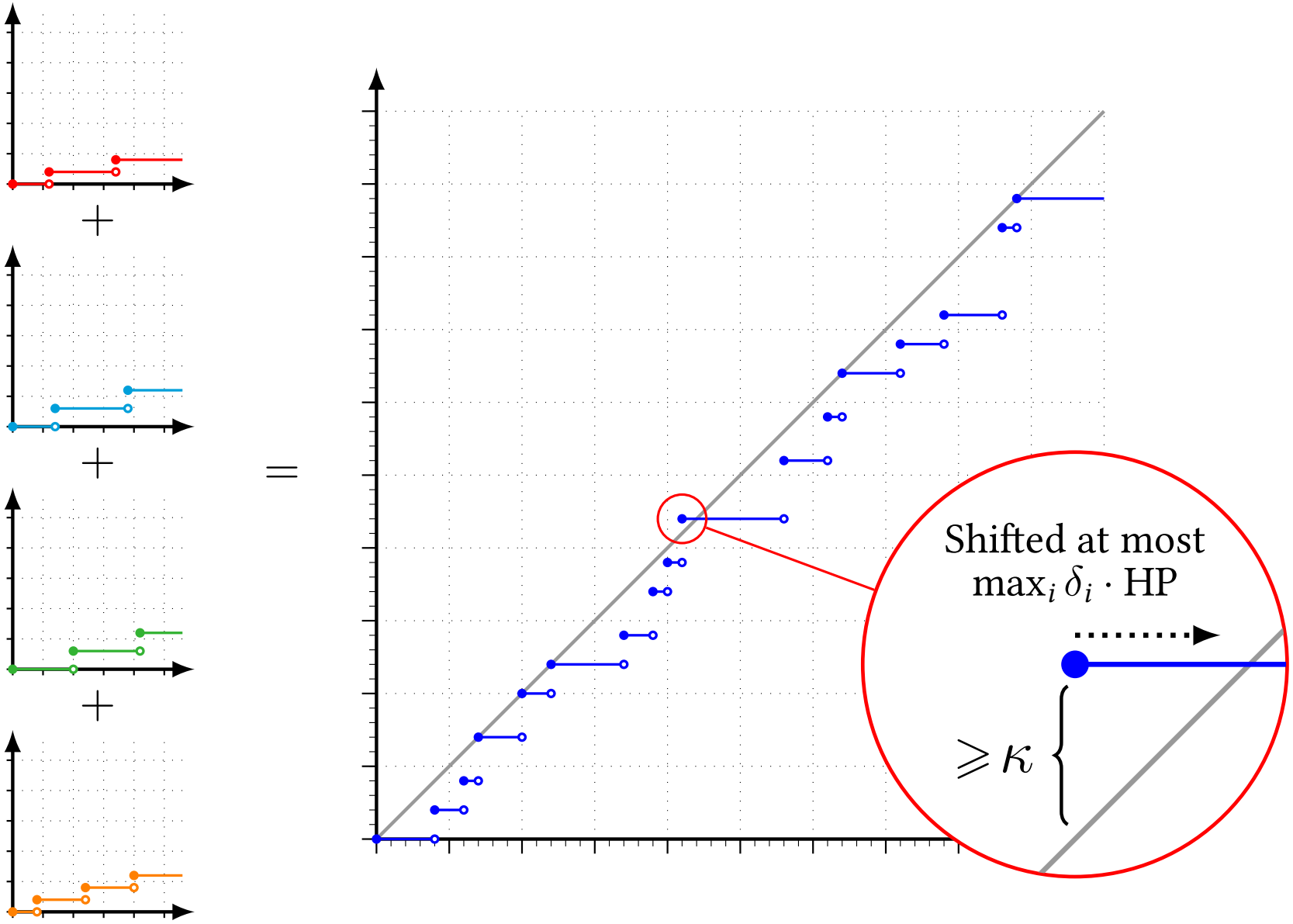
Schedulability unaffected



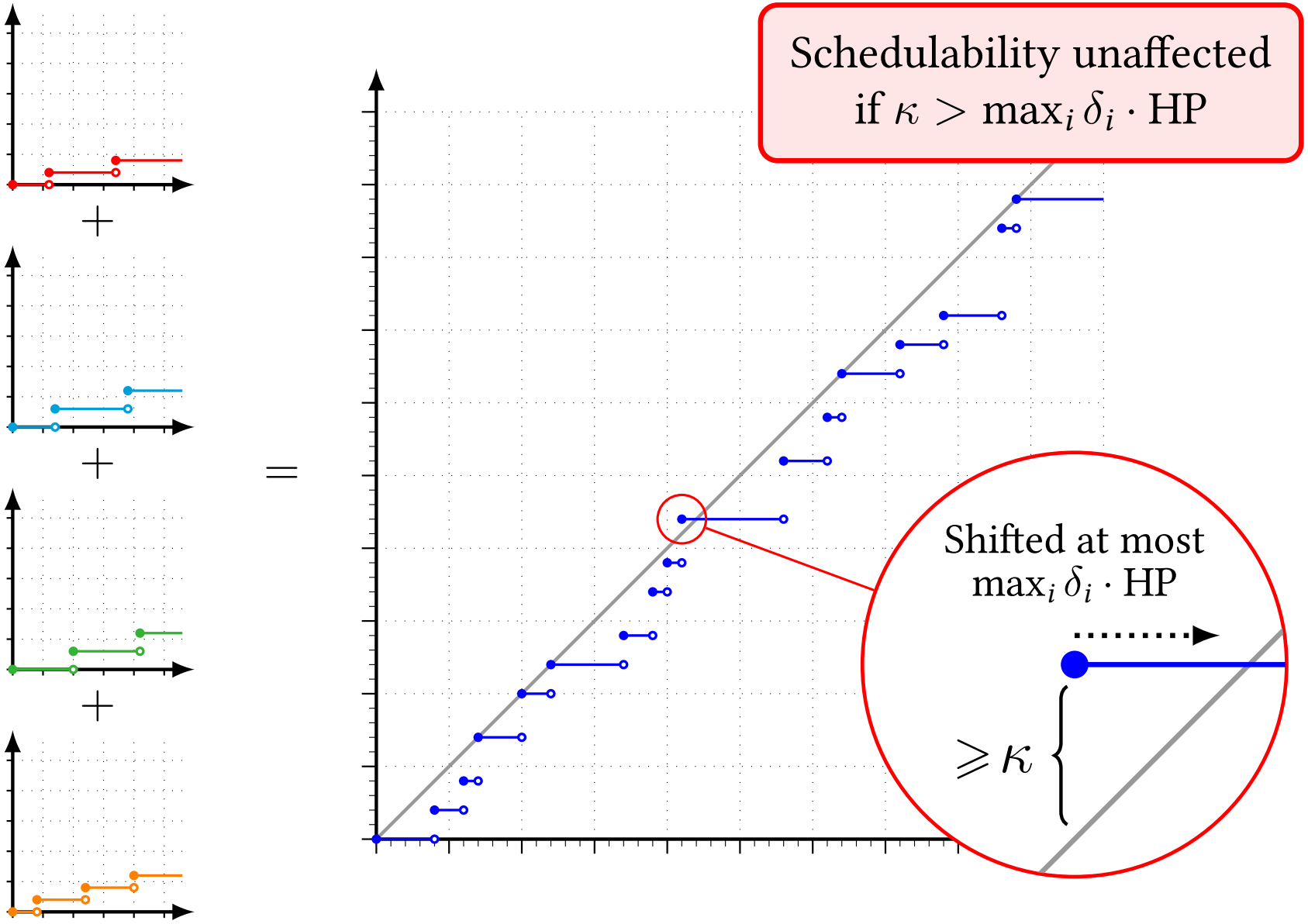
Schedulability unaffected



Schedulability unaffected

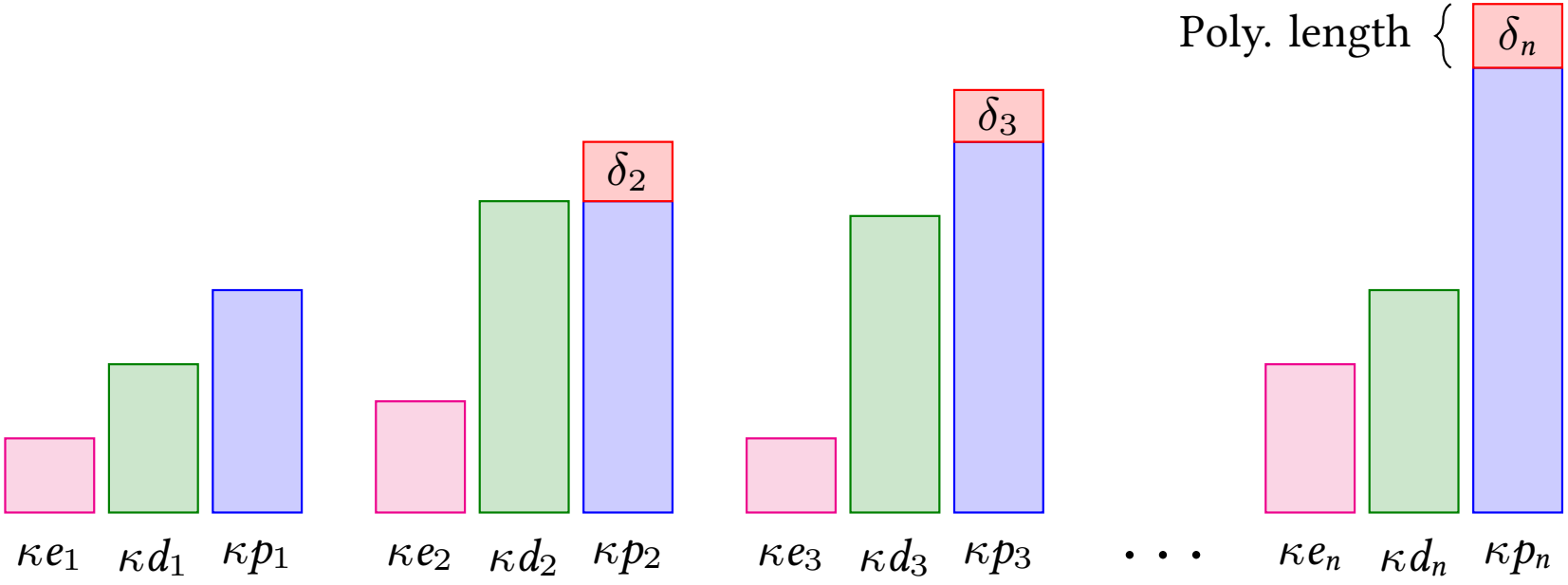


Schedulability unaffected



Outline of the Reduction

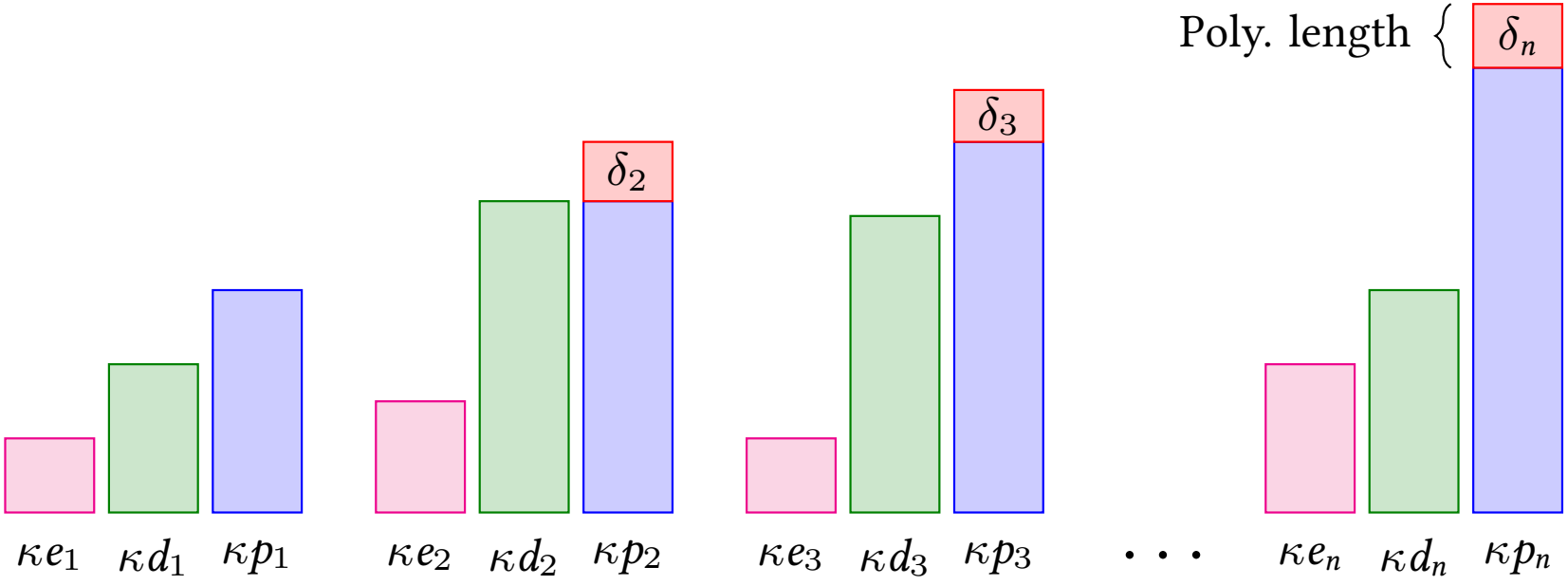
1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



- The δ_i can be found in polynomial time. ✓
- The δ_i are so small relative to κ that schedulability is *unaffected*.

Outline of the Reduction

1. Scale all task parameters uniformly by a *huge* number κ.
2. Add *small* numbers δ_i to each period so that the periods become pairwise coprime.



- The δ_i can be found in polynomial time. ✓
- The δ_i are so small relative to κ that schedulability is *unaffected*. ✓

A Tale of Two Reductions

EDF-schedulability

- Constrained deadlines
- Bounded utilization

EDF-schedulability

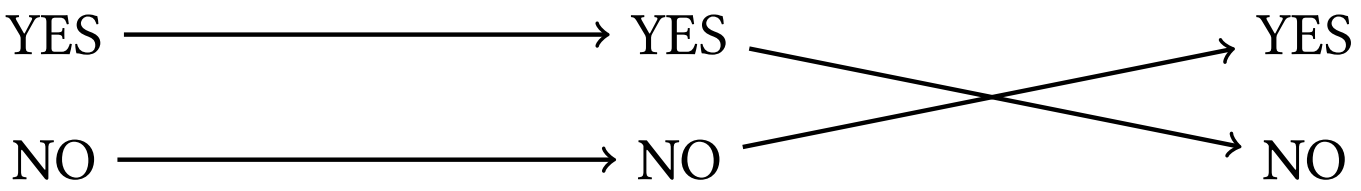
- Constrained deadlines
- Bounded utilization
- **Pairwise coprime periods**

FP-schedulability

- Implicit deadlines

—or—

- Constrained deadlines
- Bounded utilization



coNP-hard

coNP-hard

NP-hard

Conclusions

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-complete	Weakly NP-complete	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-complete for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Conclusions

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-complete	Weakly NP-complete	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-complete for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Conclusions

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-complete	Weakly NP-complete	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-complete for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Conclusions

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-complete	Weakly NP-complete	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-complete for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

Conclusions

		Implicit deadlines ($d = p$)	Constrained deadlines ($d \leqslant p$)	Arbitrary deadlines (d, p unrelated)
FP	Arbitrary utilization	Weakly NP-complete	Weakly NP-complete	Weakly NP-hard
	Utilization bounded by a constant c	Polynomial time for $c \leqslant \ln 2$ and RM priorities	Weakly NP-complete for $0 < c < 1$	Weakly NP-hard for $0 < c < 1$
EDF	Arbitrary utilization	Polynomial time	Strongly coNP-complete	Strongly coNP-complete
	Utilization bounded by a constant c	Polynomial time	Weakly coNP-complete for $0 < c < 1$	Weakly coNP-complete for $0 < c < 1$

$\forall$Thank you!

$\diamond$

$\exists$Questions?