Towards Efficient Explainability of Schedulability Properties in Real-Time Systems

Sanjoy Baruah
Washington University in Saint Louis

Pontus Ekberg
Uppsala University

ECRTS, Vienna 2023

The broader picture

The broader picture

System designer

Certification authority

The broader picture

System designer

Comes up with solution

Certification authority

The broader picture

Comes up with solution

Sends *explanation* (or proof/certificate)

The broader picture

Comes up with solution

Sends *explanation* (or proof/certificate)

Verifies solution

The broader picture

System designer

Certification authority

Comes up with solution

Sends *explanation* (or proof/certificate)

Verifies solution

Can spend lots of time/resources

Verification should be *efficient*

The broader picture

System designer

Certification authority

Comes up with solution

Sends *explanation* (or proof/certificate)

Verifies solution

Can spend lots of time/resources

Verification should be *efficient*

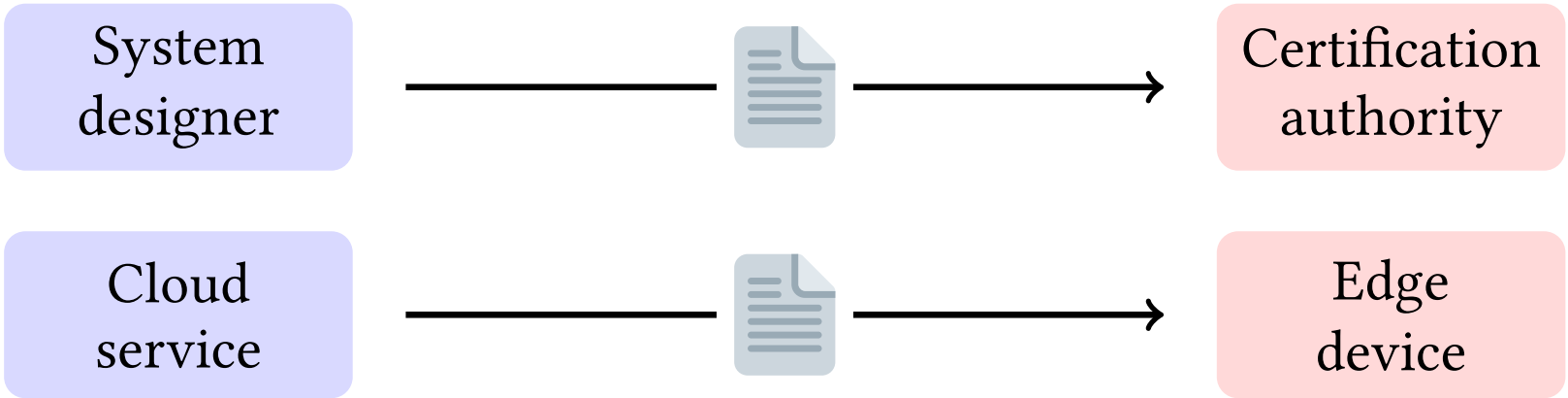
Ideally, the explanation is *formally* verifiable

Foundational Response-Time Analysis as Explainable Evidence of Timeliness
by Maida, Bozhko and Brandenburg (ECRTS 2022)

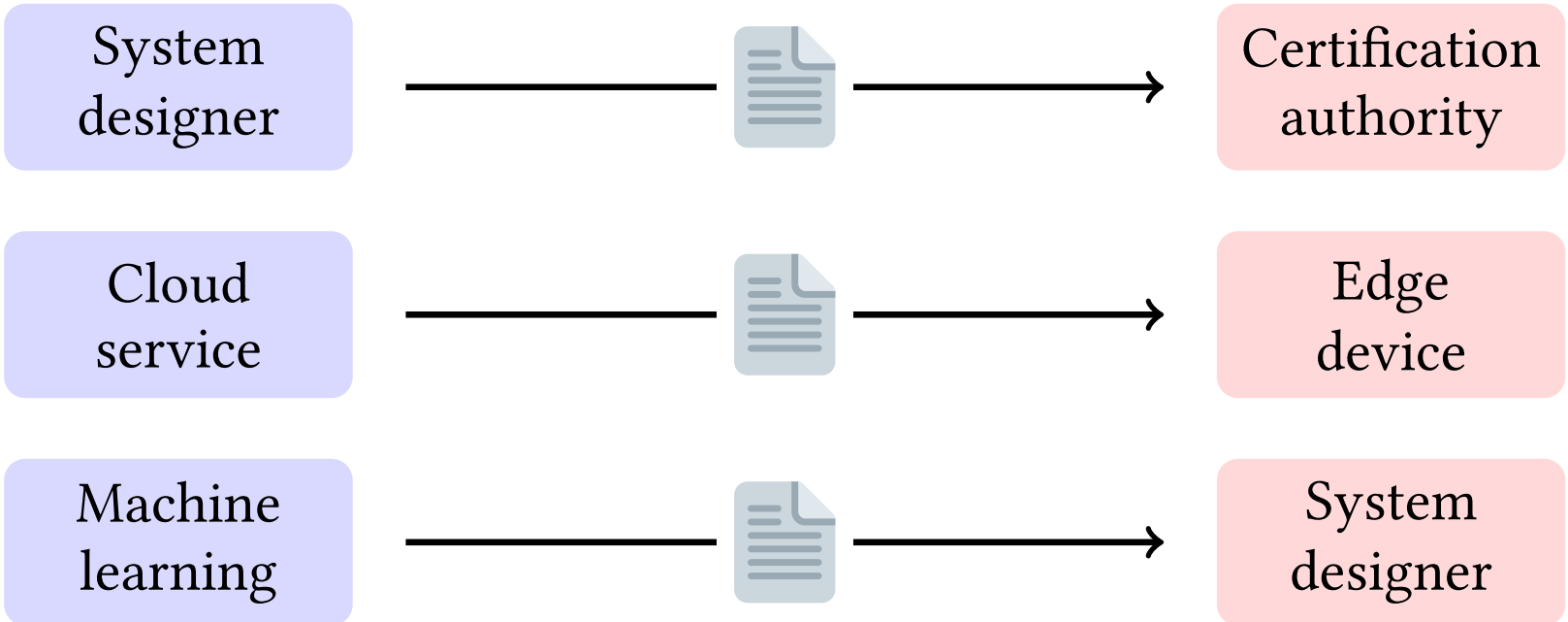
CertiCAN certifying CAN analyses and their results
by Fradet, Guo and Quinton (Real-Time Systems 2023)

The broader picture

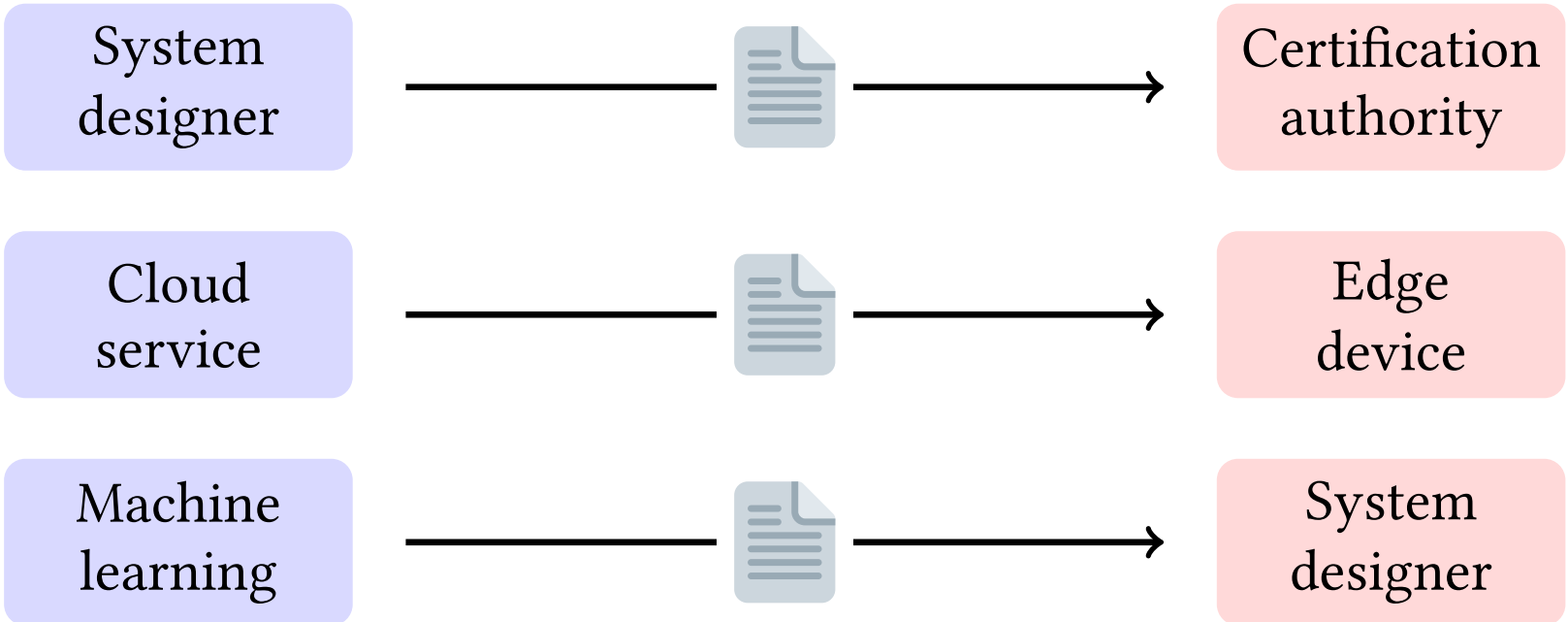
The broader picture



The broader picture



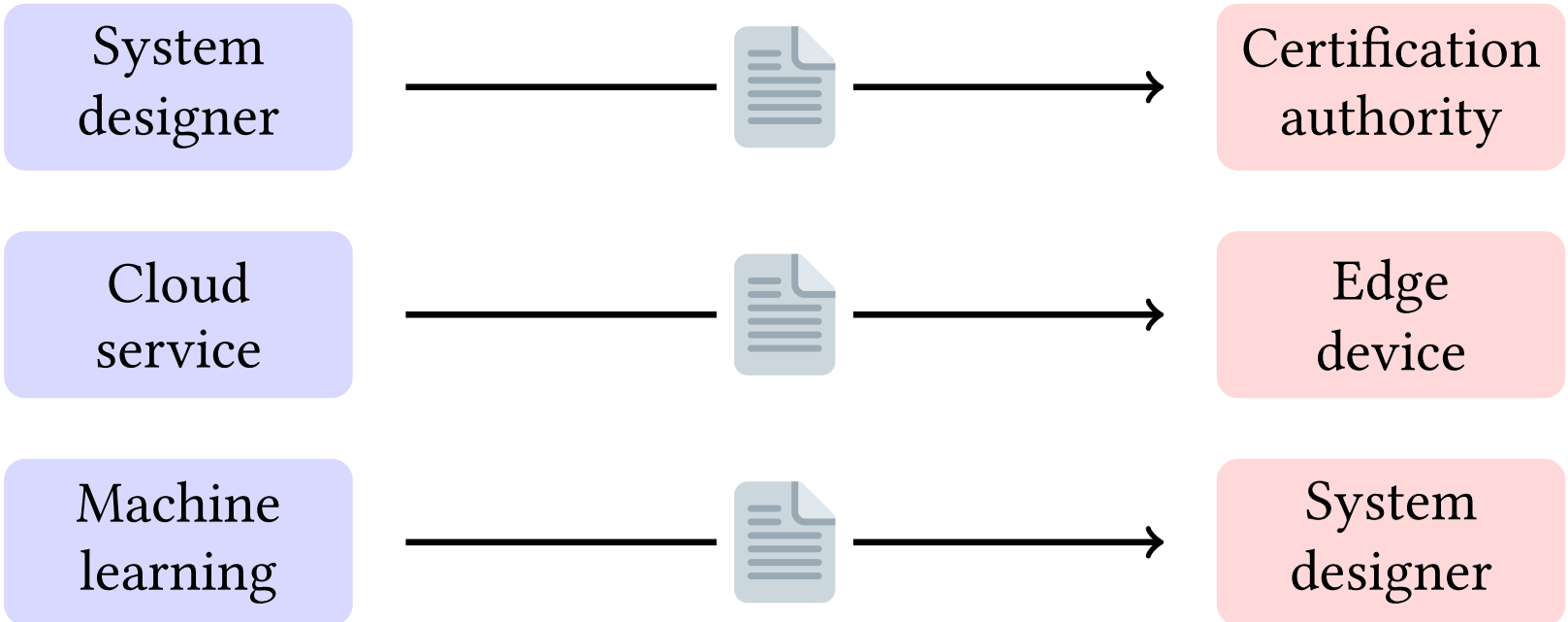
The broader picture



This work

- Classify schedulability problems as efficiently explainable or not

The broader picture



This work

- Classify schedulability problems as efficiently explainable or not
- Investigate techniques for dealing with the problems that are not

Example: Uniprocessor FP-schedulability

RTA (Joseph and Pandya, 1986, and others)

Constrained-deadline task system $\Gamma = \{\tau_1, \ldots, \tau_n\}$ is FP-schedulable iff the smallest positive fixed-point R_i,

$$R_i = C_i + \sum_{j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \times C_j,$$

is $\leqslant D_i$ for each $\tau_i \in \Gamma$.

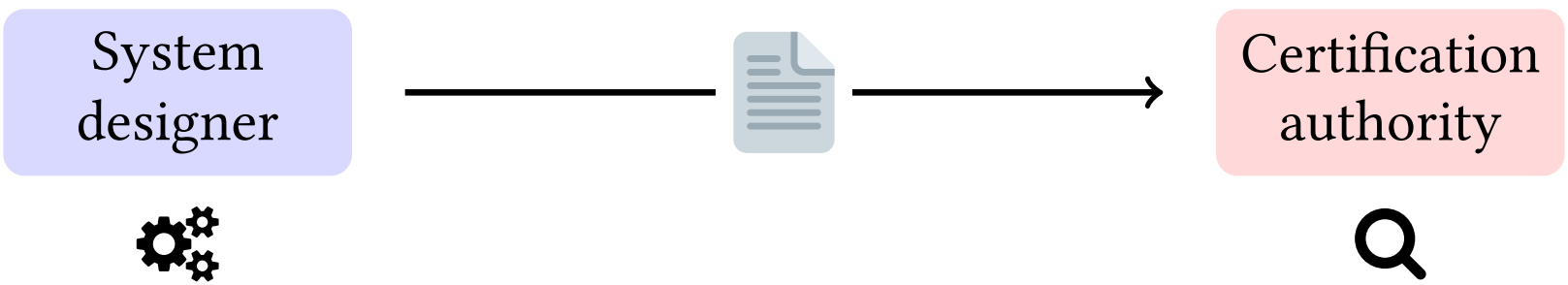
Example: Uniprocessor FP-schedulability

RTA (Joseph and Pandya, 1986, and others)

Constrained-deadline task system $\Gamma = \{\tau_1, \ldots, \tau_n\}$ is FP-schedulable iff the smallest positive fixed-point R_i,

$$R_i = C_i + \sum_{j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \times C_j,$$

is $\leqslant D_i$ for each $\tau_i \in \Gamma$.



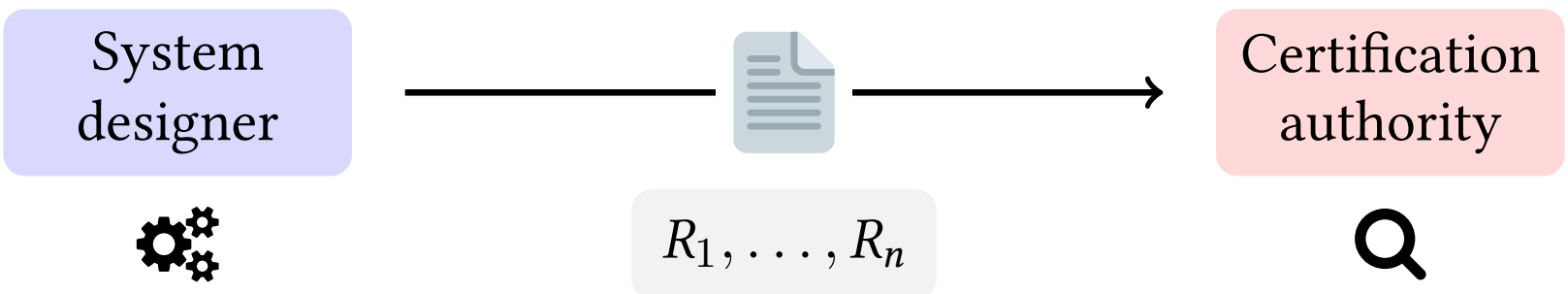
Example: Uniprocessor FP-schedulability

RTA (Joseph and Pandya, 1986, and others)

Constrained-deadline task system $\Gamma = \{\tau_1, \ldots, \tau_n\}$ is FP-schedulable iff the smallest positive fixed-point R_i,

$$R_i = C_i + \sum_{j \in \mathrm{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil \times C_j,$$

is $\leqslant D_i$ for each $\tau_i \in \Gamma$.



Example 2: Uniprocessor EDF-schedulability

PDA (Baruah et al., 1990)

Task system $\Gamma = \{\tau_1, \ldots, \tau_n\}$ is EDF-schedulable iff

$$\sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t,$$

for all $t \in \{0, 1, \ldots, \mathrm{HP}(\Gamma)\}$.

Example 2: Uniprocessor EDF-schedulability

PDA (Baruah et al., 1990)

Task system $\Gamma = \{\tau_1, \ldots, \tau_n\}$ is EDF-schedulable iff

$$\sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t,$$

for all $t \in \{0, 1, \ldots, \mathrm{HP}(\Gamma)\}$.

$$\mathrm{dbf}_i(t) \stackrel{\text{def}}{=} \max\left(\left\lfloor \frac{t - D_i}{T_i} \right\rfloor + 1, 0\right) \times C_i$$

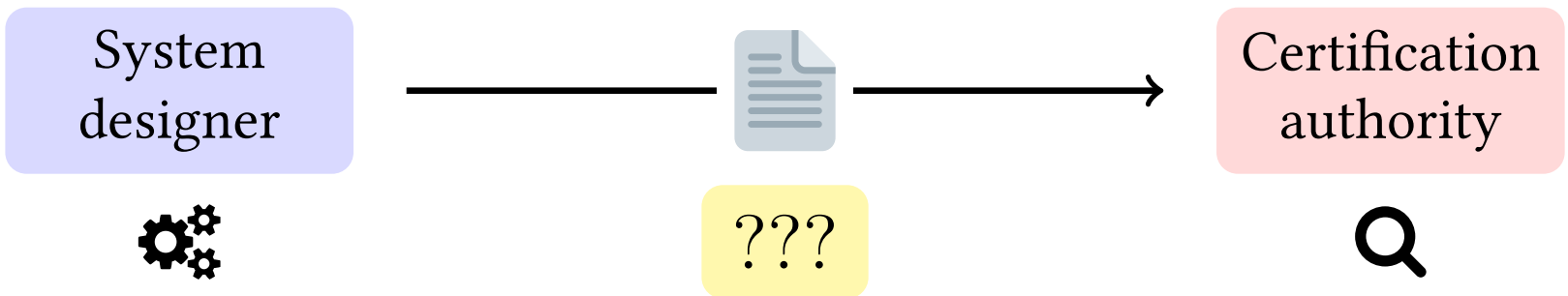
Example 2: Uniprocessor EDF-schedulability

PDA (Baruah et al., 1990)

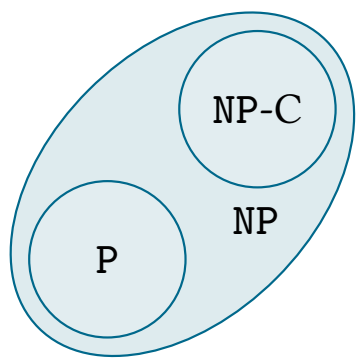
Task system $\Gamma = \{\tau_1, \ldots, \tau_n\}$ is EDF-schedulable iff

$$\sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t,$$

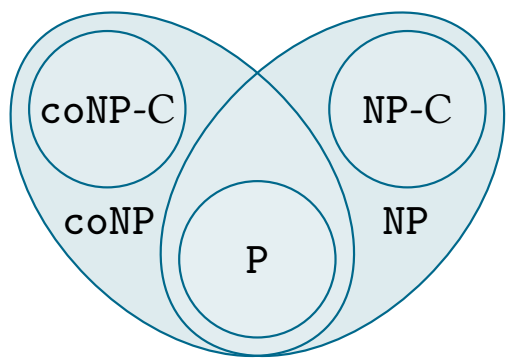
for all $t \in \{0, 1, \ldots, \mathrm{HP}(\Gamma)\}$.



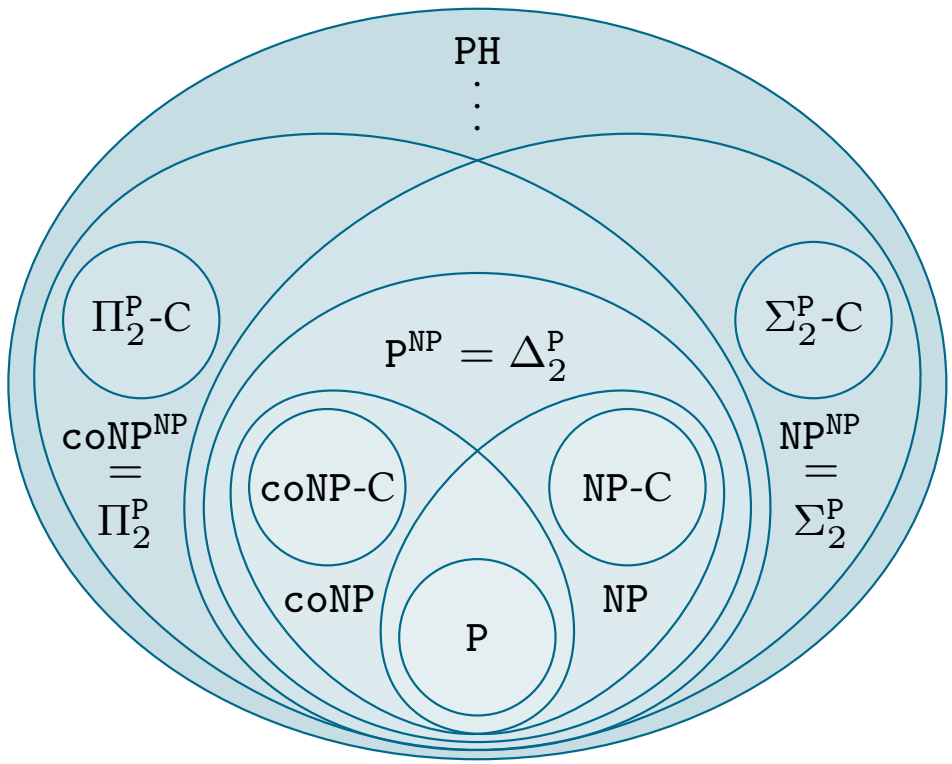
Does this relate to complexity?



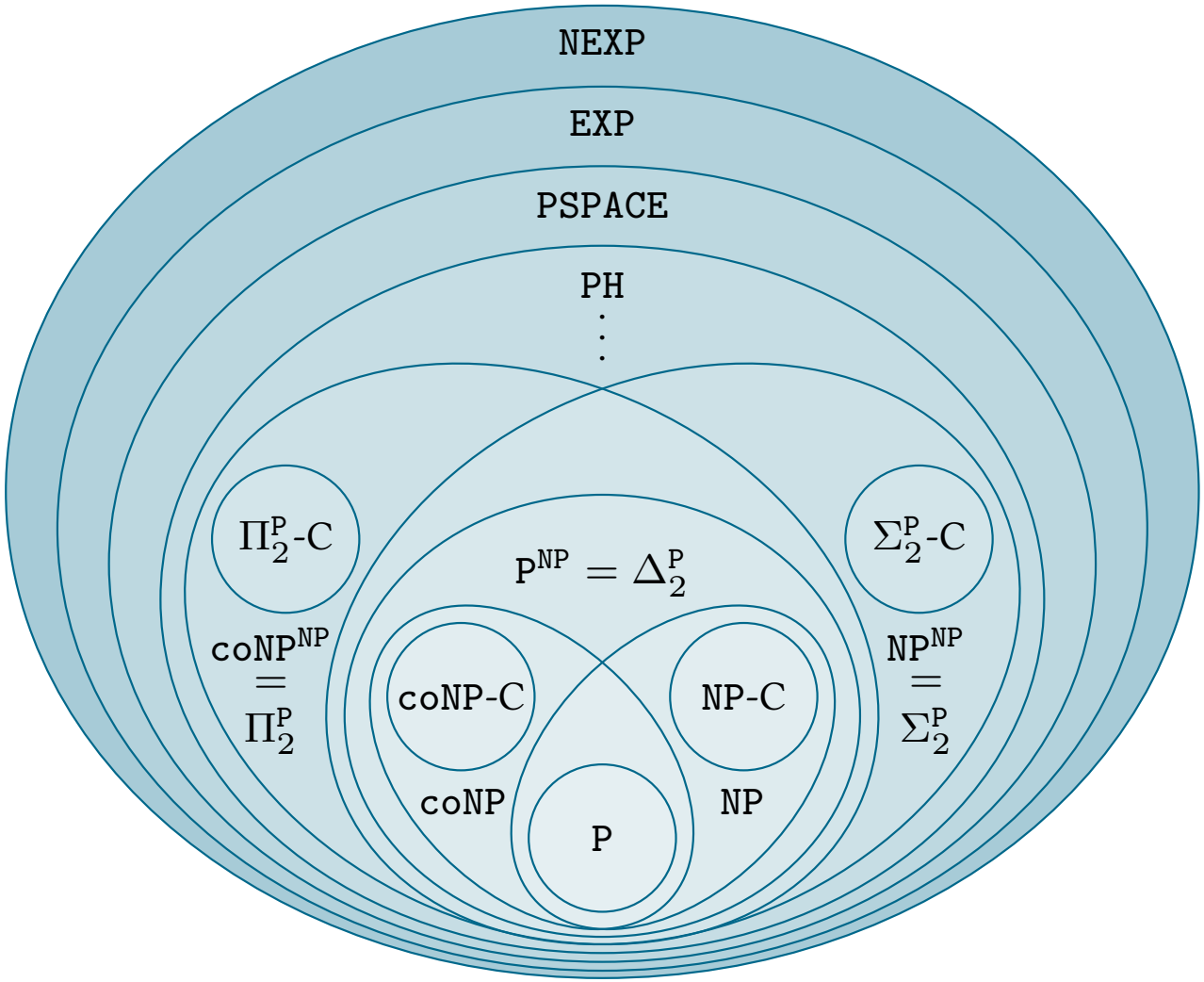
Does this relate to complexity?



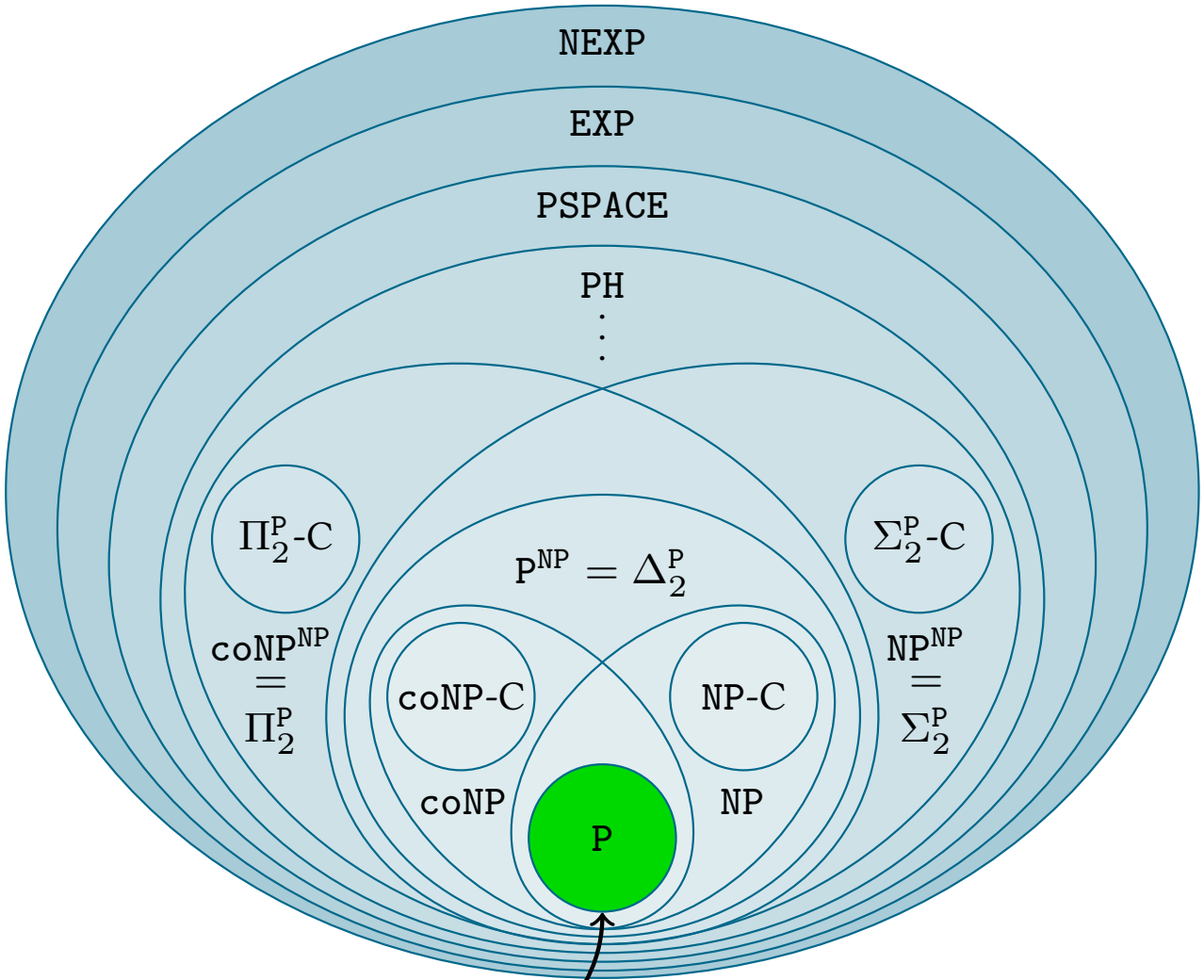
Does this relate to complexity?



Does this relate to complexity?

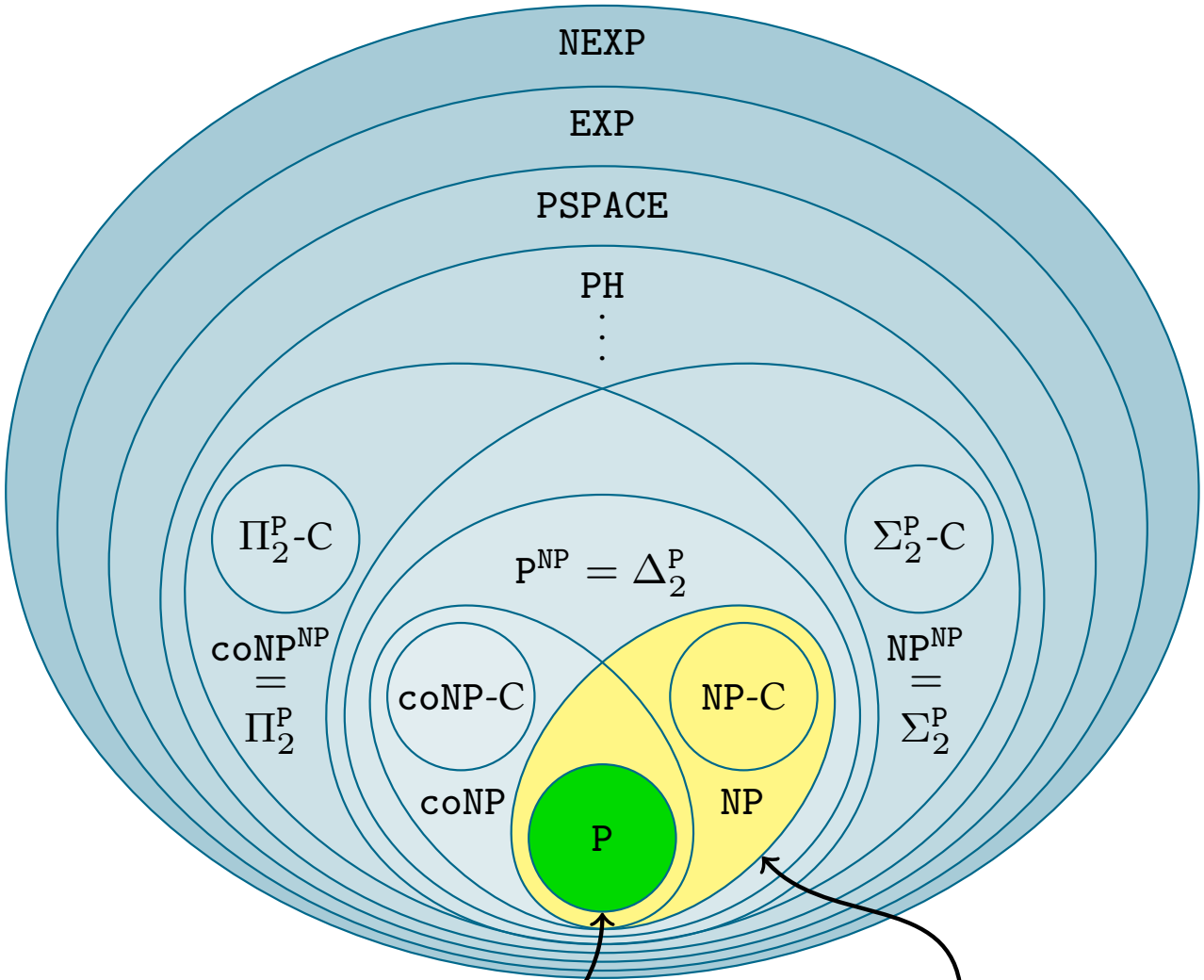


Does this relate to complexity?

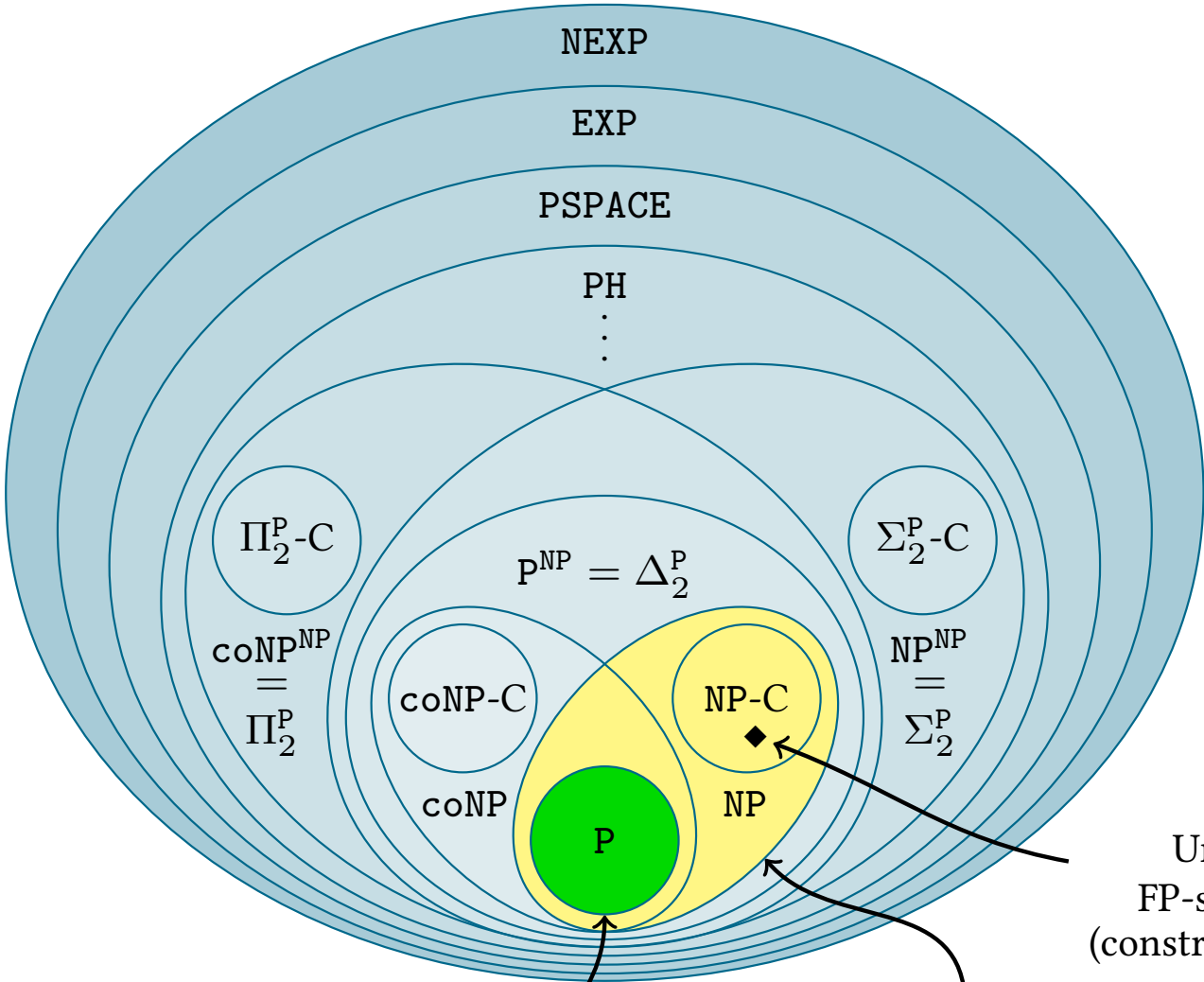


P = solvable in polynomial time

Does this relate to complexity?



Does this relate to complexity?

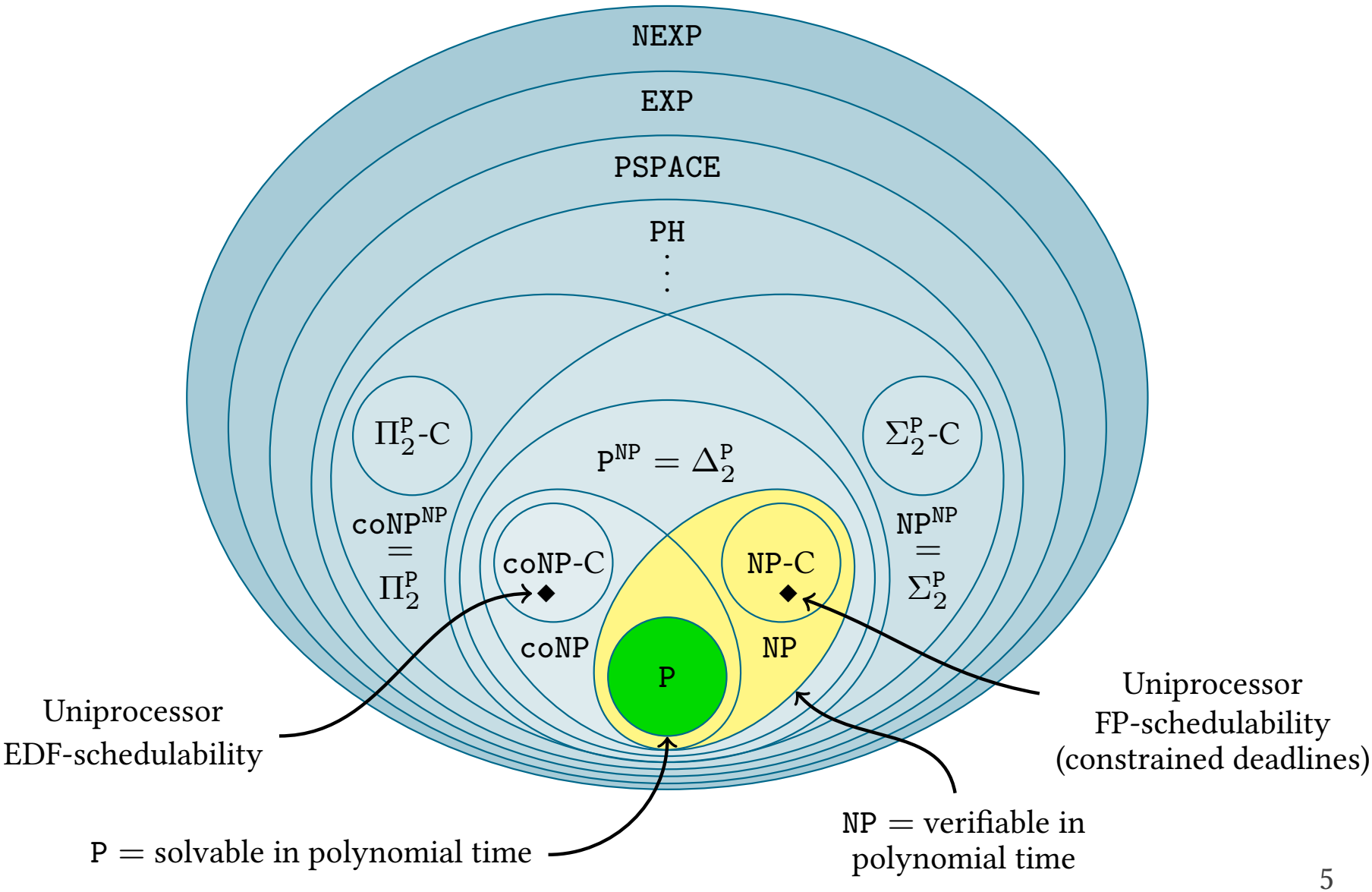


Uniprocessor FP-schedulability (constrained deadlines)

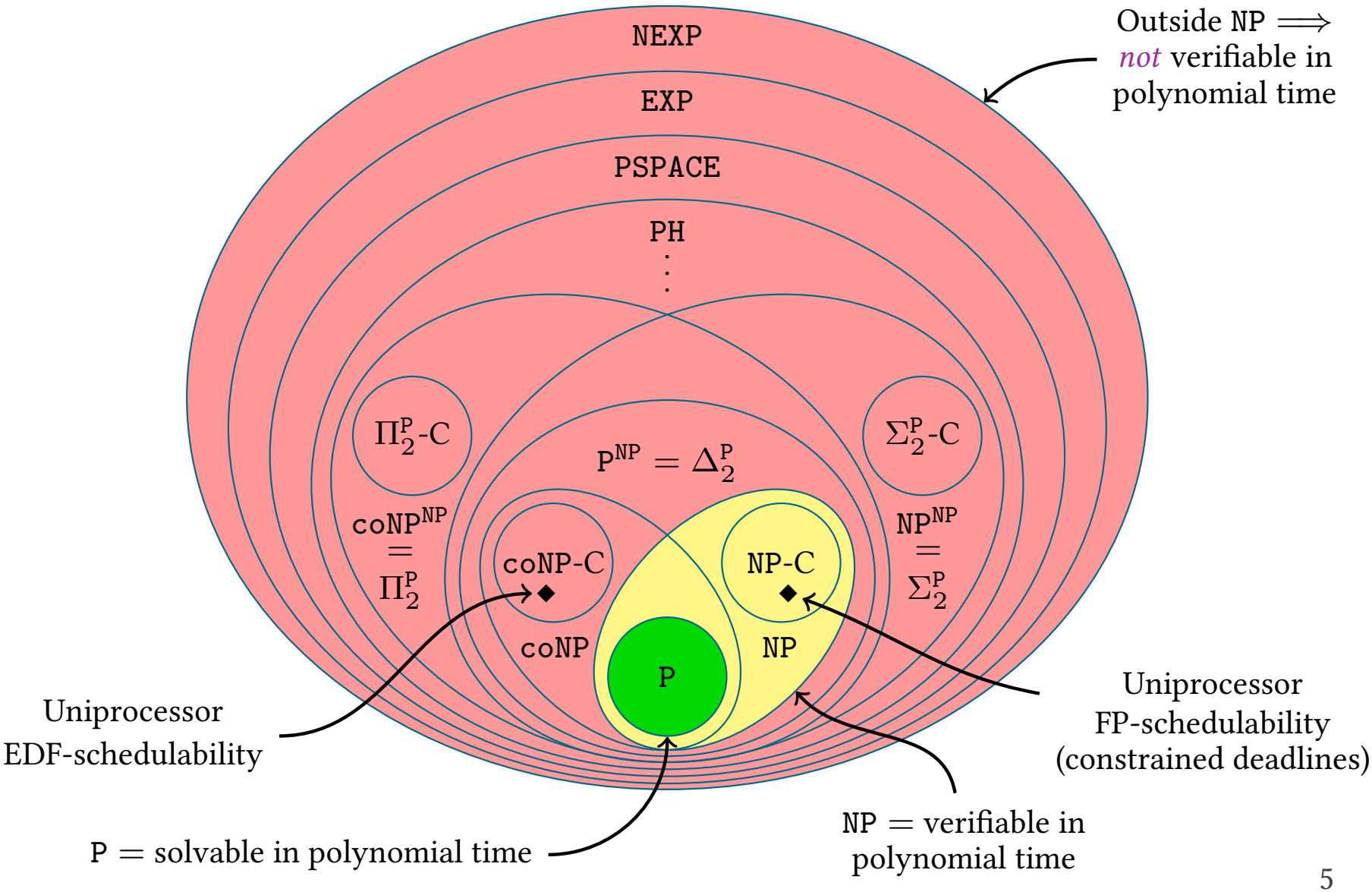
NP = verifiable in polynomial time

P = solvable in polynomial time

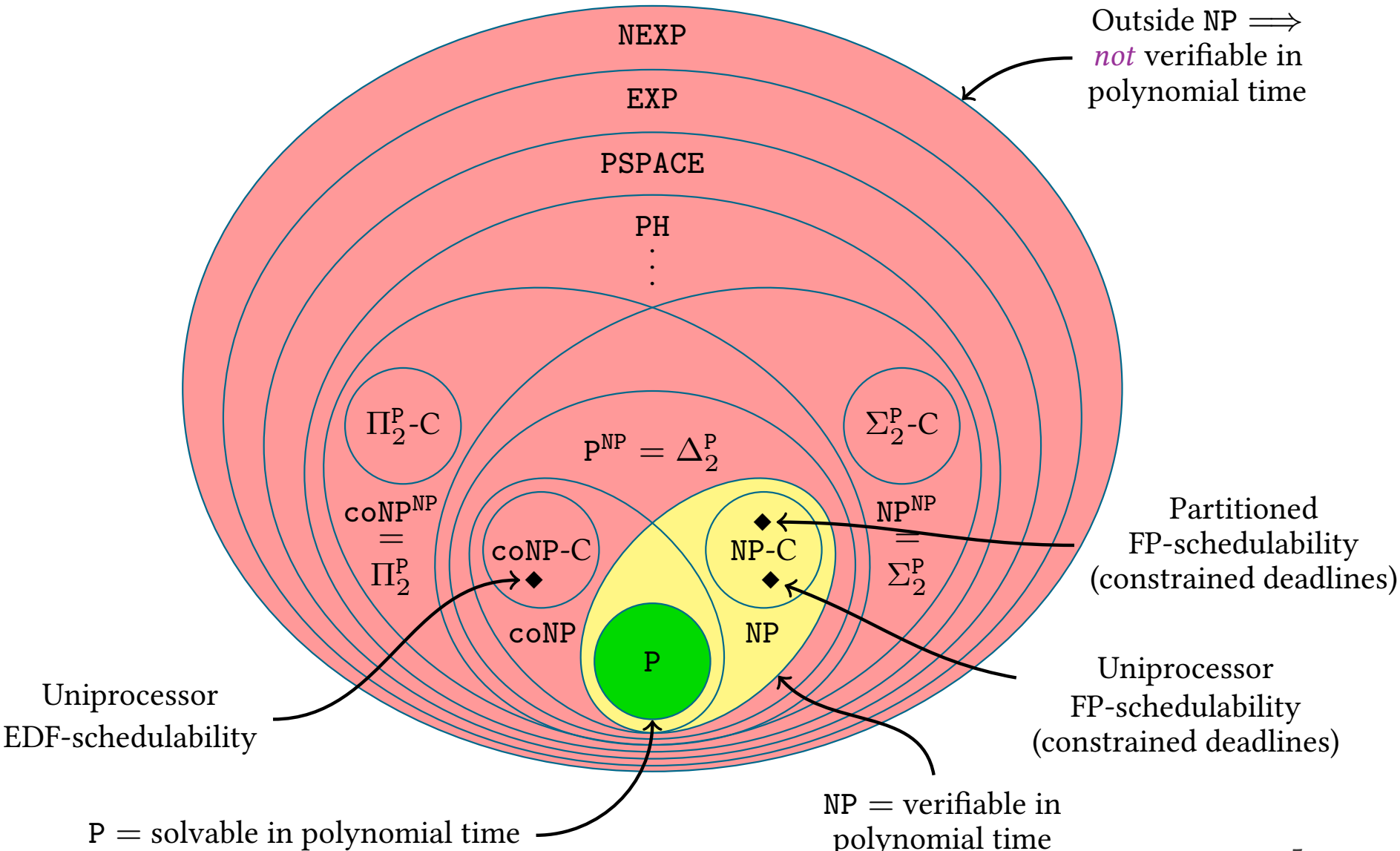
Does this relate to complexity?



Does this relate to complexity?

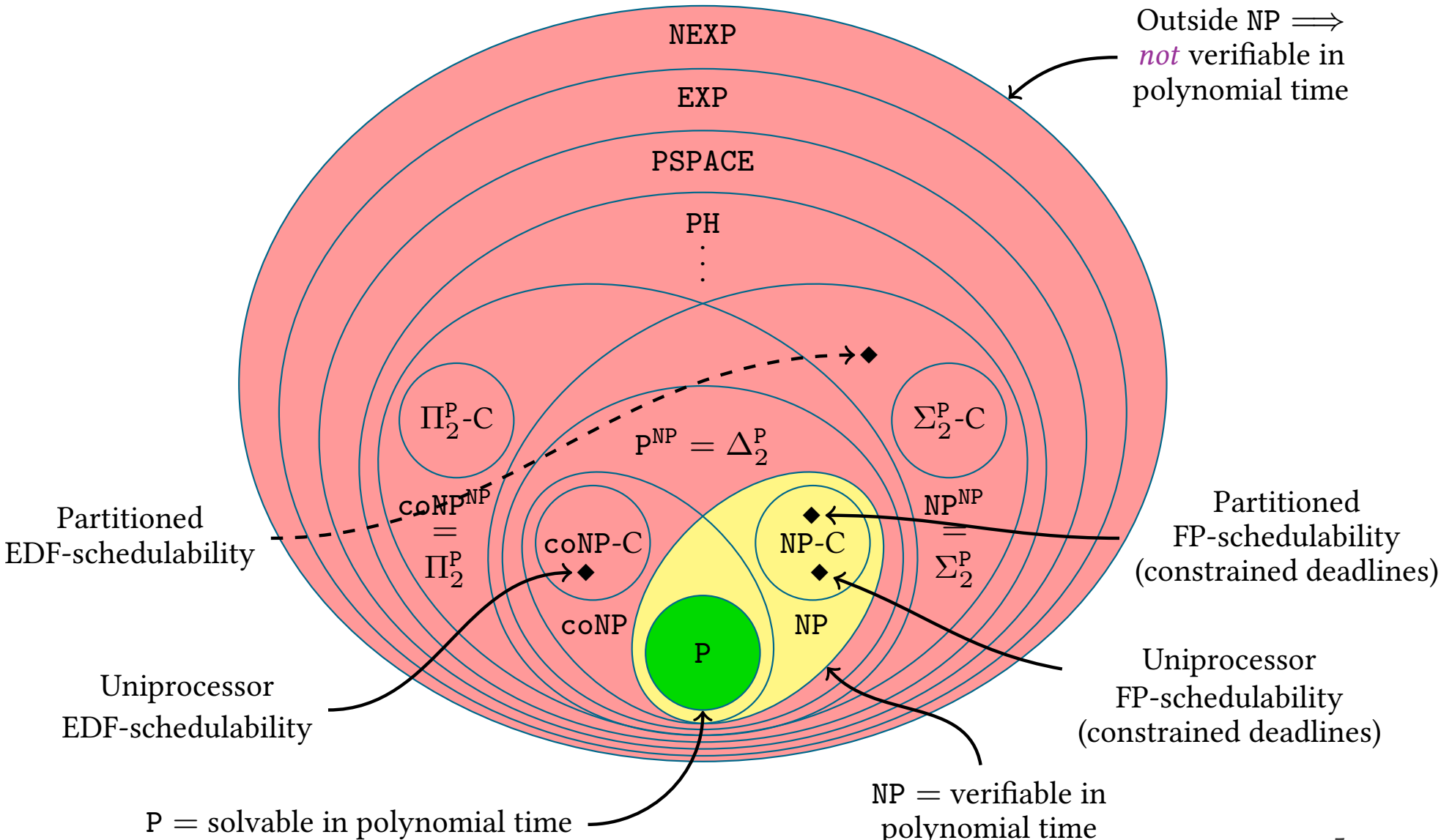


Does this relate to complexity?

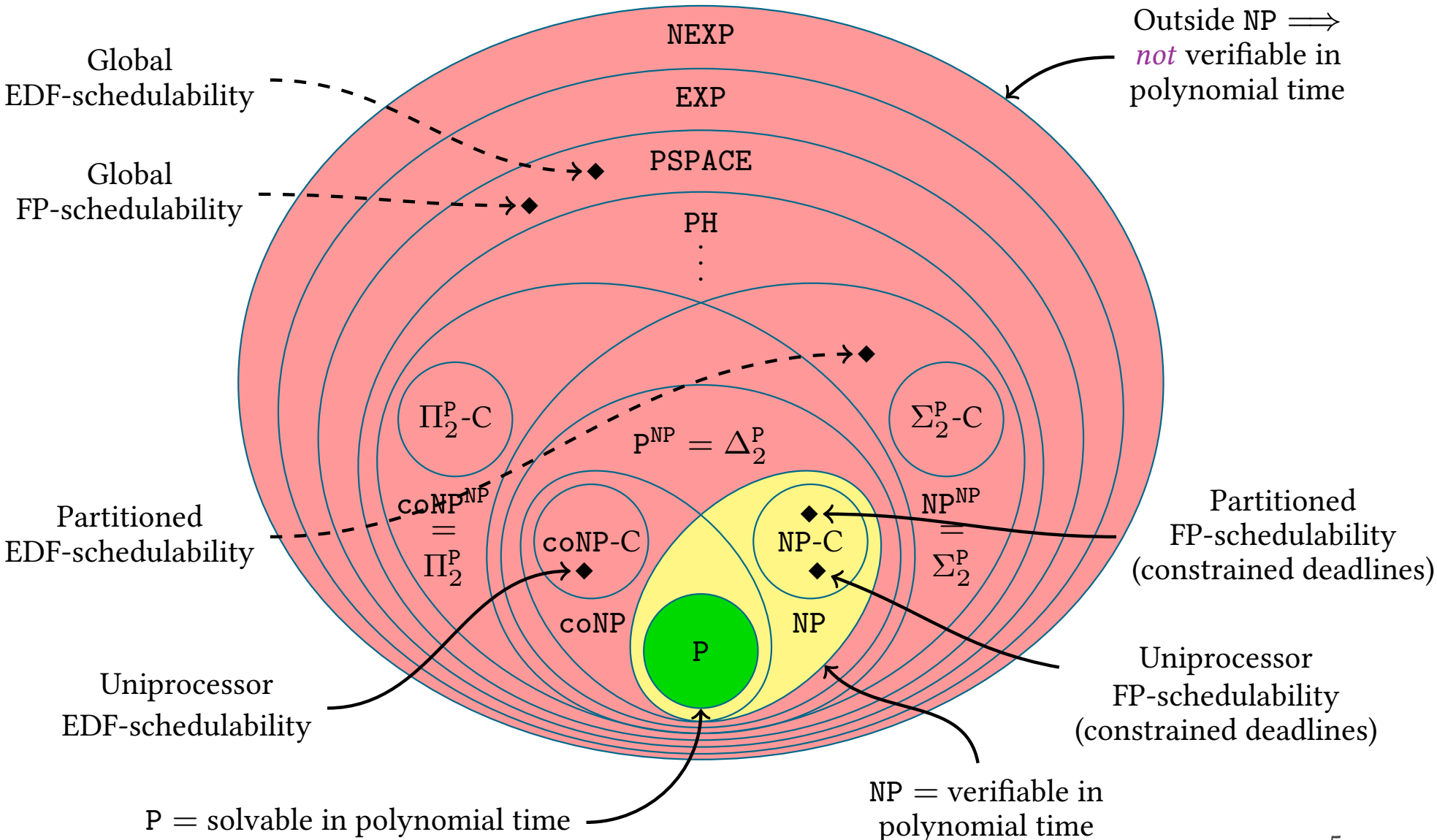


Does this relate to complexity?

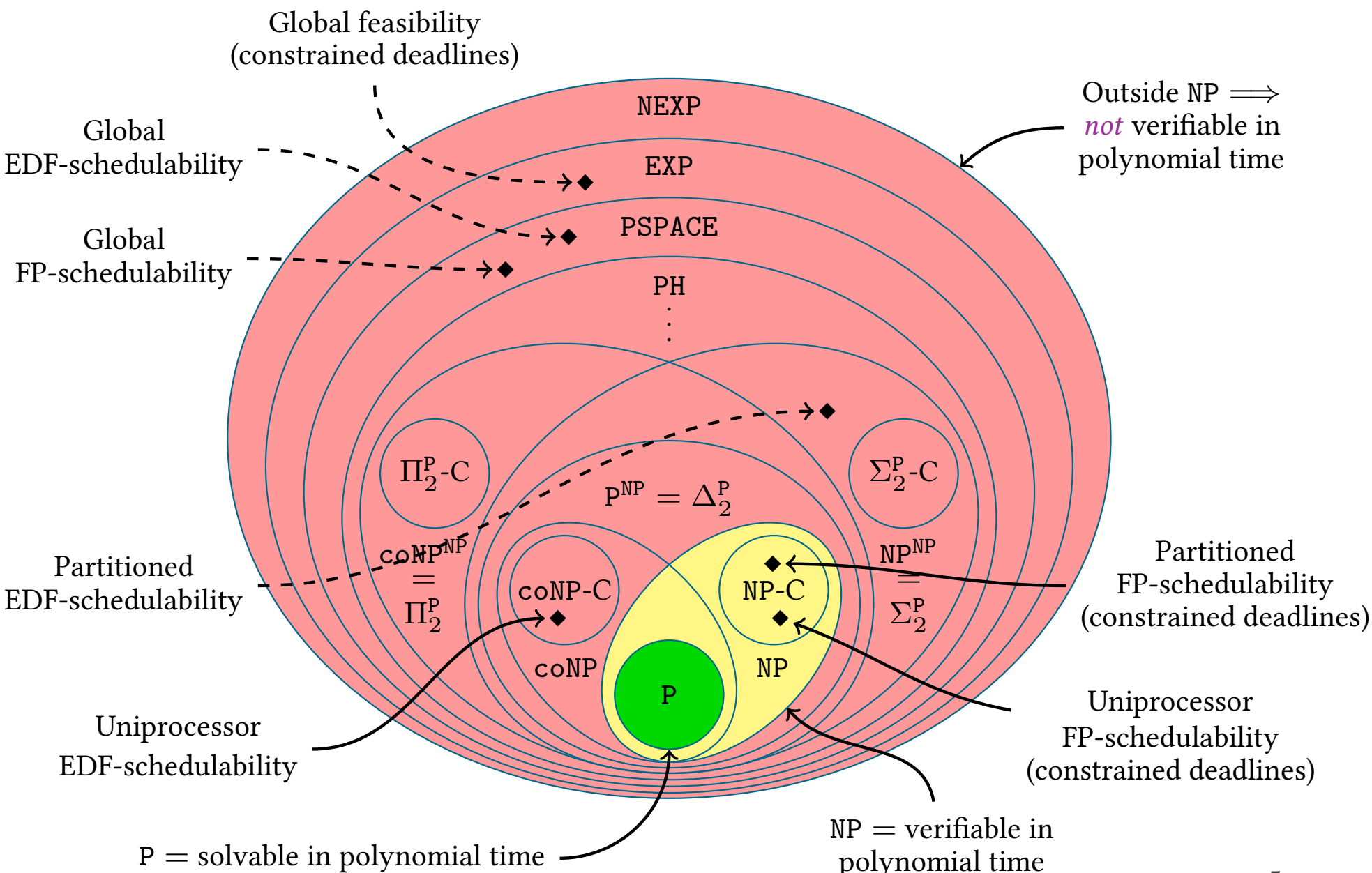
- - - - Upper bound



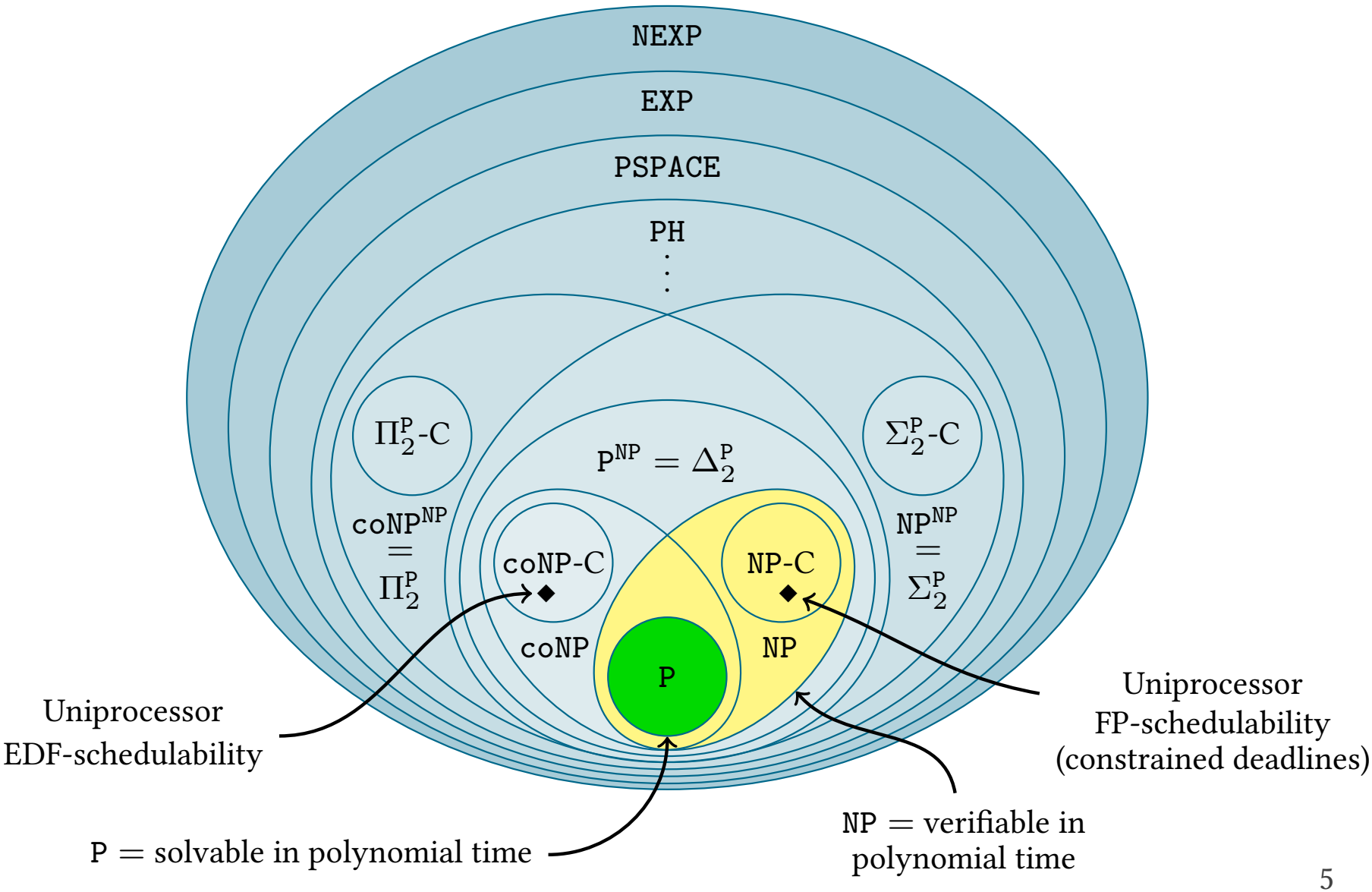
Does this relate to complexity?



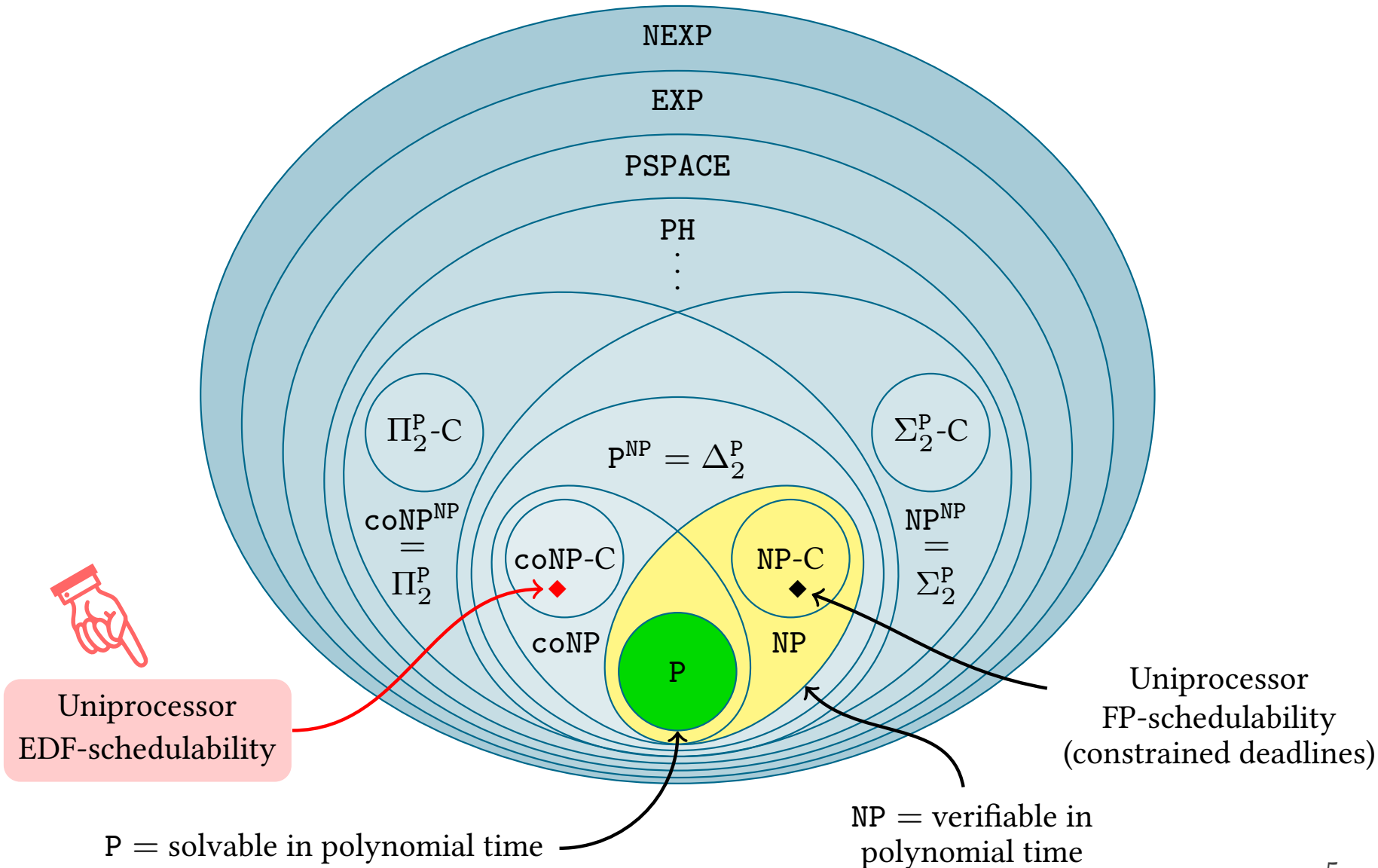
Does this relate to complexity?



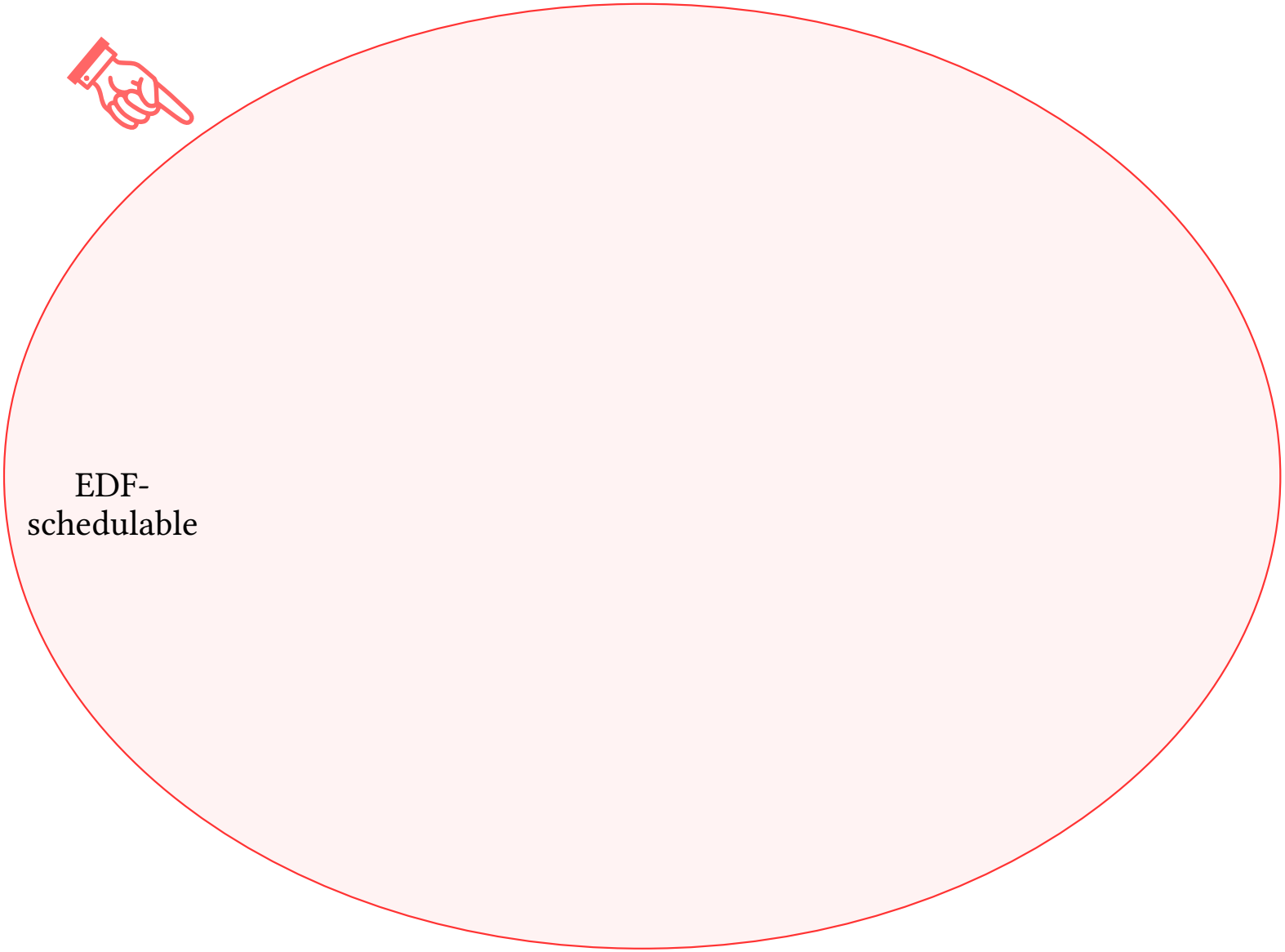
Does this relate to complexity?



Does this relate to complexity?

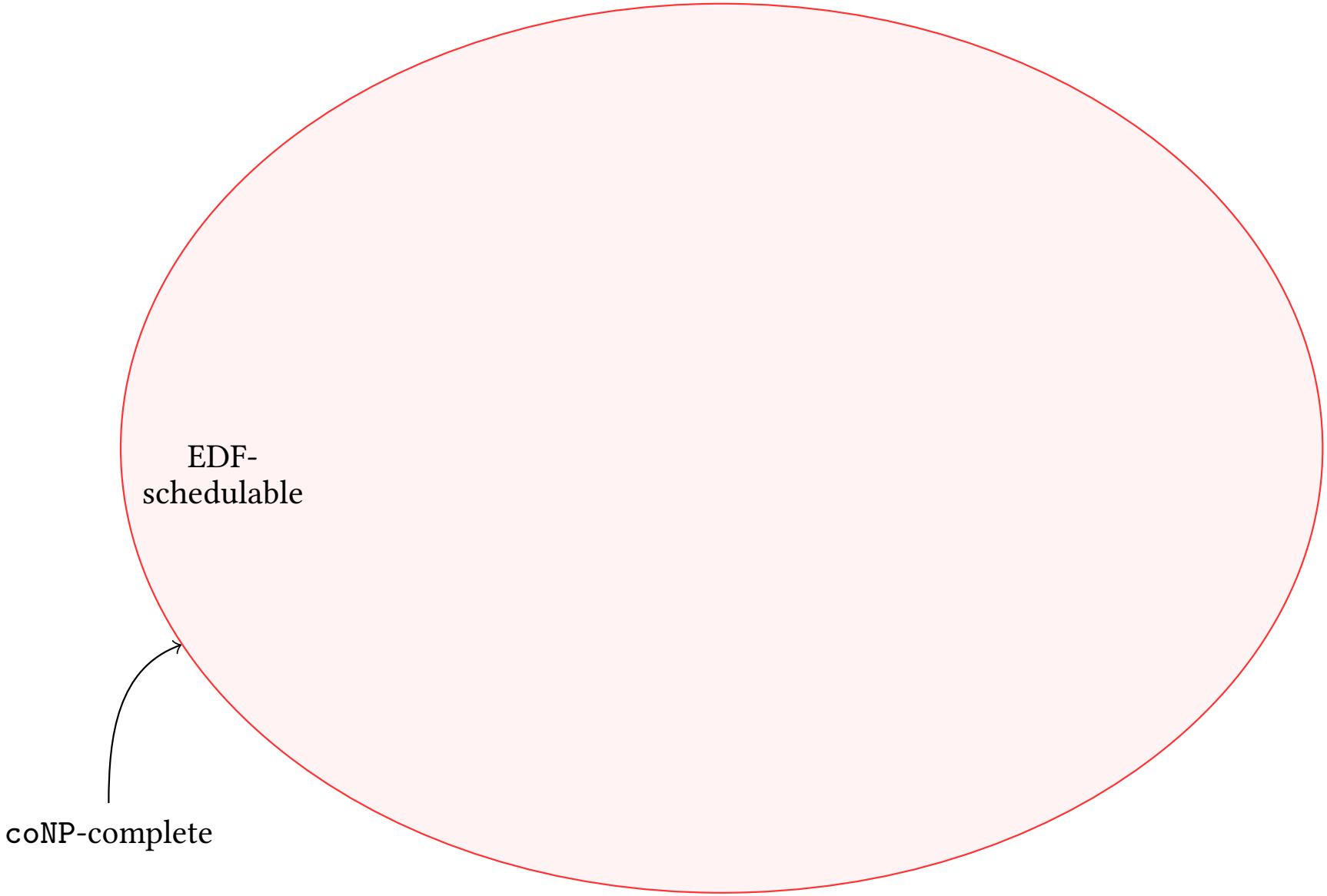


Dealing with EDF

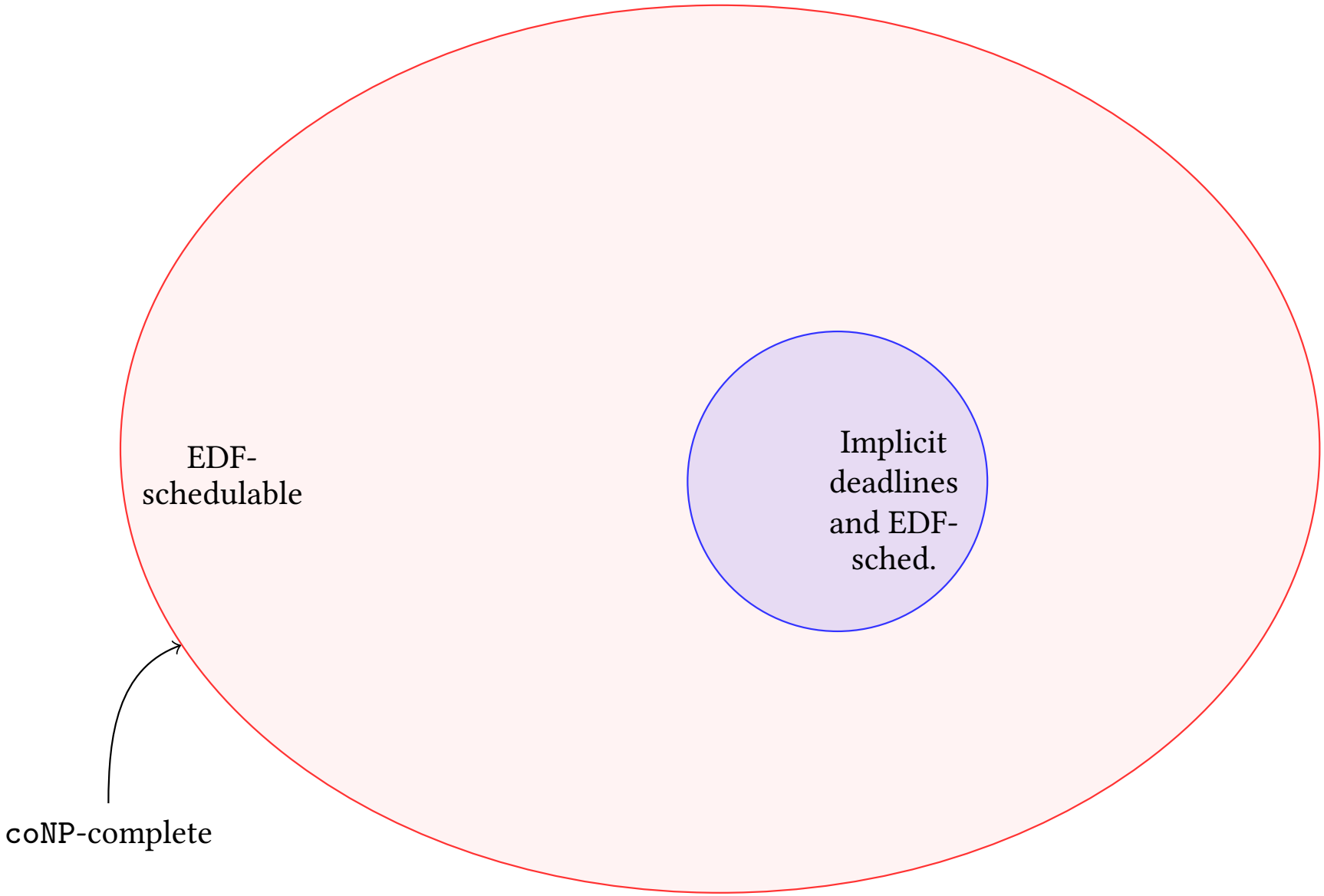


(Sporadic tasks, preemptive uniprocessor)

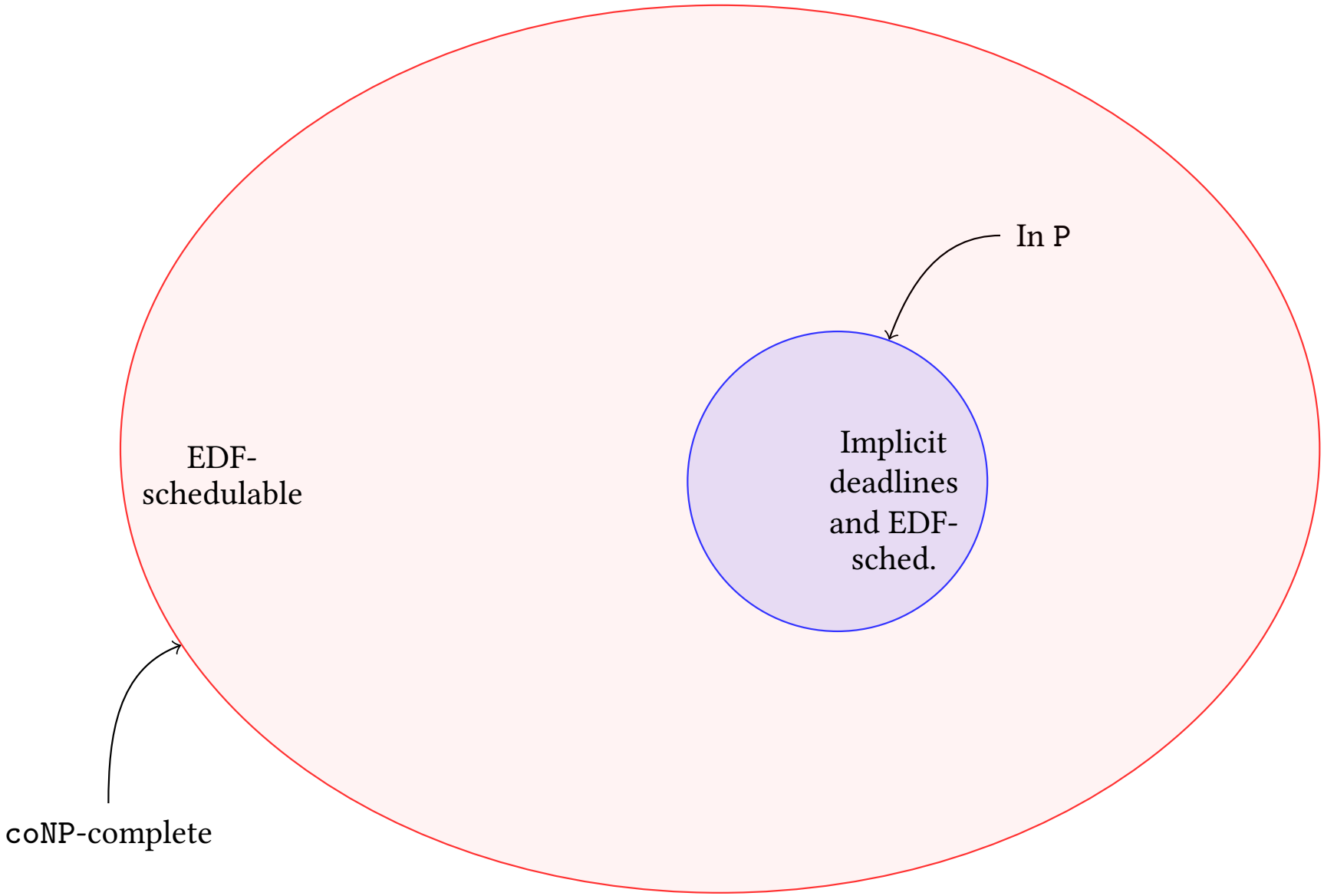
Dealing with EDF



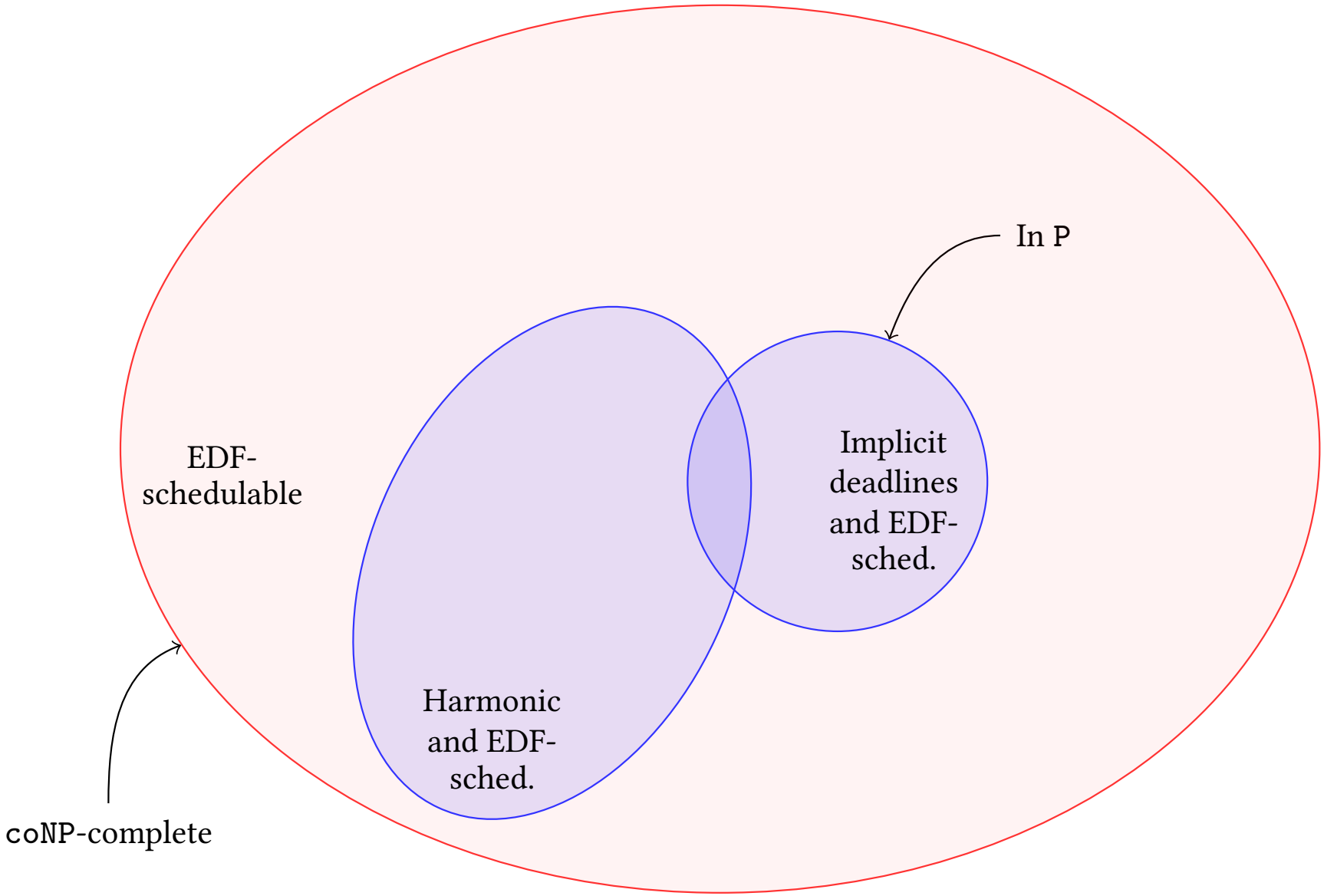
Dealing with EDF



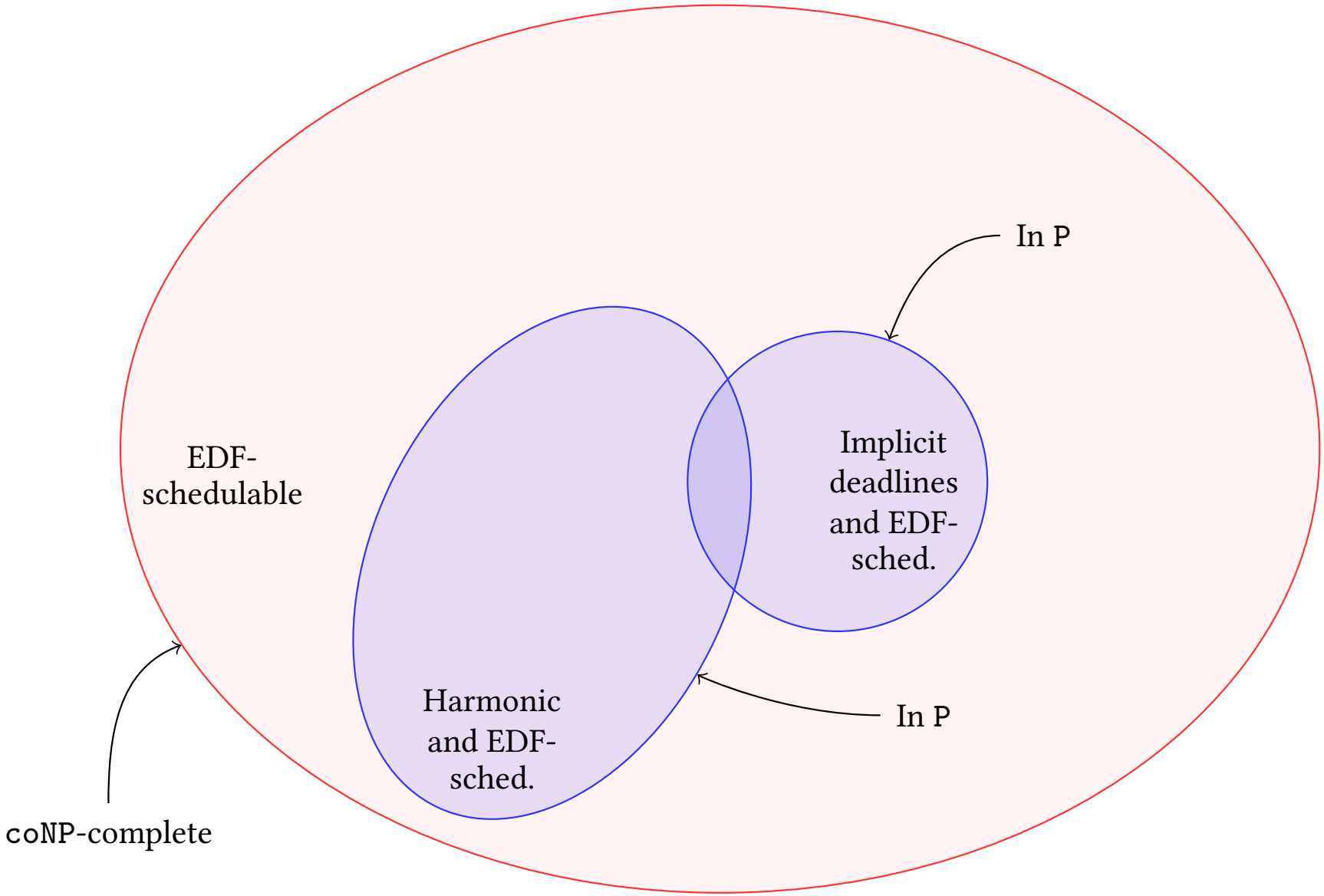
Dealing with EDF



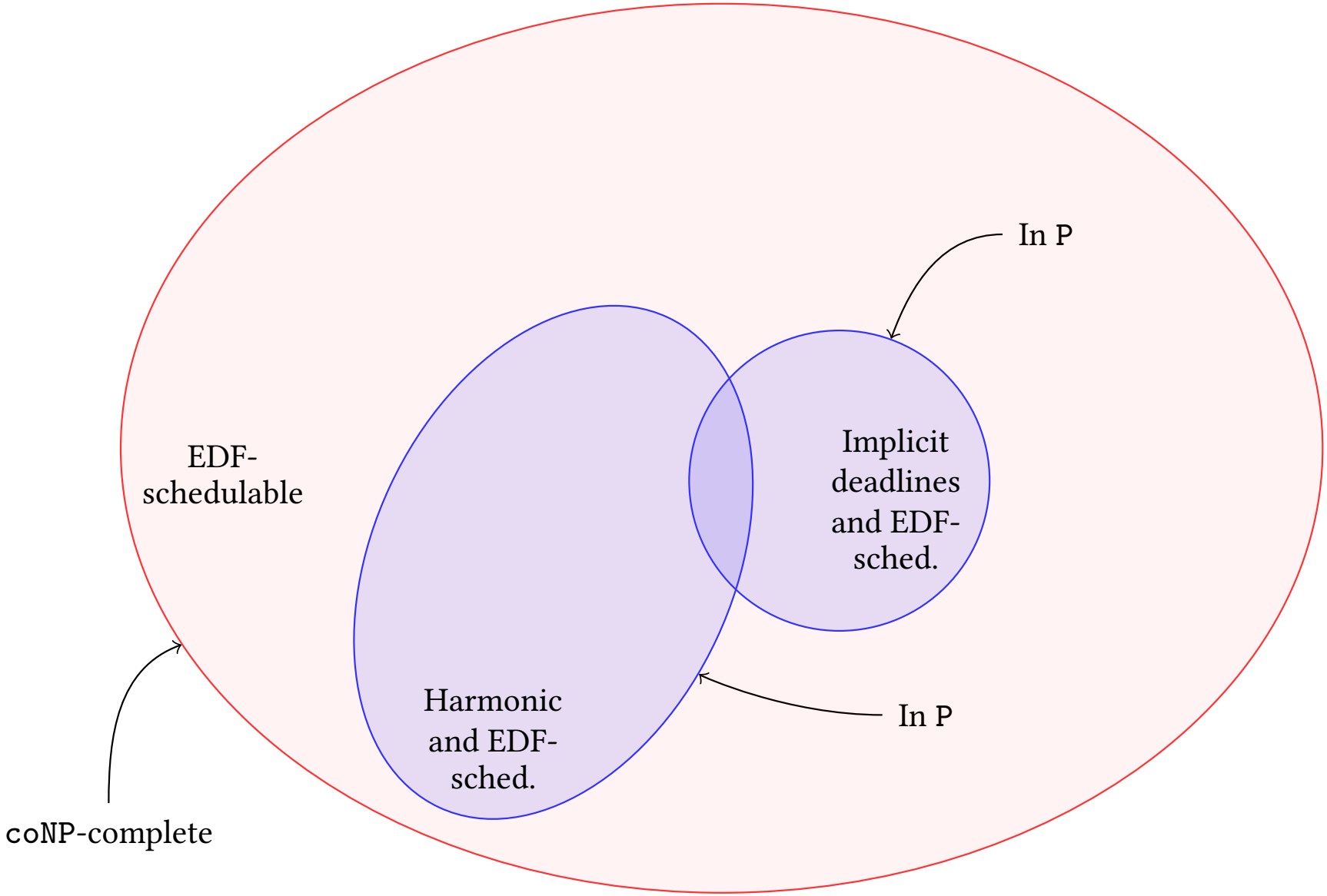
Dealing with EDF



Dealing with EDF

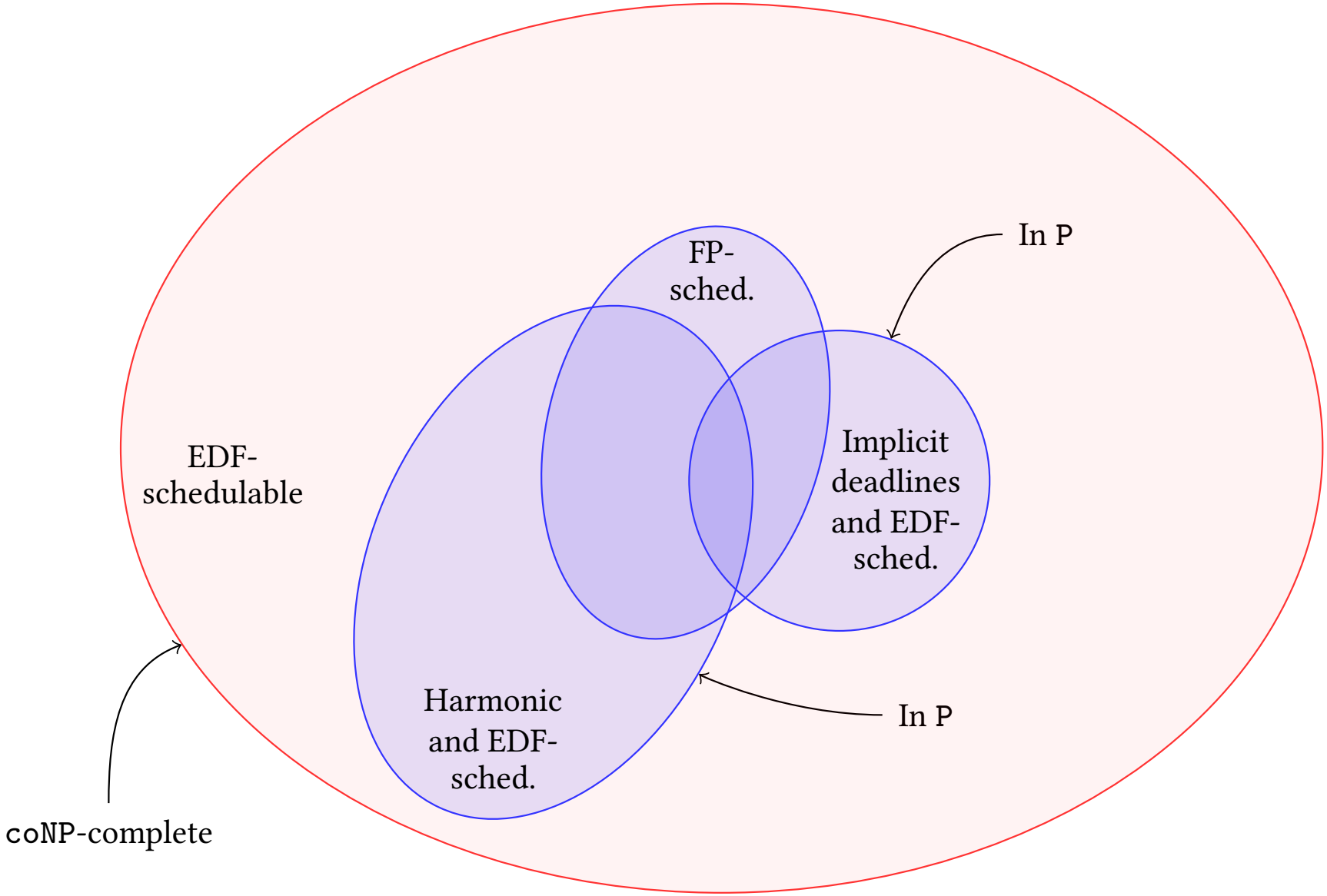


Dealing with EDF



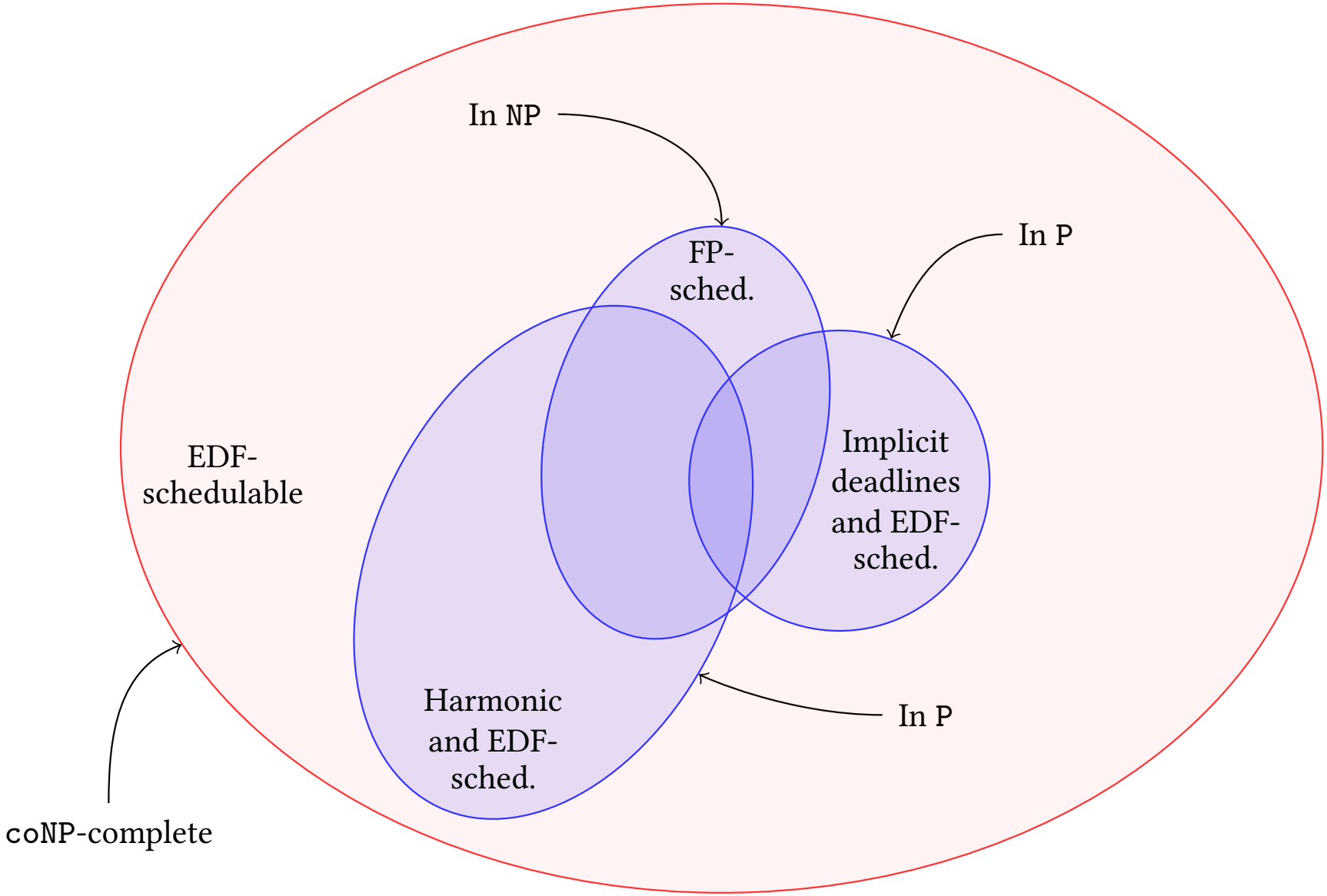
FP-schedulability $\Longrightarrow$ EDF-schedulability

Dealing with EDF



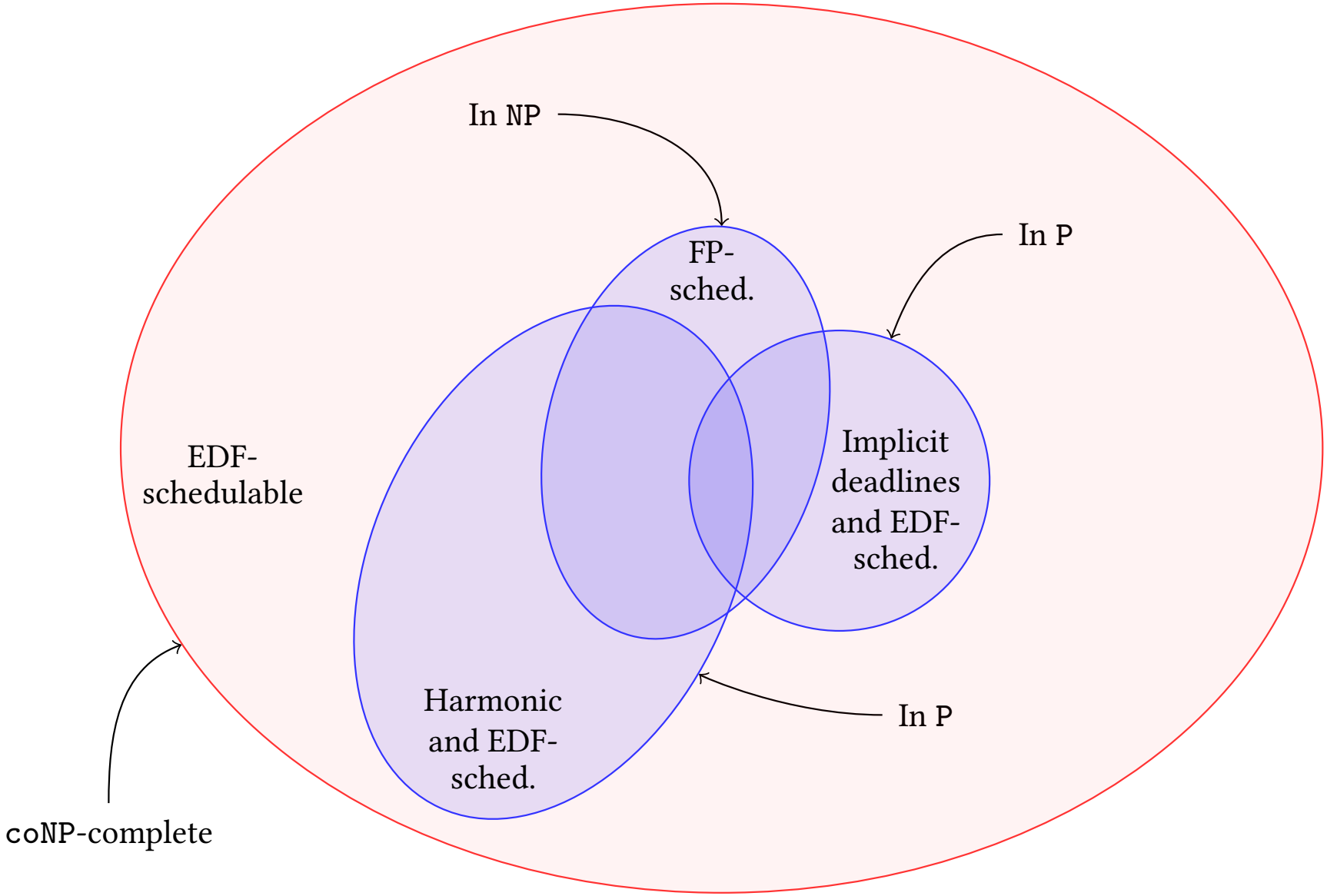
FP-schedulability $\Longrightarrow$ EDF-schedulability

Dealing with EDF



FP-schedulability $\Longrightarrow$ EDF-schedulability

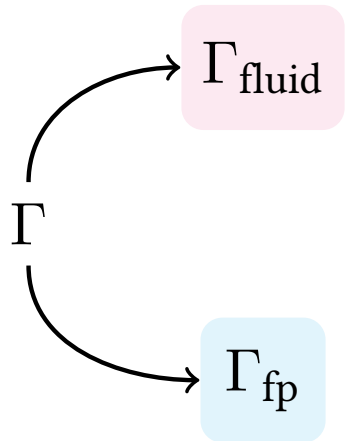
Dealing with EDF



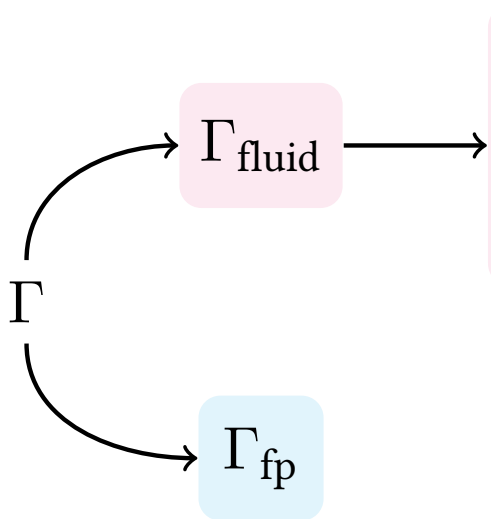
$$\forall \mathcal{A} : \quad \mathcal{A}\text{-schedulability} \Longrightarrow \text{EDF-schedulability}$$

Let's make up a scheduler: FP+fluid

Let's make up a scheduler: FP+fluid



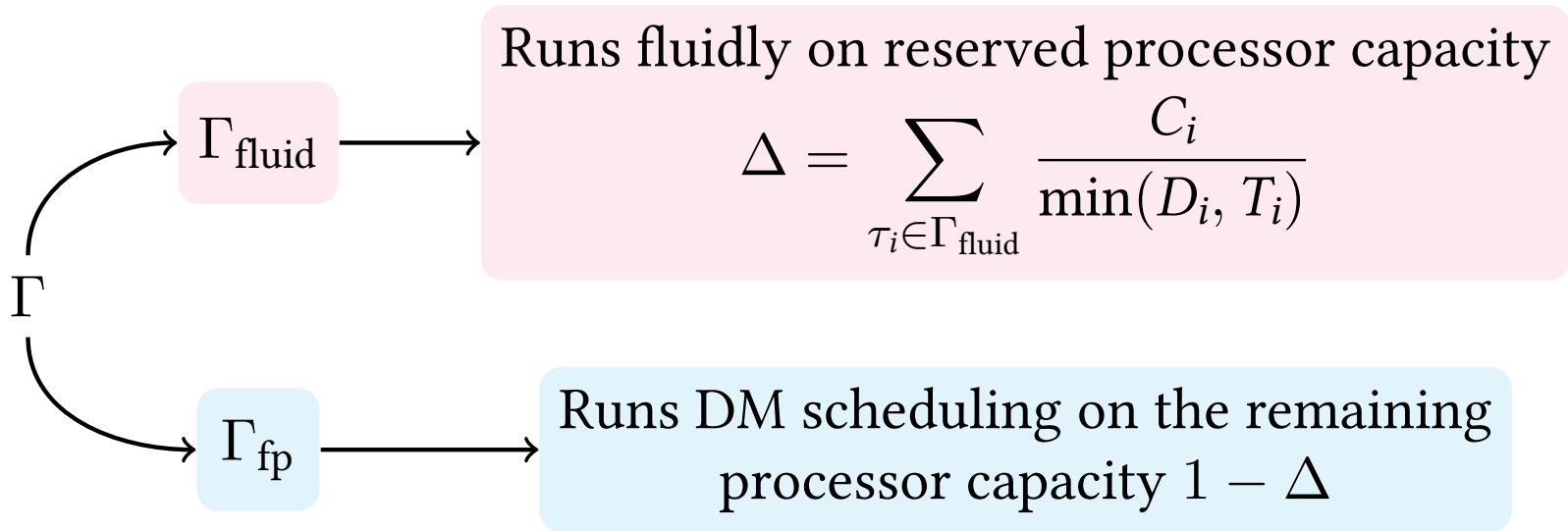
Let's make up a scheduler: FP+fluid



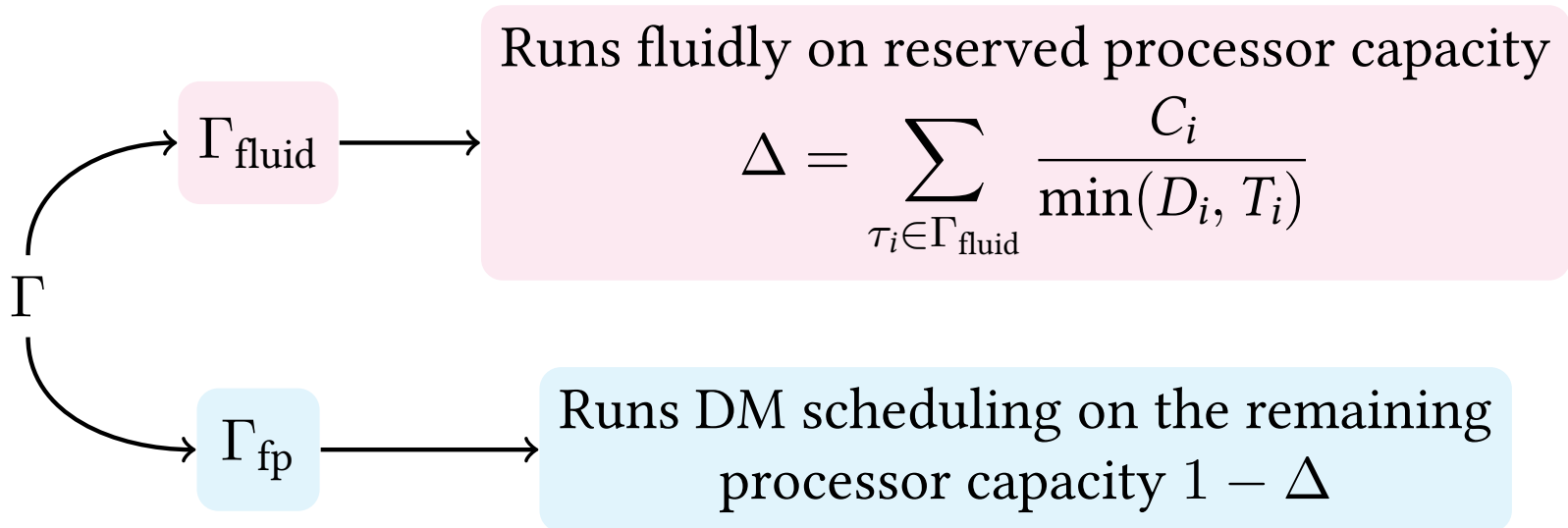
Runs fluidly on reserved processor capacity

$$\Delta = \sum_{\tau_i \in \Gamma_{\text{fluid}}} \frac{C_i}{\min(D_i, T_i)}$$

Let's make up a scheduler: FP+fluid

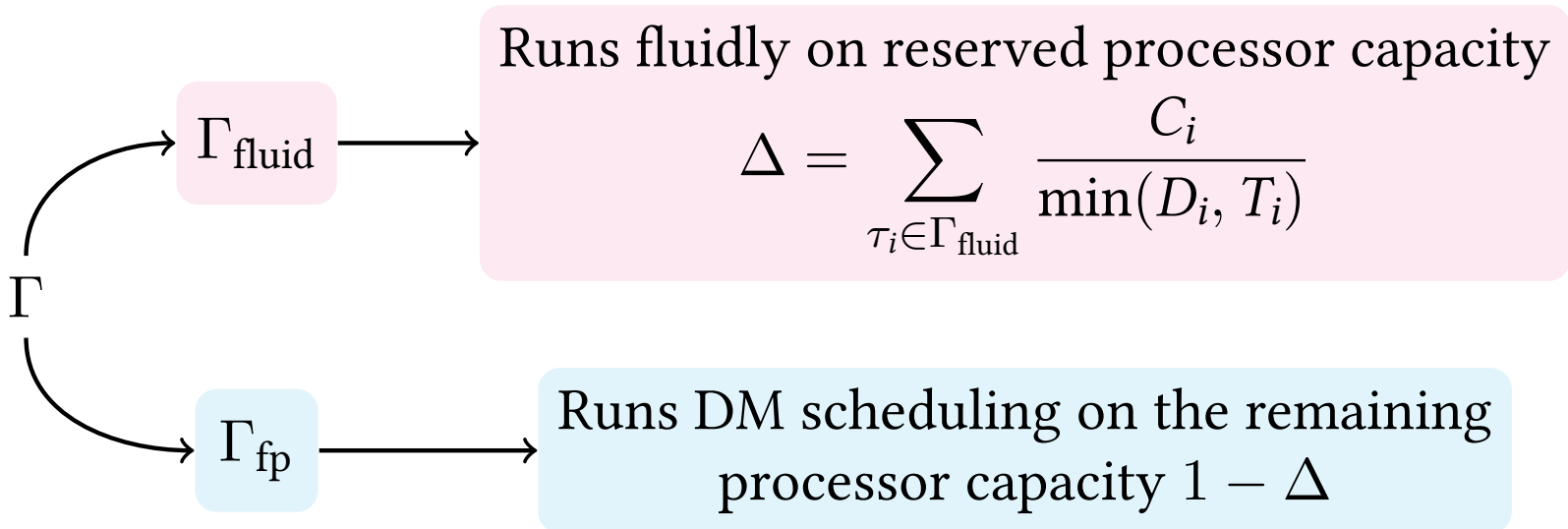


Let's make up a scheduler: FP+fluid



(1) The partitioning of Γ into Γ_{fluid} and Γ_{fp}
(2) RTA fixed-points for Γ_{fp} on a speed-$(1 - \Delta)$ processor

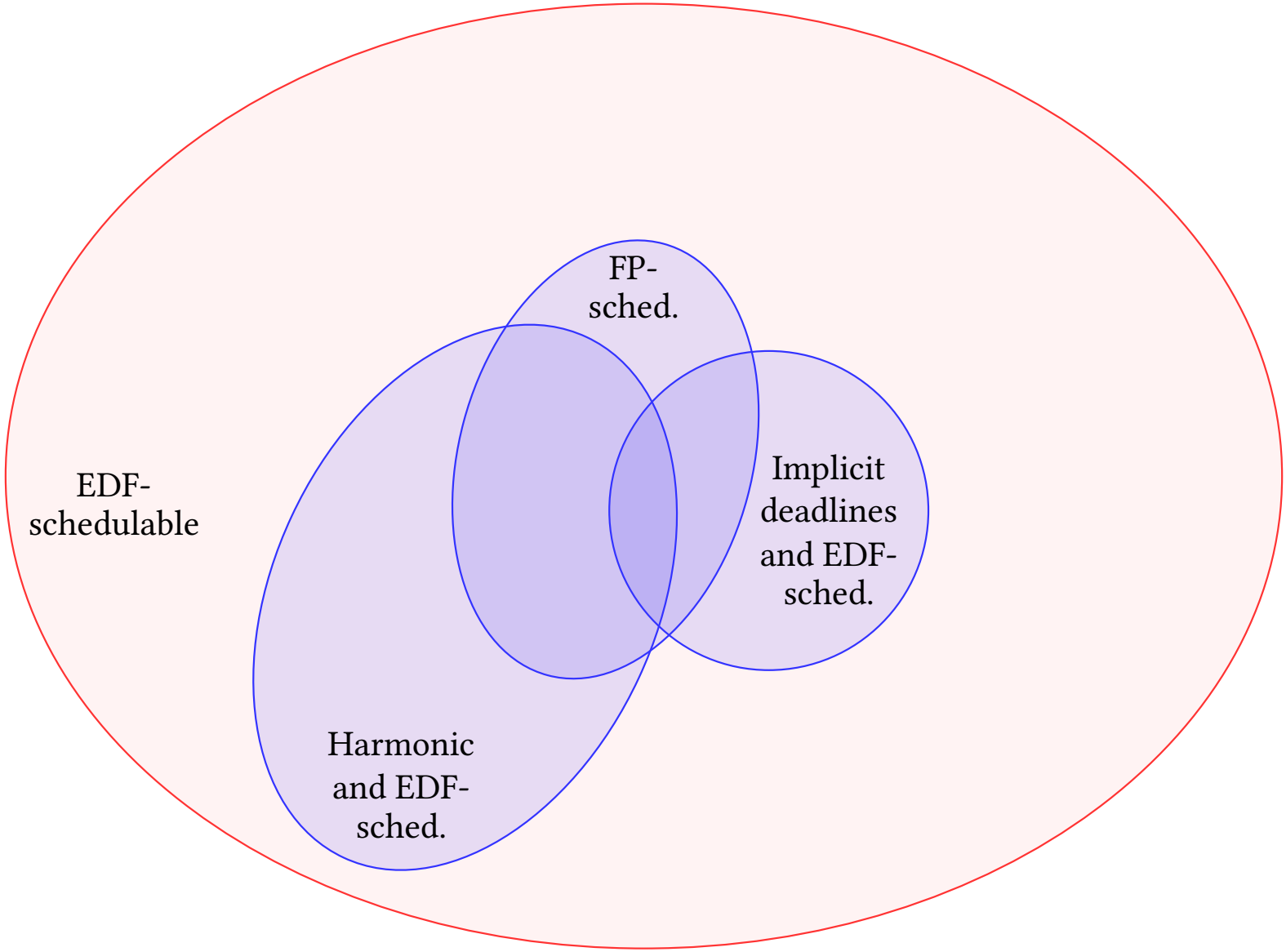
Let's make up a scheduler: FP+fluid



(1) The partitioning of Γ into Γ_{fluid} and Γ_{fp}
(2) RTA fixed-points for Γ_{fp} on a speed-$(1 - \Delta)$ processor

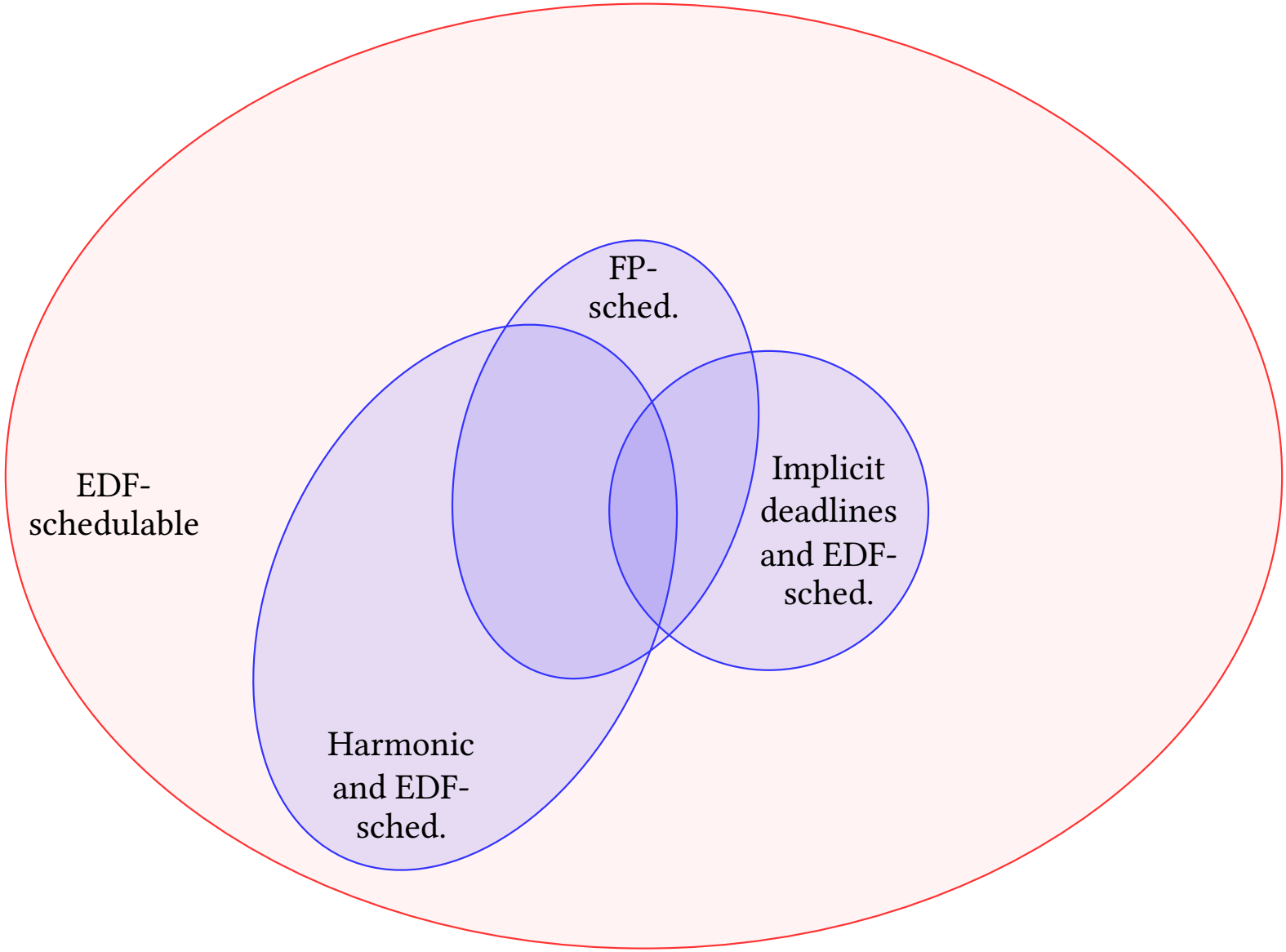
Verifiable in polynomial time $\iff$ FP+fluid-schedulability $\in$ NP

Dealing with EDF



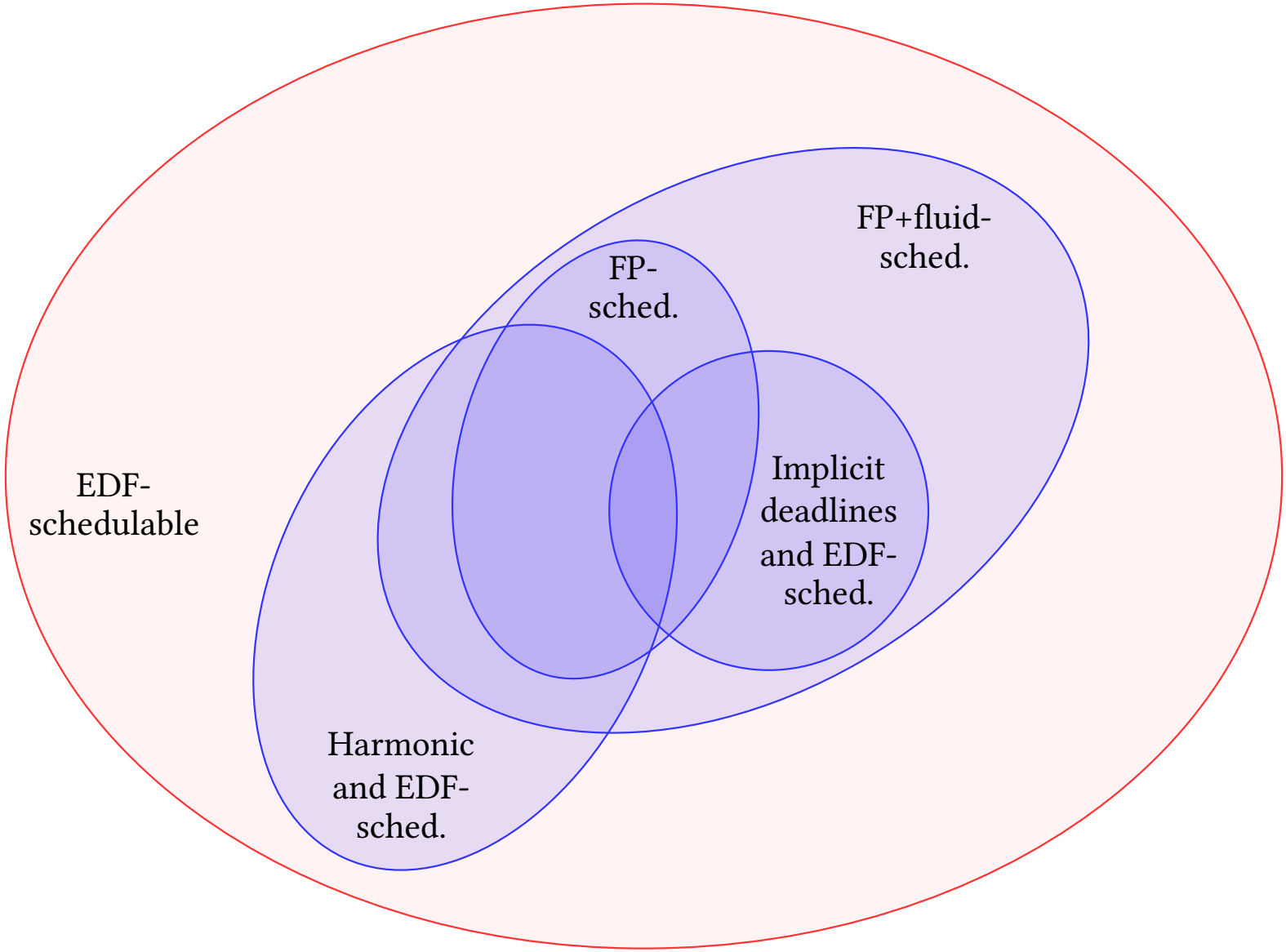
$$\forall \mathcal{A}: \quad \mathcal{A}\text{-schedulability} \Longrightarrow \text{EDF-schedulability}$$

Dealing with EDF



FP+fluid-schedulability $\Longrightarrow$ EDF-schedulability

Dealing with EDF



FP+fluid-schedulability $\Longrightarrow$ EDF-schedulability

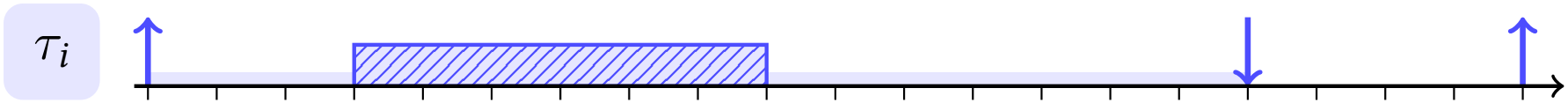
Another one: FP+split

Another one: FP+split

Split each task τ_i into $k_i \in \mathbb{N}_+$ pieces, then schedule with DM.

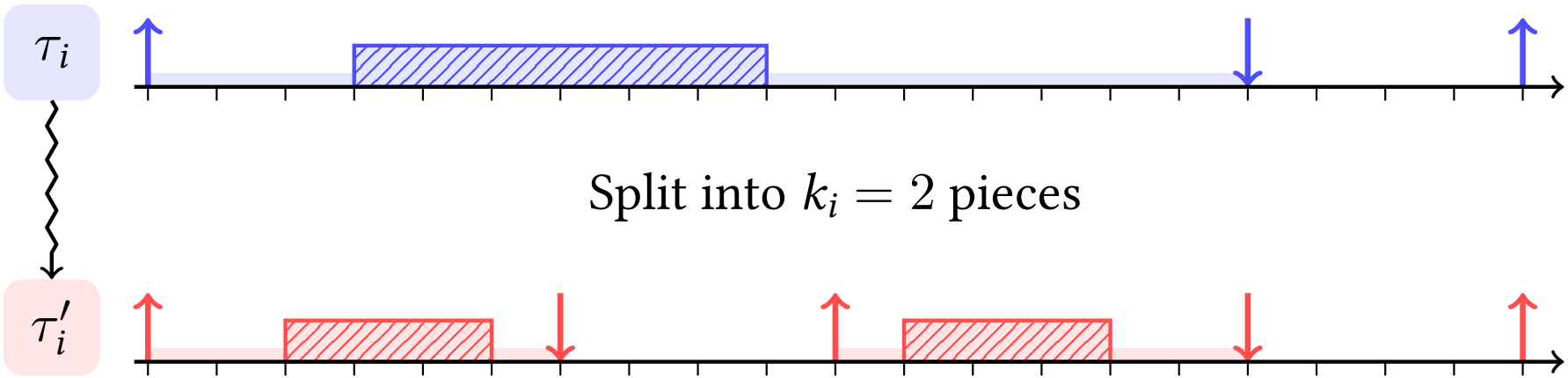
Another one: FP+split

Split each task τ_i into $k_i \in \mathbb{N}_+$ pieces, then schedule with DM.



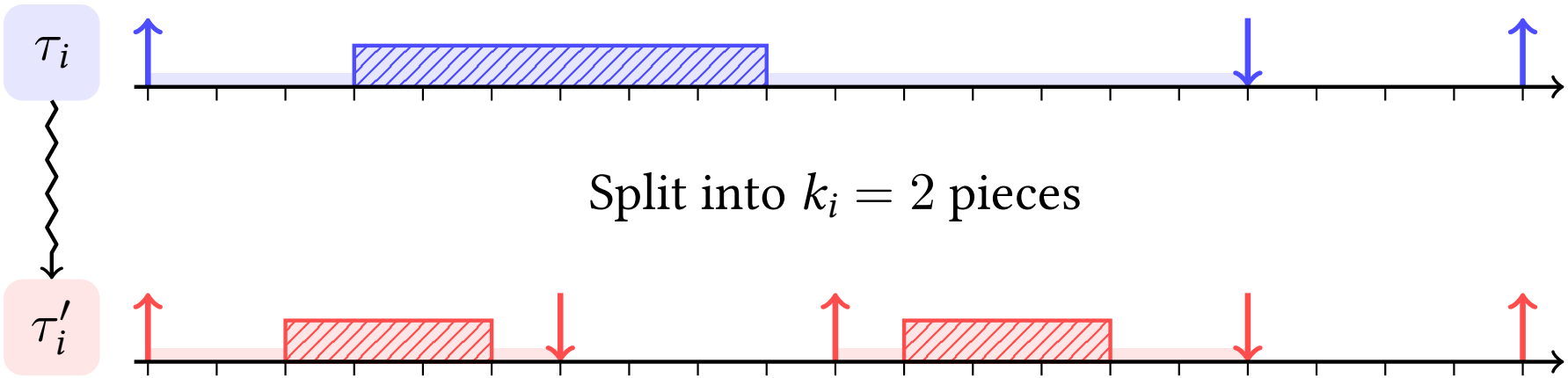
Another one: FP+split

Split each task τ_i into $k_i \in \mathbb{N}_+$ pieces, then schedule with DM.



Another one: FP+split

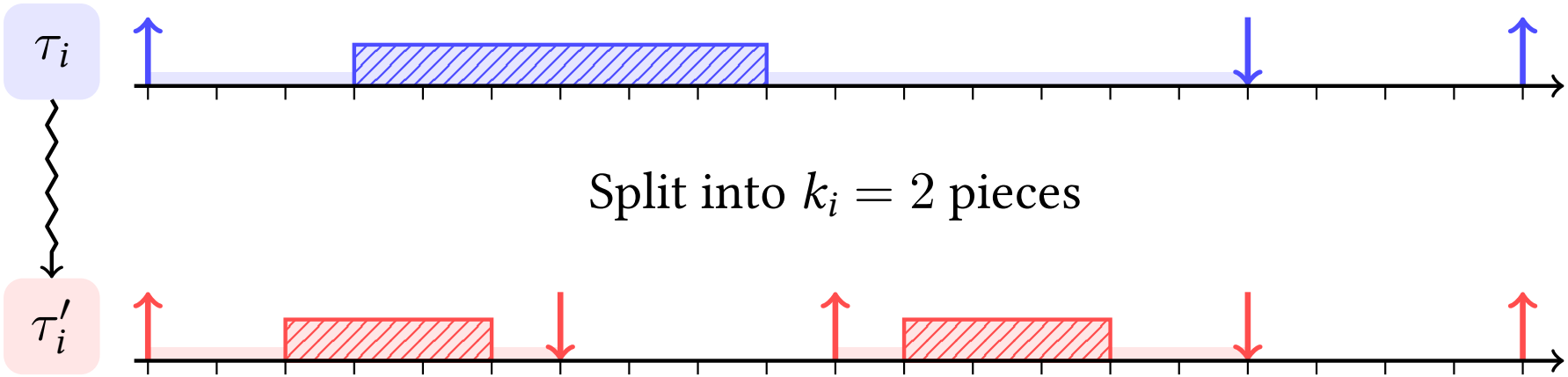
Split each task τ_i into $k_i \in \mathbb{N}_+$ pieces, then schedule with DM.



(1) The k_i for each $\tau_i \in \Gamma$
(2) RTA fixed-points for Γ after splitting each τ_i into k_i pieces

Another one: FP+split

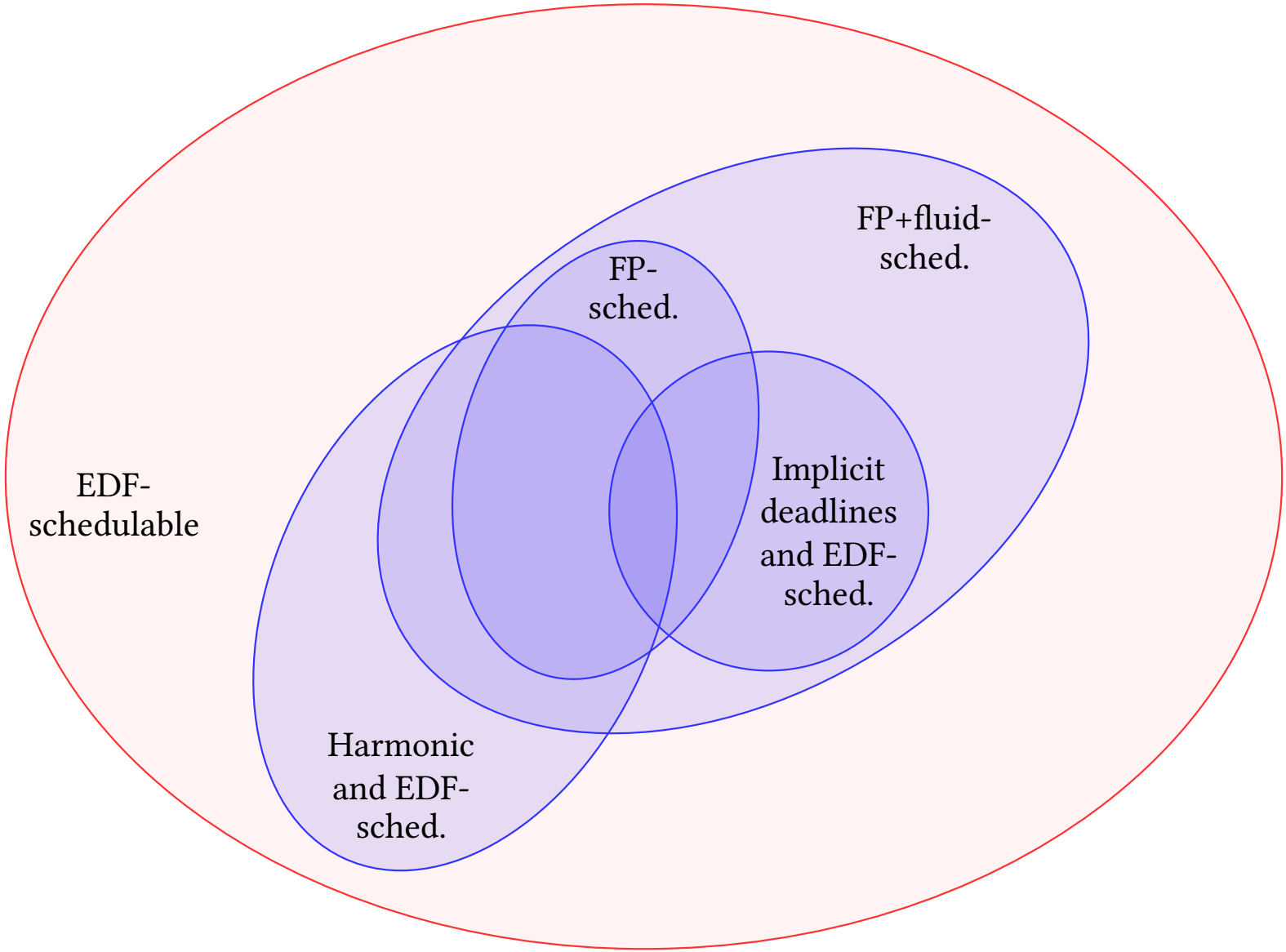
Split each task τ_i into $k_i \in \mathbb{N}_+$ pieces, then schedule with DM.



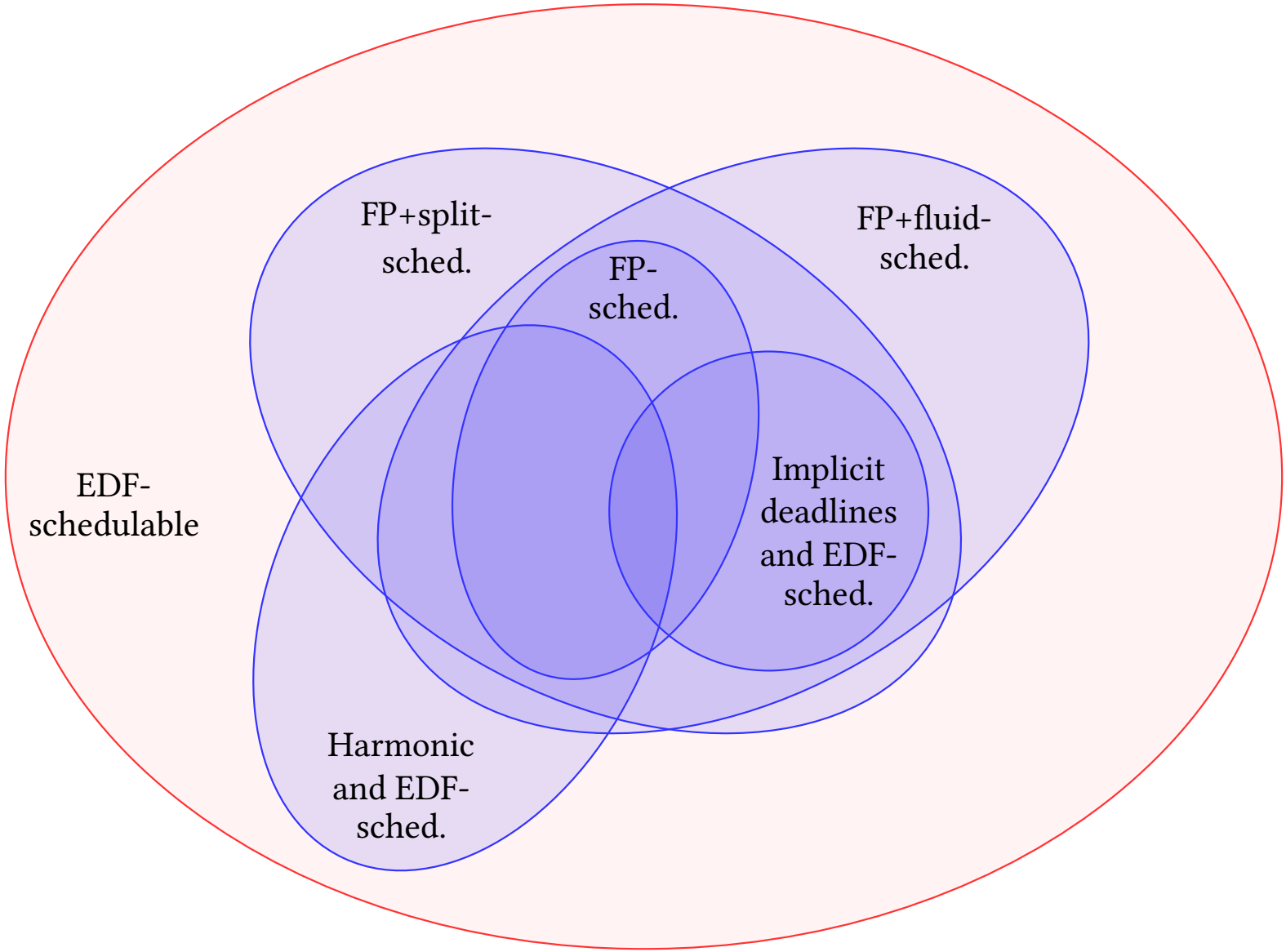
(1) The k_i for each $\tau_i \in \Gamma$
(2) RTA fixed-points for Γ after splitting each τ_i into k_i pieces

Verifiable in polynomial time $\iff$ FP+split-schedulability $\in$ NP

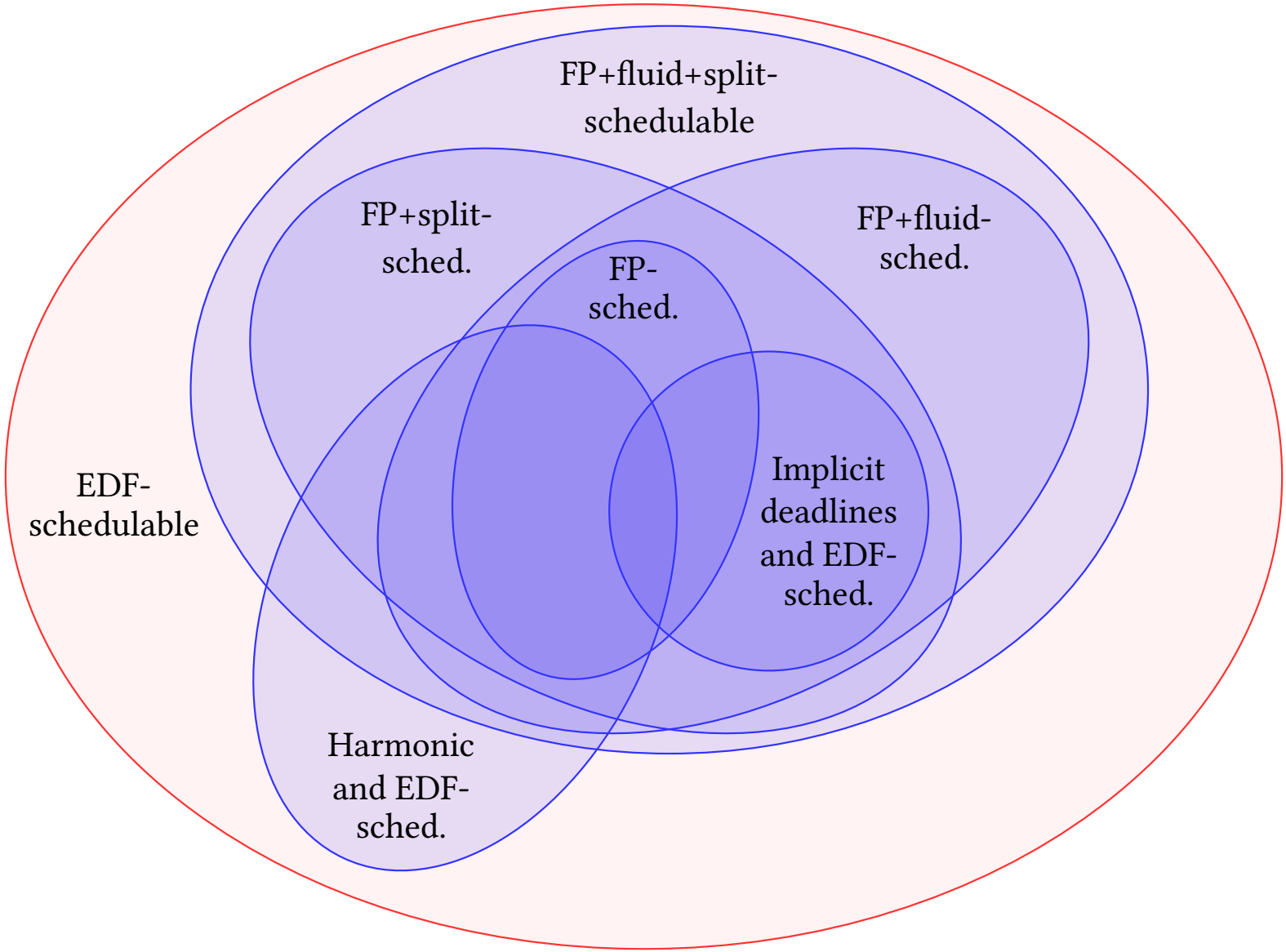
Dealing with EDF



Dealing with EDF



Dealing with EDF



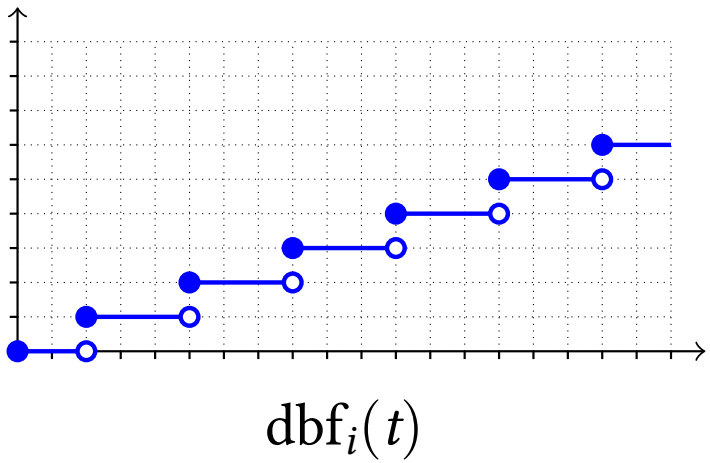
Approximation algorithms

Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$

Approximation algorithms

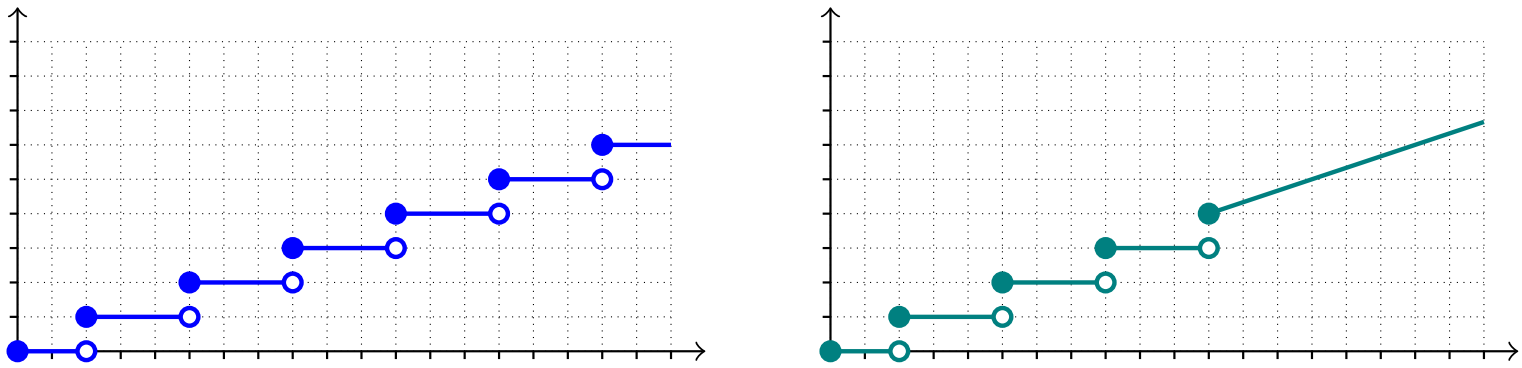
Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$



$\mathrm{dbf}_i(t)$

Approximation algorithms

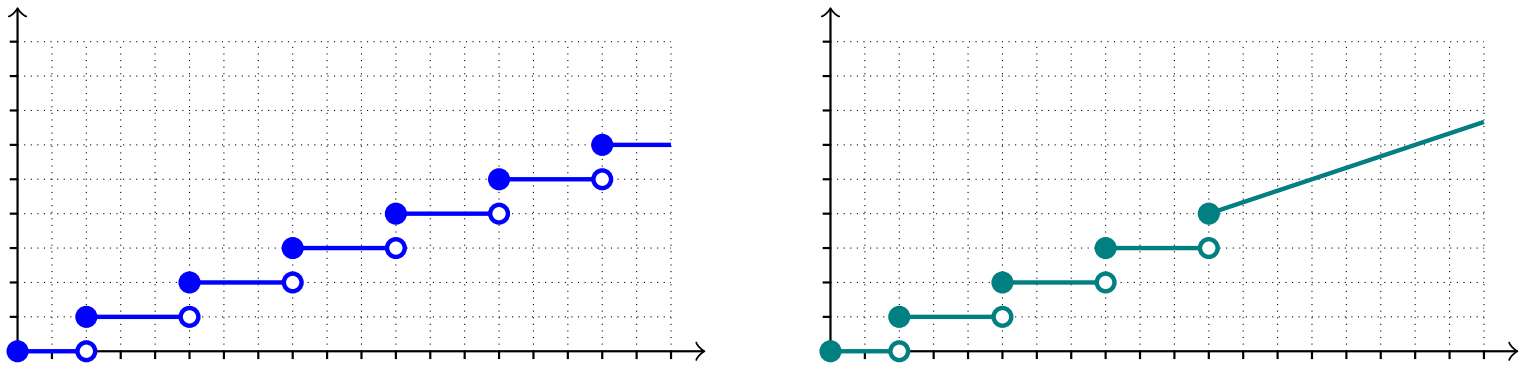
Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$



$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$

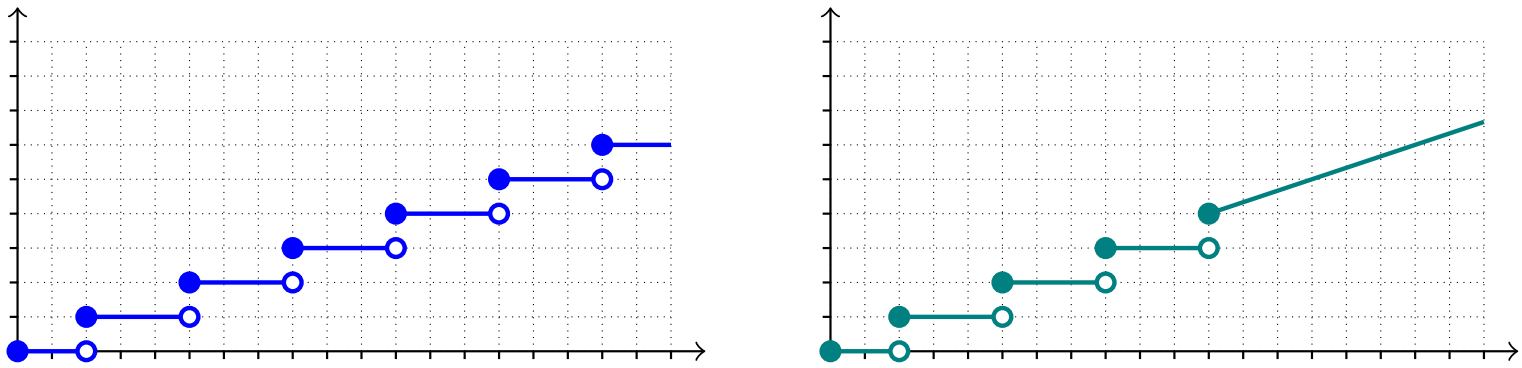


$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t.$

Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$



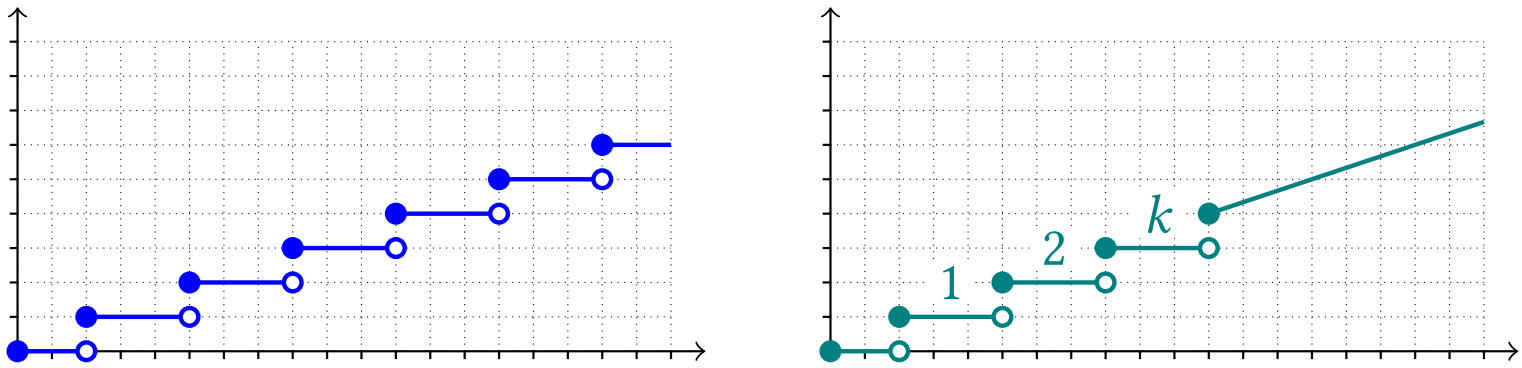
$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t.$

(by Albers and Slomka, ECRTS 2004)

Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$



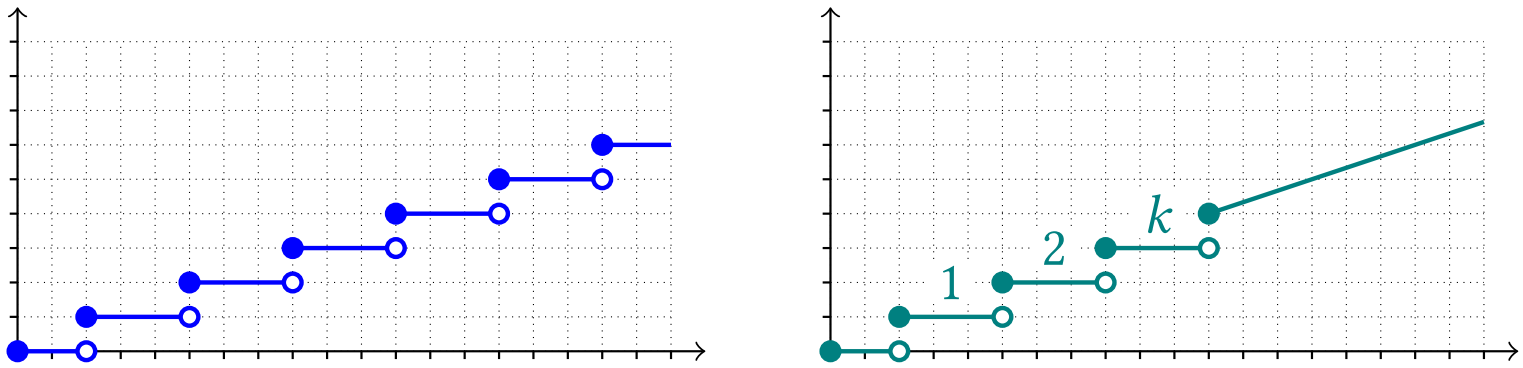
$\mathrm{dbf}_i(t) \qquad \leqslant \qquad \overline{\mathrm{dbf}}_i(t)$

Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t.$

(by Albers and Slomka, ECRTS 2004)

Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t.$



$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

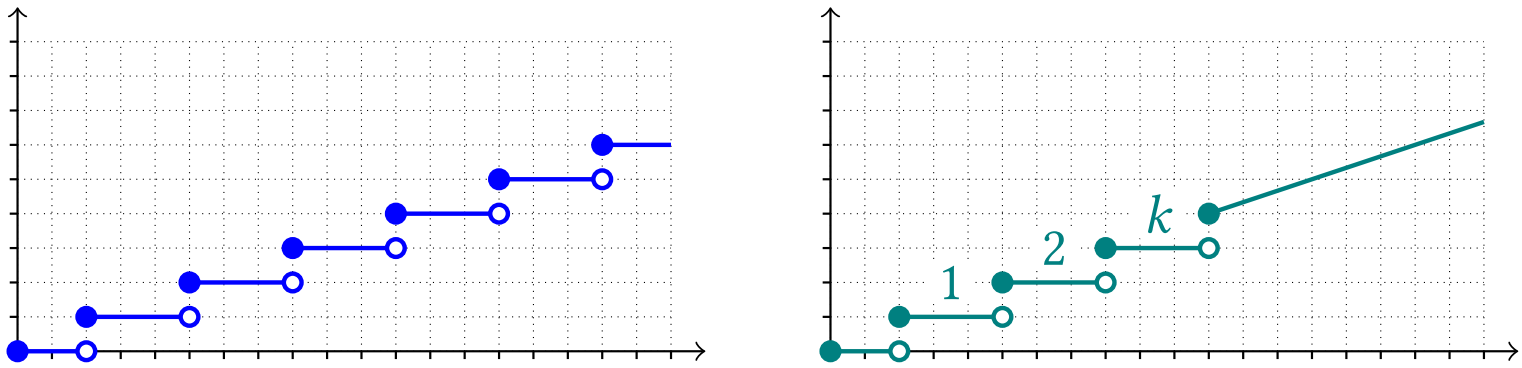
Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t.$

$\Gamma, k \longrightarrow$ A&S FPTAS

(by Albers and Slomka, ECRTS 2004)

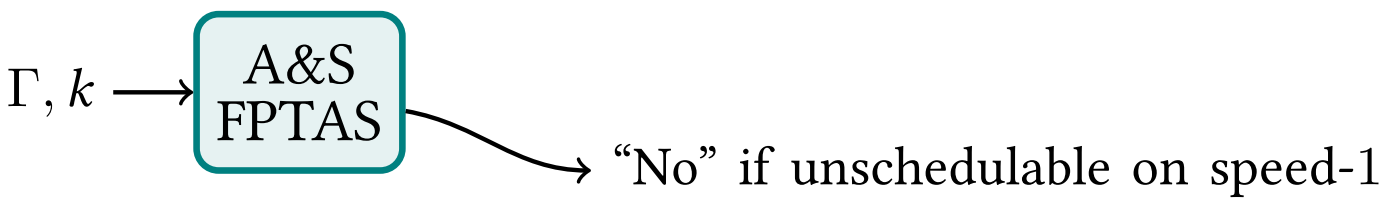
Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t$.



$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

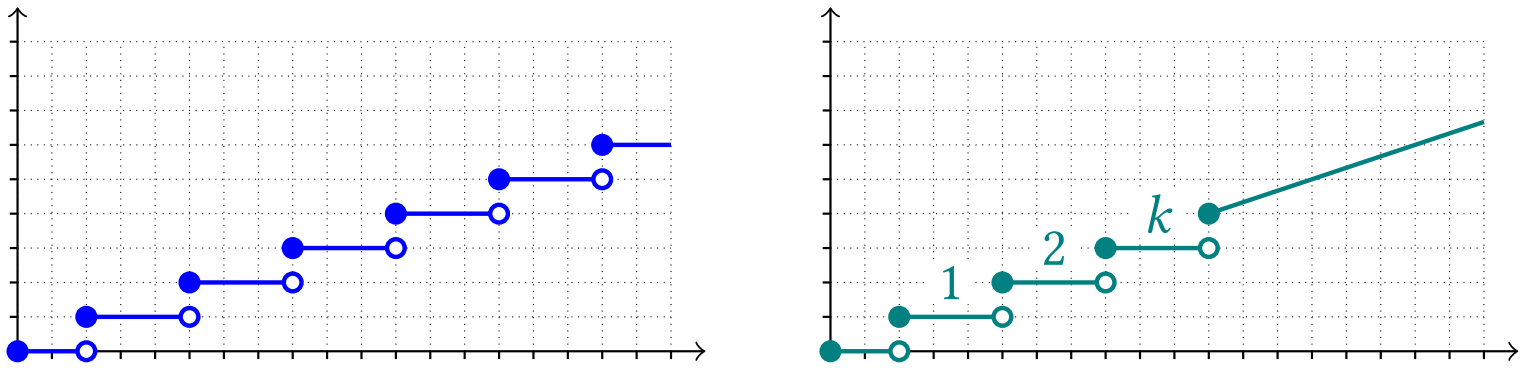
Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t$.



(by Albers and Slomka, ECRTS 2004)

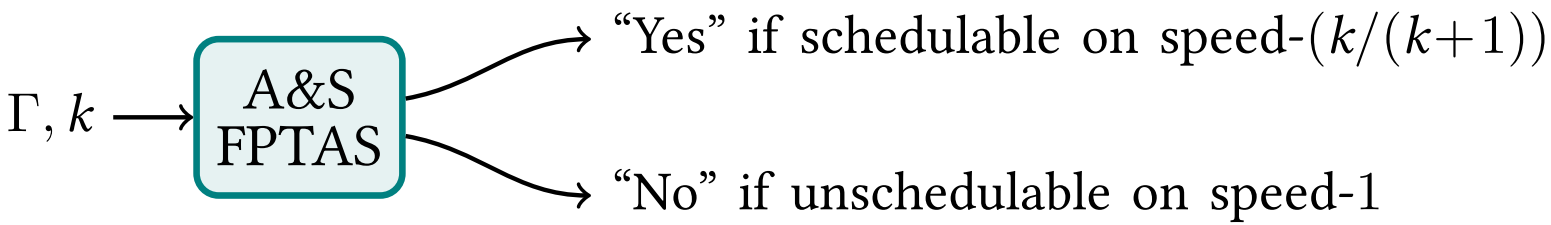
Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t$.



$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

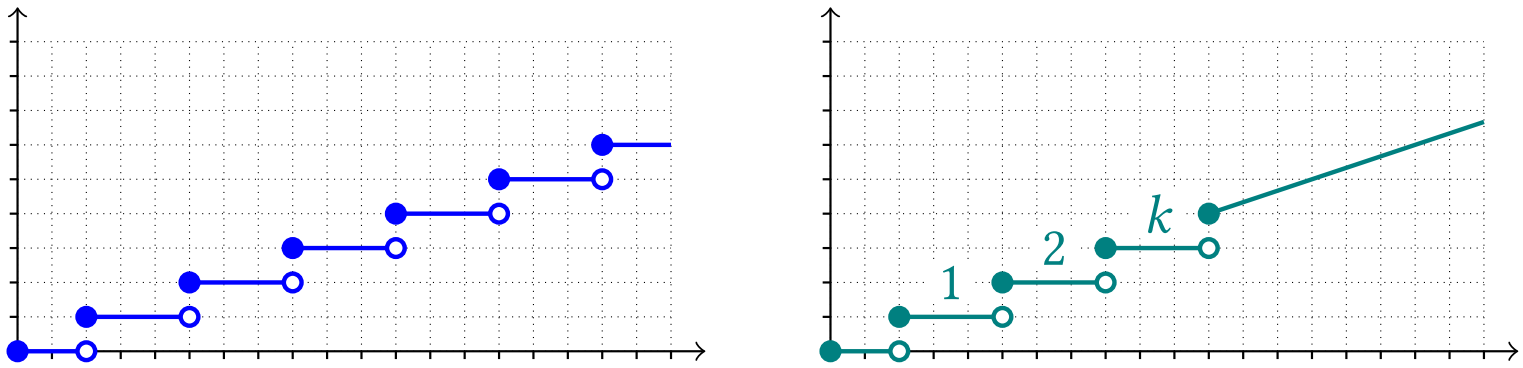
Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t$.



(by Albers and Slomka, ECRTS 2004)

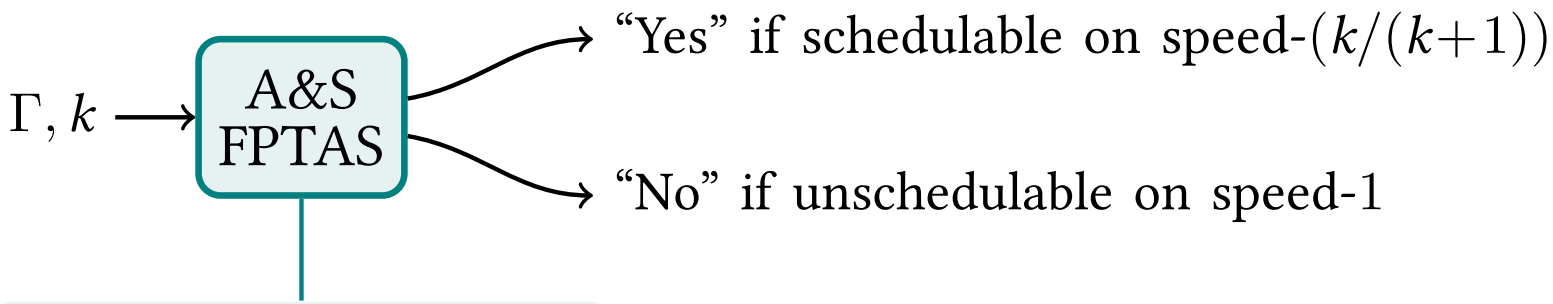
Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t$.



$\mathrm{dbf}_i(t) \quad \leqslant \quad \overline{\mathrm{dbf}}_i(t)$

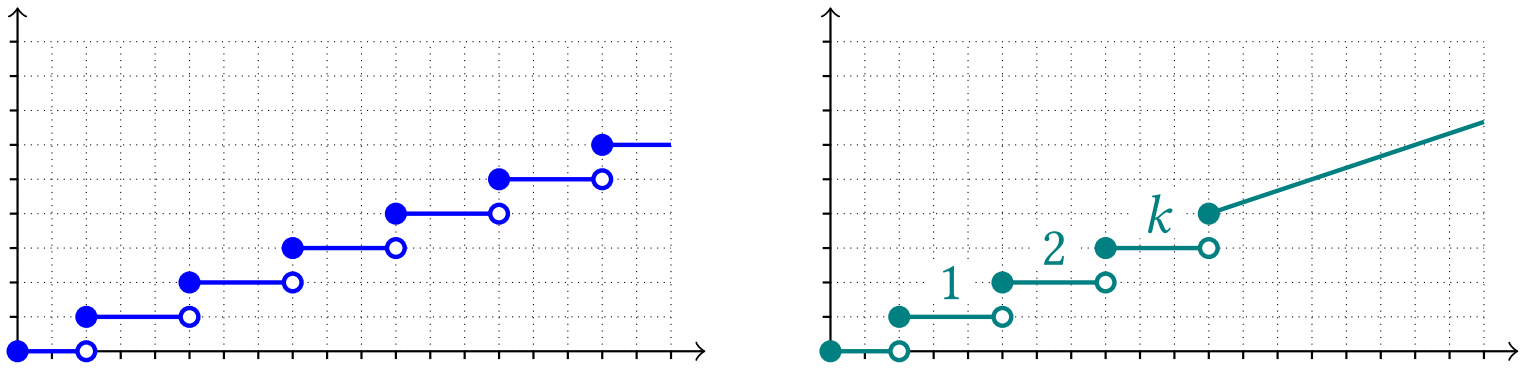
Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t$.



(by Albers and Slomka, ECRTS 2004)

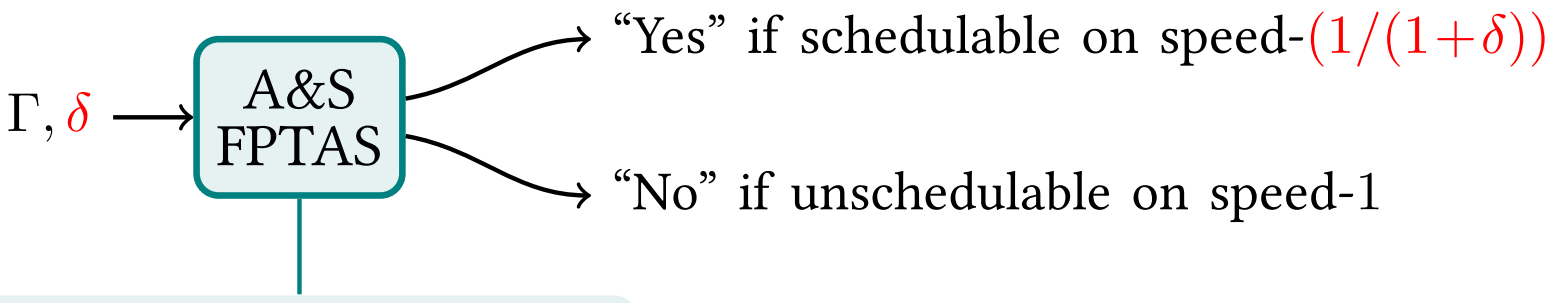
Approximation algorithms

Γ is EDF-schedulable iff $\forall t : \sum_{\tau_i \in \Gamma} \mathrm{dbf}_i(t) \leqslant t$.



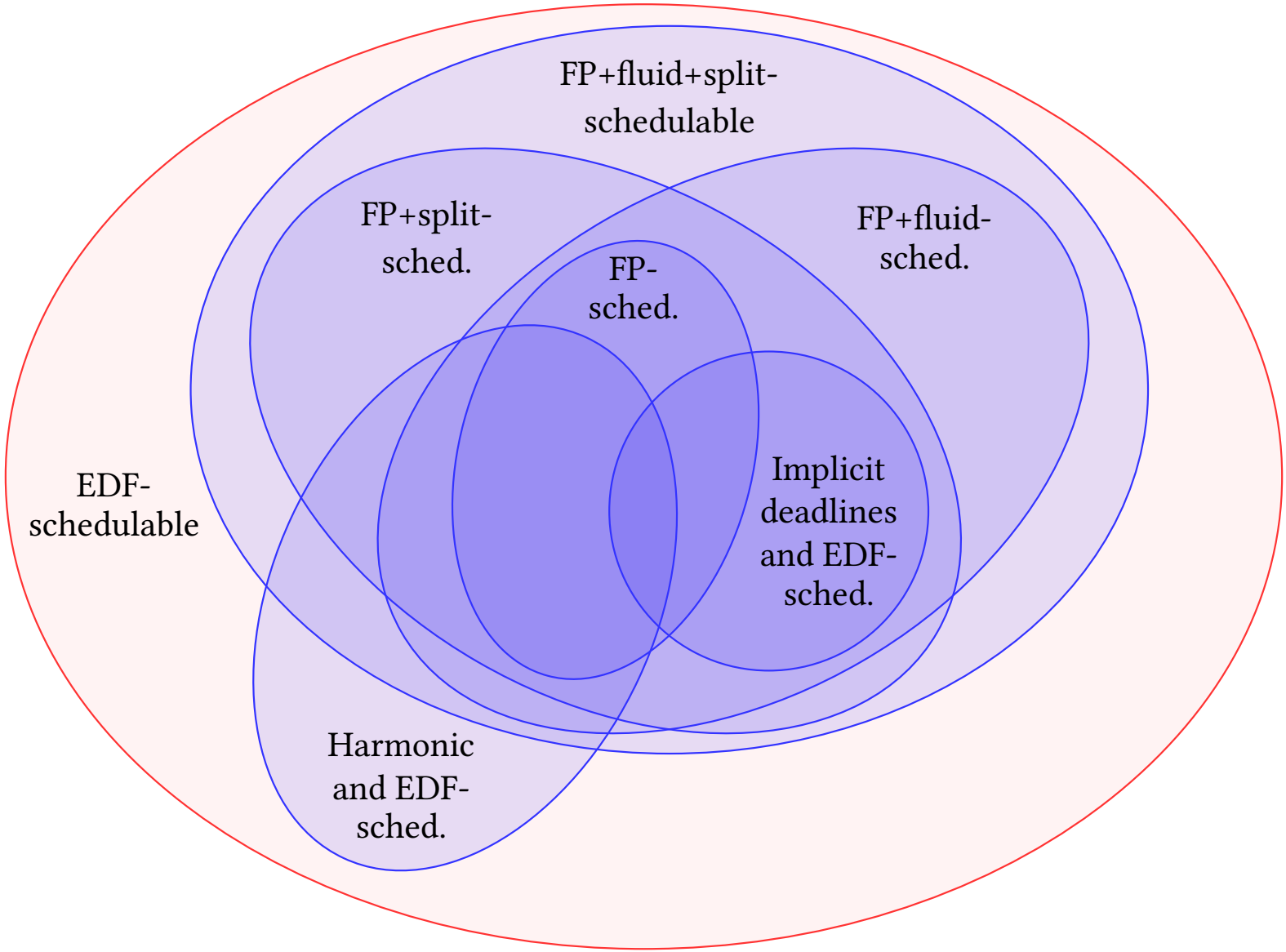
$\mathrm{dbf}_i(t) \leqslant \overline{\mathrm{dbf}}_i(t)$

Γ is EDF-schedulable **if** $\forall t : \sum_{\tau_i \in \Gamma} \overline{\mathrm{dbf}}_i(t) \leqslant t$.

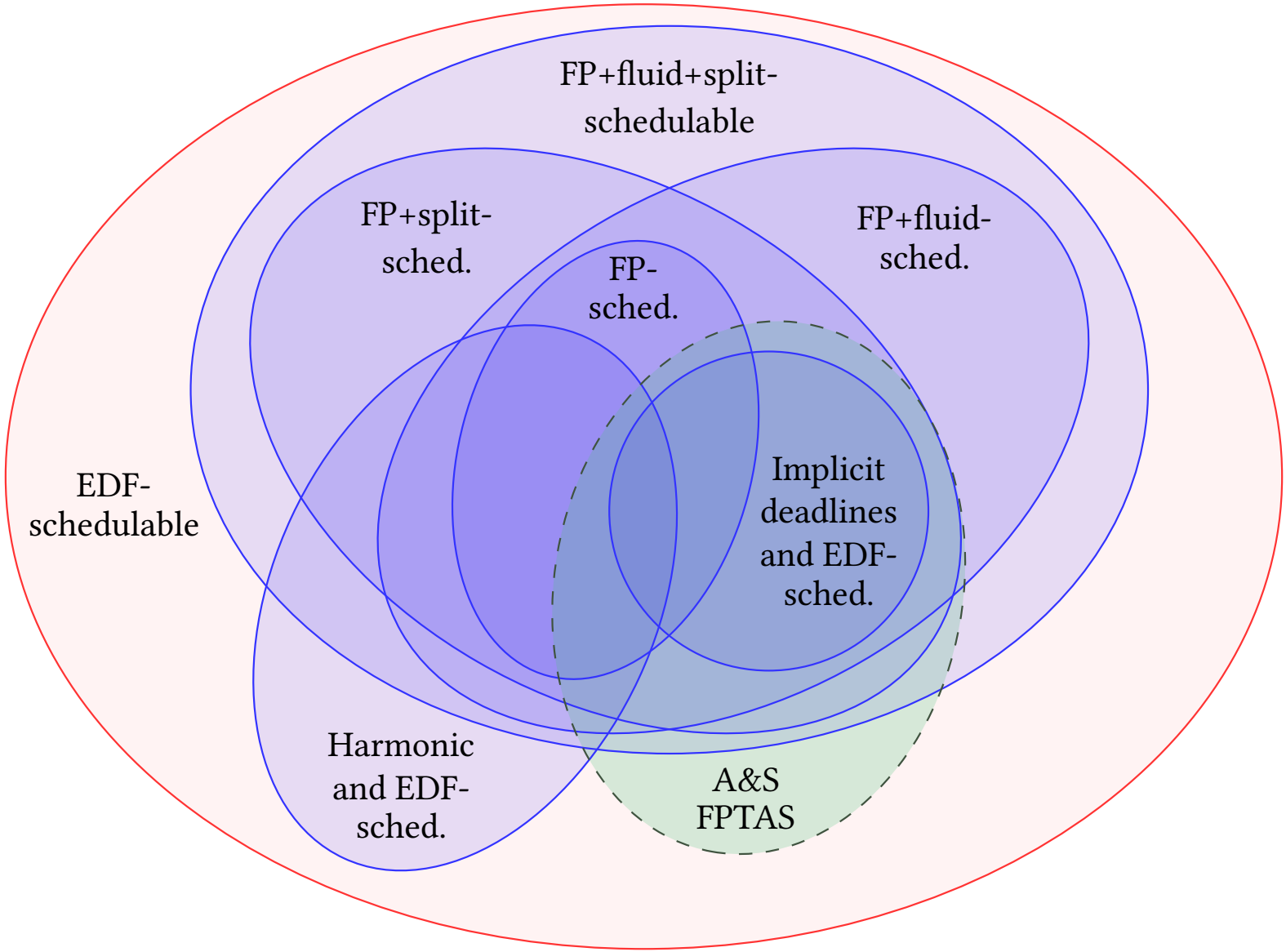


(by Albers and Slomka, ECRTS 2004)

Dealing with EDF

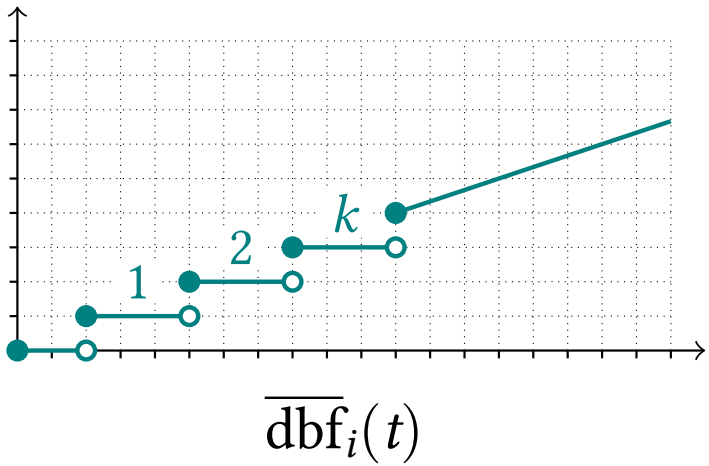


Dealing with EDF



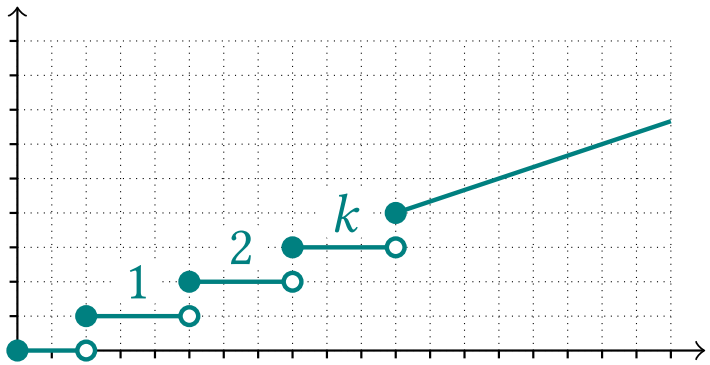
Approximation algorithms for explainability?

Approximation algorithms for explainability?

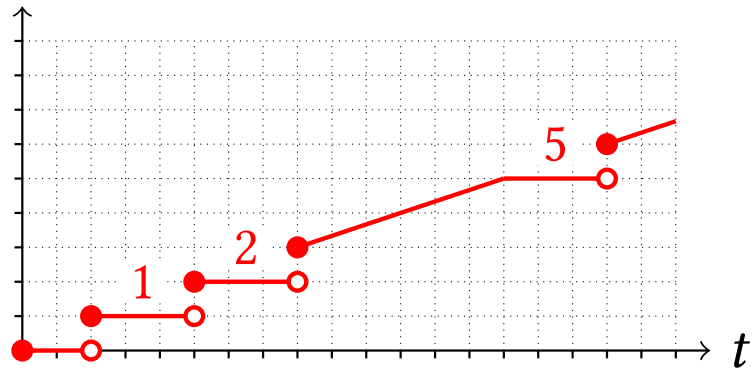


Approximation algorithms for explainability?

Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



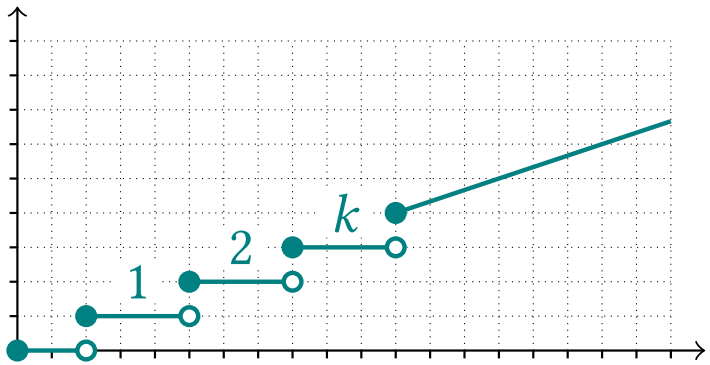
$\overline{\text{dbf}}_i(t)$



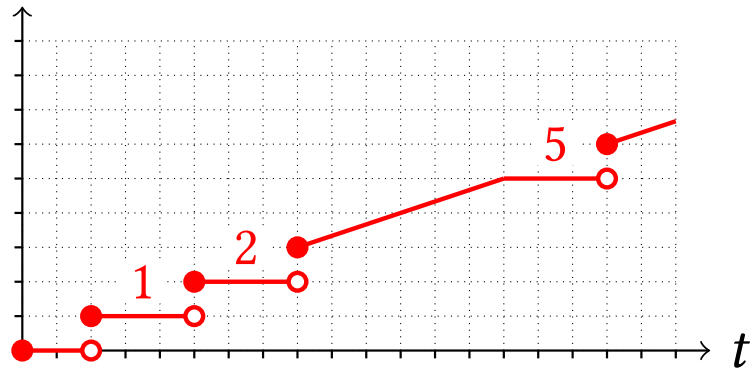
$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

Approximation algorithms for explainability?

Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



$\overline{\text{dbf}}_i(t)$

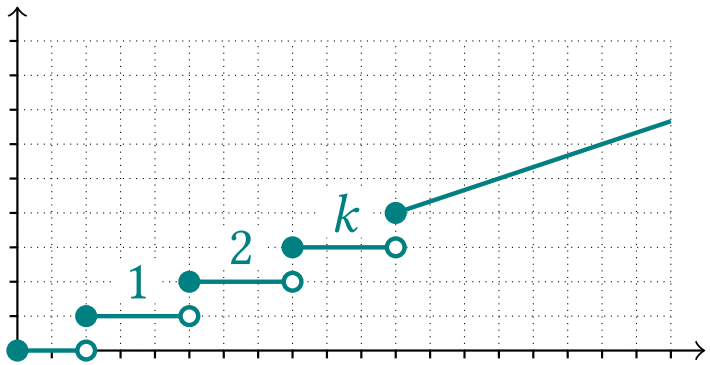


$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

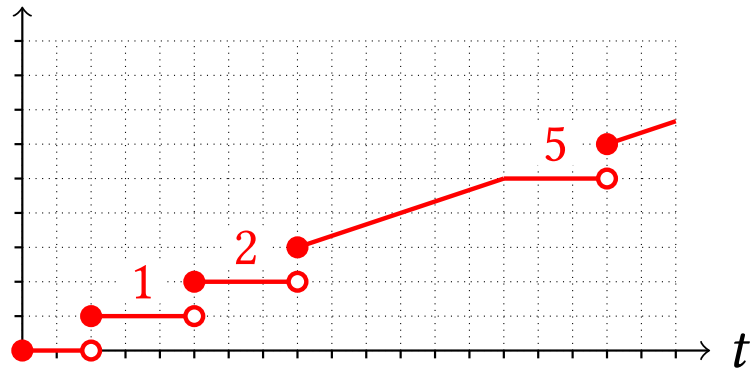
$\Gamma, \delta, \langle \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle \longrightarrow$ FPTVAS

Approximation algorithms for explainability?

Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



$\overline{\text{dbf}}_i(t)$



$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

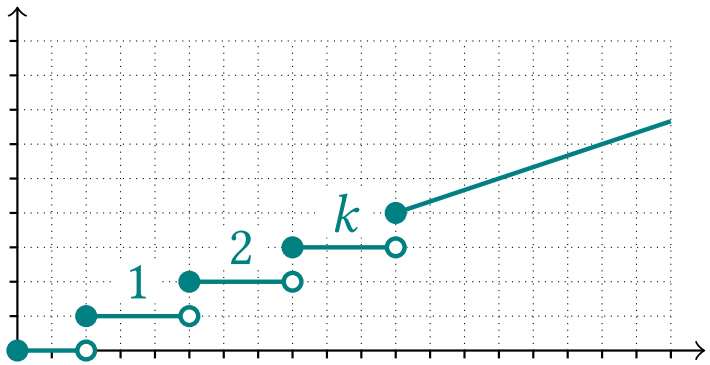
$\Gamma, \delta, \langle \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle \longrightarrow$ FPTVAS

FPTVAS =

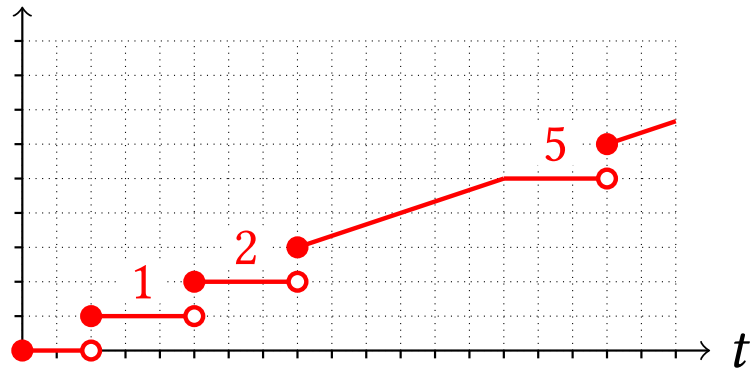
Fully Polynomial-Time
Verification Approximation Scheme

Approximation algorithms for explainability?

Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



$\overline{\text{dbf}}_i(t)$

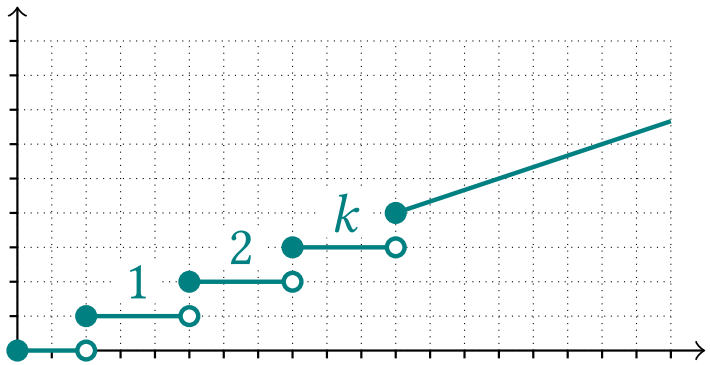


$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

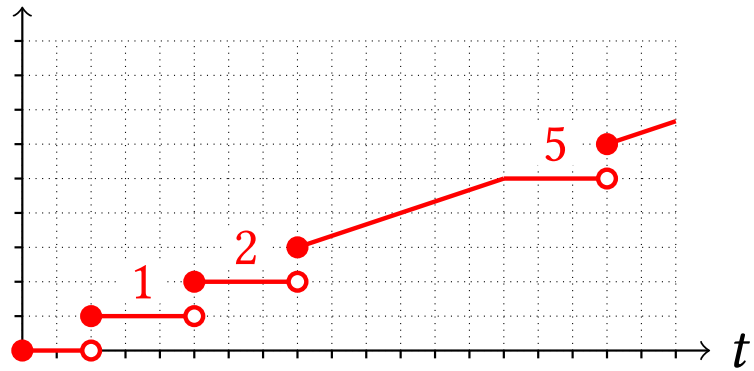
$\Gamma, \delta, \langle \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle \longrightarrow$ FPTVAS

Approximation algorithms for explainability?

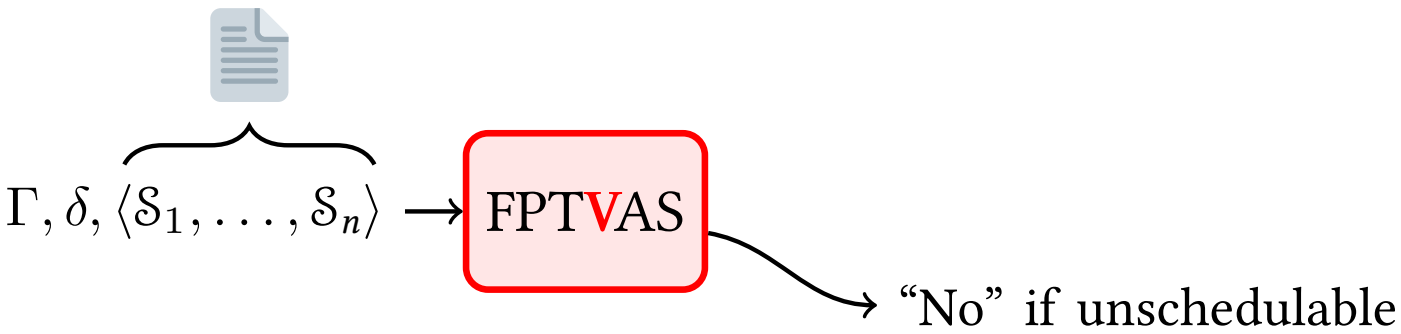
Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



$\overline{\text{dbf}}_i(t)$

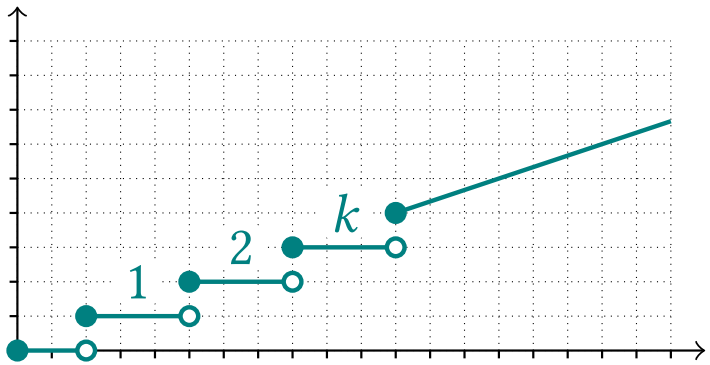


$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

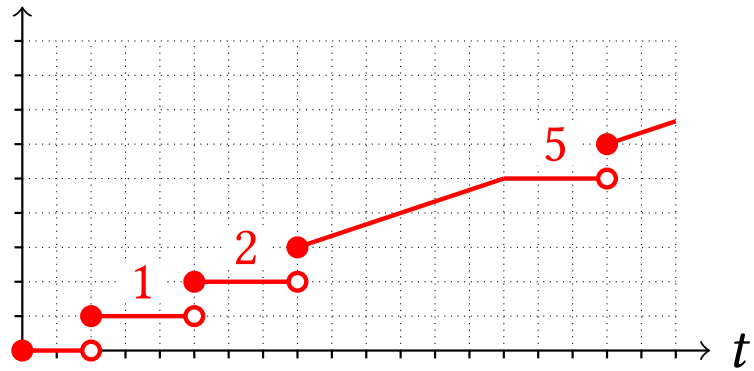


Approximation algorithms for explainability?

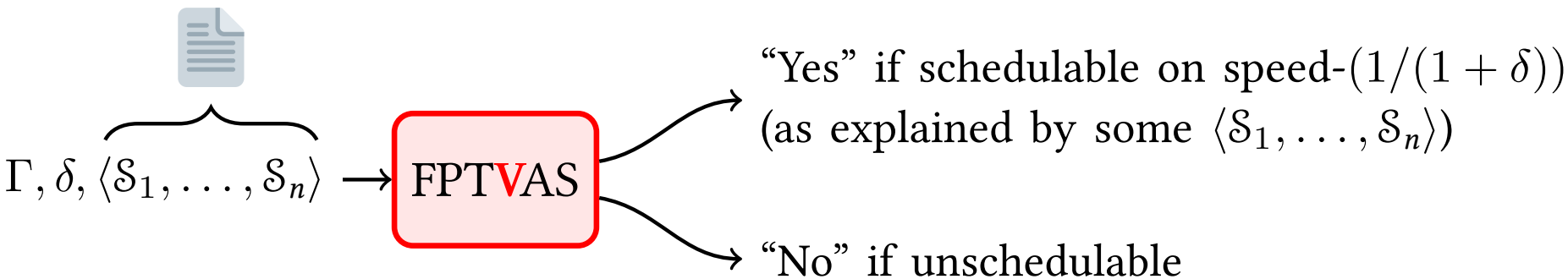
Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



$\overline{\text{dbf}}_i(t)$

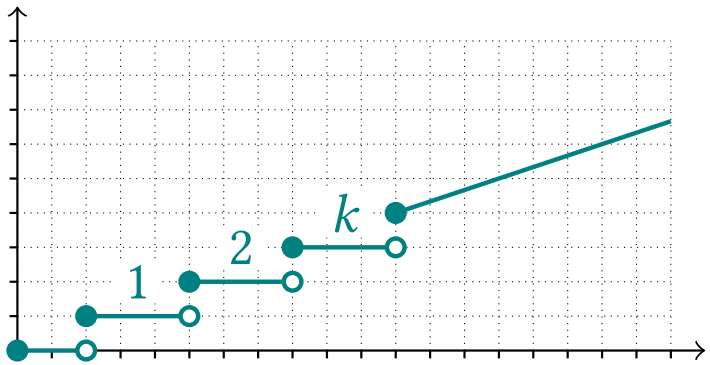


$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

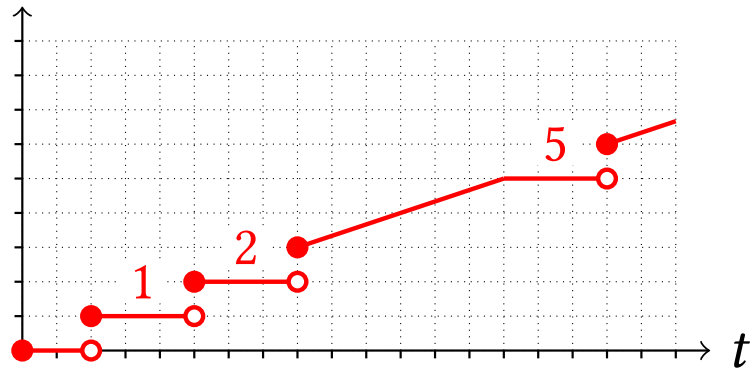


Approximation algorithms for explainability?

Idea: Let $\mathcal{S}_i \subset \mathbb{N}$ be the steps to keep



$\overline{\text{dbf}}_i(t)$



$\overline{\text{dbf}}_i(\mathcal{S}_i, t)$

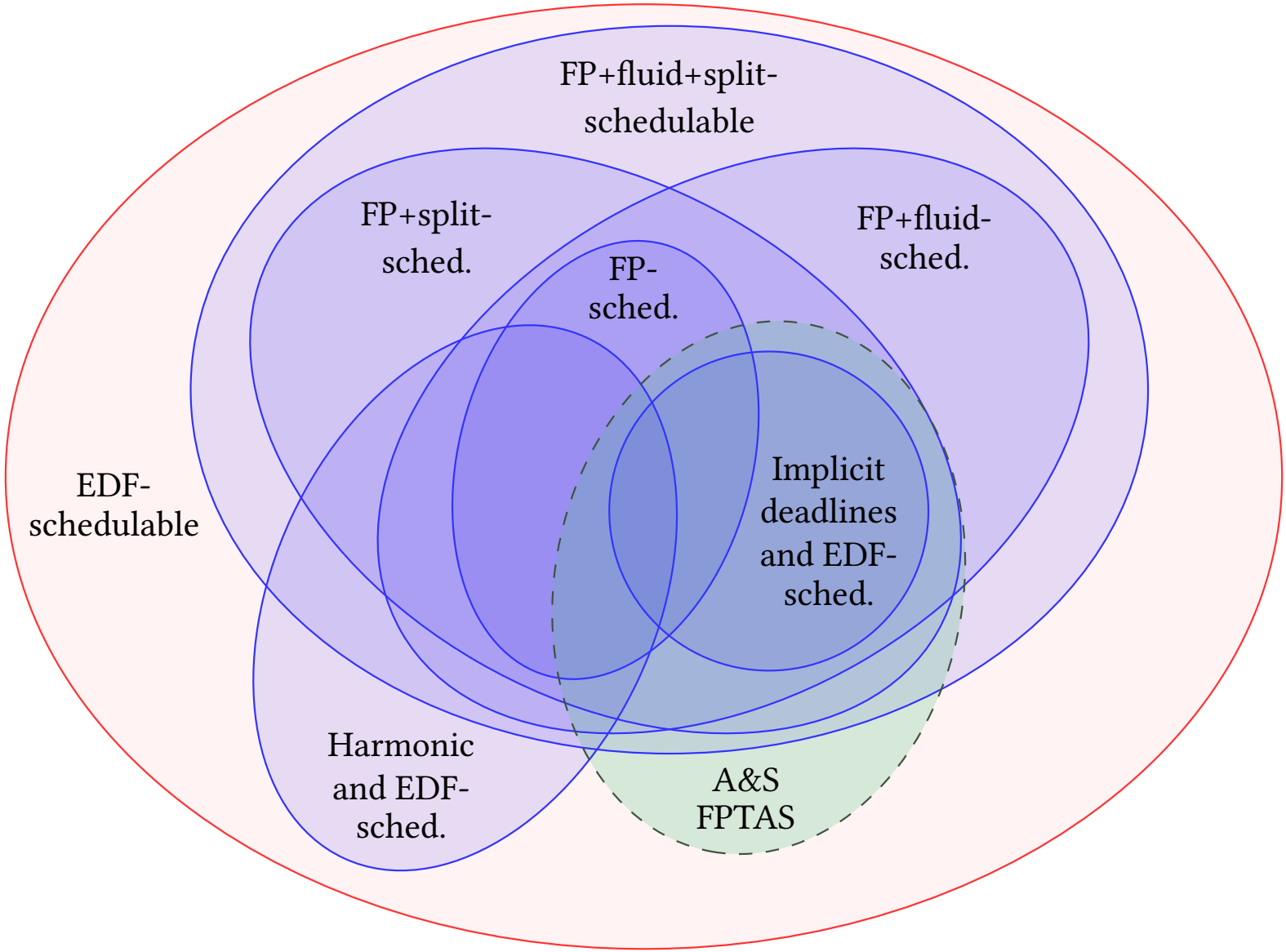
$\Gamma, \delta, \langle \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle \rightarrow$ FPTVAS

"Yes" if schedulable on speed-$(1/(1+\delta))$ (as explained by some $\langle \mathcal{S}_1, \ldots, \mathcal{S}_n \rangle$)

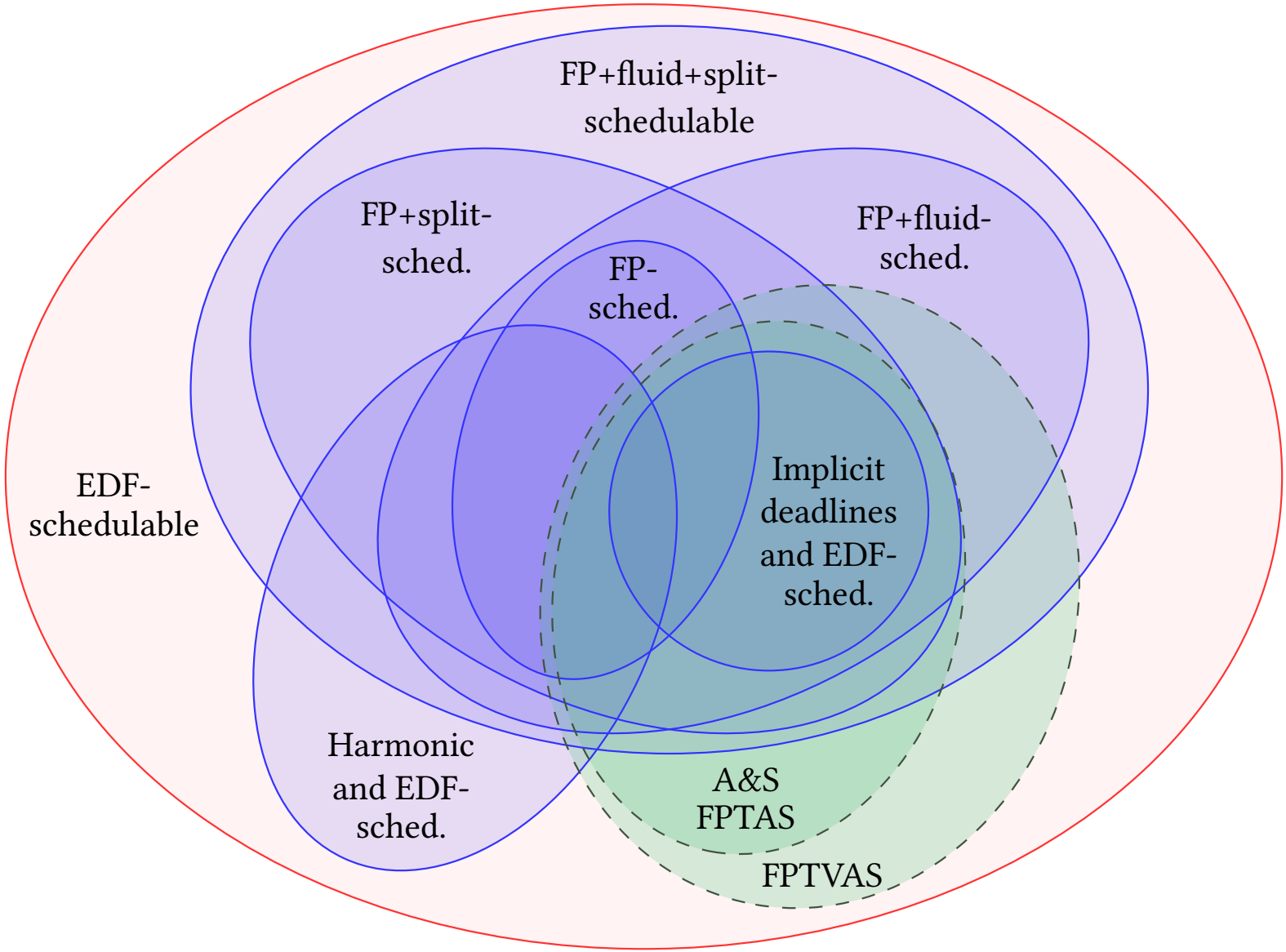
"No" if unschedulable

Polynomial in $|\Gamma|$ and $(\frac{1}{\delta})$

Dealing with EDF



Dealing with EDF



Pseudo-polynomial time verification

Pseudo-polynomial time verification

Definition: `pseudoNP`

A problem is in `pseudoNP` if solutions can be verified in pseudo-polynomial time.

Pseudo-polynomial time verification

Definition: `pseudoNP`

A problem is in `pseudoNP` if solutions can be verified in pseudo-polynomial time.

Is Γ schedulable by partitioned-EDF on m processors such that the utilization per partition is at most $c < 1$?

Pseudo-polynomial time verification

Definition: pseudoNP

A problem is in pseudoNP if solutions can be verified in pseudo-polynomial time.

Is Γ schedulable by partitioned-EDF on m processors such that the utilization per partition is at most $c < 1$?

✗ Not in NP (if NP $\neq$ coNP)

Pseudo-polynomial time verification

Definition: pseudoNP

A problem is in pseudoNP if solutions can be verified in pseudo-polynomial time.

Is Γ schedulable by partitioned-EDF on m processors such that the utilization per partition is at most $c < 1$?

✗ Not in NP (if NP $\neq$ coNP)

✗ Not solvable in pseudo-polynomial time (if P $\neq$ NP)

Pseudo-polynomial time verification

Definition: pseudoNP

A problem is in pseudoNP if solutions can be verified in pseudo-polynomial time.

Is Γ schedulable by partitioned-EDF on m processors such that the utilization per partition is at most $c < 1$?

✗ Not in NP (if NP $\neq$ coNP)

✗ Not solvable in pseudo-polynomial time (if P $\neq$ NP)

✓ In pseudoNP!

Conclusions

Conclusions

- Classification of schedulability problems as *efficiently explainable*

Conclusions

- Classification of schedulability problems as *efficiently explainable*
- Methods for those that are not inherently efficiently explainable:

Conclusions

- Classification of schedulability problems as *efficiently explainable*
- Methods for those that are not inherently efficiently explainable:
  - Identification of efficiently explainable subproblems

Conclusions

- Classification of schedulability problems as *efficiently explainable*
- Methods for those that are not inherently efficiently explainable:
  - Identification of efficiently explainable subproblems
  - A concept for efficiently explainable approximation: FPTVAS

Conclusions

- Classification of schedulability problems as *efficiently explainable*
- Methods for those that are not inherently efficiently explainable:
  - Identification of efficiently explainable subproblems
  - A concept for efficiently explainable approximation: FPTVAS
  - A relevant complexity class: `pseudoNP`

$\forall$Thank you!

$\exists$Questions?