Periodic Task Scheduling

CPEN 432 Real-Time System Design

Arpan Gujarati
University of British Columbia

Utilization Bounds [1/2]

- For any algorithm A, a utilization bound of U_{ub}^{A} implies
  - Task sets with a total processor utilization of $U \leq U_{ub}^{A}$ can be scheduled successfully
- EDF bound: $U_{ub}^{EDF} = 1 \ (100\%)$
- RM bound (simple): $U_{ub,simple}^{RM} = n(2^{1/n} - 1)$
  - If $n = 2$, then $0.828 \ (82.8\%)$
  - If $n \to \infty$, then $ln\, 2 = 0.693 \ (69.3\%)$
- RM bound (hyperbolic): $\prod_{i=1}^{n} (U_i + 1) \leq 2$
  - Not just a function of n, but of task-specific utilizations U_i

Utilization Bounds [2/2]

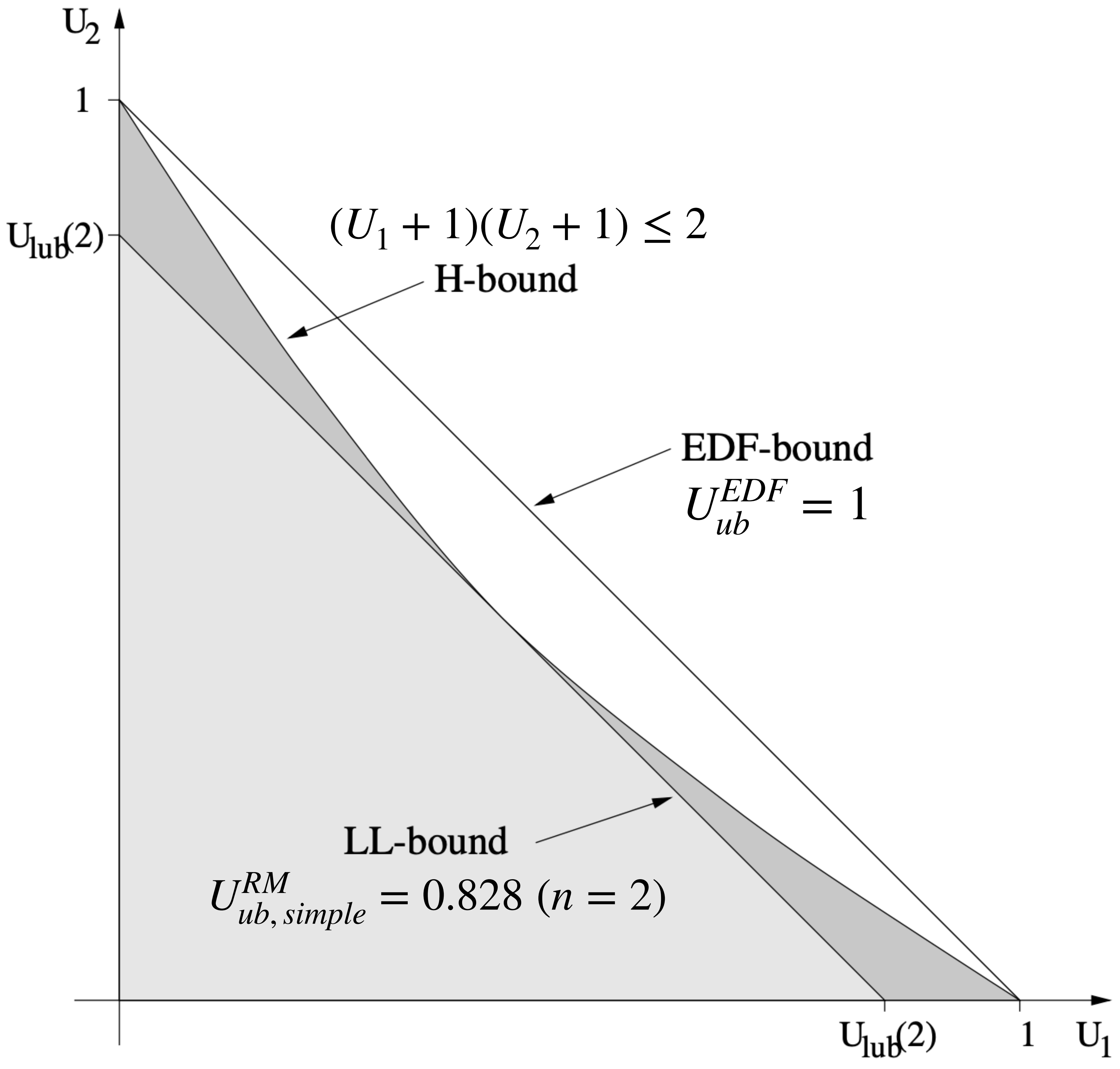
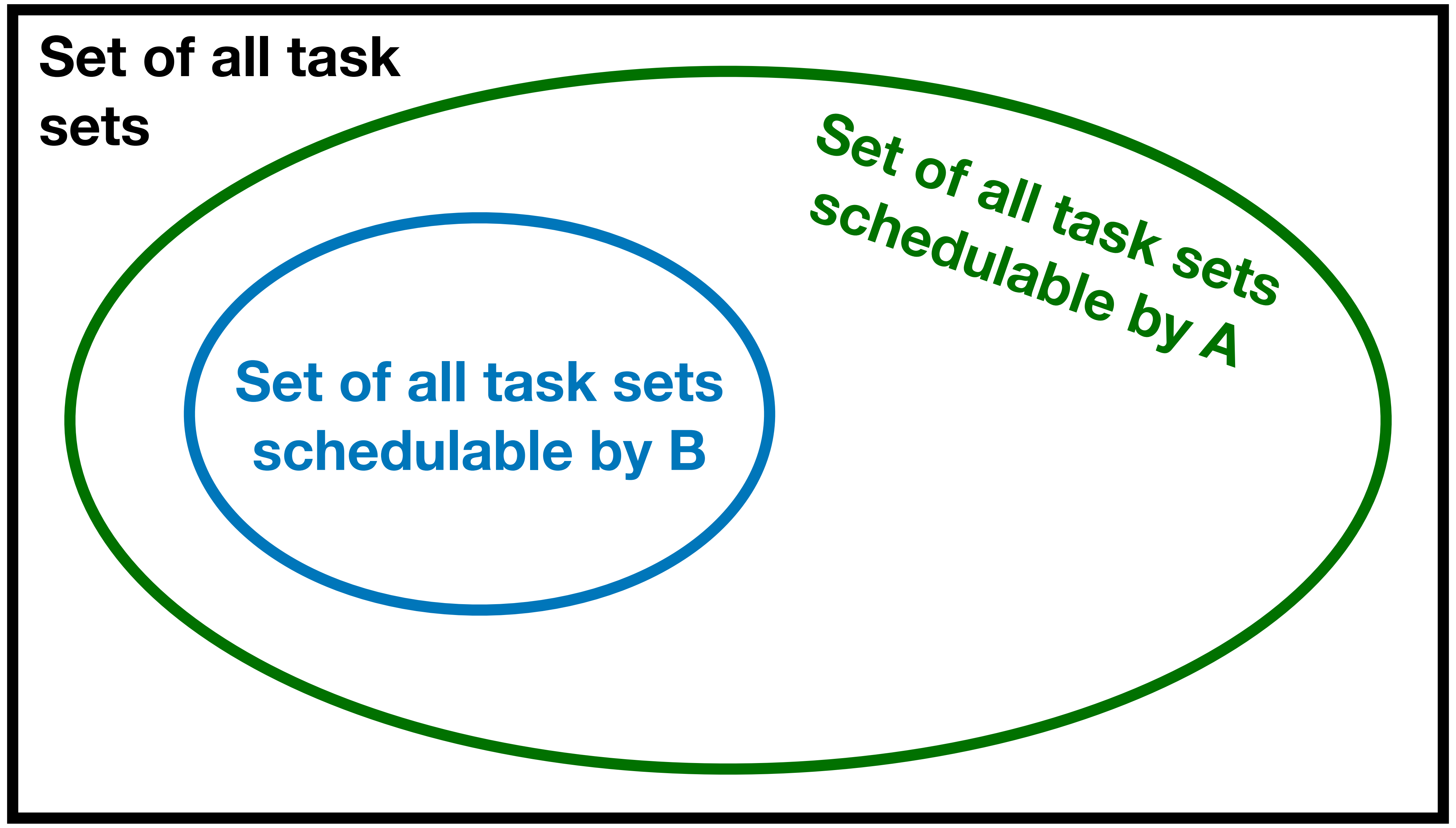


Figure 4.11 Schedulability bounds for RM and EDF in the utilization space.

- $U \leq U_{ub,simple}^{RM}$ is a **"sufficient"**, but **not a "necessary"** condition
  - Sufficient to ensure that the task set can be scheduled by RM
  - But not always necessary; many task sets with $U > U_{ub}^{A}$ can also be scheduled by RM
    - e.g., we saw earlier a task set with $U = 90\,\%$ that was schedulable by RM; some may even have $U = 100\,\%$
  - In other words: $U \leq U_{ub,simple}^{RM} \implies success$, however, $success \not\Longrightarrow U \leq U_{ub,simple}^{RM}$
- $U \leq U_{ub}^{EDF}$ is a **"sufficient"** and **also a "necessary"** condition
  - $U \leq U_{ub}^{EDF} \iff success$
- Question: Find a simple **necessary** but **not sufficient** test for RM?
  - ____________ $\not\Longrightarrow success$, however, $success \implies$ ____________
- Question: Is RM's hyperbolic bound, e.g., the condition $(U_1 + 1)(U_2 + 1) \leq 2$,
  1. sufficient but not necessary?
  2. not sufficient but necessary?
  3. both sufficient and necessary?
  4. neither sufficient nor necessary?
- How are RM's simple and hyperbolic bounds related?
  - The hyperbolic bound is a **"tight"** bound
    - Cannot be improved any further; beyond this bound, we can always find a task set that RM cannot schedule
  - The simple bound $U_{ub,simple}^{RM}$ is **"conservative"** or **"pessimistic"**

Comparing scheduling algorithms A and B [1/2]

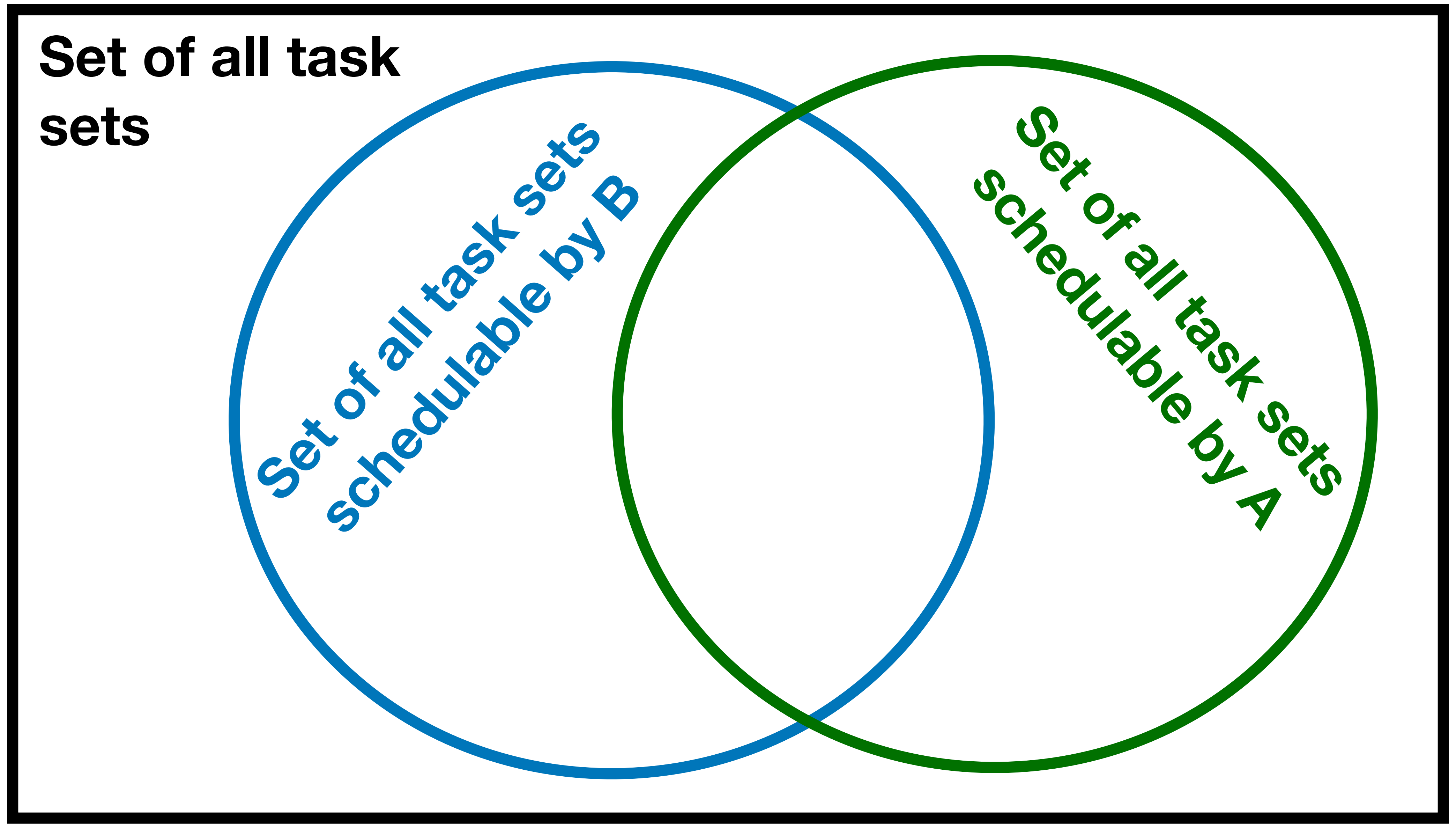
- $A \geq B$, i.e., A **"dominates"** B
  - Task set τ is schedulable using $B \implies$ Task set τ is schedulable using A
  - If B can successfully schedule τ, then A can also successfully schedule τ
  - Venn diagram: $S_{blue} \subseteq S_{green}$



- $A > B$, i.e., A **"strictly dominates"** B
  - At least one task set τ such that
    - A can successfully schedule τ
    - B cannot successfully schedule τ
  - $S_{blue} \subset S_{green}$

Comparing scheduling algorithms A and B [2/2]

- $A \neq B$, i.e., A and B are **"incomparable"**
  - At least two task sets τ and τ' such that
    - A can successfully schedule τ but B cannot
    - B can successfully schedule τ' but A cannot
  - Venn diagram: $S_{blue} \not\subseteq S_{green}$ and $S_{green} \not\subseteq S_{blue}$



How do RM and EDF compare?

- EDF ________________ RM!
- Question:
  - Two unknown algorithms A and B with utilization bounds of U_{ub}^{A} and U_{ub}^{B}, respectively
    - Every task set τ whose utilization $U \leq U_{ub}^{A}$ is schedulable using A
    - Every task set τ whose utilization $U \leq U_{ub}^{B}$ is schedulable using B
  - How are A and B related if $U_{ub}^{A} = 95\,\%$ and $U_{ub}^{B} = 69\,\%$?
    1. A dominates B
    2. A strictly dominates B
    3. B dominates A
    4. B strictly dominates A
    5. A and B are incomparable
    6. None of the above

How do RM and EDF compare?

- In terms of runtime overheads, which is better? Why?

What if $D_i < T_i$?

Recap: Assumptions

A1: All jobs of τ_i are regularly activated at a **constant frequency** of $1/T_i$

A2: All jobs of τ_i have the **same worst-case execution time** C_i

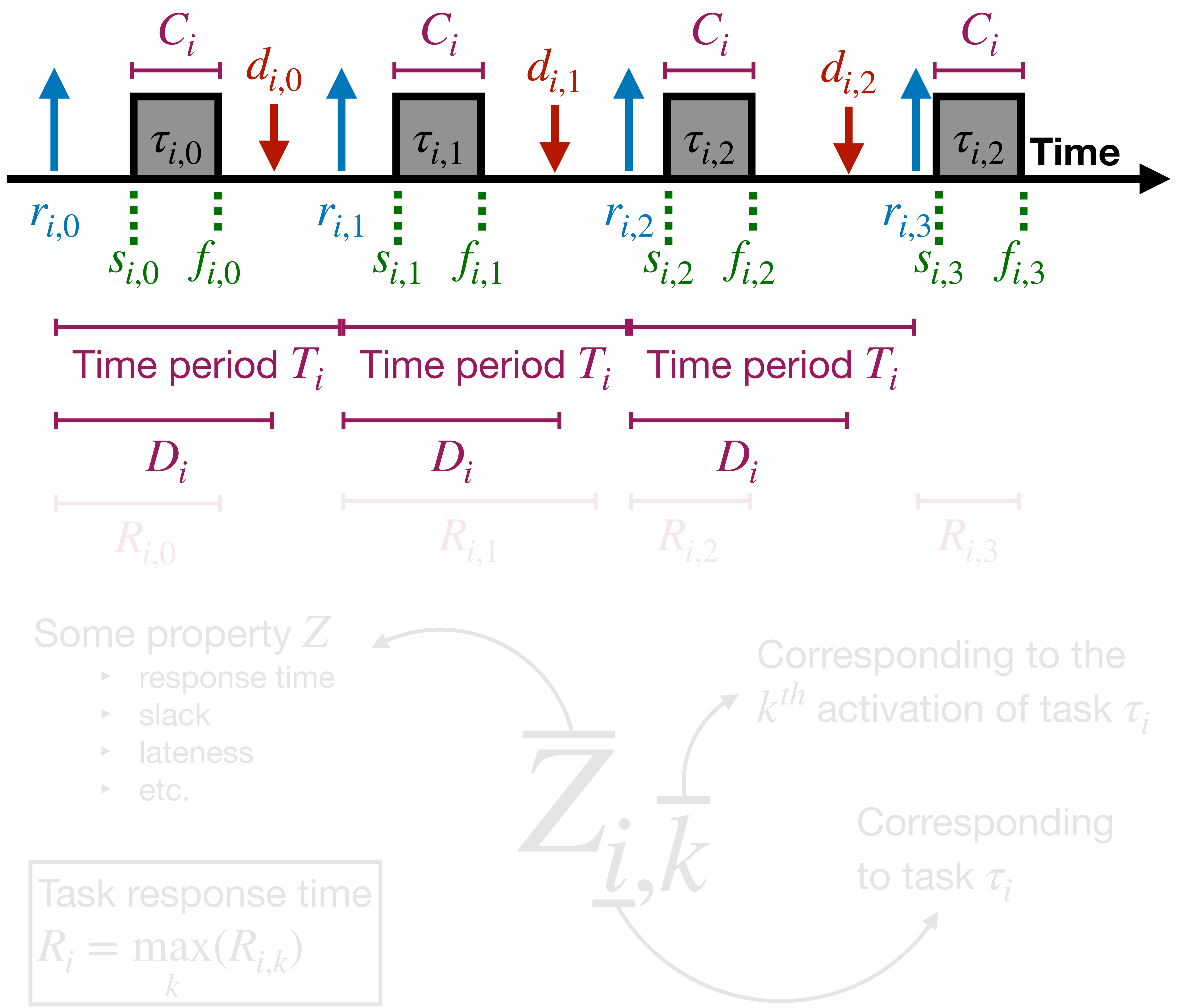
A3. All jobs of τ_i have the **same relative deadline** ~~$D_i = T_i$~~ $C_i \leq D_i \leq T_i$

A4. All tasks in τ are **independent** (no dependencies, no shared resources)

Implicit Deadlines

Constrained Deadlines

Recap: Aperiodic Job vs. Periodic Task



Why would you want $D_i \leq T_i$?

Is RM still optimal?

Do RM utilization bounds still hold?

~~Rate~~ Deadline Monotonic Scheduling (DM)

Is DM optimal among all fixed-priority algorithms?

- Like RM, **DM is optimal**
  - If a task set is schedulable by some fixed priority algorithm, it is also schedulable by DM
  - Proof of DM's optimality is similar to the one for RM

Schedulability Analysis for DM

- For simplicity: $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$ such that $D_1 < D_2 < \ldots < D_n$
- DM assigns the highest priority to τ_1, then to τ_2, and so on …
- Sketch:
  - Analyze one task at a time
    - E.g., let's analyze whether τ_i will meet all its deadlines.
  - Consider any arbitrary job of τ_i
    - E.g., let's consider its k^{th} job $\tau_{i,k}$, which is released at $r_{i,k}$ and whose deadline is at $d_{i,k} = r_{i,k} + D_i$
  - Consider another arbitrary task with a higher priority
    - E.g., let's consider τ_a $(a < i)$; from our model above, this implies that $D_a < D_i$ and thus τ_a has a higher priority than τ_i
  - What is the maximum duration for which task τ_a can **"interfere"** with job $\tau_{i,k}$
    - In other words, how often does DM schedule jobs of τ_a while job $\tau_{i,k}$ is still **"pending"**
    - We will refer to this quantify as **"interference"** from τ_a to $\tau_{i,k}$ and denote its value using $I^a_{i,k}$
  - Schedulability analysis:
    - Job $\tau_{i,k}$ does not miss its deadline if $C_i + \sum_{a<i} I^a_{i,k} \leq D_i$
  - Final step: Iterate!