IMPLEMENTING CUMULATIVE FUNCTIONS FOR CONDITIONAL TASK INTERVALS USING A GENERALIZED CUMULATIVE CONSTRAINT

Pierre Schaus[1], Roger Kameugne[1,2], **Charles Thomas**[1]

1: UCLouvain, Louvain-la-Neuve, Belgique

2: University of Maroua, Faculty of Sciences, Department of Mathematics and Computer Science

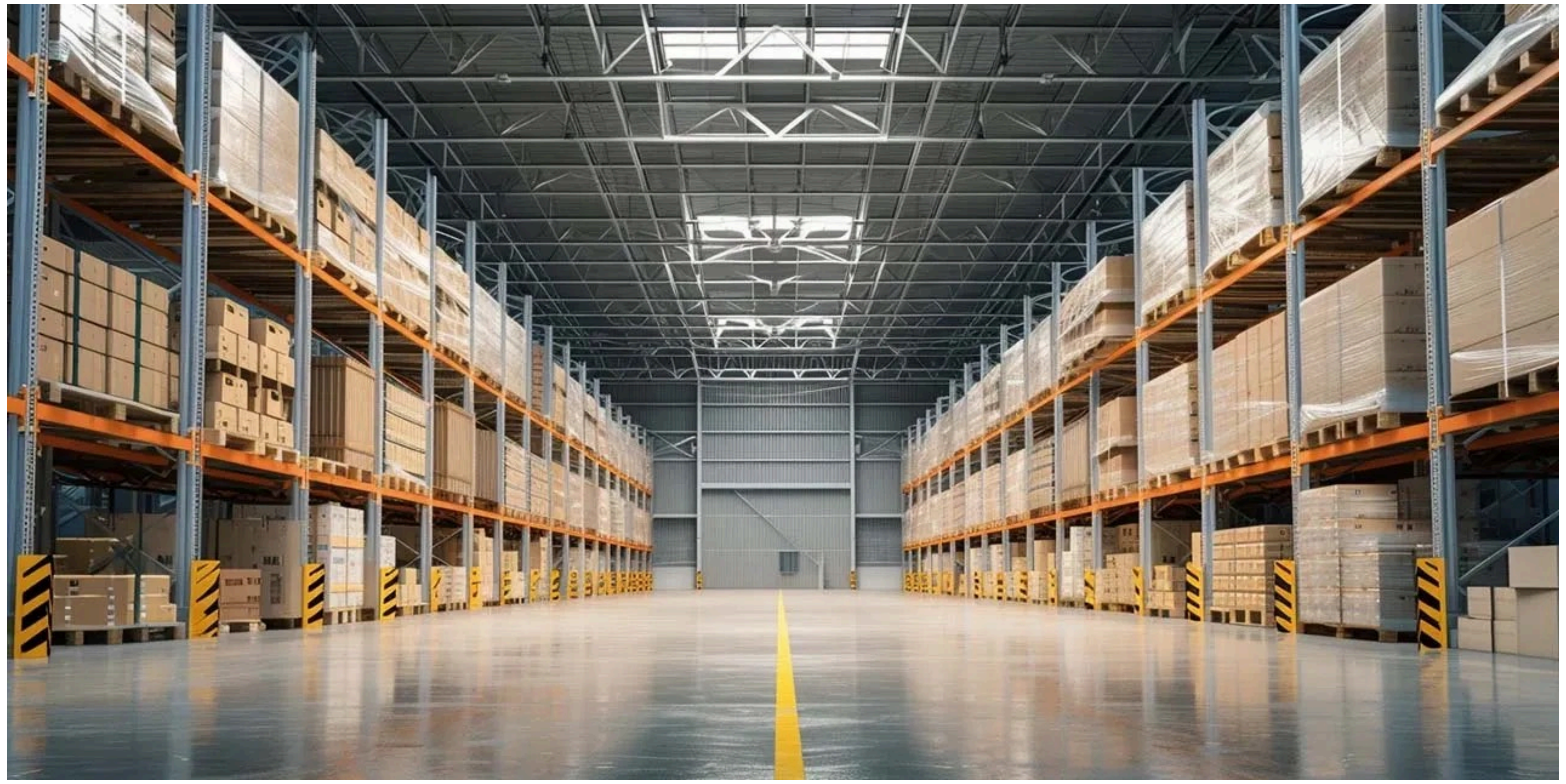
Artificial Intelligence & Algorithms

icteam

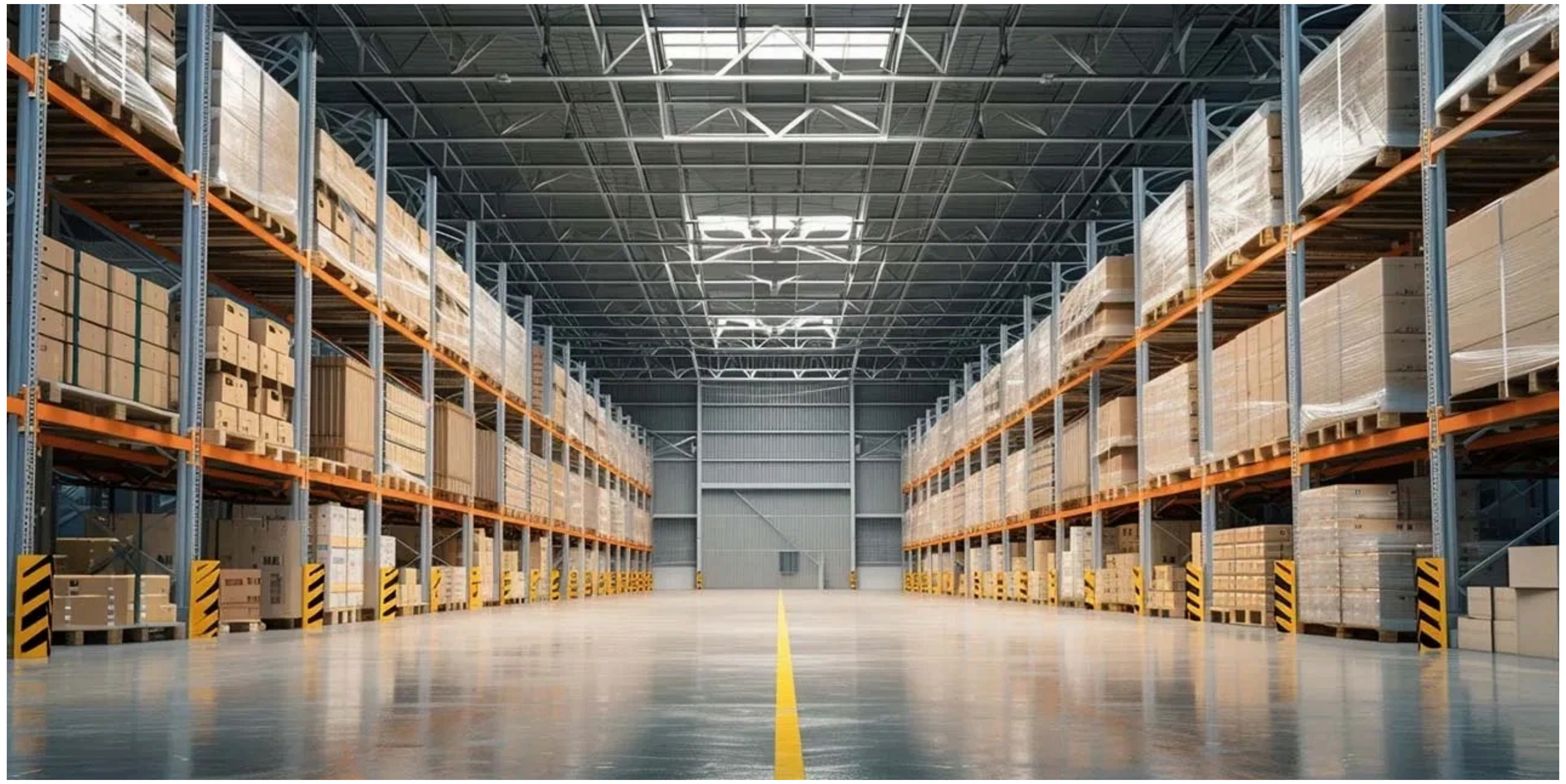
UCLouvain

SCHEDULING WITH RESOURCES

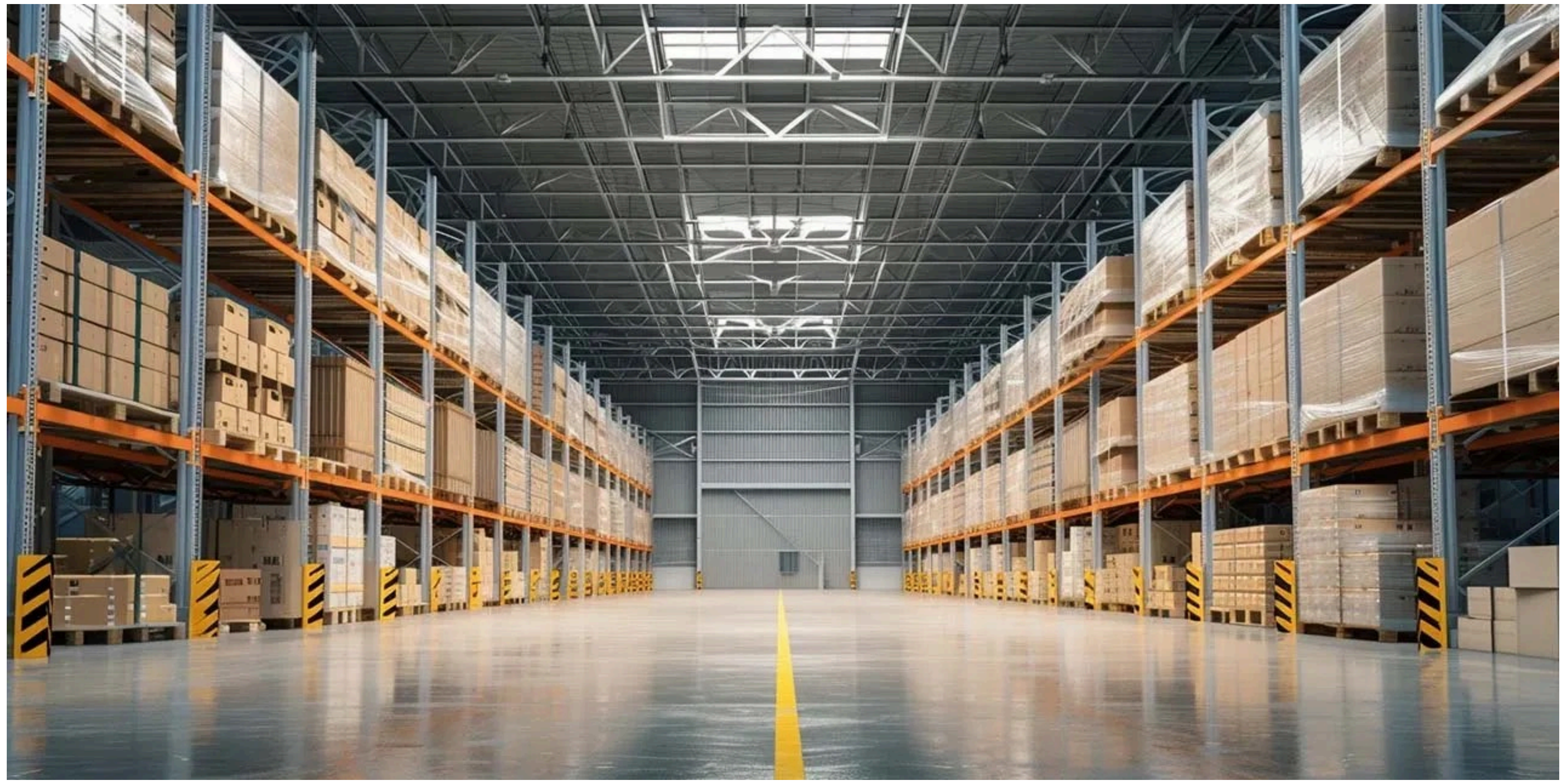
SCHEDULING WITH RESOURCES



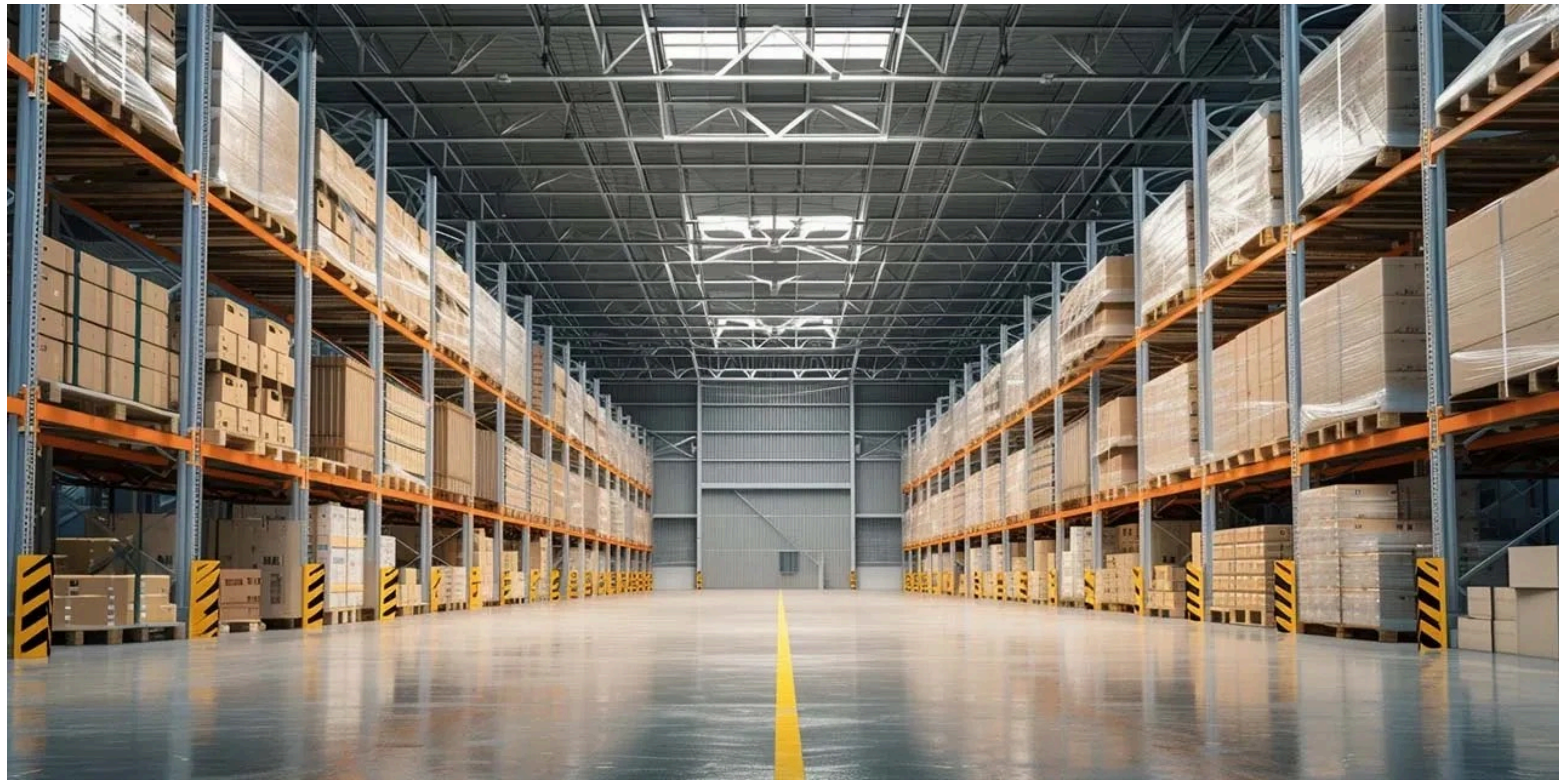
SCHEDULING WITH RESOURCES



SCHEDULING WITH RESOURCES



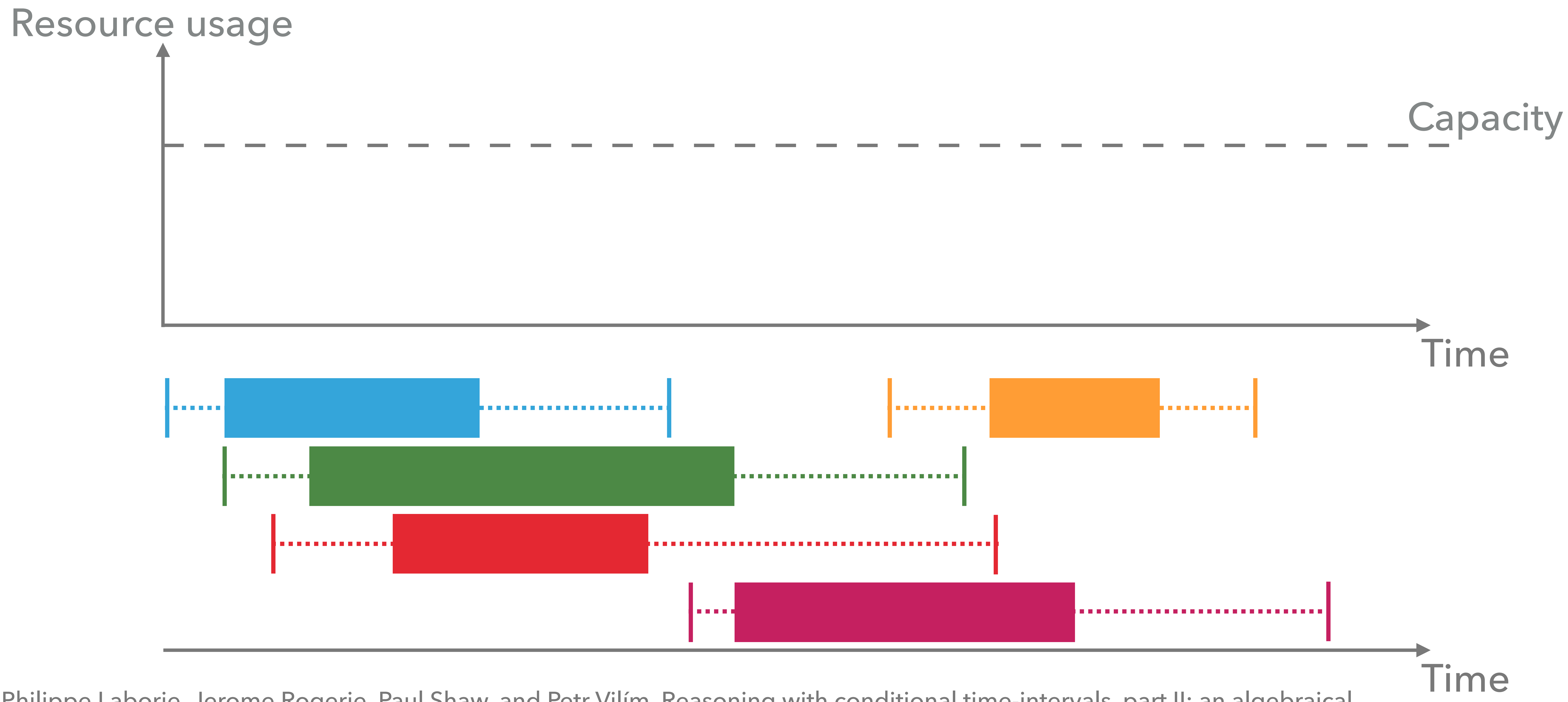
SCHEDULING WITH RESOURCES



CUMULATIVE FUNCTIONS

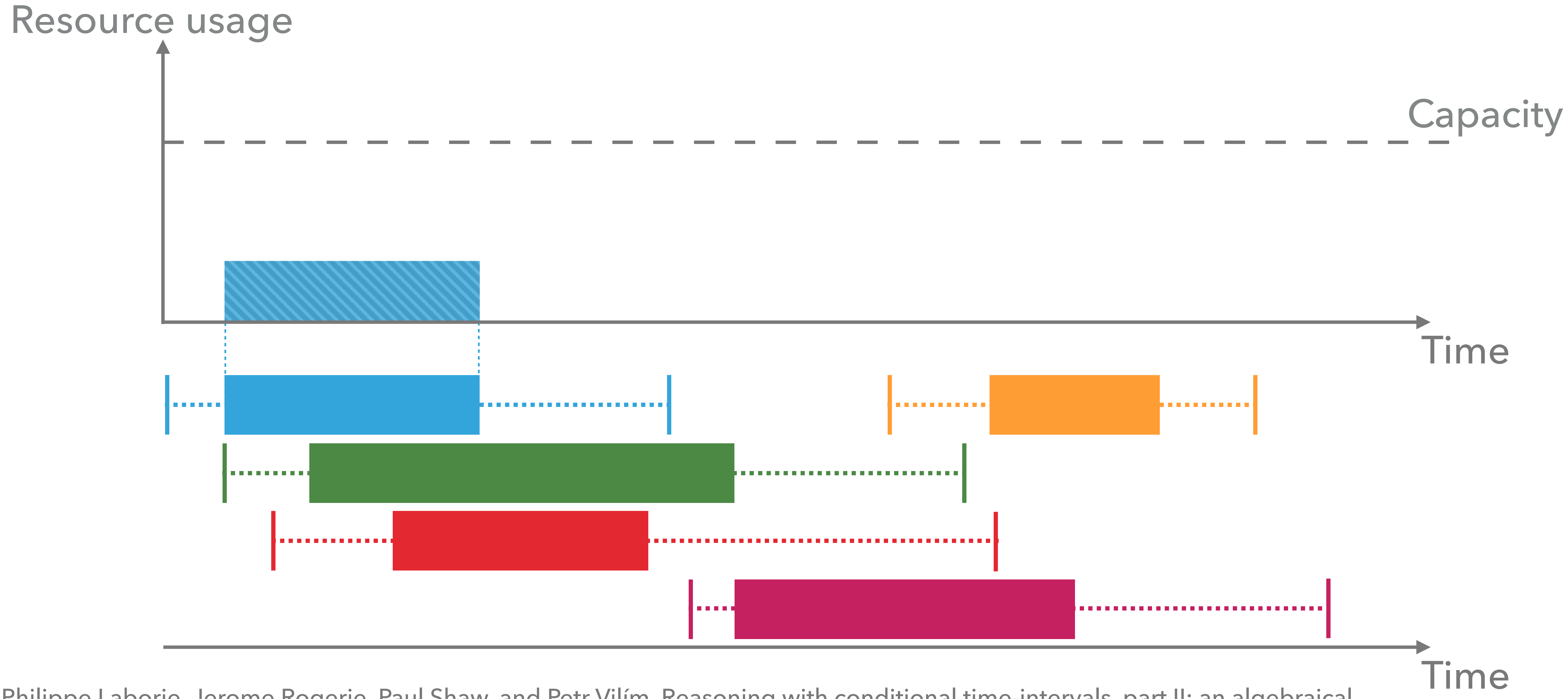
Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

CUMULATIVE FUNCTIONS



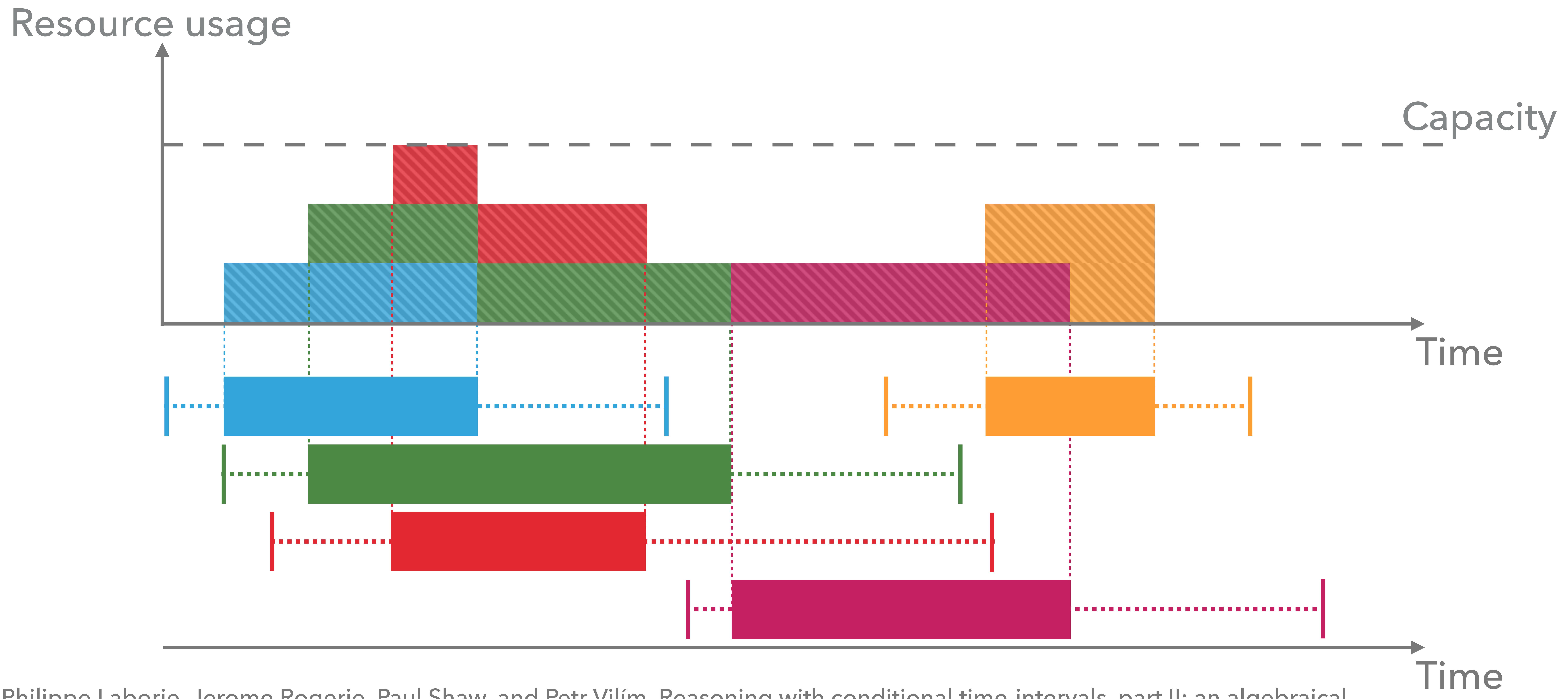
Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

CUMULATIVE FUNCTIONS



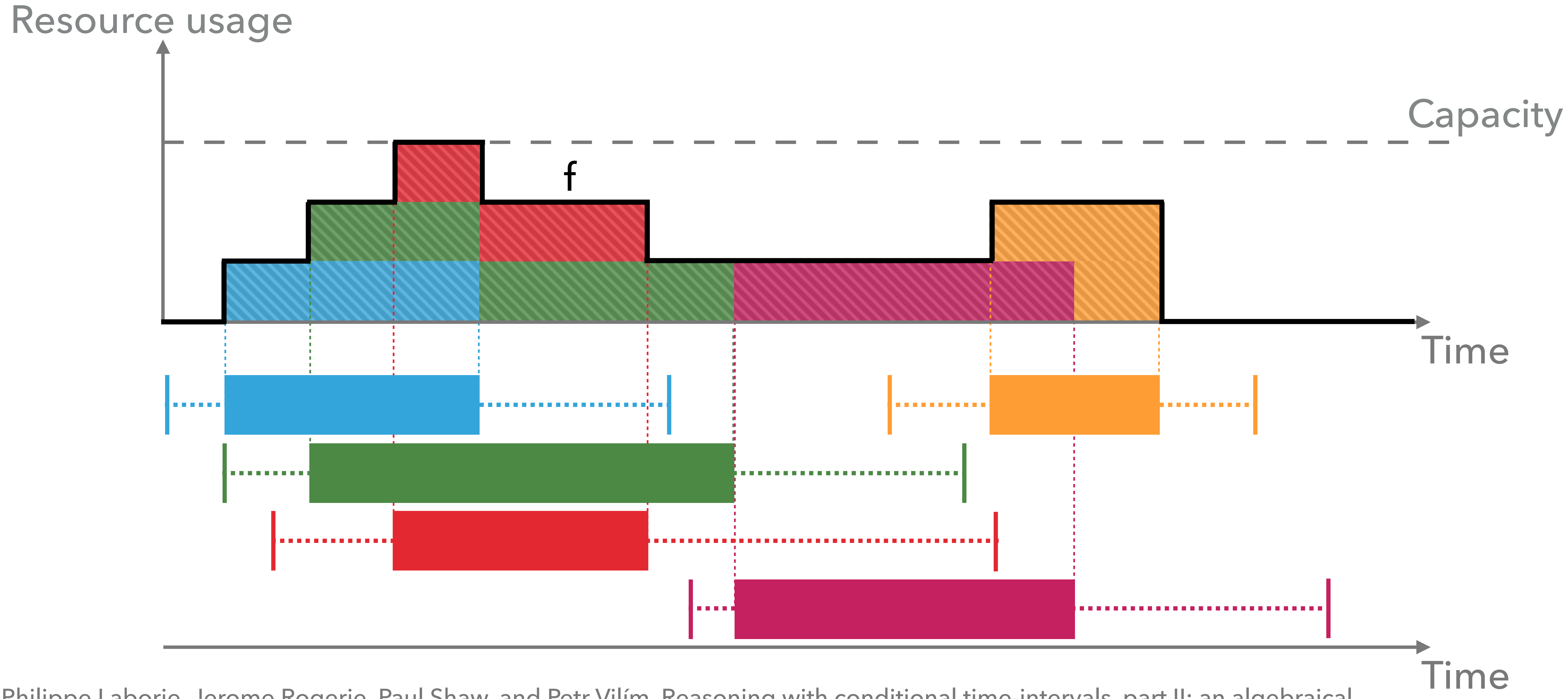
Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

CUMULATIVE FUNCTIONS



Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

CUMULATIVE FUNCTIONS

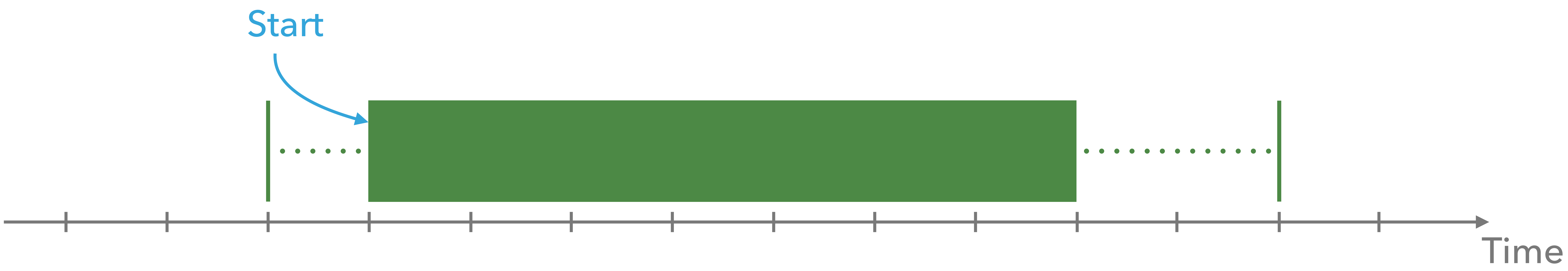


Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

CONDITIONAL TIME INTERVALS

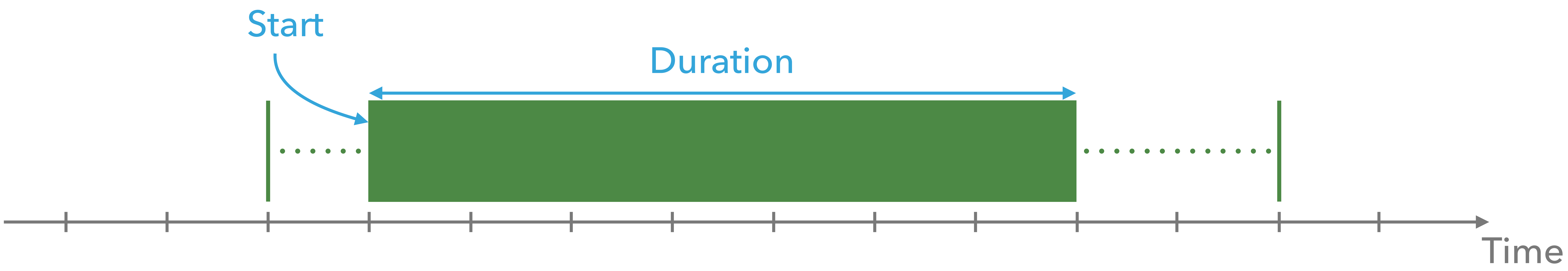
Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

CONDITIONAL TIME INTERVALS



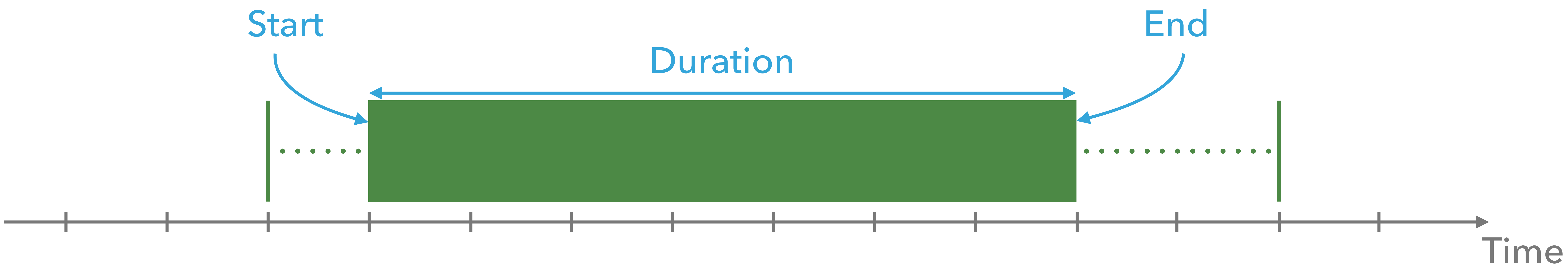
Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

CONDITIONAL TIME INTERVALS



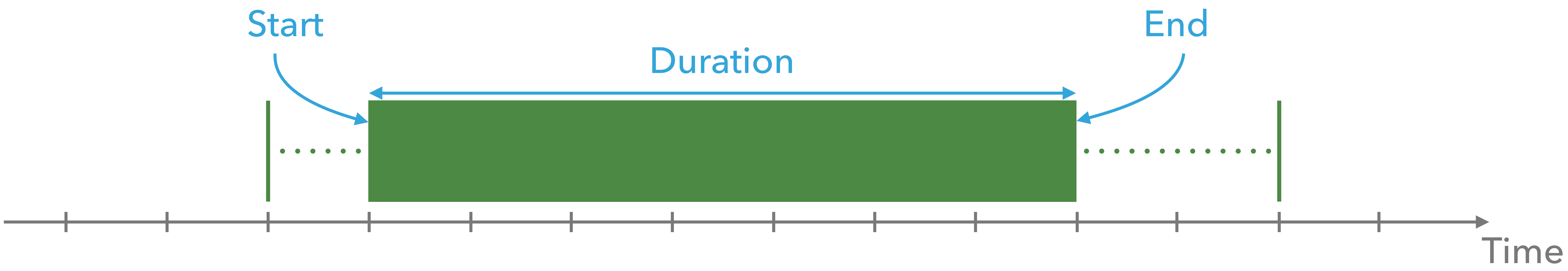
Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

CONDITIONAL TIME INTERVALS



Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

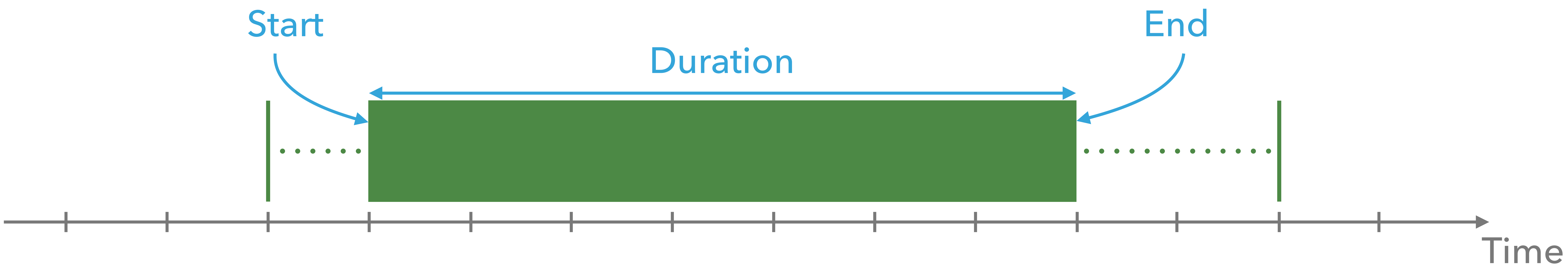
CONDITIONAL TIME INTERVALS



- **Optional**: can take the value $\perp$ to indicate that the interval is not present

Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

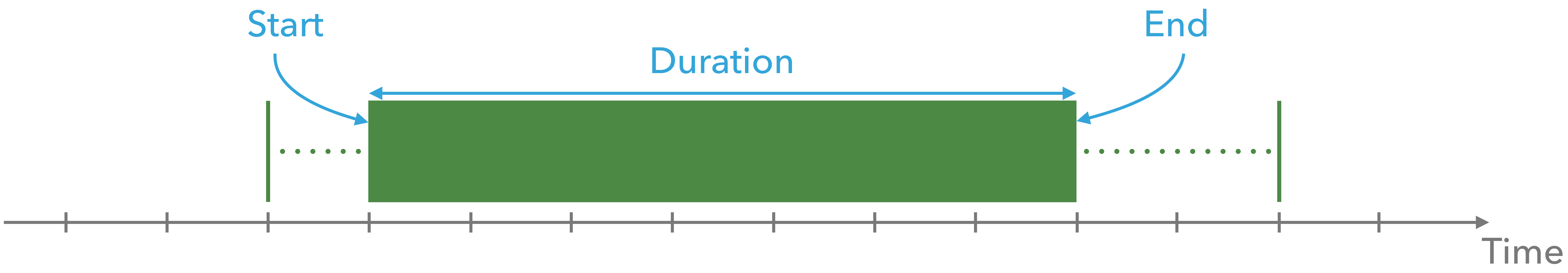
CONDITIONAL TIME INTERVALS



- **Optional**: can take the value ⊥ to indicate that the interval is not present
- Internal filtering:

Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

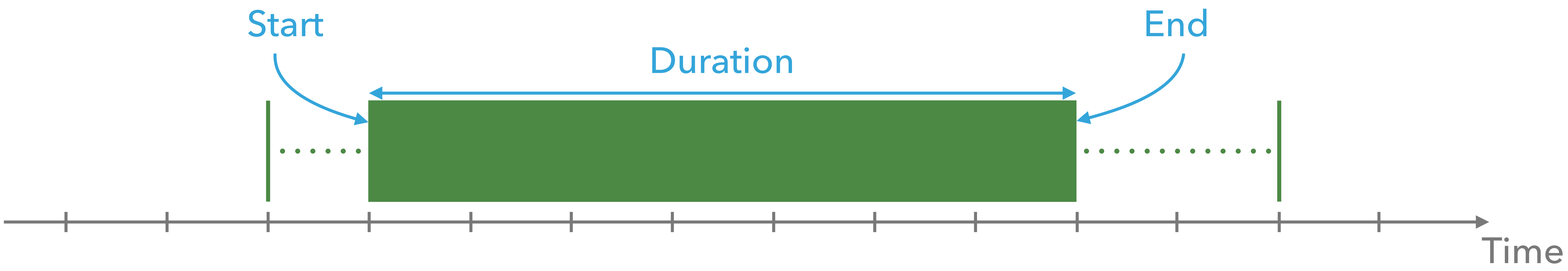
CONDITIONAL TIME INTERVALS



- **Optional**: can take the value ⊥ to indicate that the interval is not present
- Internal filtering:
  - **start + duration = end**

Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

CONDITIONAL TIME INTERVALS



- **Optional**: can take the value ⊥ to indicate that the interval is not present
- Internal filtering:
  - **start + duration = end**
  - **start = {}** ∨ **duration = {}** ∨ **end = {}** ⇒ ⊥

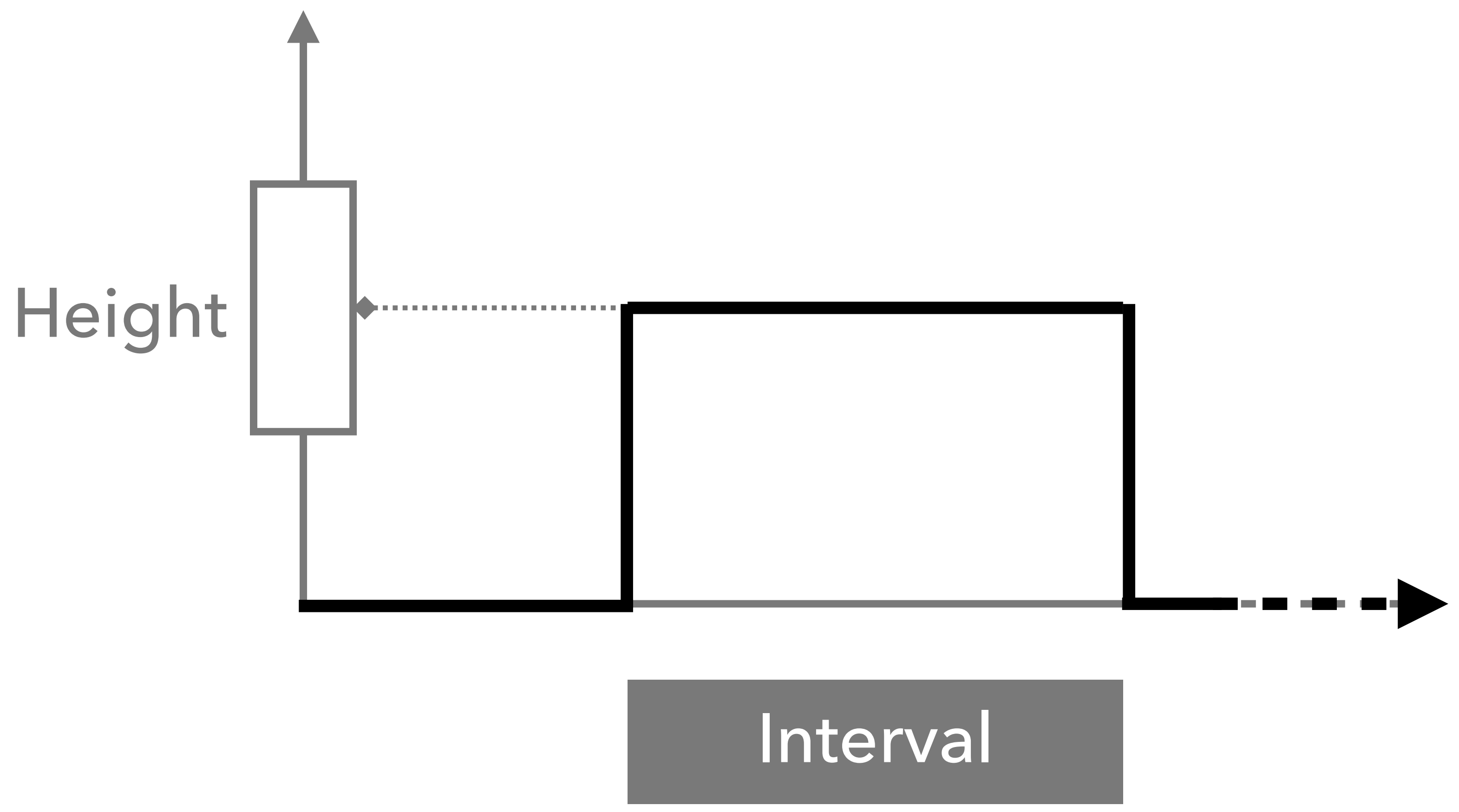
Philippe Laborie and Jerome Rogerie. Reasoning with conditional time-intervals. In *FLAIRS*, pages 555–560. AAAI Press, 2008.

ELEMENTARY CUMULATIVE FUNCTIONS

Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

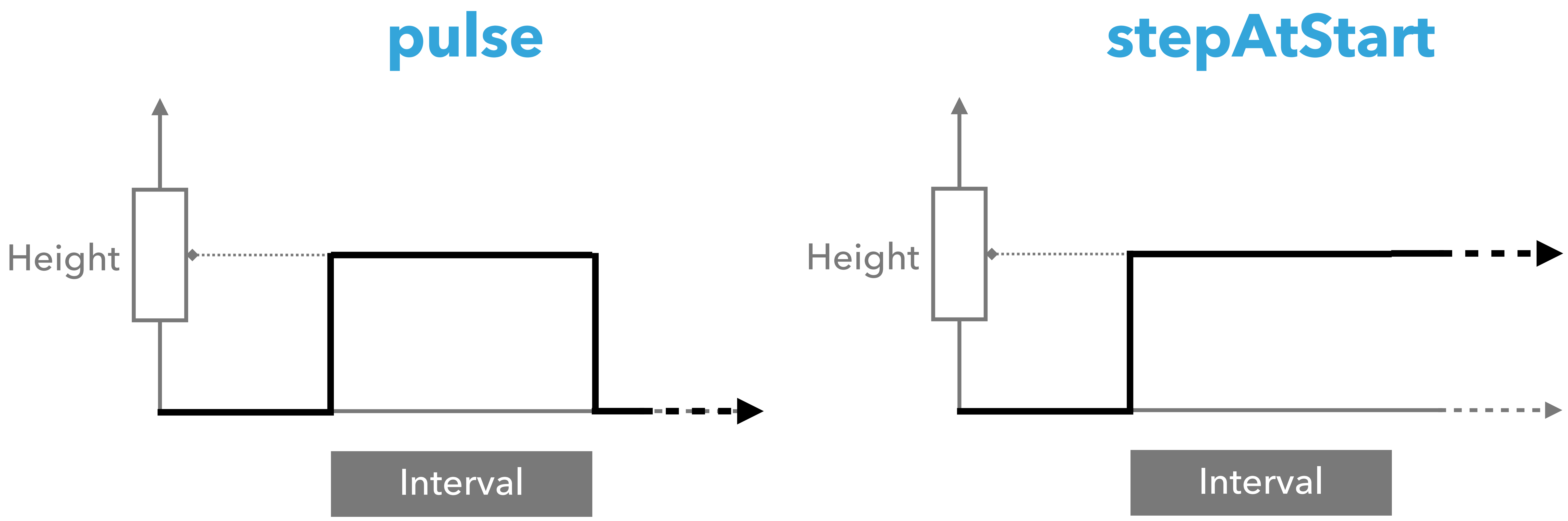
ELEMENTARY CUMULATIVE FUNCTIONS

pulse



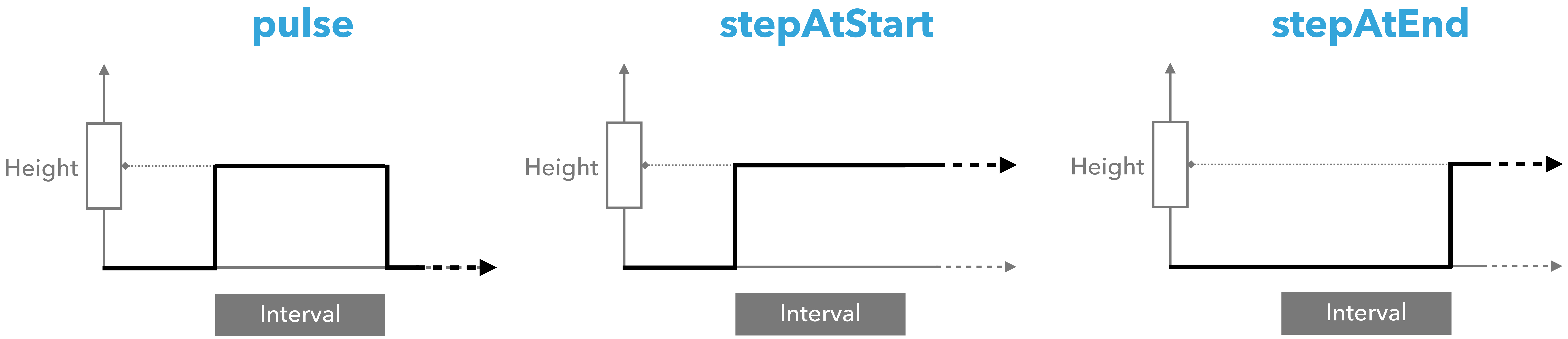
Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

ELEMENTARY CUMULATIVE FUNCTIONS



Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

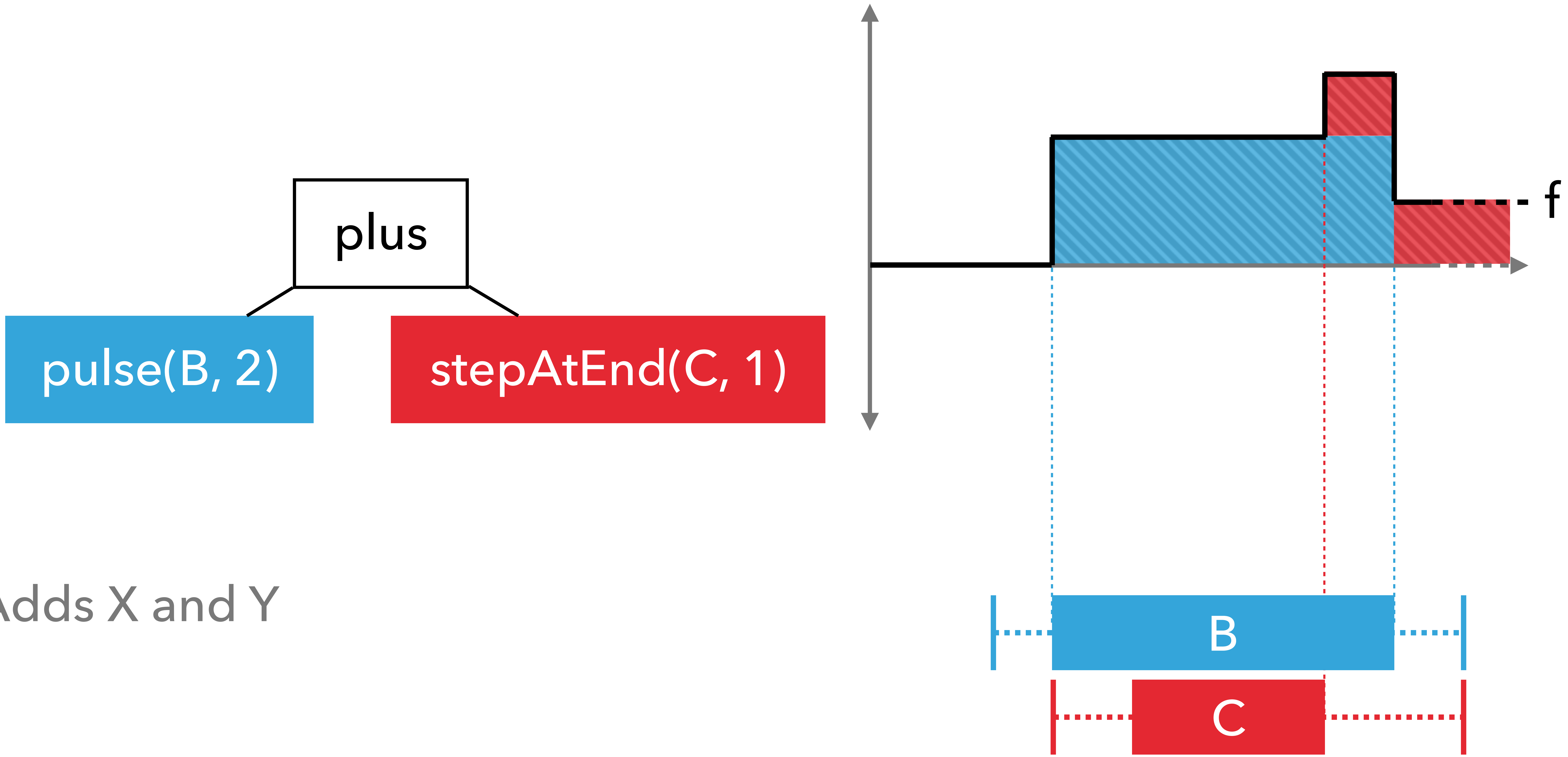
ELEMENTARY CUMULATIVE FUNCTIONS



Philippe Laborie, Jerome Rogerie, Paul Shaw, and Petr Vilím. Reasoning with conditional time-intervals. part II: an algebraical model for resources. In FLAIRS. AAAI Press, 2009.

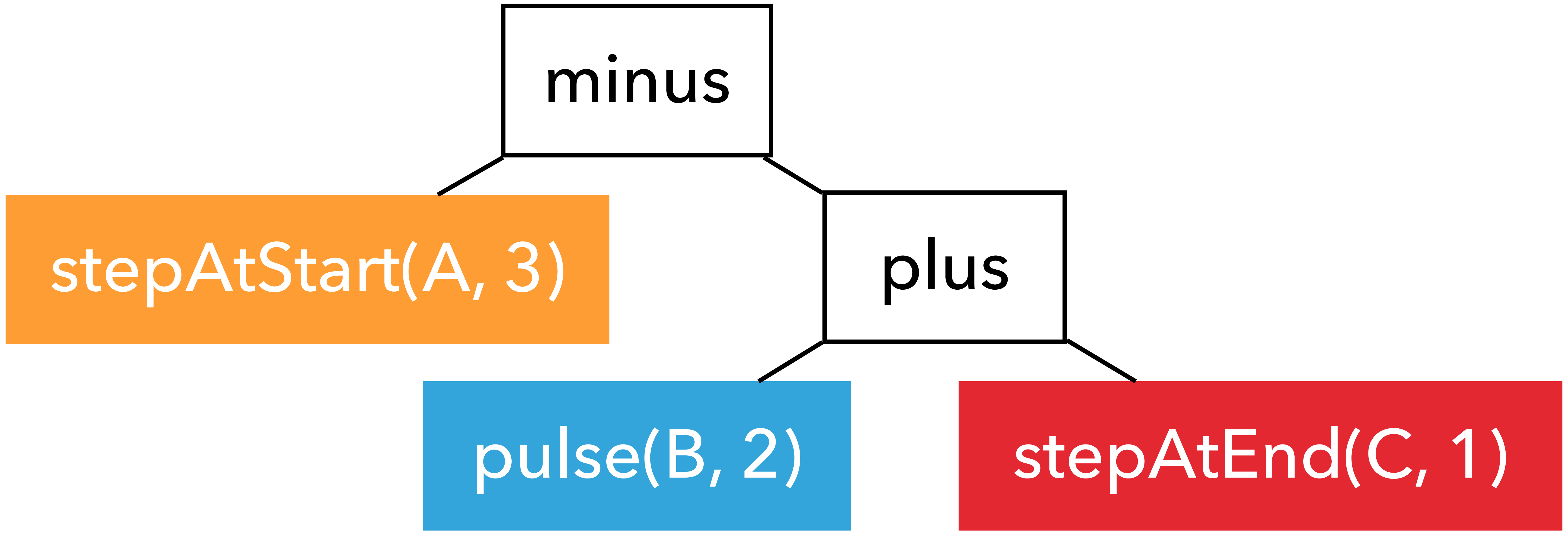
COMBINING CUMULATIVE FUNCTIONS

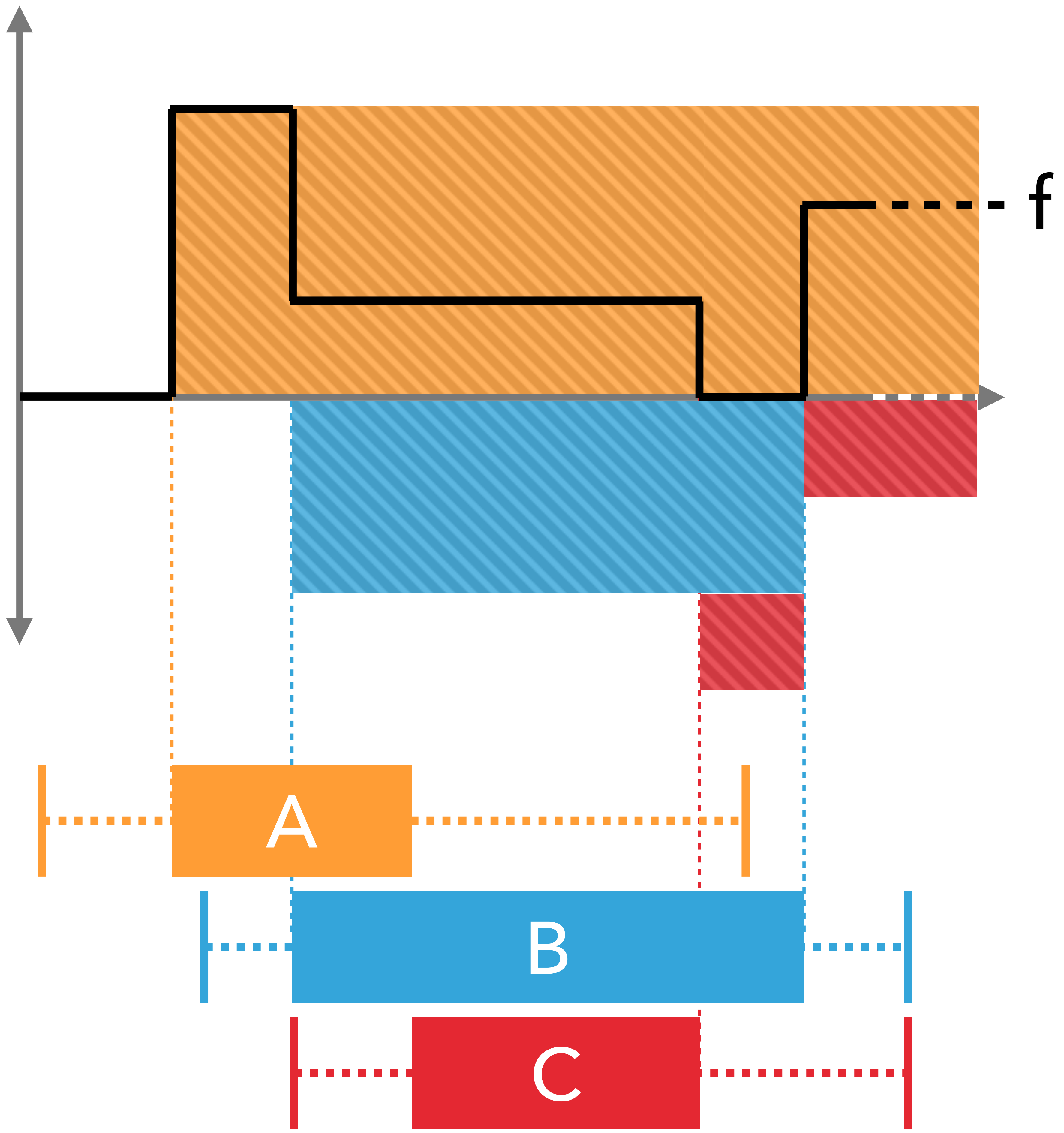
COMBINING CUMULATIVE FUNCTIONS



- **Plus(X, Y)** : Adds X and Y

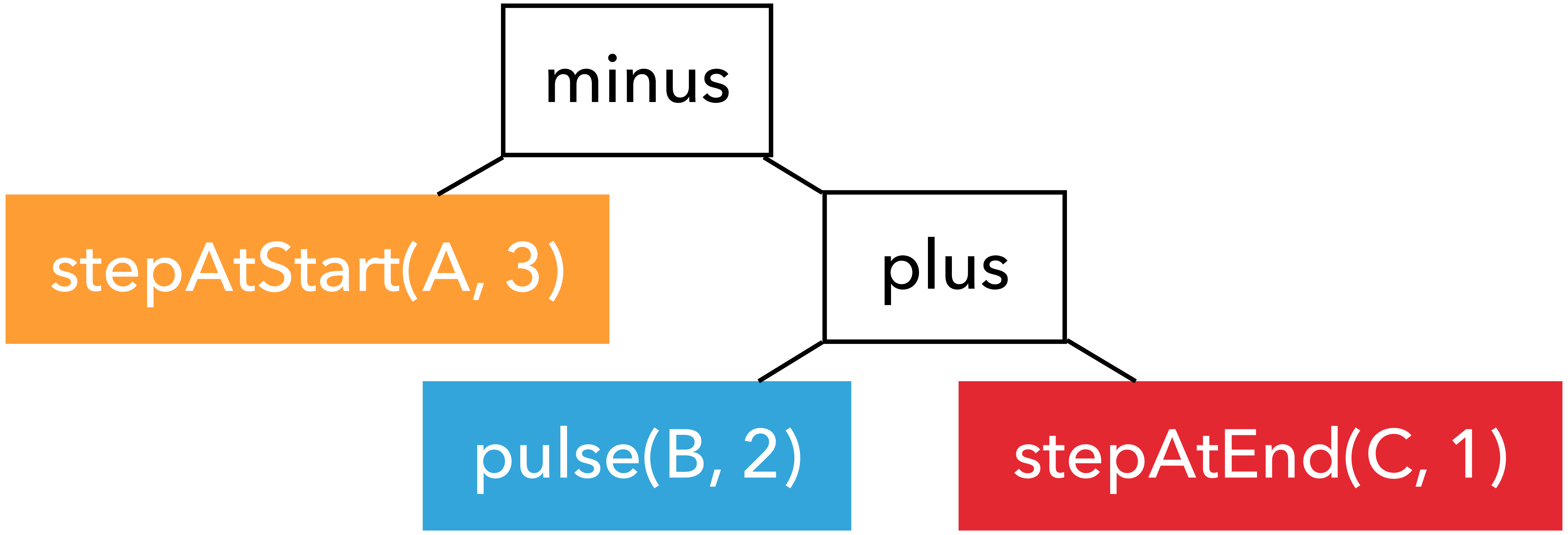
COMBINING CUMULATIVE FUNCTIONS

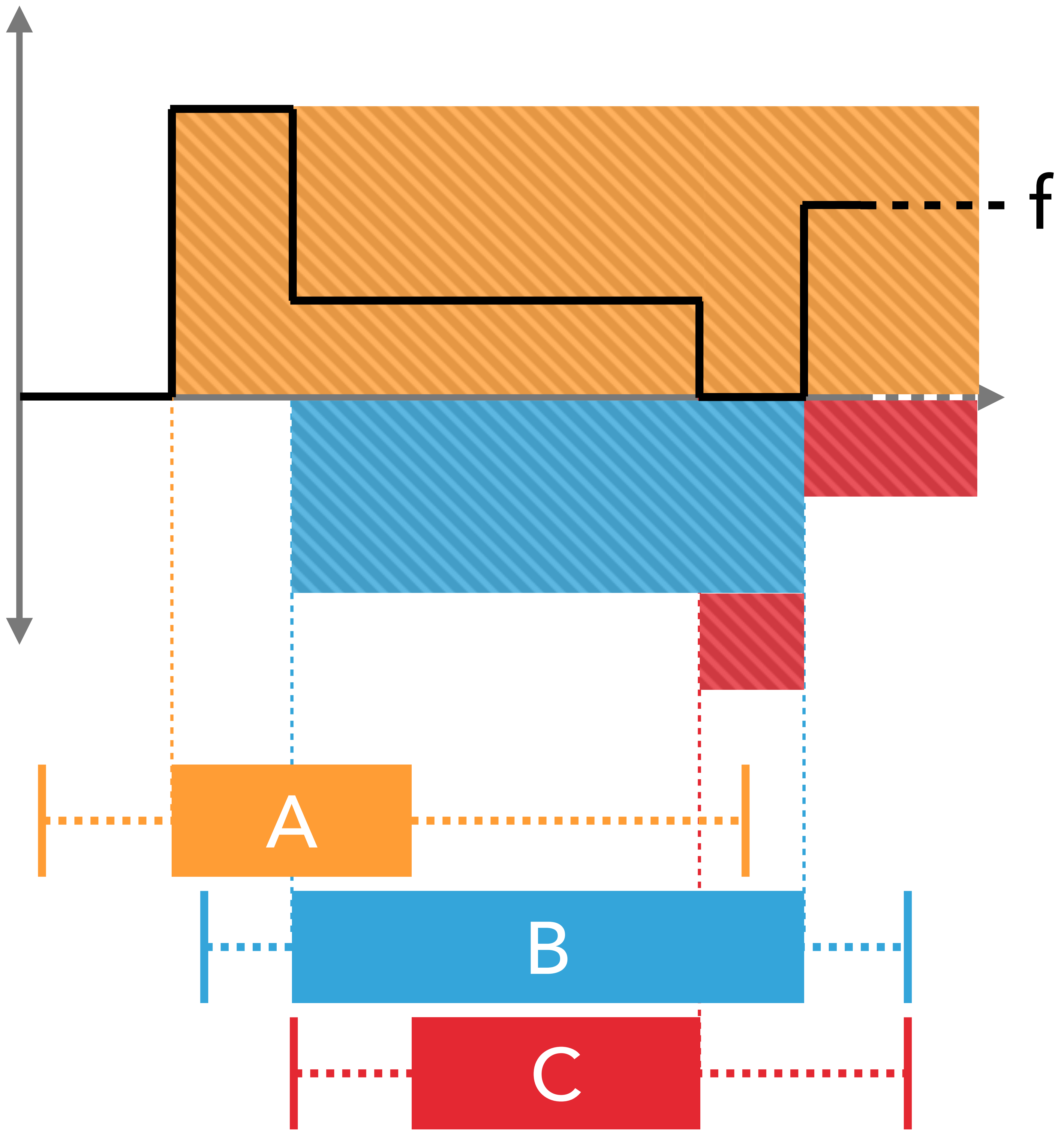




- **Plus(X, Y)** : Adds X and Y
- **Minus(X, Y)** : Subtracts Y from X

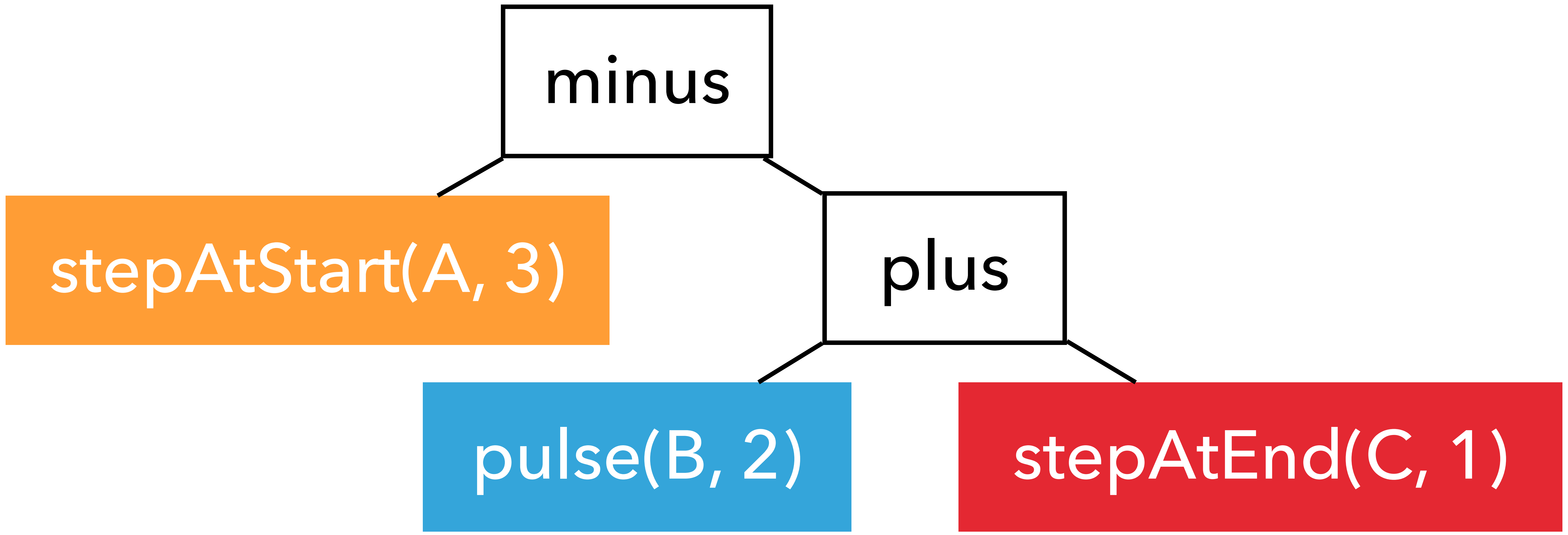
COMBINING CUMULATIVE FUNCTIONS

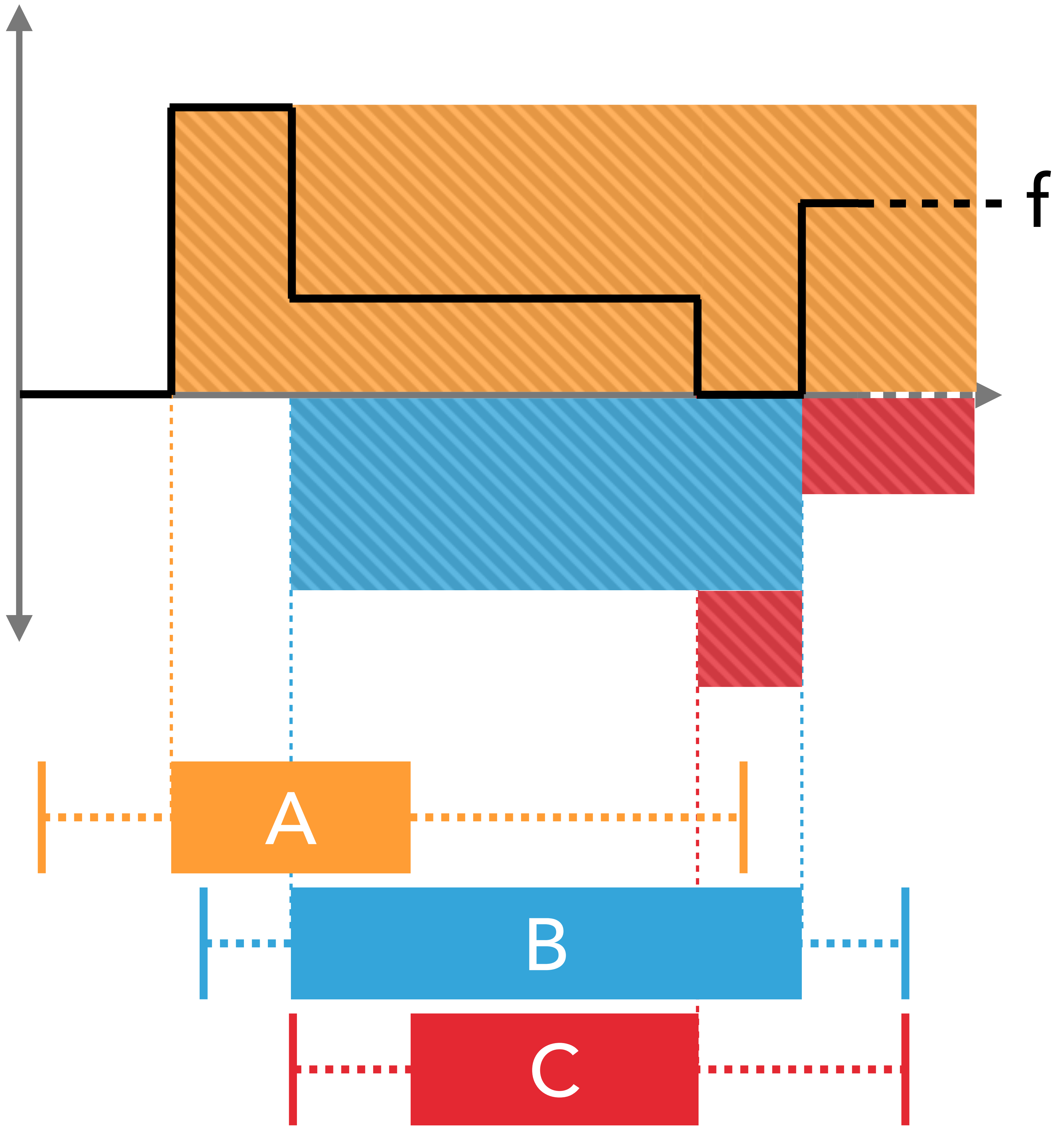




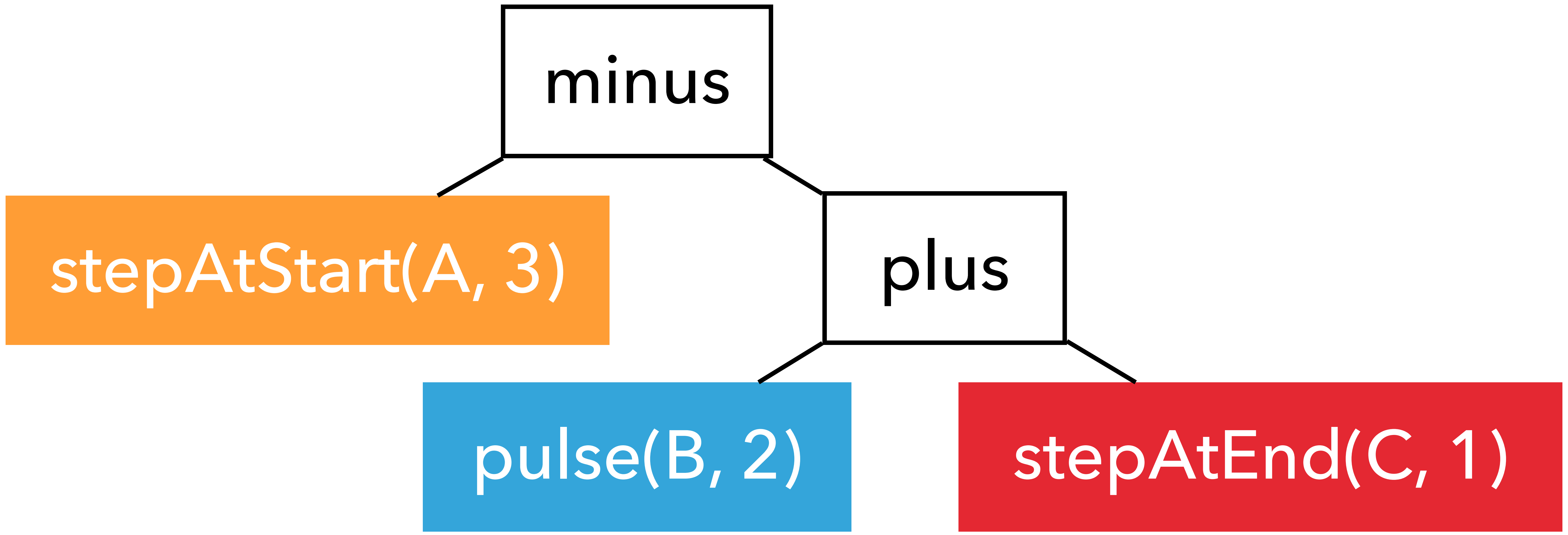
- **Plus(X, Y)** : Adds X and Y
- **Minus(X, Y)** : Subtracts Y from X
- Forms an **abstract syntax tree** (AST)

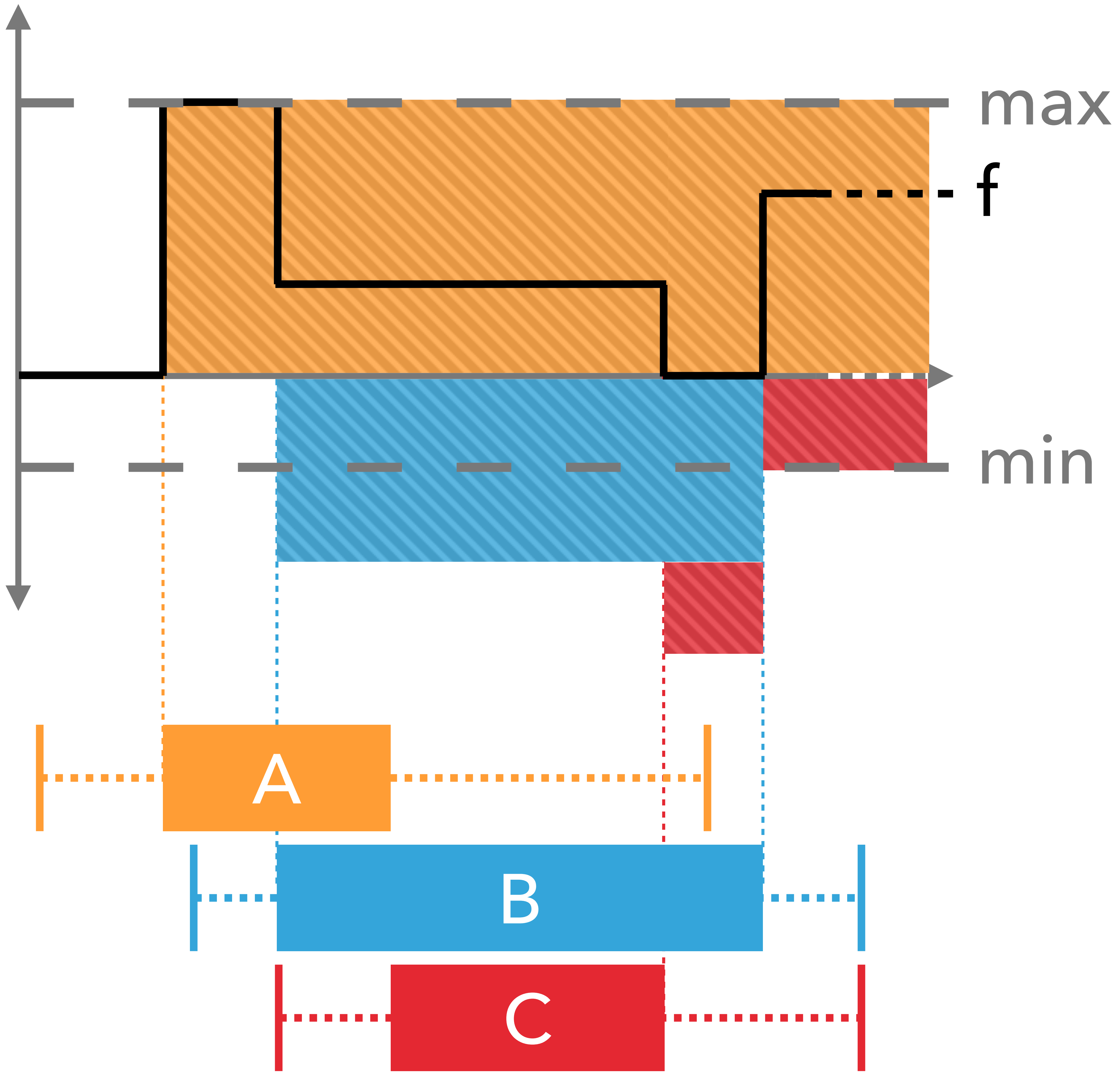
CONSTRAINTS ON CUMULATIVE FUNCTIONS





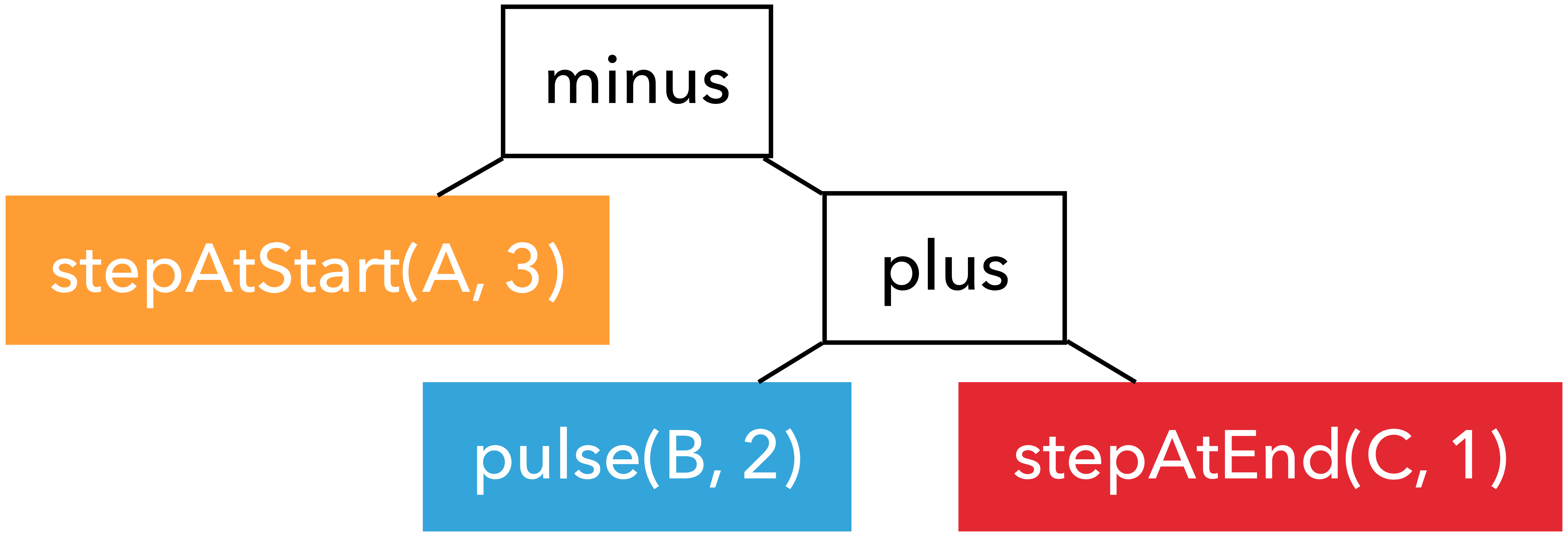
CONSTRAINTS ON CUMULATIVE FUNCTIONS

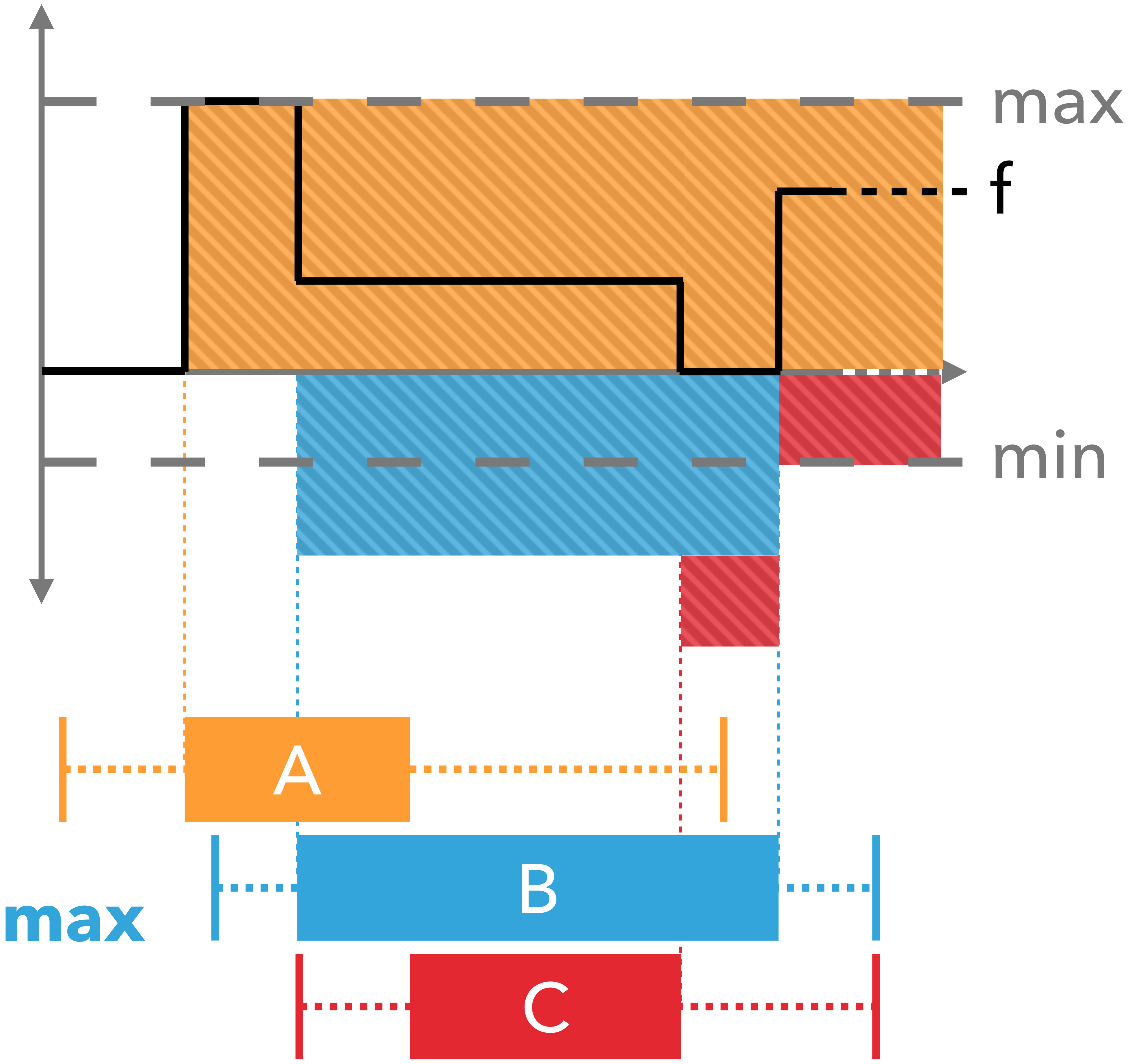




- **alwaysIn(f, min, max) : min ≤ f ≤ max**

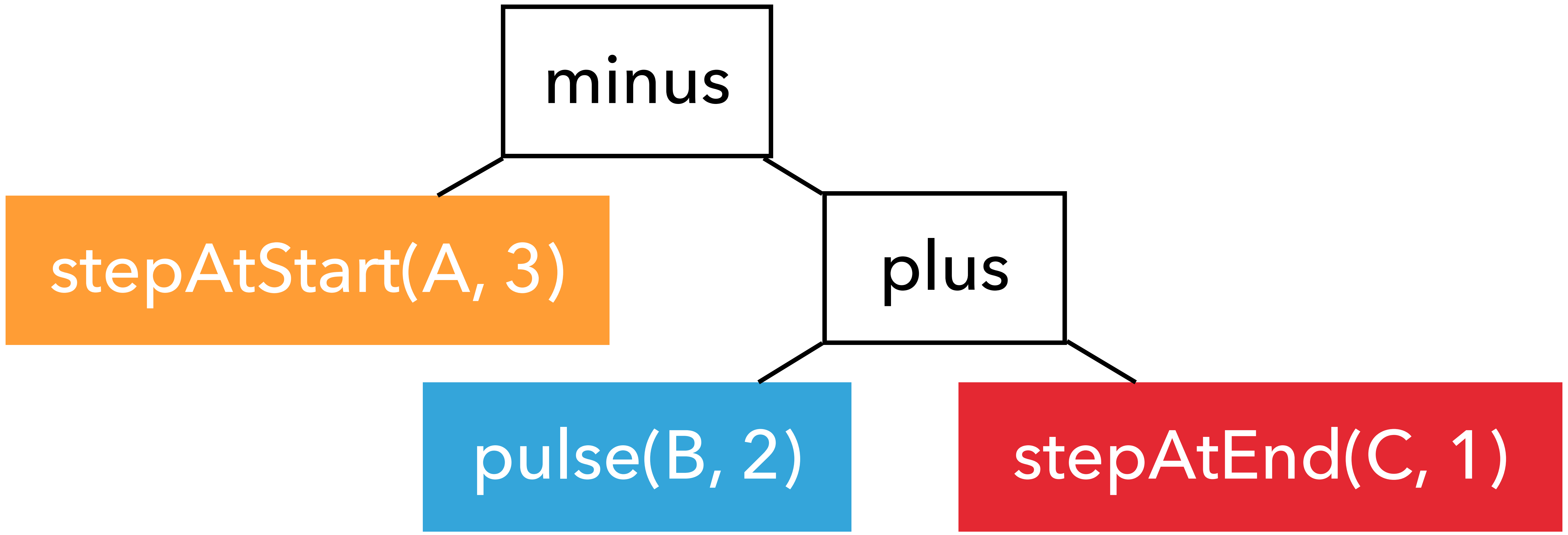
CONSTRAINTS ON CUMULATIVE FUNCTIONS

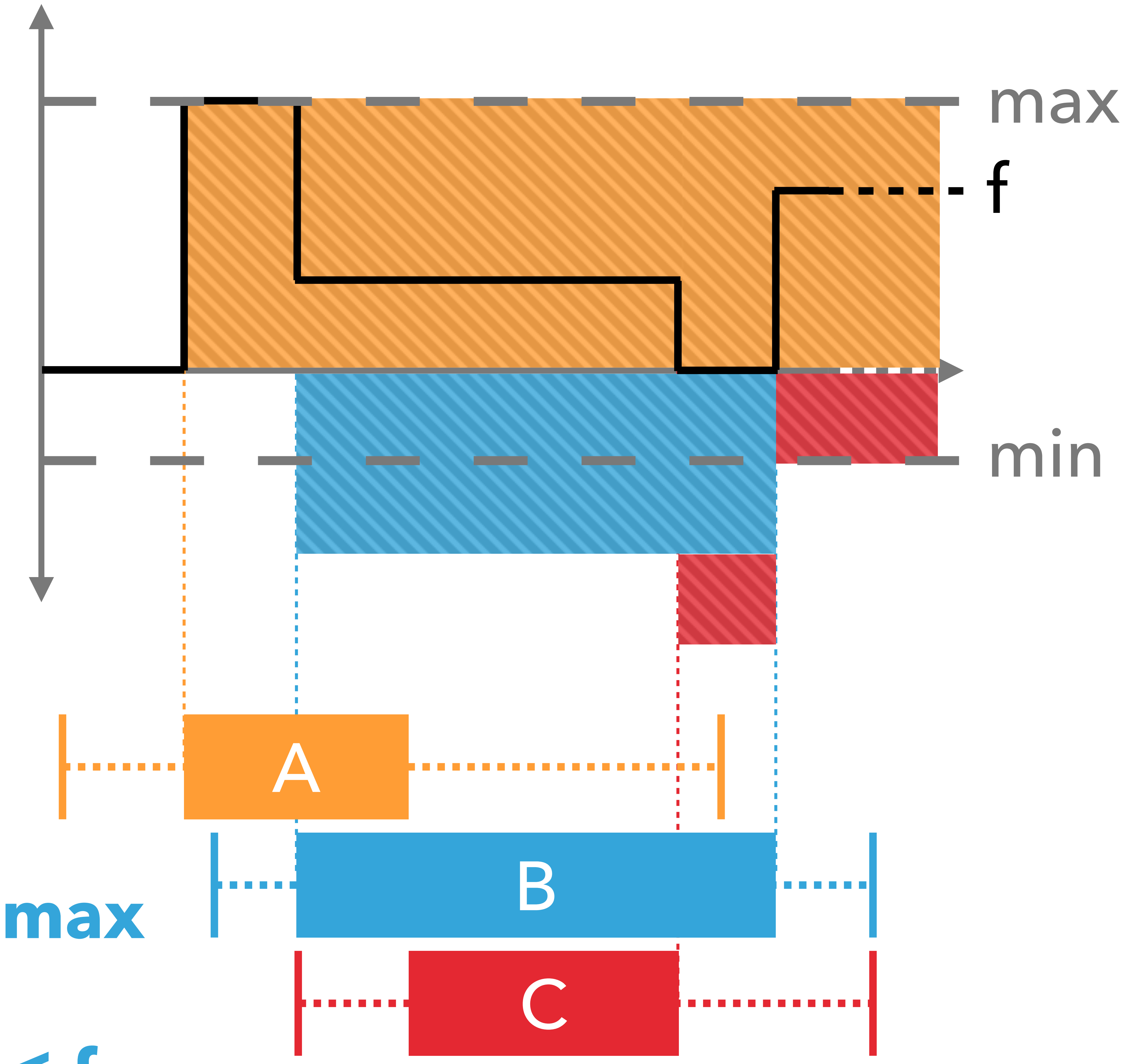




- **alwaysIn(f, min, max)** : **min ≤ f ≤ max**
- **lesserOrEqual(f, max)** = alwaysIn(f, -∞, max) : **f ≤ max**

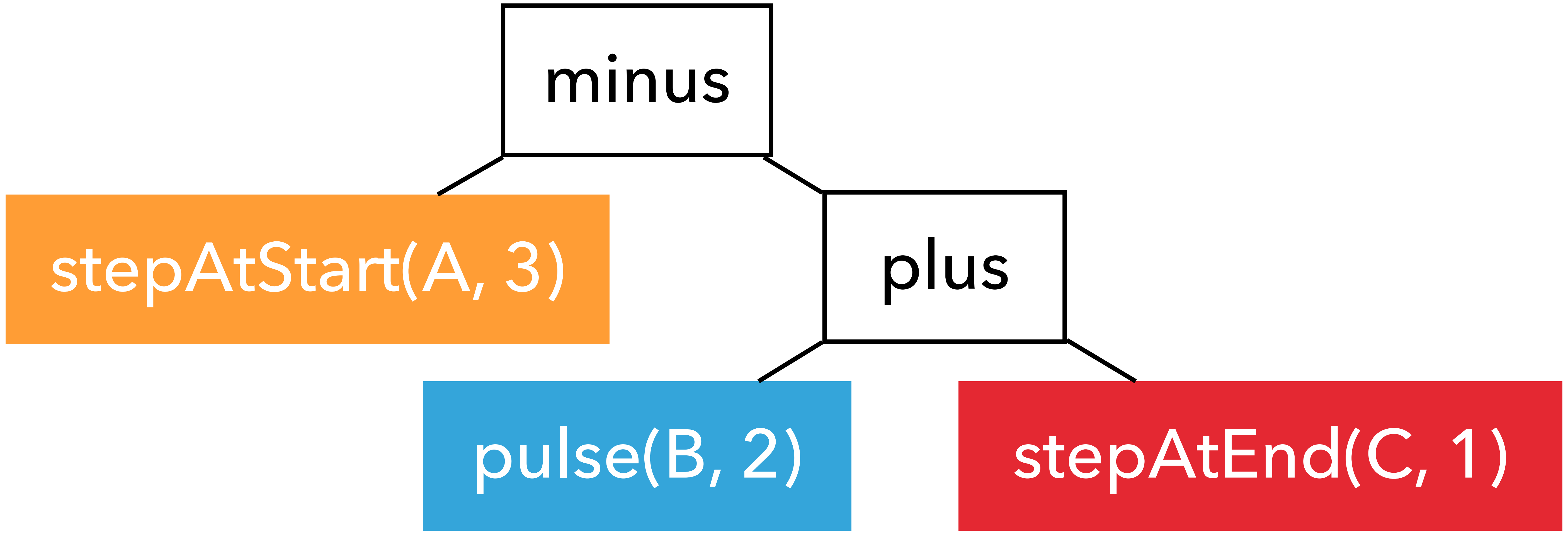
CONSTRAINTS ON CUMULATIVE FUNCTIONS

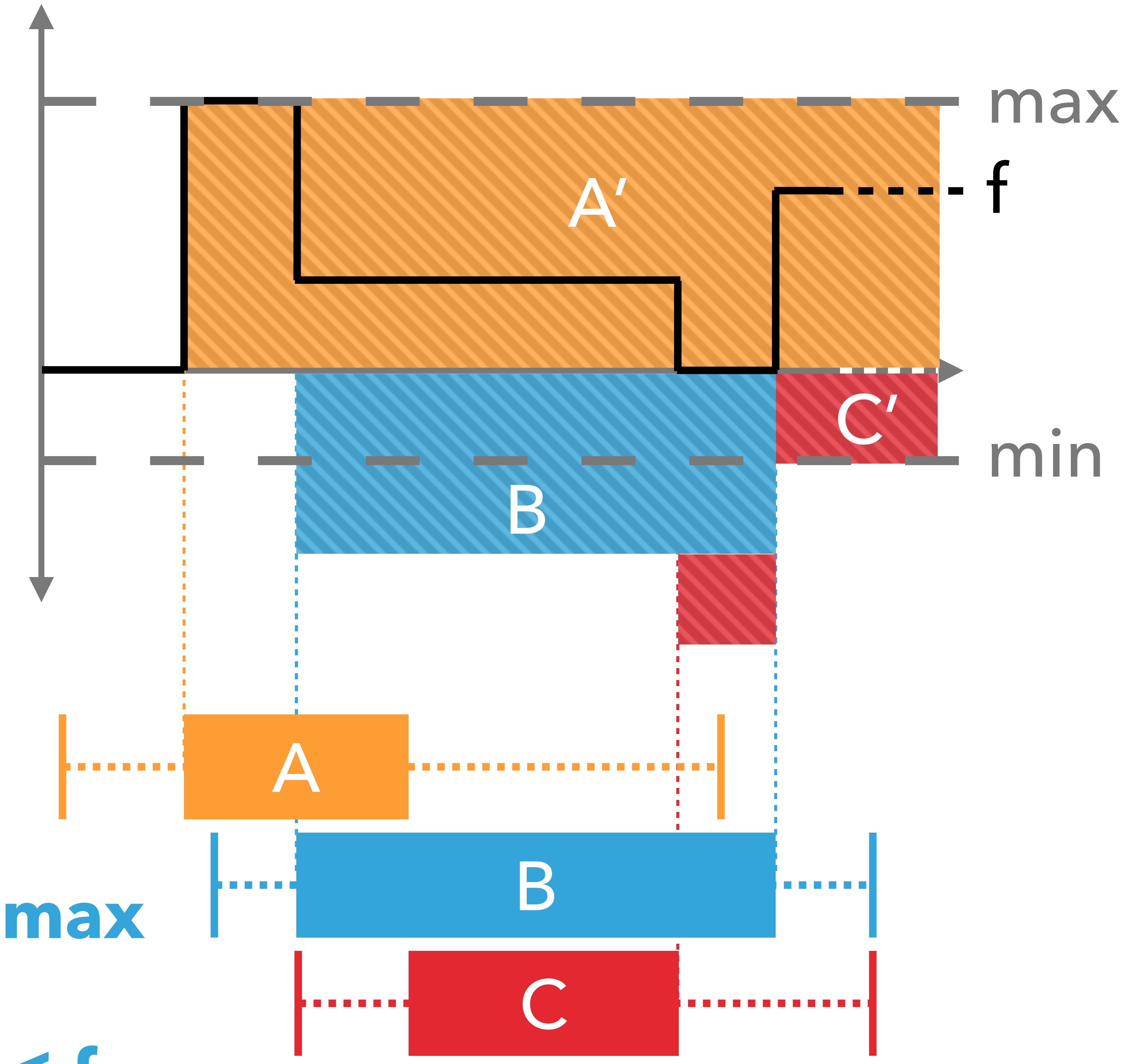




- **alwaysIn(f, min, max)** : **min ≤ f ≤ max**
- **lesserOrEqual(f, max)** = alwaysIn(f, -∞, max) : **f ≤ max**
- **higherOrEqual(f, min)** = alwaysIn(f, min, ∞) : **min ≤ f**

CONSTRAINTS ON CUMULATIVE FUNCTIONS





- **alwaysIn(f, min, max)** : **min ≤ f ≤ max**
- **lesserOrEqual(f, max)** = alwaysIn(f, -∞, max) : **f ≤ max**
- **higherOrEqual(f, min)** = alwaysIn(f, min, ∞) : **min ≤ f**
- Collects tasks by **traversing** the AST: {<A′, 3>, <B, -2>, <C′ , -1>}

GENERALIZED CUMULATIVE CONSTRAINT

GENERALIZED CUMULATIVE CONSTRAINT

- **GCC(T, min, max)** : ensures that **min ≤ f(T) ≤ max** at any time when at least one interval is present

GENERALIZED CUMULATIVE CONSTRAINT

- **GCC(T, min, max)** : ensures that **min ≤ f(T) ≤ max** at any time when at least one interval is present
- **T** is a set of **tasks** (each characterized by a **tuple <interval, height>**)

GENERALIZED CUMULATIVE CONSTRAINT

- **GCC(T, min, max)** : ensures that **$min \leq f(T) \leq max$** at any time when at least one interval is present
- **T** is a set of **tasks** (each characterized by a **tuple <interval, height>**)
- Intervals can be **optional** and can have **variable starts**, **durations** and **ends**

GENERALIZED CUMULATIVE CONSTRAINT

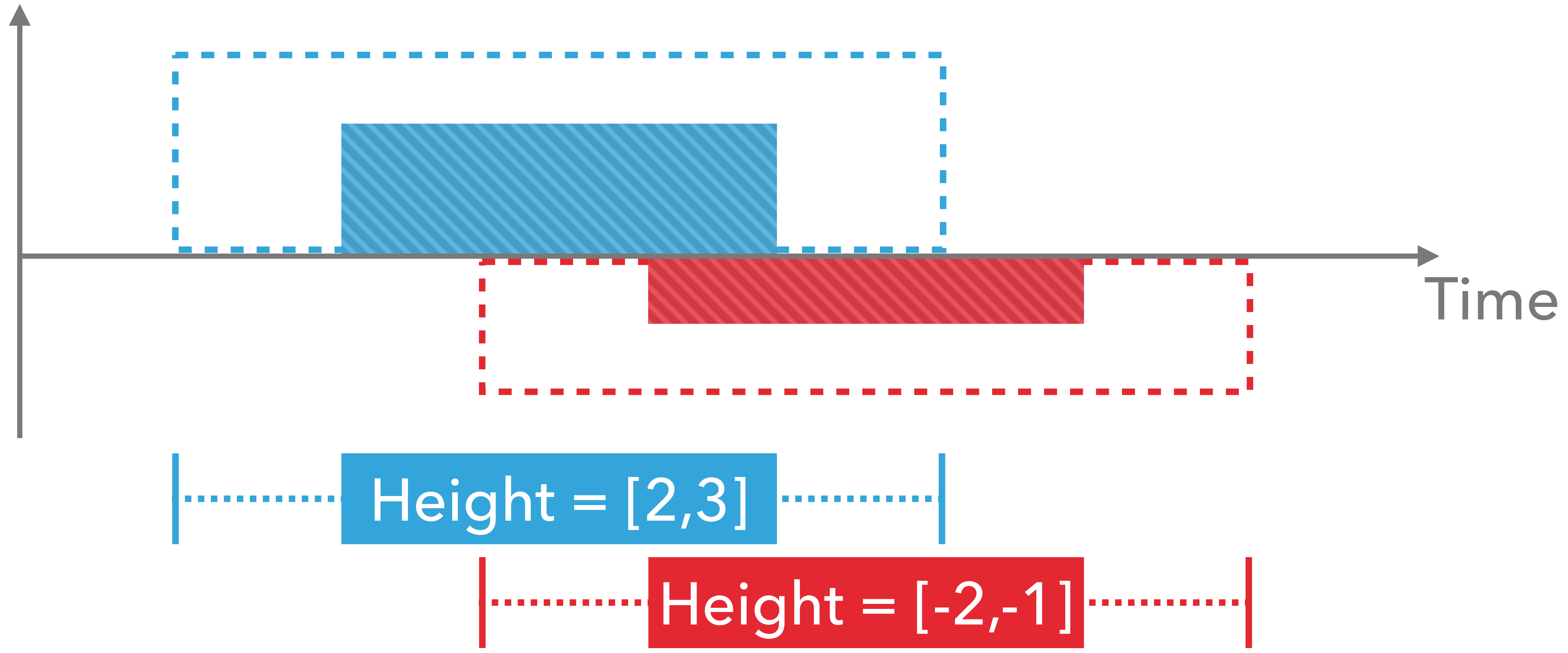
- **GCC(T, min, max)** : ensures that **min ≤ f(T) ≤ max** at any time when at least one interval is present
- **T** is a set of **tasks** (each characterized by a **tuple <interval, height>**)
- Intervals can be **optional** and can have **variable starts**, **durations** and **ends**
- **Heights** can be **variable** too

BUILDING THE PROFILES

BUILDING THE PROFILES

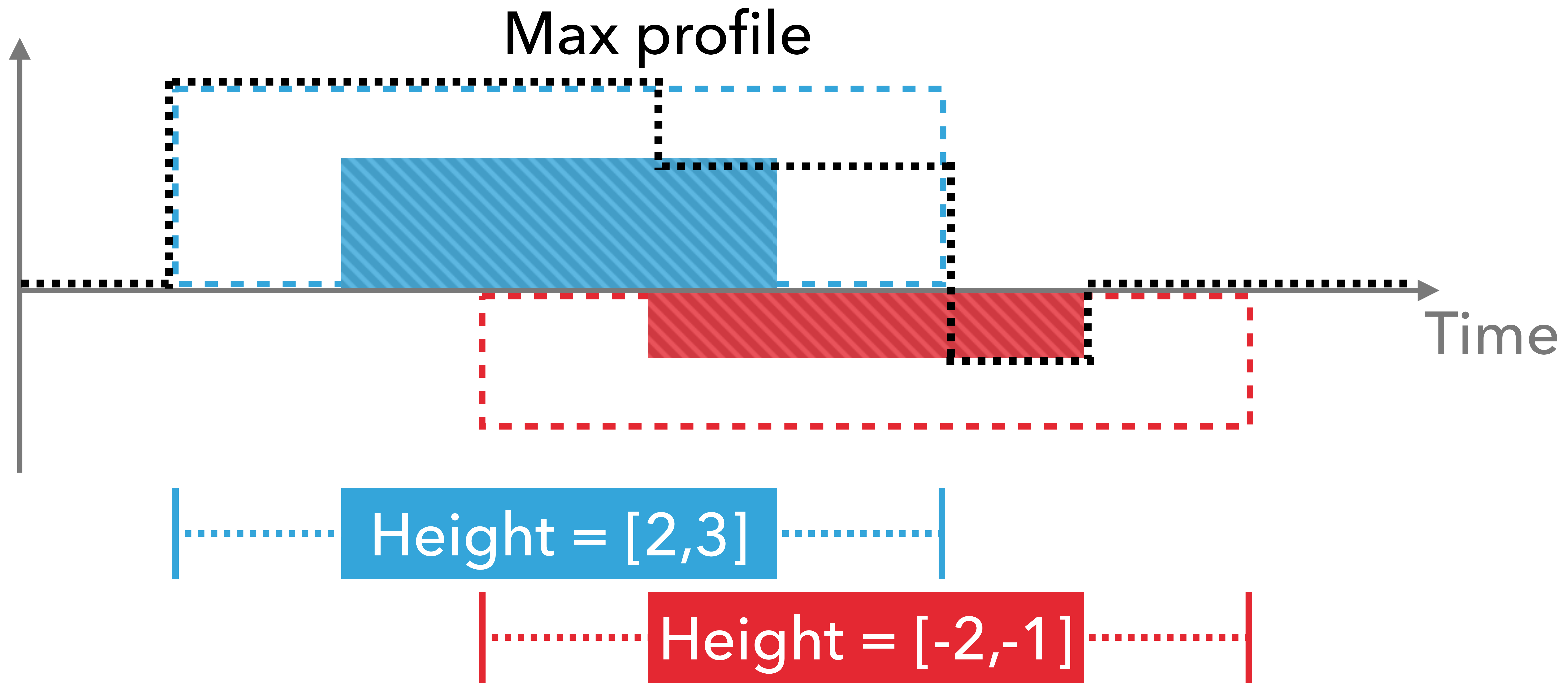
- Build **minimum** and **maximum profiles** with contributions of tasks:

BUILDING THE PROFILES



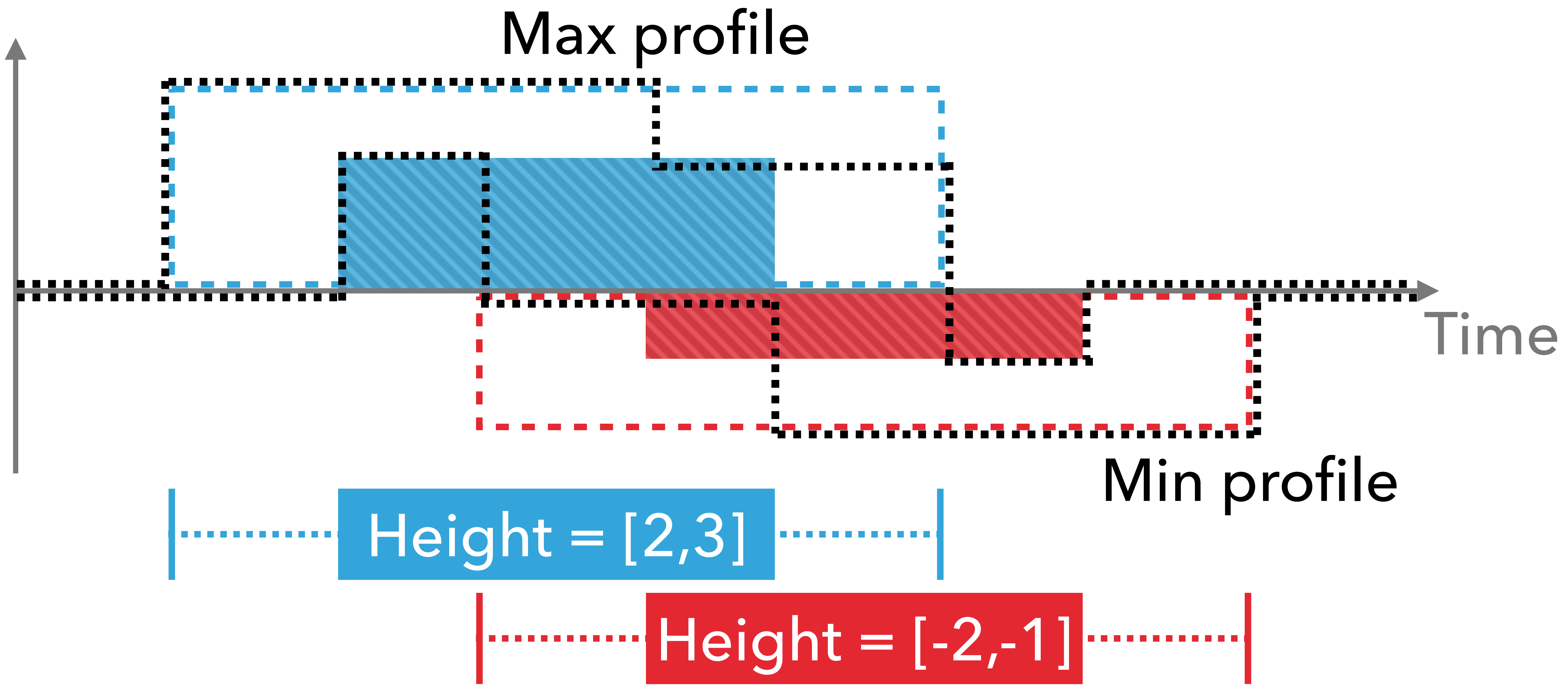
- Build **minimum** and **maximum profiles** with contributions of tasks:

BUILDING THE PROFILES



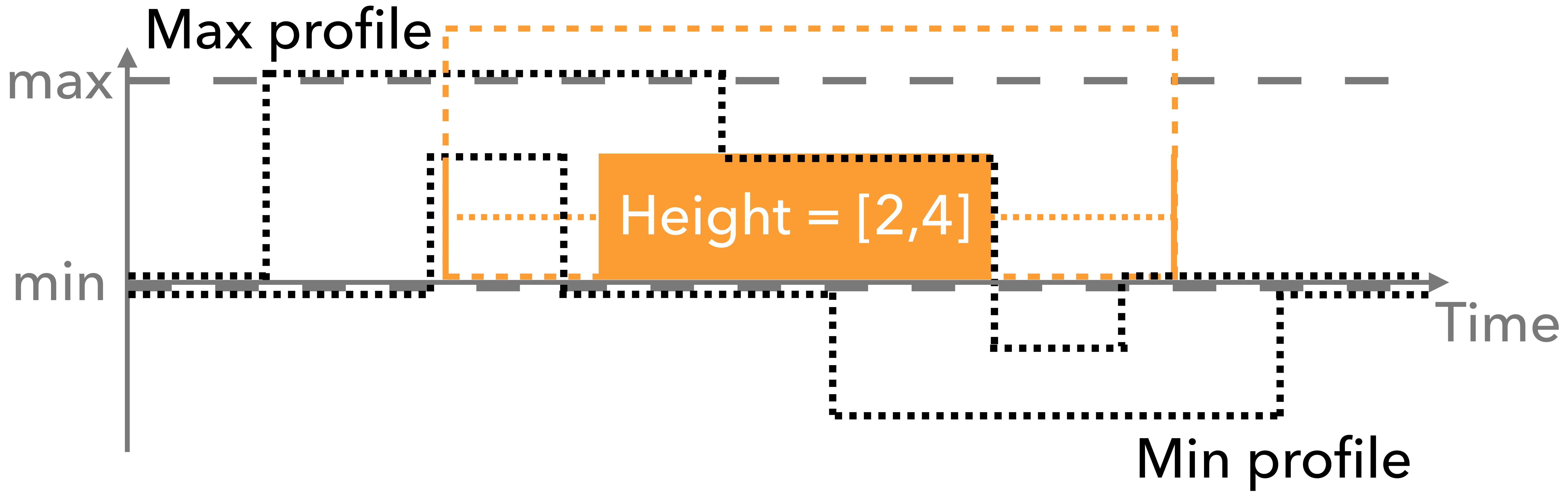
- Build **minimum** and **maximum profiles** with contributions of tasks:
  - Max profile = **max positive** contributions + **min negative** contributions

BUILDING THE PROFILES

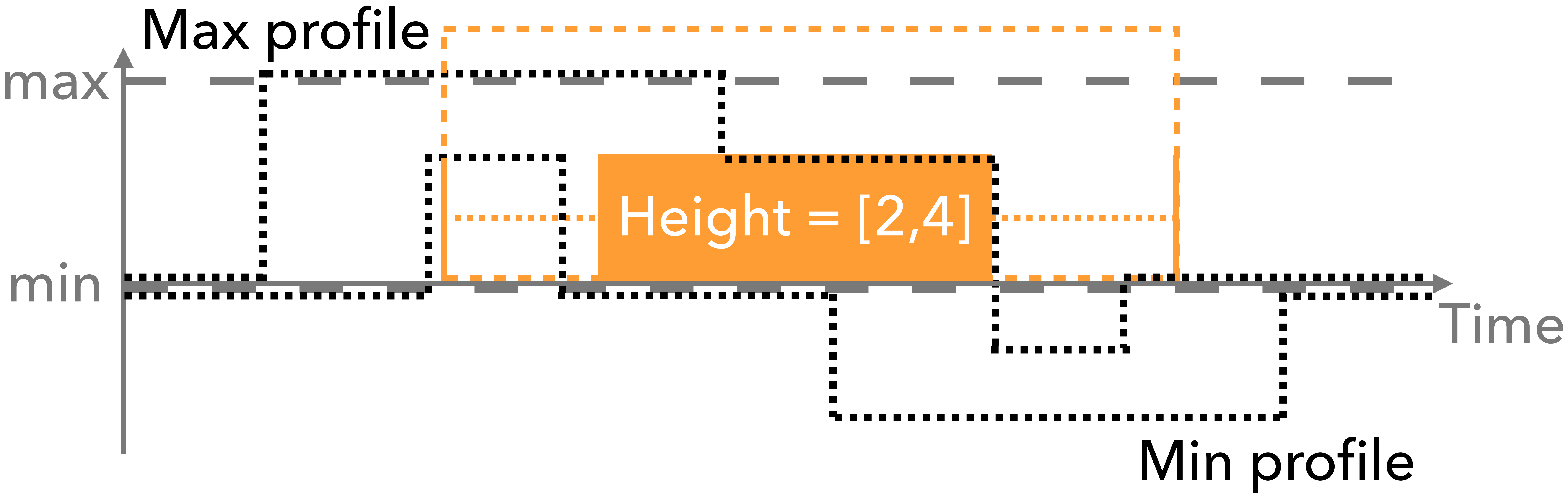


- Build **minimum** and **maximum profiles** with contributions of tasks:
  - Max profile = **max positive** contributions + **min negative** contributions
  - Min profile = **max negative** contributions + **min positive** contributions

TIMETABLING

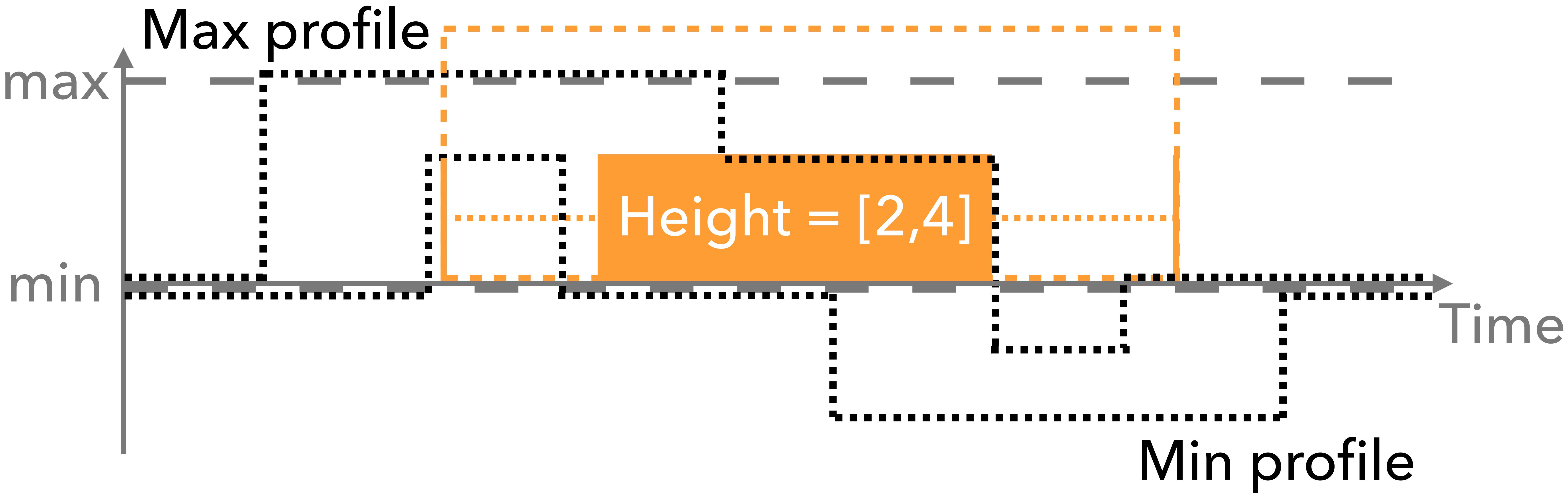


TIMETABLING



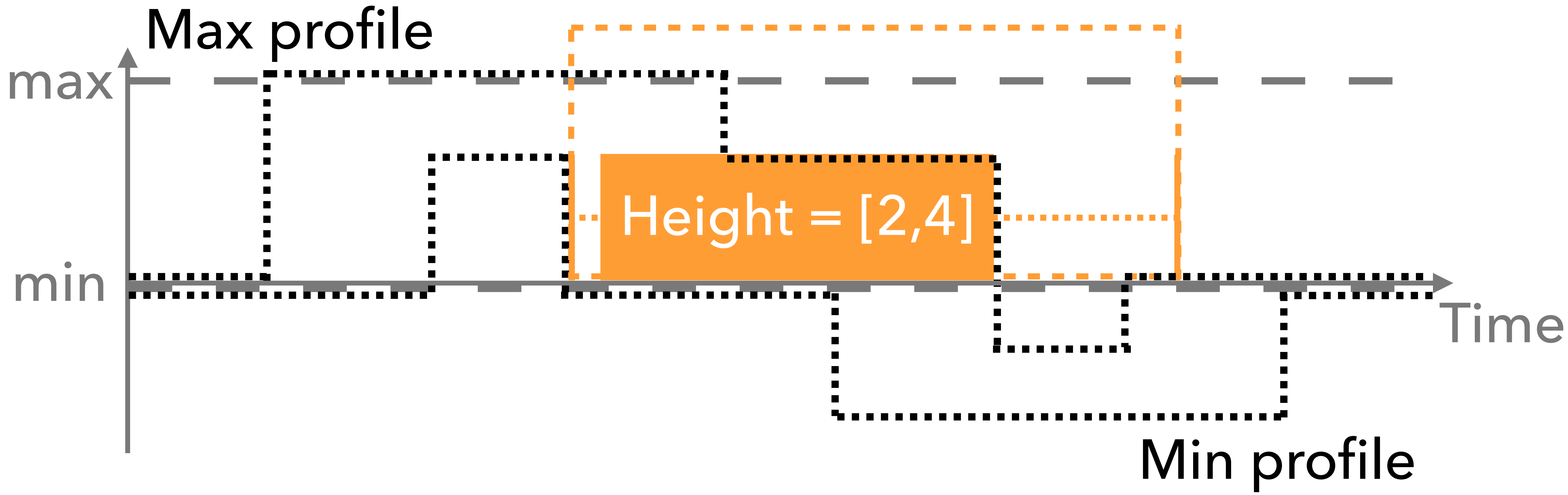
- Compare **each task** during its whole time window with **min** and **max profiles**

TIMETABLING



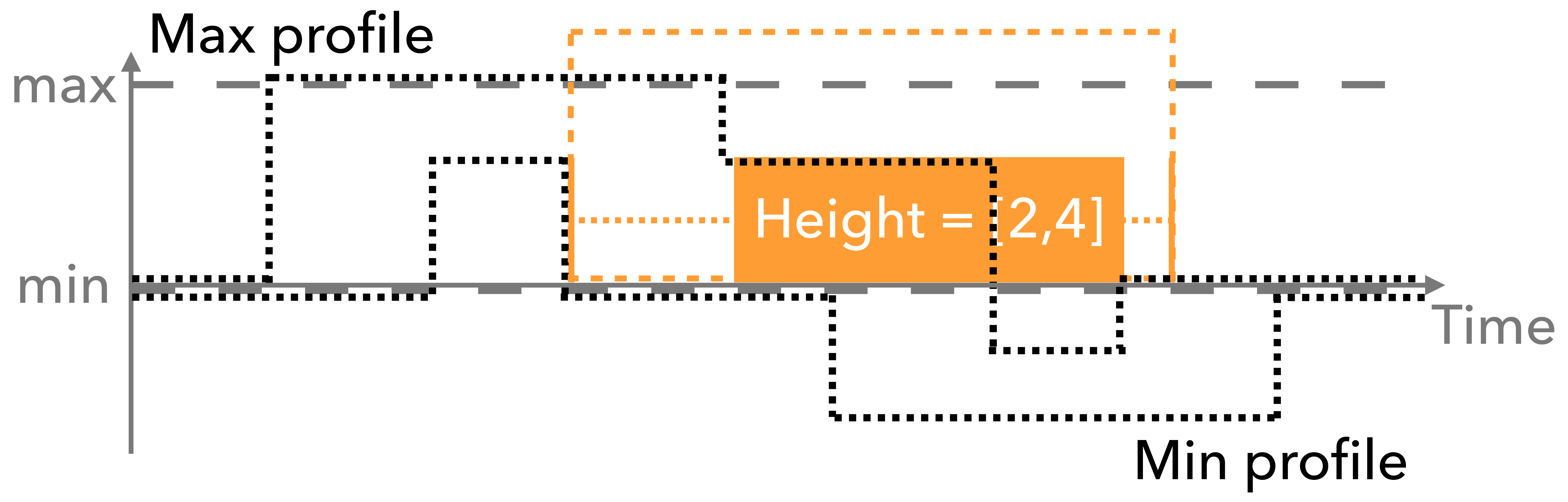
- Compare **each task** during its whole time window with **min** and **max profiles**
- 4 propagation rules:

TIMETABLING



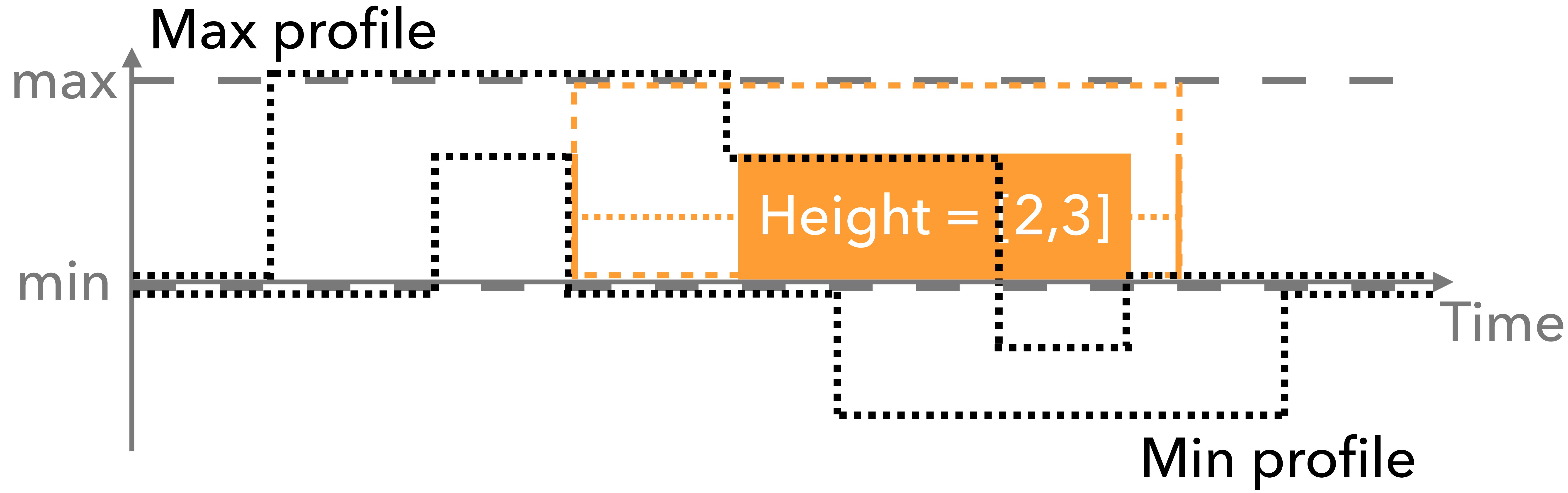
- Compare **each task** during its whole time window with **min** and **max profiles**
- 4 propagation rules:
  - **Forbid** : Task is not feasible at some time ⇒ adjust time window

TIMETABLING



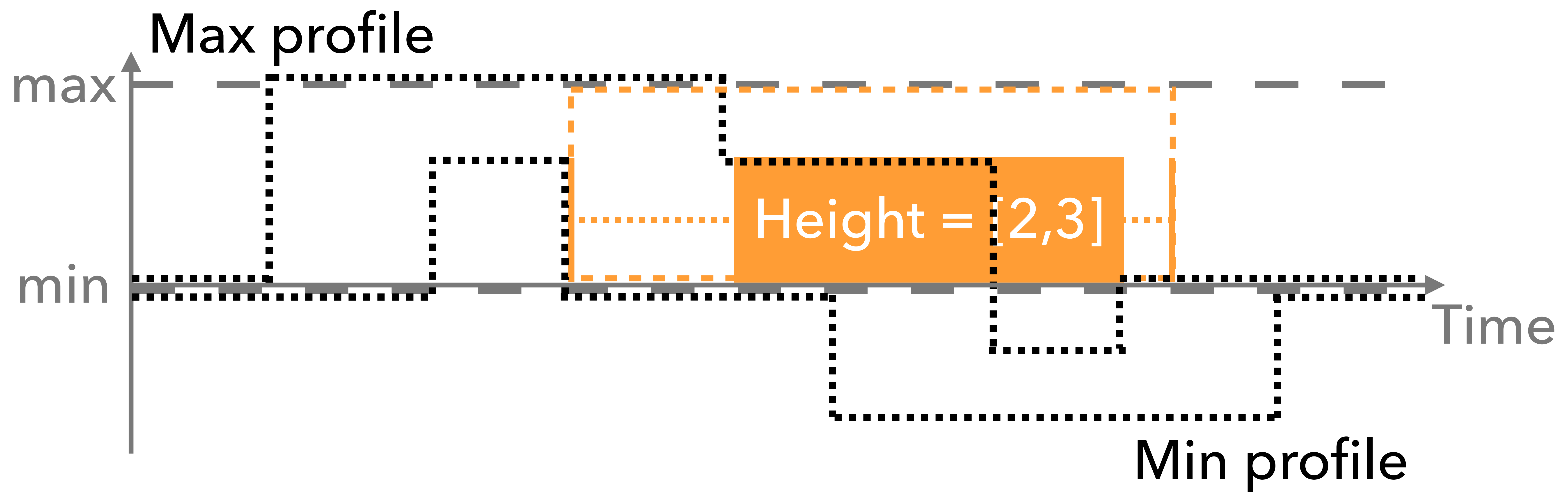
- Compare **each task** during its whole time window with **min** and **max profiles**
- 4 propagation rules:
  - **Forbid** : Task is not feasible at some time ⇒ adjust time window
  - **Mandatory**: Task is necessary to satisfy capacity ⇒ set task present & cover deficit

TIMETABLING



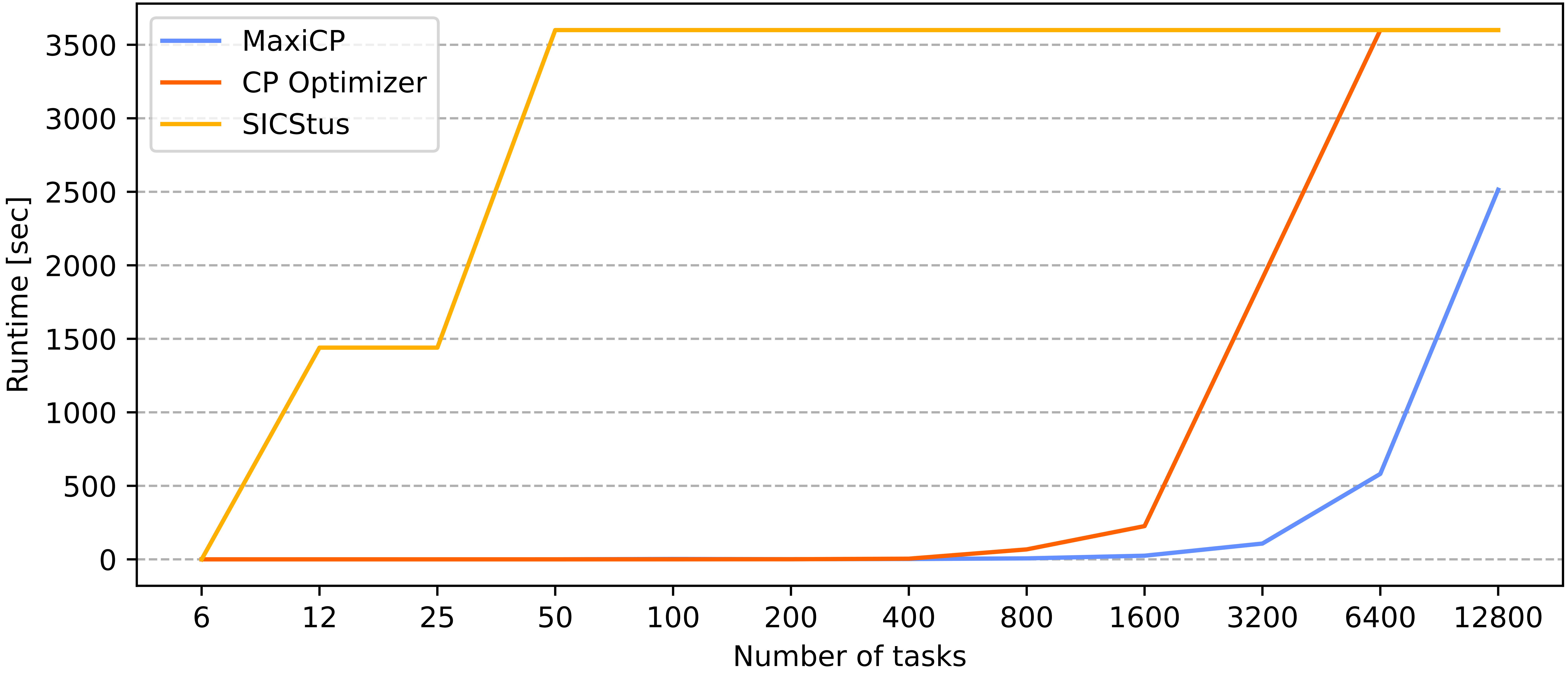
- Compare **each task** during its whole time window with **min** and **max profiles**
- 4 propagation rules:
  - **Forbid** : Task is not feasible at some time ⇒ adjust time window
  - **Mandatory**: Task is necessary to satisfy capacity ⇒ set task present & cover deficit
  - **Height**: Adjust min & max height of task

TIMETABLING



- Compare **each task** during its whole time window with **min** and **max profiles**
- 4 propagation rules:
  - **Forbid** : Task is not feasible at some time ⇒ adjust time window
  - **Mandatory**: Task is necessary to satisfy capacity ⇒ set task present & cover deficit
  - **Height**: Adjust min & max height of task
  - **Length**: Adjust max length of task

PERFORMANCES



- Maximum Energy Scheduling Problem
- Average time to 1st solution with greedy search, timeout at 3600s

CONCLUSION

CONCLUSION

- **Cumulative functions** are a powerful modeling tool for scheduling problems with **resources** constraints

CONCLUSION

- **Cumulative functions** are a powerful modeling tool for scheduling problems with **resources** constraints
- Can be easily implemented with an **AST**

CONCLUSION

- **Cumulative functions** are a powerful modeling tool for scheduling problems with **resources** constraints
- Can be easily implemented with an **AST**
- A **generalized cumulative constraint** can be used to model **capacity ranges**

CONCLUSION

- **Cumulative functions** are a powerful modeling tool for scheduling problems with **resources** constraints
- Can be easily implemented with an **AST**
- A **generalized cumulative constraint** can be used to model **capacity ranges**

THANKS FOR YOUR ATTENTION!