# Einführung in Scheduling

Dr. Julien Bidot

Sommersemester 2008

Institut für Künstliche Intelligenz

# Inhalt

# I. Definition und Formulierung des Scheduling-Problems

1. Was ist Scheduling?
2. Einige Anwendungsgebiete für Scheduling
3. Constraints in Scheduling
4. Mathematische Formulierung

# Scheduling?

## Definition

Das Scheduling-Problem besteht darin, eine Menge von Tasks zeitlich so zu organisieren, dass temporale und Ressourcen-Constraints erfüllt sind.

Eine Menge von **Tasks**

Eine Menge von **Ressourcen**

**Zeitliche Constraints** und **Ressourcen-Constraints**

Ein oder mehrere **Optimierungskriterien**

- ➔ Bestimmung der Ausführungsdaten und/oder die Ressourcenzuweisungen für die Tasks

# Scheduling vs. Planung

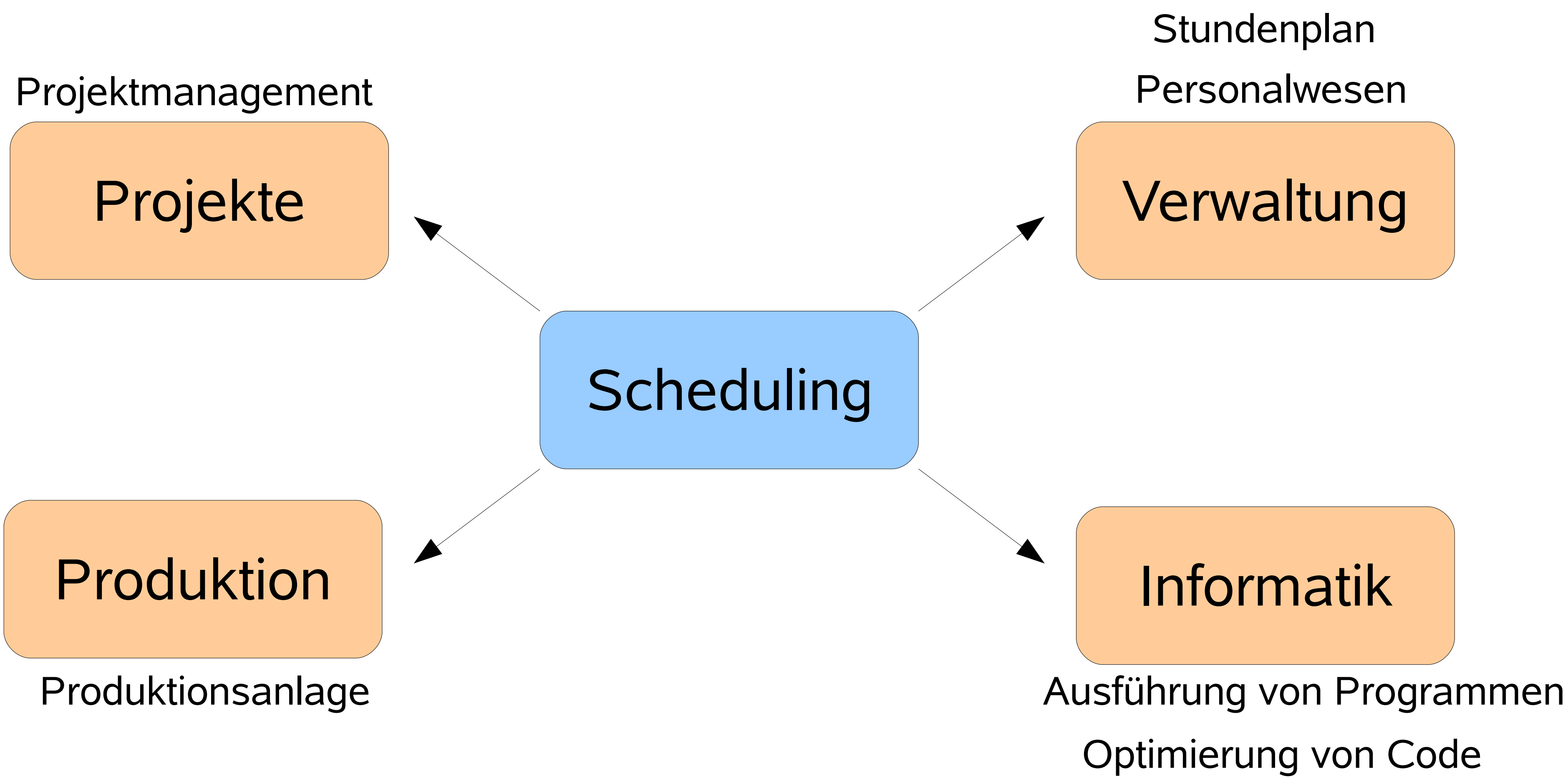
- Scheduling:
  - hohe Komplexität
  - **numerische** Berechnung
  - Constraints erfüllt + Optimierung
  - operationale Tasks **schon** bekannt

- Planung:
  - sehr hohe Komplexität
  - **symbolische** Berechnung
  - Constraints erfüllt
  - operationale Tasks **nicht** bekannt

# Begriffe

- Scheduling
- Sequenzierung (Produktion)
- Ablaufplanung
- Planung
- Zeitplanung
- Terminierung

# Anwendungsgebiete



# Mathematische Formulierung

- $n$ Tasks ausführen + 2 fiktive Tasks 0 und $n$+1 mit Zeitdauer 0
- Gesucht $(t_0, t_1, \ldots, t_n, t_{n+1})$, so dass $f(t_0, t_1, \ldots, t_n, t_{n+1})$ minimiert ist mit folgenden zeitlichen Constraints:
  - $t_j - t_i \geq a_{ij}$ Potenzial-Constraints
  - $t_0, t_1, \ldots, t_n, t_{n+1} \geq 0$ positive Constraints
  - $t_j - t_i \geq p_i$ oder $t_i - t_j \geq p_j$ disjunktive Constraints
  - idem kumulative Constraints
- $f$ hängt von Anfangszeitpunkten von Tasks ab.

# II. Projektplanung

5. Kernproblem des Scheduling: Ressourcen mit unendlichen Kapazitäten
6. Im Allgemeinen: Ressourcen mit endlichen Kapazitäten

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- Kontext:

  Ein Projekt besteht aus einer Menge von $n$ Tasks verbunden mit **zeitlichen Constraints**

- Ziel:
  - Berechne die minimale Zeitdauer des Projektes, angenommen, die Ressourcen haben **unendliche Kapazitäten** $\rightarrow$ minimiere $(t_{n+1} - t_0)$, so dass zeitliche Constraints erfüllt sind
  - Bestimme die frühestmöglichen und spätest zulässigen Anfangszeitpunkte von allen Tasks
  - Bestimme die kritischen Tasks

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- Mathematische Formulierung
  - Bestimme $(t_0, t_1, \ldots, t_n, t_{n+1})$ so dass $(t_{n+1} - t_0)$ minimiert ist mit folgenden zeitlichen Constraints:
    - $t_j - t_i \geq a_{ij}$
    - $t_0, t_1, \ldots, t_n, t_{n+1} \geq 0$

  Das ist ein lineares Gleichungssystem.

Aber es gibt etwas einfacheres!

➔ Metra-Potenzial-Methode und PERT Methode

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- MPM: Modellierung mit einem Netzplan

  Wir verwenden einen gerichteten Graph $G=(X,U)$

  - $X$: Menge von Punkten (Knoten oder Ecken)
  - $U$: Menge von gerichten Kanten

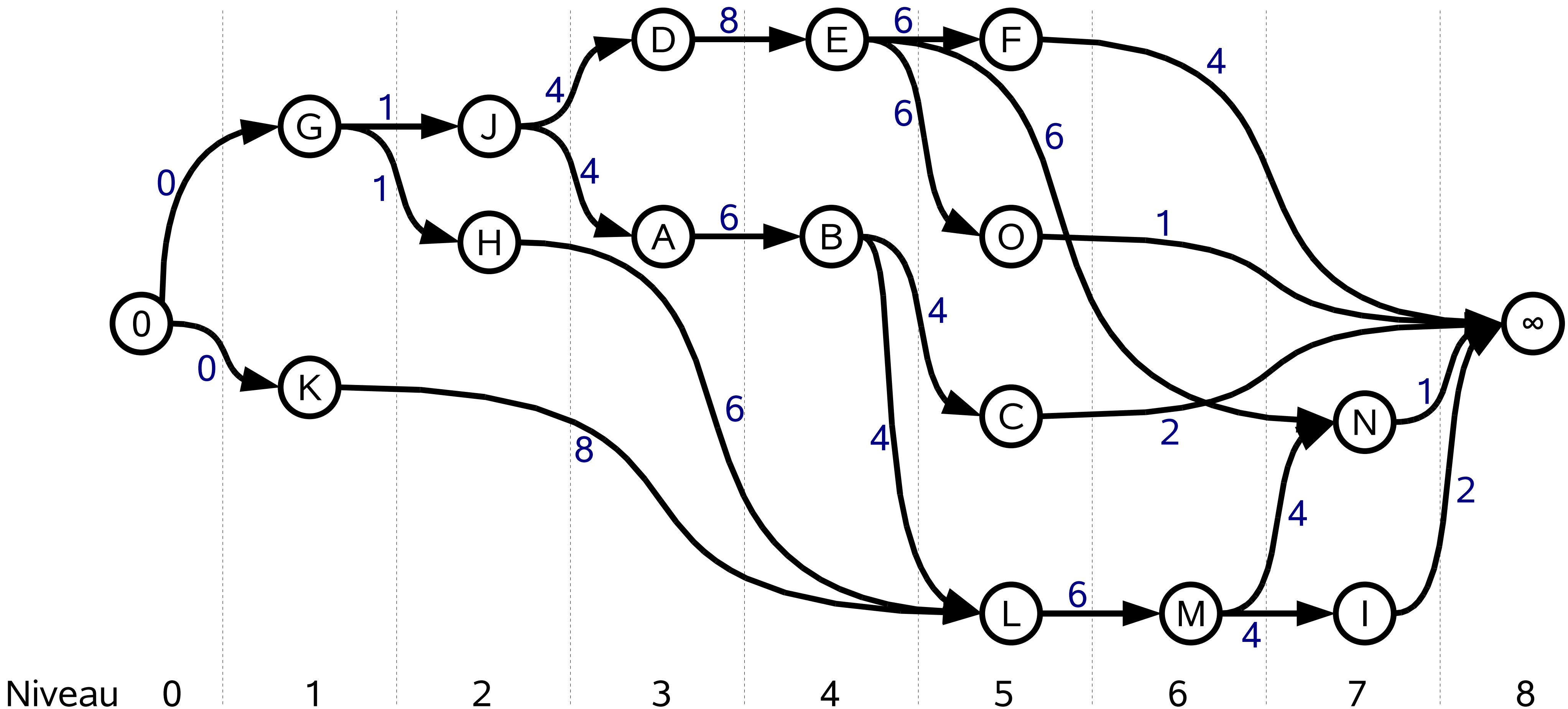
$$U=\{(i,j)\in X\times X, \exists t_j - t_i \geq a_{ij}\}$$

Der Wert einer Kante $(i,j)$ ist $v_{(ij)}=a_{(ij)}$

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

| Task | Beschreibung | Dauer | Vorgänger |
|---|---|---|---|
| Anfang | Projektanfang | 0 | - |
| A | Ausbau der alten Fliese | 6 | J |
| B | Aufbau der neuen Fliese | 4 | A |
| C | Verbindungsstellen für die Fliese | 2 | B |
| D | Ausbau der alten Tapete | 8 | J |
| E | Aufbau der Keramikfliese für Wand | 6 | D |
| F | Aufbau der neuen Tapete | 4 | E |
| G | Ausbau des alten Waschbecken | 1 | Anfang |
| H | Verschiebung von Waschbecken | 6 | G |
| I | Aufbau des neuen Waschbecken | 2 | M |
| J | Ausbau der alten Möbel | 4 | G |
| K | Zusammenbau der Kisten und Schubladen | 8 | Anfang |
| L | Aufbau der tiefliegenden Möbel | 6 | B,J,H,K |
| M | Aufbau der Arbeitsfläche | 4 | L |
| N | Abdichtung der Arbeitsfläche | 1 | E,M |
| O | Aufbau der hochliegenden Möbel | 1 | E,J |
| Ende | Projektende | 0 | C,F,I,N,O |

# Projektplanung – Ressourcen mit unendlichen Kapazitäten



# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- **Suche nach einem ausführbaren Ablaufplan**
- Theorem:

  Ein ausführbare Ablaufplan existiert gdw. es keinen streng positiven Zyklus in $G$ gibt.

  Wenn die Bedingung erfüllt ist gibt es im Allgemeinen mehrere Ablaufpläne mit einer minimalen Dauer.

  Zwei Sonderfälle:

  - Frühestmöglicher Ablaufplan
  - Spätest zulässiger Ablaufplan (mit einer Frist für das Ende des Projektes)

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- **Frühestmögliche Anfangszeitpunkte**
- Definition:

Der frühestmögliche Anfangszeitpunkt $\underline{t}_i$ eines Tasks $i$ ist gleich der Länge des kürzesten Pfades vom Task Anfang (oder 0) bis $i$.

- Theorem:

Wenn der Graph keinen Zyklus hat, dann:

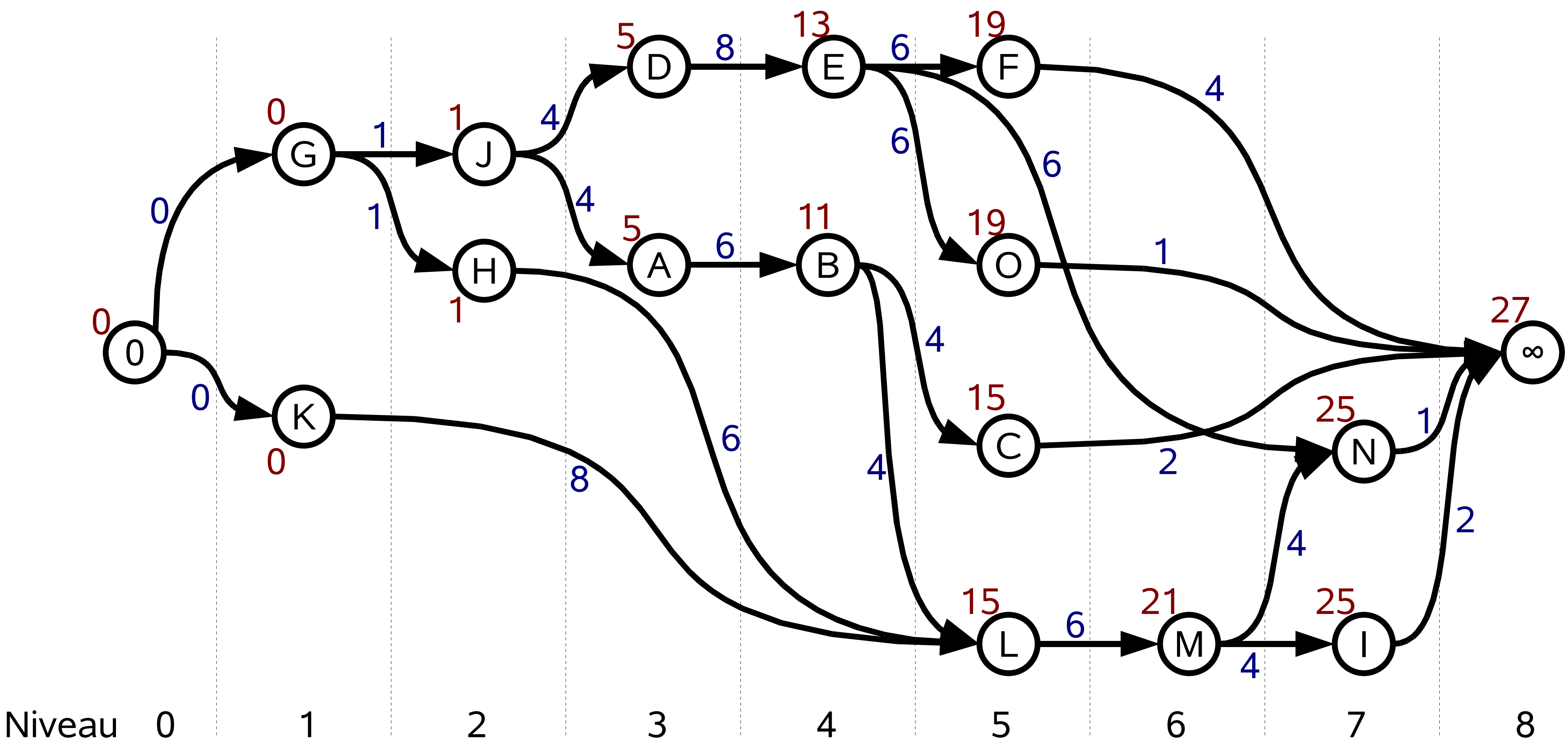
$$\underline{t}_0 = 0 \qquad \text{und} \qquad \underline{t}_i = \max_{j \in \Gamma^{-1}(i)} \left( \underline{t}_j + v_{ji} \right)$$

$v_{ji}$ ist der Wert der Kante $(j, i)$ (z. B. die Dauer von $j$)

$\Gamma^{-1}(i)$ ist die Menge der Vorgängern von $i$.

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

## Frühestmögliche Anfangszeitpunkte



# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- **Spätest zulässige Anfangszeitpunkte**

  Wir möchten das Projekt spätestens an der Frist $D \geq \underline{t}_{n+1}$ abschliessen.

- Definition:

  Der spätest zulässige Anfangszeitpunkt $\bar{t}_i$ eines Tasks $i$ ist das Datum an dem wir $i$ ausführen können ohne das Projekt zu verzögern (Ende des Projektes später als $D$).

- Theorem:

$$\bar{t}_{n+1} = D$$

$\bar{t}_i = \bar{t}_{n+1}$ ist der Wert eines der längsten Pfade zwischen $i$ und $n$+1.

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- **Spätest zulässige Anfangszeitpunkte**

- Theorem:

Wenn der Graph keinen Zyklus hat, dann:

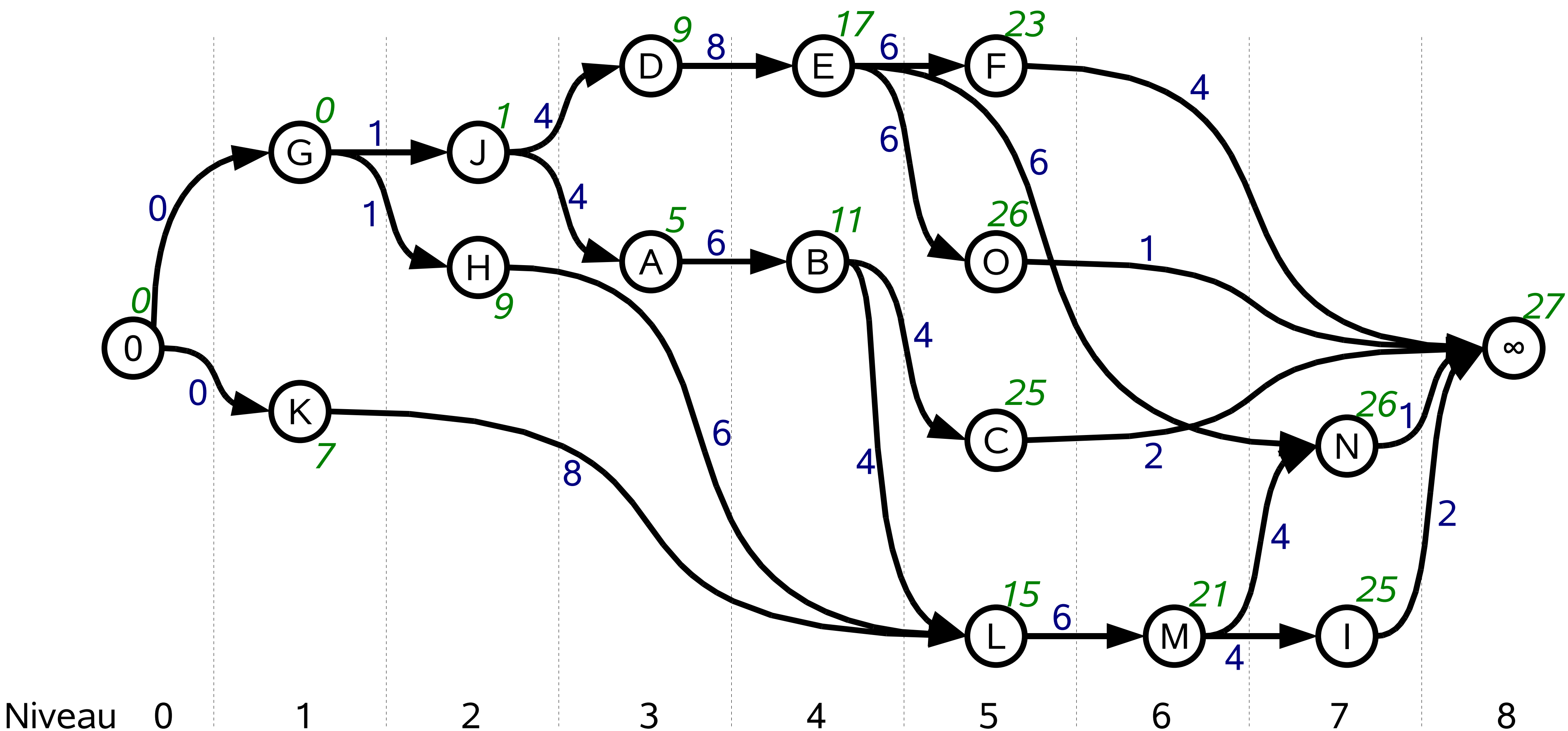
$$\bar{t}_{\infty} = D \quad \text{und} \quad \bar{t}_i = \min_{j \in \Gamma(i)} (\bar{t}_j - v_{ij})$$

$v_{ij}$ ist der Wert der Kante $(i, j)$ (z. B. die Dauer von $i$)

$\Gamma(i)$ ist die Menge der Nachfolgern von $i$.

# Projektplanung – Ressourcen mit unendlichen Kapazitäten

## Spätest zulässige Anfangszeitpunkte



# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- **Gesamte Pufferzeiten und kritischer Pfad**
- Definition

Die *gesamte Pufferzeit* eines Tasks $i$ ist die gesamte Verzögerung von $i$ ohne das Projekt zu verzögern.

$$GP_i = \bar{t}_i - \underline{t}_i$$

- Definition

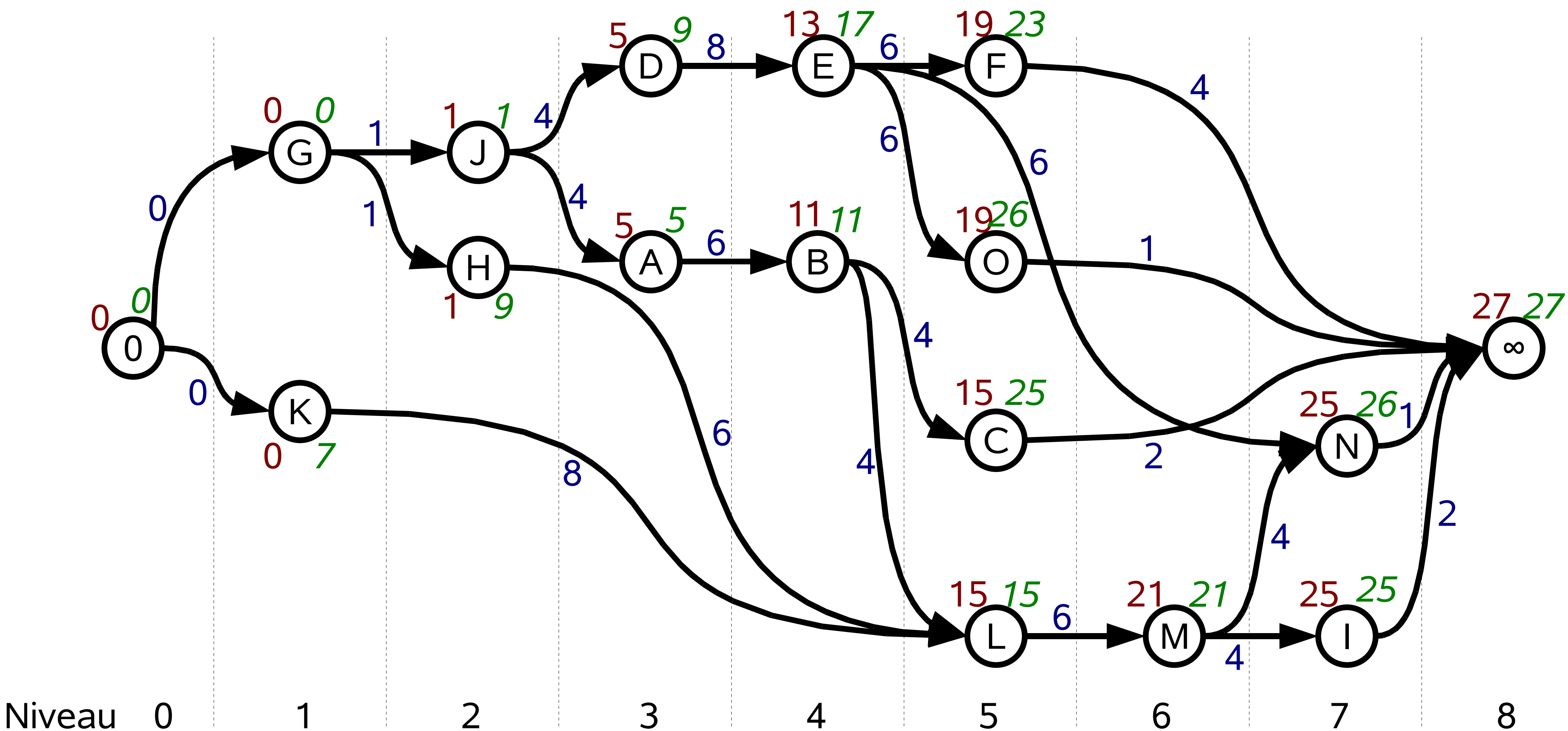
Ein *kritisches Task* hat eine gesamte Pufferzeit von null.

→ Eine Verzögerung eines solchen Tasks verursacht eine Verzögerung des Projektes.

Ein *Pfad ist kritisch* gdw. er nur kritische Tasks von Anfang bis Ende verbindet.

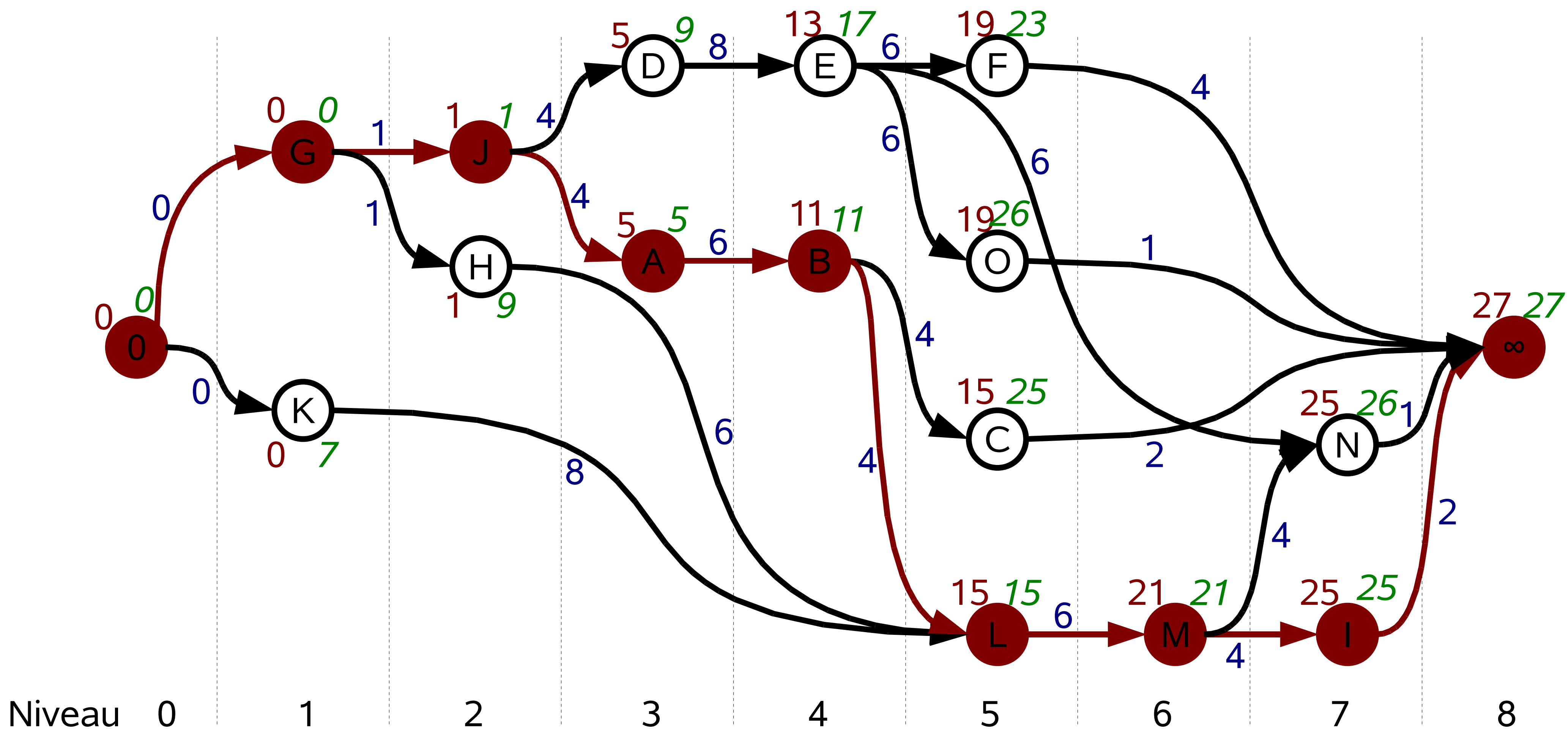
# Projektplanung – Ressourcen mit unendlichen Kapazitäten

## Gesamte Pufferzeiten und kritischer Pfad



# Projektplanung – Ressourcen mit unendlichen Kapazitäten

## Gesamte Pufferzeiten und kritischer Pfad



# Projektplanung – Ressourcen mit unendlichen Kapazitäten

- **Freie Pufferzeiten**
- Definition

  Die *freie Pufferzeit* eines Tasks $i$ ist die gesamte Verzögerung von $i$ ohne ein anderes Task zu verzögern (im Vergleich zu frühestmöglichen Anfangszeitpunkten).

- Theorem

$$FP_i = \min_{j \in \Gamma(i)} \left( \underline{t}_j - \underline{t}_i - v_{ij} \right)$$

# Projektplanung – Ressourcen mit endlichen Kapazitäten

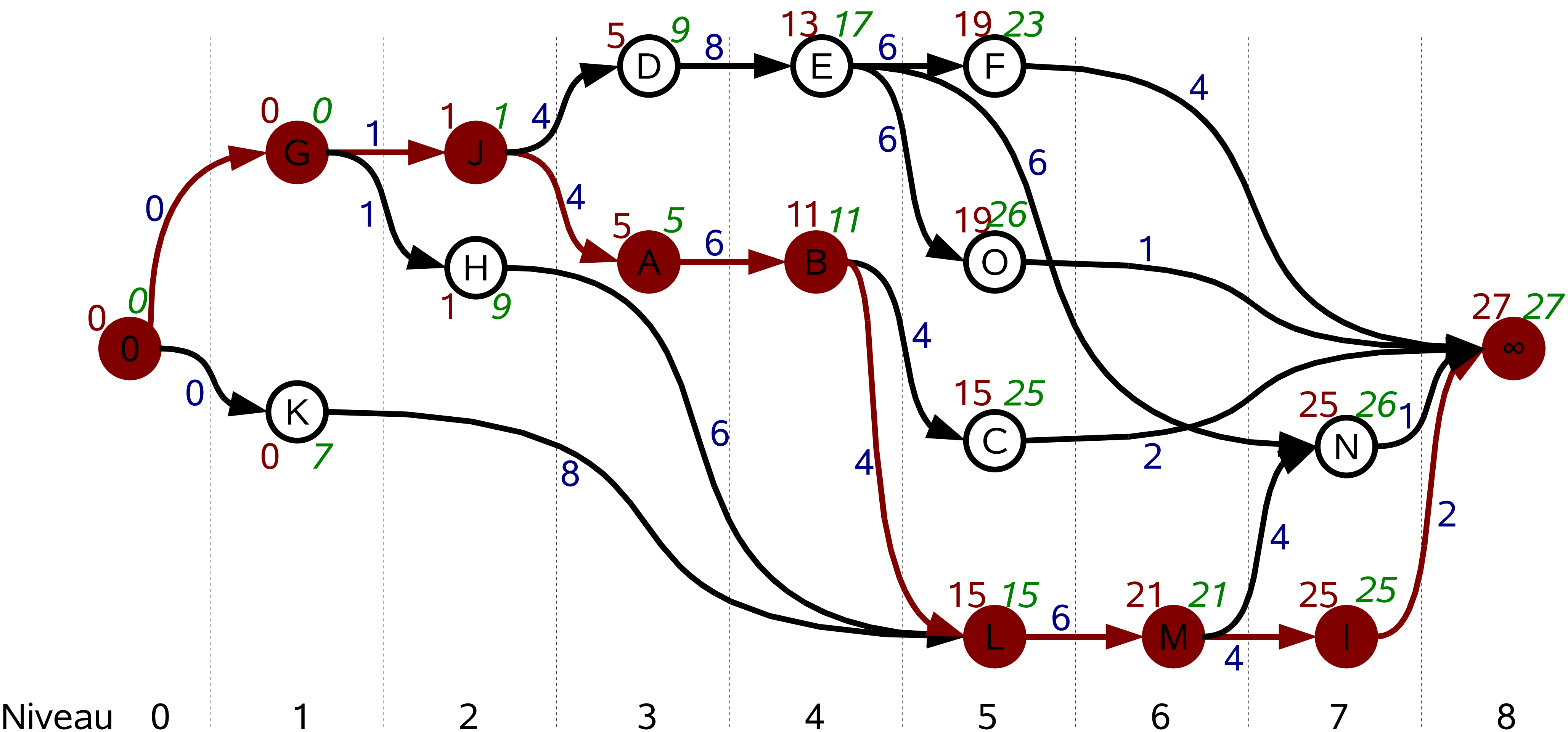
- Kontext:

  Ein Projekt besteht aus einer Menge von $n$ Tasks verbunden mit zeitlichen Constraints und **Ressourcen-Constraints**

- Objektiv:
  - Berechne die minimale Zeitdauer des Projektes, angenommen, die Ressourcen haben **endliche Kapazitäten** $\rightarrow$ minimiere $(t_{n+1} - t_0)$, so dass zeitliche Constraints und Ressourcen-Constraints erfüllt sind
  - Bestimme die frühestmöglichen und spätest zulässigen Anfangszeitpunkte von allen Tasks
  - Bestimme die kritischen Tasks

# Projektplanung – Ressourcen mit endlichen Kapazitäten

**Nur eine Person für die Tasks D und A → ordnen D und A**



# Projektplanung – Ressourcen mit endlichen Kapazitäten

**Allgemeine Prozedur für die Auflösung:**

1) Aufzählen der verschiedenen Möglichkeiten:
   - wenn A vor D ist, dann fügen wir eine Kante von A zu D ein, deren Wert gleich der Dauer von A ist;
   - wenn D vor A ist, dann fügen wir eine Kante von D zu A ein, deren Wert gleich der Dauer von D ist.
2) Lösen der entsprechenden Probleme
3) Treffen der besten Entscheidung

Wenn die Menge der Tasks, die Ressourcen verteilen, groß ist, dann dauert die Auflösung sehr lange, weil die Anzahl der Möglichkeiten riesig ist!

# III. Produktionsplanung

- Kontext
- Beispiel
- Industrielle Software

# Produktionsplanung – Kontext

- Eine Menge von $n$ **Tasks** ist auszuführen.
- Die **Ressourcen** sind Maschinen mit endlichen Kapazitäten.
- Die **Optimierungskriterien** hängen von Zeitpunkten, Ablieferungsterminen oder laufenden Vorräten ab; Fertigungsaufträge sind mehr oder weniger dringend...

# Produktionsplanung – Beispiel

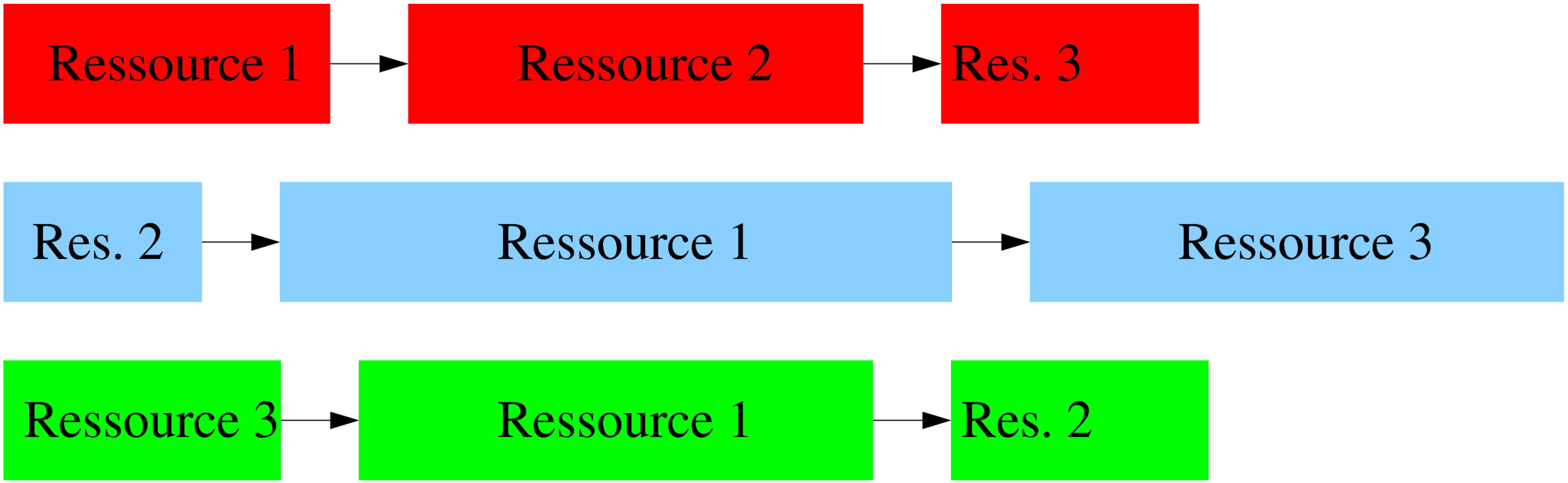
Problembeschreibung:

- 3 Fertigungsaufträge und 3 Tasks je Fertigungsauftrag
- 3 Maschinen

  jede Maschine kann nur ein Task auf einmal durchführen
- Minimiere die gesamte Produktionsdauer (Makespan)

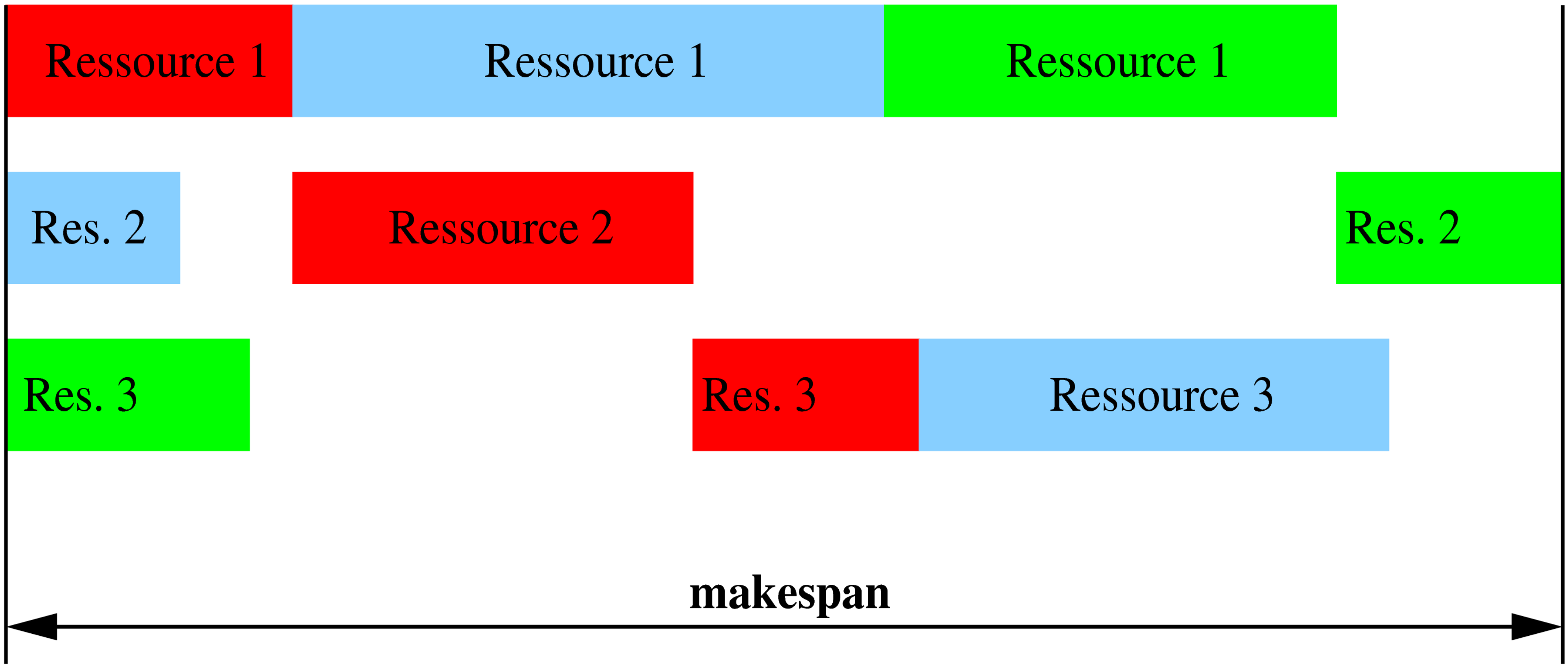
# Produktionsplanung – Beispiel

## Problembeschreibung



# Produktionsplanung – Beispiel

## Optimaler Ablaufplan



# Scheduling in der Industrie

- Programmbibliotheken:
  - ILOG Scheduler
  - Koalog Constraint Solver
- Enterprise Resource Planning (ERP):
  - SAP AG
  - Oracle Applications
  - Infor Global Solutions