



# Management großer Softwareprojekte

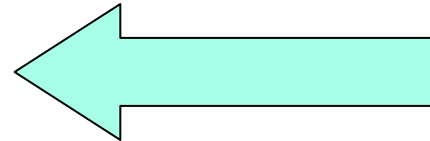
Prof. Dr. Holger Schlingloff

Humboldt-Universität zu Berlin,  
Institut für Informatik

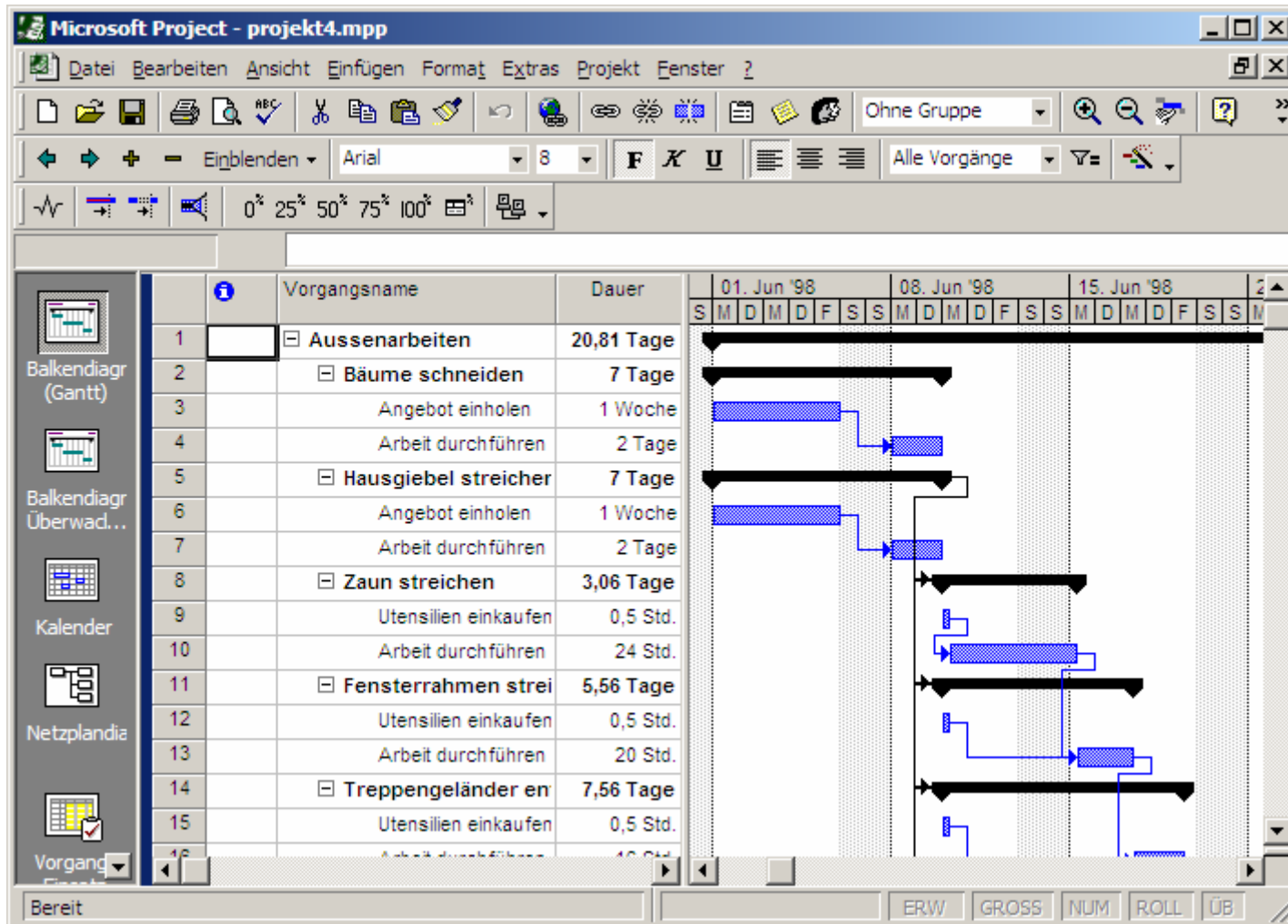
Fraunhofer Institut für Rechnerarchitektur  
und Softwaretechnik FIRST

# Wo stehen wir?

- 1. Einleitung**
- 2. Projektphasen**
- 3. Projektorganisation**
- 4. Aufwandsschätzung**
- 5. Planung**
  - Planungsarten
  - Netzplantechnik, Gantt-Pläne
  - Werkzeuge und Algorithmen
- 6. Risikoanalyse**
  - Risikoarten
  - Risikoermittlung und –quantifizierung
- 7. Projektdurchführung**
  - Projektablauf, Meilensteine
  - Managementaufgaben
- 8. Personalführung**
  - ...



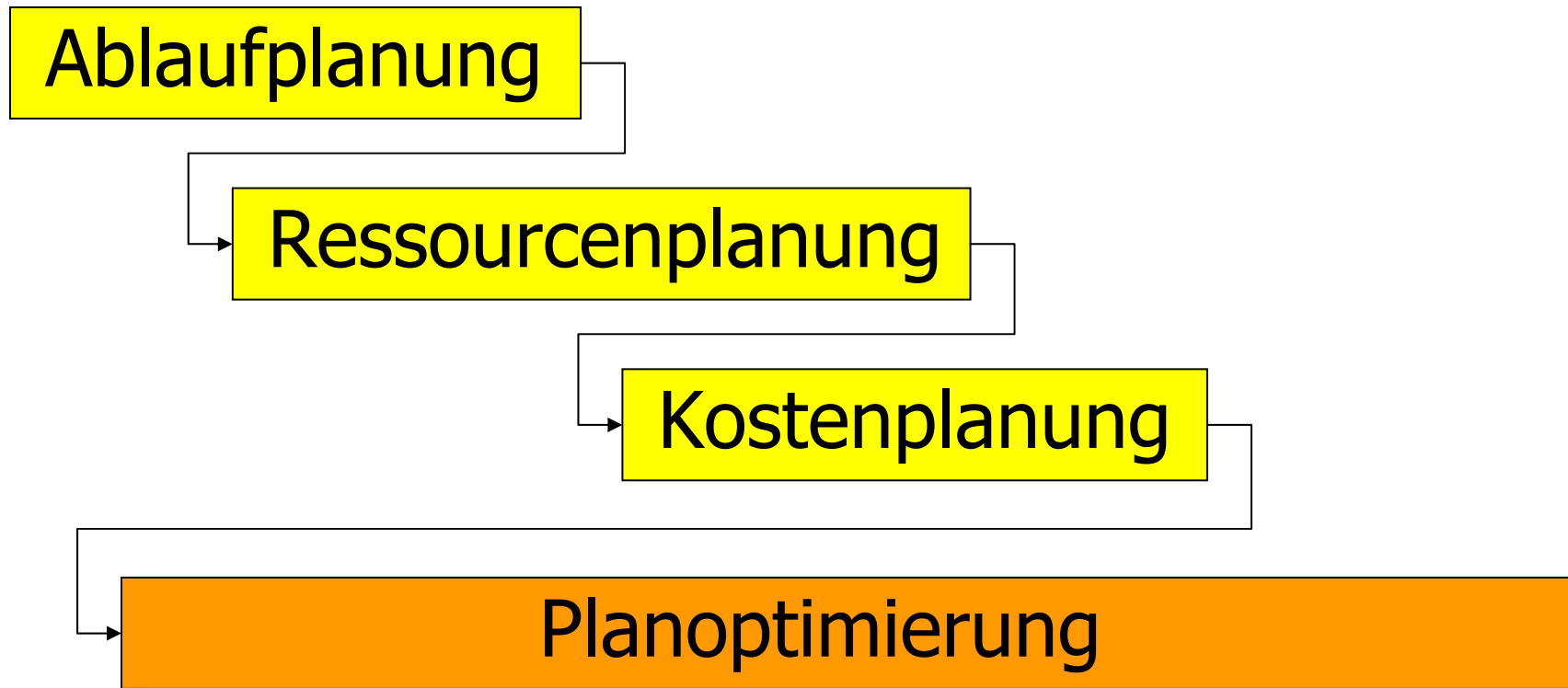
# Werkzeugbeispiel: MS Project



<http://www.waehlich.fh-aachen.de/lehrgebiete/winproj/index.html>



# Planungsprozess



# Optimierungskriterien

---

Die für eine Planoptimierung relevanten Größen können sich auf eine oder mehrere der drei wichtigsten Planungselemente beziehen:

- **Termine** - z.B. in Form von vereinbarten Meilenstein-Terminen
- **Ressourcen** - z.B. durch die begrenzte Verfügbarkeit, Urlaub oder Abwesenheiten
- **Kosten** - z.B. durch eine Liquiditätsplanung

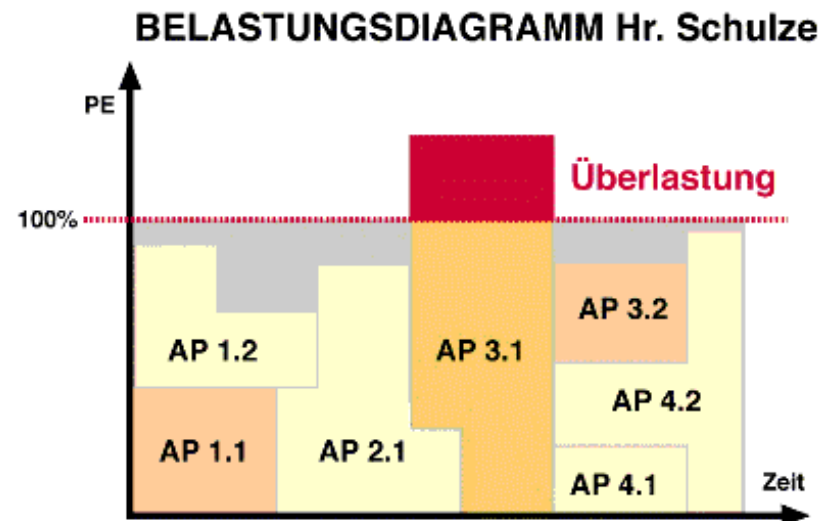


# manuelle Optimierung

**Ziel:** Auslastung von Ressourcen maximieren

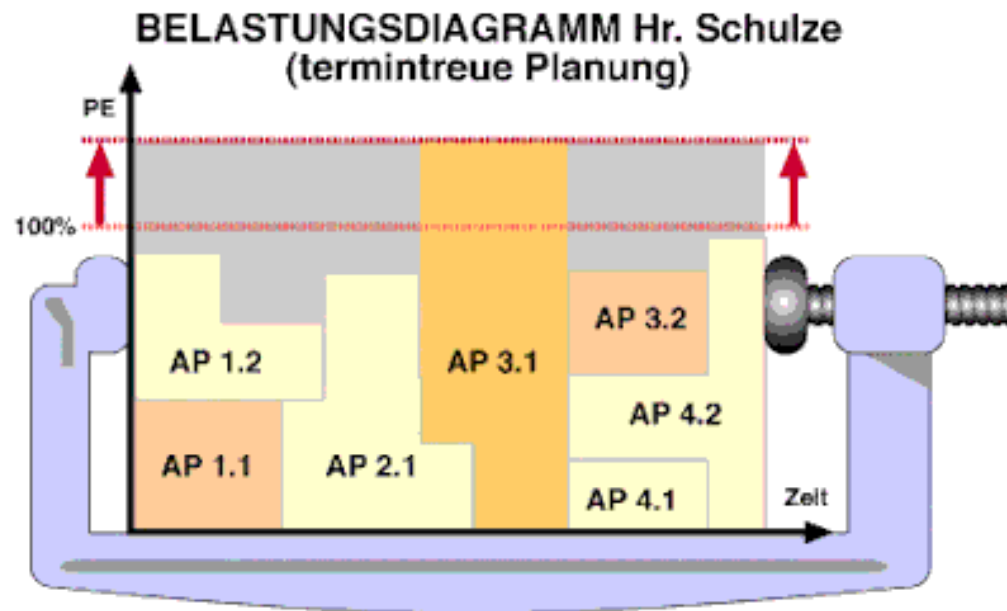
**Vorgehensweise: Belastungsdiagramm**  
bestehend aus

- **Ressourcenverfügbarkeit** und
- **Ressourcenbedarf**



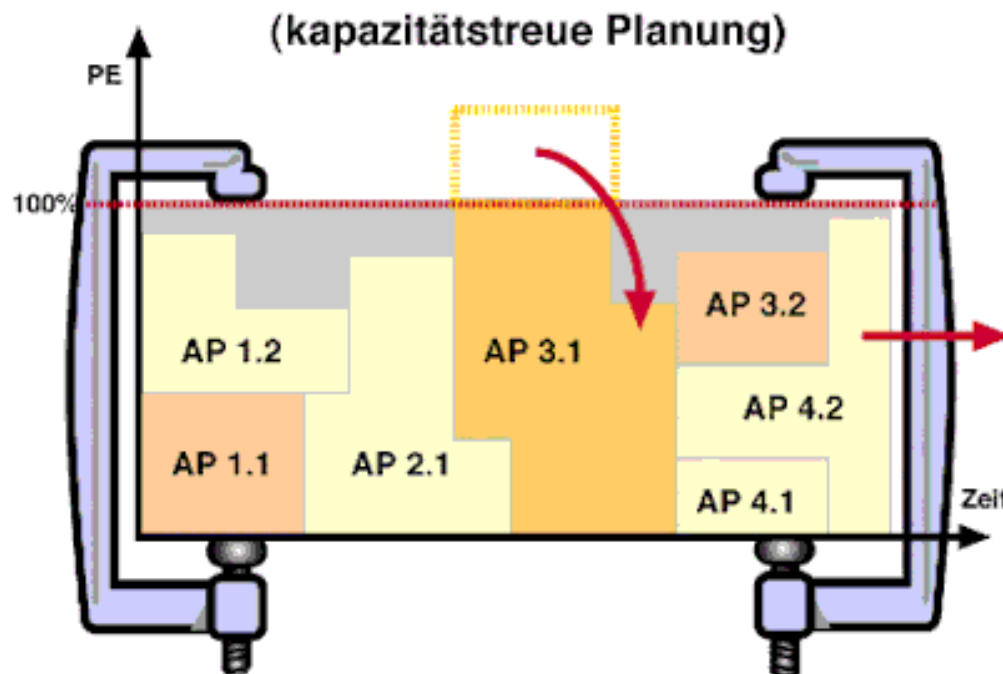
# termintreue Planung

- Die Verfügbarkeit wird geändert, weil der Termin beibehalten werden muss
  - Überstunden
  - zusätzliche Ressourcen



# kapazitäts- oder ressourcentreue Planung

- Aufgaben werden nach hinten geschoben, weil die Verfügbarkeit nicht erhöht werden kann
  - Verschiebung des Fertigstellungstermins!





- Führen beide Planungsansätze nicht zum Ziel, nämlich das geforderte Ergebnis (Projektziel) in der geforderten Zeit zu erbringen, muss die Höhe des Aufwands, also das Arbeitsvolumen reduziert werden
  - Verminderung der Funktionalität
  - Verschlechterung der Qualität

# Plananpassung nach Terminen

- **Ziel:** bereits vereinbarte Termine (Meilensteine) in Planung einbeziehen
- Eingreifen in die Netzplanlogik – eigentlich zu vermeiden
- Meilensteine sind wichtig als Frühwarnsystem
- **Beispiel:**  
Meilenstein laut Netzplanterminierung frühestens am 15.08. abgeschlossen. Laut Vereinbarung mit Auftraggeber muss Meilenstein spätestens am 01.09. erreicht sein. Fügt man diese Prämisse in den Netzplan ein (spätestes Ende des Meilensteins am 01.09.), erhalten alle Vorgänge, die im Pfad vor dem Meilenstein liegen, zwei Wochen Puffer. Sind diese zwei Wochen Puffer aufgebraucht (z.B. durch unvorhergesehene Verzögerungen), werden die Vorgänge kritisch.



# Plananpassung nach Kosten

---

- meist nur in Spezialfällen
  - bei sehr großen Projekten
  - bei vereinbartem Zahlungsplan
- Netzplan als Ausgangspunkt
- kostenintensive Arbeitspakete so spät wie möglich
  - Vorfinanzierungskosten!
- Visualisierung z.B. durch Gegenüberstellung von geplanten Projekt-Kosten und Zahlungseingang



# algorithmische Planung

---

Quelle: Problemlösen mit  
constraintbasierter  
Programmierung; H.-J. Goltz,  
FhG FIRST

- verschiedene mögliche Ansätze; heute:  
constraintbasierte logische Programmierung  
(CLP)
- Constraints als Repräsentation relationaler  
Beziehungen zwischen Variablen
- Verarbeitung durch einen Constraint-Solver
  - Constraint Satisfaction Problem (CSP)



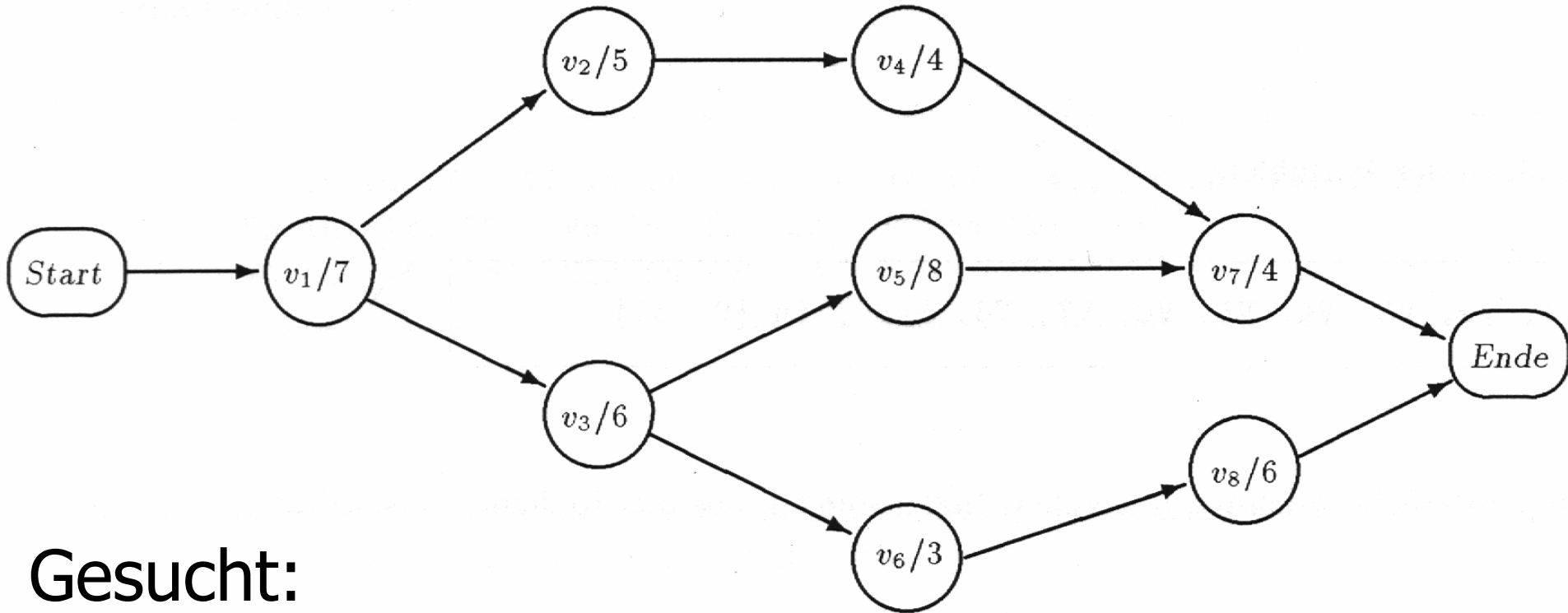
# Verarbeitungsschritte

---

- Vereinfachen von Constraint-mengen (z.B. durch Addition von Gleichungen)
- Konsistenzprüfung
- Erzeugen von Konsequenzen
- inkrementelle Verarbeitung von Constraints
- Backtracking und Cut



# Beispiel: Netzplan als CSP



Gesucht:

- früheste und späteste Startzeitpunkte
- kritischer Weg

# Aufstellen der Gleichungen

- Startzeitpunkt 0, spätester Endtermin ca. 32
- $V_1, V_2, \dots, E$ : Variablen für Startzeiten  $\{0..32\}$
- Ungleichungen:

$$\begin{aligned} V_1 + 7 &\leq V_2, & V_1 + 7 &\leq V_3, \\ V_2 + 5 &\leq V_4, \\ V_3 + 6 &\leq V_5, & V_3 + 6 &\leq V_6, \\ V_4 + 4 &\leq V_7, & V_5 + 8 &\leq V_7, \\ V_6 + 3 &\leq V_8, \\ V_7 + 4 &\leq E, & V_8 + 6 &\leq E \end{aligned}$$



# Berechnungsweg

$V1+7 \leq V2$ ,  $V2+5 \leq V4$ ,  
 $V4+4 \leq V7$ ,  $V7+4 \leq E$ , ...

- $V1$  in  $\{0..32\}$ ,  $V2$  in  $\{0..32\}$ ,  $V1+7 \leq V2$   
→  $V1$  in  $\{0..25\}$ ,  $V2$  in  $\{7..32\}$   
Begründung:  $V2 \leq 32 \rightarrow V1+7 \leq 32 \rightarrow V1 \leq 25$
- $V2$  in  $\{7..32\}$ ,  $V4$  in  $\{0..32\}$ ,  $V2+5 \leq V4$   
→  $V2$  in  $\{7..27\}$ ,  $V4$  in  $\{12..32\}$
- $V4$  in  $\{12..32\}$ ,  $V7$  in  $\{0..32\}$ ,  $V4+4 \leq V7$   
→  $V4$  in  $\{12..28\}$ ,  $V7$  in  $\{16..32\}$
- $V7$  in  $\{16..32\}$ ,  $E$  in  $\{0..32\}$ ,  $V7+4 \leq E$   
→  $V7$  in  $\{16..28\}$ ,  $E$  in  $\{20..32\}$





# kritischer Pfad

- **Ergebnis der Berechnung**

$V1$  in  $\{0..7\}$ ,  $V2$  in  $\{7..19\}$ ,  $V3$  in  $\{7..14\}$ ,  
 $V4$  in  $\{12..24\}$ ,  $V5$  in  $\{13..20\}$ ,  $V6$  in  $\{13..23\}$ ,  
 $V7$  in  $\{21..28\}$ ,  $V8$  in  $\{16..26\}$ ,  $E$  in  $\{25..32\}$

- **Hinzufügen der Gleichung  $E=25$  ergibt**

$V1 = 0$ ,  $V2$  in  $\{7..12\}$ ,  $V3 = 7$ ,  
 $V4$  in  $\{12..17\}$ ,  $V5 = 13$ ,  $V6$  in  $\{13..16\}$ ,  
 $V7 = 21$ ,  $V8$  in  $\{16..19\}$ ,  $E = 25$



# Modellierung von Bedingungen

- **A vor B:**  $\text{Start}_A + \text{Dauer}_A \leq \text{Start}_B$
- **B direkt nach A:**  $\text{Start}_A + \text{Dauer}_A = \text{Start}_B$
- **B spätestens nach A:**  $\text{Start}_B \leq \text{Start}_A + \text{Dauer}_A$
- **B zeitversetzt nach A:**  $\text{Start}_A + \text{Dauer}_A + t = \text{Start}_B$
- **späteste Endzeit:**  $\text{Start}_A + \text{Dauer}_A \leq \text{Ende}$
- **Ende-Anfang:**  $\text{Start}_A + \text{Dauer}_A \leq \text{Start}_B + t$
- **Ende-Ende:**  $\text{Start}_A + \text{Dauer}_A = \text{Start}_B + \text{Dauer}_B$

USW.



# Aufgabe

---

- Modellieren Sie den Strukturplan vom 18.12. als CSP und berechnen Sie die Lösung:

/1/ Vorgang 1, Aufwand 3 MT, fester Anfang am 21.10.96

/2/ Vorgang 2, Aufwand 20 MT

/3/ Vorgang 3, Aufwand 15 MT

/4/ Vorgang 4, Aufwand 5 MT, festes Ende am 13.12.96

Zwischen den Vorgängen existieren folgende Abhängigkeiten:

/2/ kann sofort nach Ende von /1/ beginnen

/3/ kann erst 5 Tage nach Ende von /1/ beginnen

/4/ kann erst beginnen, wenn /3/ beendet ist

/4/ kann frühestens 5 Tage vor dem Ende von /2/ beginnen



# Grenzen der Formulierung als CSP

---

- Disjunktion: z.B. „**A** und **B** können nicht gleichzeitig durchgeführt werden“
- Zuordnung von beschränkten Ressourcen
- Kumulative Constraints,  
z.B. „Aufgabe  $A_1$ ,  $A_2$ ,  $A_3$ ... benötigen gleichartige beschränkte Ressourcen“



# Constraintlogische Programmierung

---

- Erweiterung von Prolog um Verarbeitung von Constraints
- Inferenzsystem mit Backtracking
- Beim Backtracking Zurücksetzen des Zustandes der Constraintvariablen



# Beispiel

```
test1(X,Y,Z) :- [X,Y,Z]::1..20, X + 5 #< Y, Z + 10 #< X, indomain(X).
```

Trace-Protokoll mit zusätzlicher Ausschrift über die Domänen und erzwungenem Backtracking:

```
?- test1(X,Y,Z).  
CALL    test1(X, Y, Z)  
CALL    [X, Y, Z] :: 1..20  
EXIT    [X in {1..20}, Y in {1..20}, Z in {1..20}] :: 1..20  
CALL    X in {1..20} + 5 #< Y in {1..20}  
EXIT    X in {1..14} + 5 #< Y in {7..20}  
Domains [X in {1..14}, Y in {7..20}, Z in {1..20}]  
CALL    Z in {1..20} + 10 #< X in {1..14}  
WAKE    X in {12..14} + 5 #< Y in {7..20}  
EXIT    Z in {1..3} + 10 #< X in {12..14}  
Domains [X in {12..14}, Y in {18..20}, Z in {1..3}]  
CALL    indomain(X in {12..14})  
WAKE    Z in {1..3} + 10 #< 12  
WAKE    12 + 5 #< Y in {18..20}  
EXIT    indomain(12)  
EXIT    test1(12, Y in {18..20}, 1)
```

X = 12      Y = Y in {18..20}      Z = 1      ;



# Beispiel (Forts.)

```
CALL    fail
FAIL    fail
RETRY   indomain(X in {12..14})
WAKE    Z in {1..3} + 10 #< 13
WAKE    13 + 5 #< Y in {18..20}
EXIT    indomain(13)
EXIT    test1(13, Y in {19..20}, Z in {1..2})
```

X = 13      Y = Y in {19..20}      Z = Z in {1..2}      ;

```
CALL    fail
FAIL    fail
RETRY   indomain(X in {12..14})
WAKE    Z in {1..3} + 10 #< 14
WAKE    14 + 5 #< Y in {18..20}
EXIT    indomain(14)
EXIT    test1(14, 20, Z in {1..3})
```

X = 14      Y = 20      Z = Z in {1..3}      ;

```
CALL    fail
FAIL    fail
RETRY   indomain(X in {12..14})
FAIL    indomain(X in {12..14})
FAIL    test1(X in {12..14}, Y in {18..20}, Z in {1..3})
no (more) solutions
```



# Definition disjunktiver Constraints

- „Aktivität **A** mit Dauer **DA** und Aktivität **B** mit Dauer **DB** können nicht gleichzeitig ausgeführt werden, da sie die selbe Ressource verwenden“

$\text{disjunct}(A, DA, B, DB) :- A + DA \leq B.$

$\text{disjunct}(A, DA, B, DB) :- B + DB \leq A.$

- „Aktivitäten **A<sub>i</sub>** mit Startzeiten **S<sub>i</sub>**, Dauern **D<sub>i</sub>** und Bedarf **H<sub>i</sub>** dürfen nicht mehr als **L** Ressourcen gleichzeitig verwenden“



$\text{cumulative}([S1, S2, \dots, Sn], [D1, D2, \dots, Dn], [H1, H2, \dots, Hn], L)$