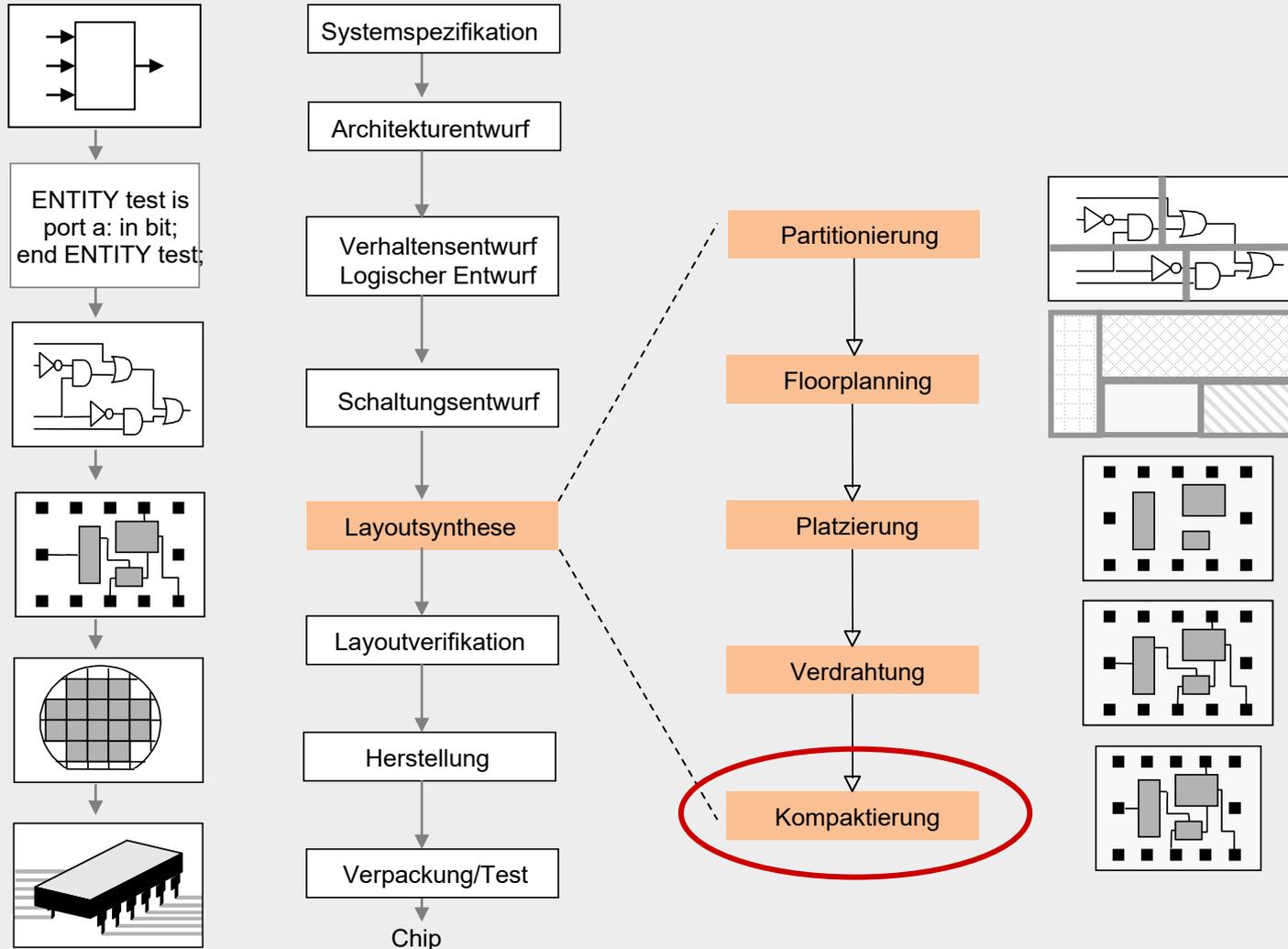


- 8.1 Einführung
- 8.2 Begriffe, Modelle, Datenstrukturen
- 8.3 Symbolische Kompaktierung
- 8.4 Kompaktierungsalgorithmen
  - 8.4.1 Schnittkompaktierung
  - 8.4.2 Abstandsgraph-Kompaktierung

# 8.1 Einführung



Gegeben ist

- ein **entwurfsregel-korrektes Schaltungslayout** (Maskenlayout) mit der Platzierung aller Komponenten und der Verdrahtung aller Netze oder
- ein **symbolisches Layout** mit einer abstrakten Darstellung aller Komponenten und deren Verdrahtung.

Gesucht ist ein kompaktiertes Layout mit

- minimaler Fläche,
- invarianter Struktur von Platzierung und Verdrahtung sowie
- strikter Einhaltung von Entwurfsregeln.

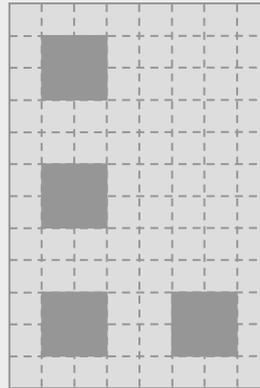
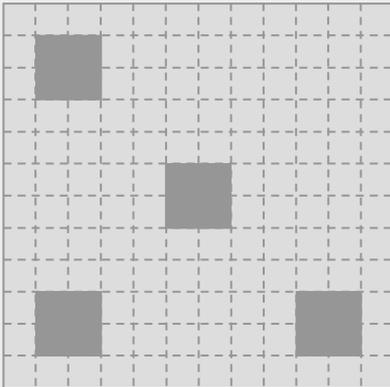
- Kompaktierungsalgorithmen werden seit den 1970er Jahren aktiv angewendet und sind heute sehr weit ausgereift
- Während anfänglich ausschließlich die Verringerung der Layoutfläche im Vordergrund stand, kamen in den letzten Jahren aufgrund der wachsenden Schaltungsanforderungen weitere Zielstellungen hinzu
  - Optimierung kritische Parameter einer Schaltung, z.B. Verbindungslängen, kritische Netze, Signallaufzeiten, ...
  - „Performance-driven Compaction“.

### 1- und 2-dimensionale Kompaktierung

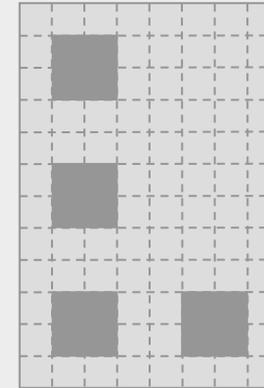
- Bei der **1-dimensionalen Kompaktierung** werden die Layoutelemente nur in einer Richtung bewegt bzw. „zusammengeschoben“ (z.B. zuerst in  $x$ -Richtung, dann in  $y$ -Richtung)
- Bei der **2-dimensionalen Kompaktierung** werden  $x$ - und  $y$ -Verschiebung gleichzeitig betrachtet.

## 1D-Kompaktierung

Ausgangslayout  
 $12 \lambda \times 12 \lambda$



1. x-Kompaktierung  
 $12 \lambda \rightarrow 8 \lambda$



2. y-Kompaktierung  
 $12 \lambda \rightarrow 12 \lambda$

Ergebnis  
 $8 \lambda \times 12 \lambda$

Entwurfsregeln:

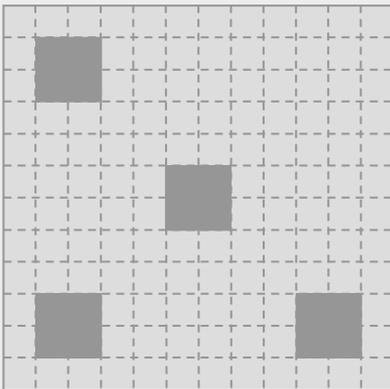
  $2 \lambda \times 2 \lambda$

Mindestabstand  $2 \lambda$

Randabstand  $1 \lambda$

### 1D-Kompaktierung

Ausgangslayout  
 $12\lambda \times 12\lambda$

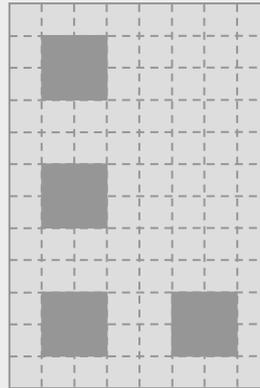


Entwurfsregeln:

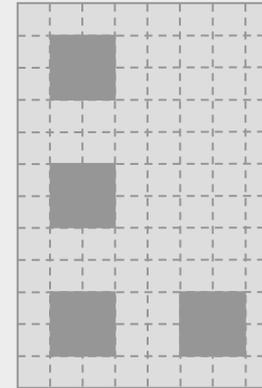
  $2\lambda \times 2\lambda$

Mindestabstand  $2\lambda$

Randabstand  $1\lambda$

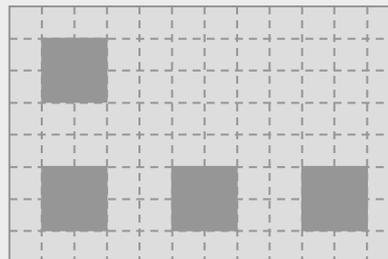


1. x-Kompaktierung  
 $12\lambda \rightarrow 8\lambda$

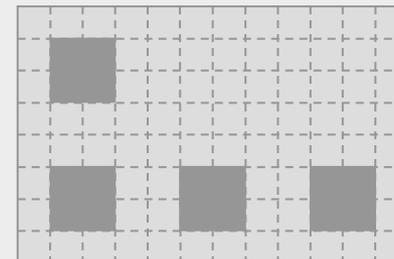


2. y-Kompaktierung  
 $12\lambda \rightarrow 12\lambda$

Ergebnis  
 $8\lambda \times 12\lambda$



1. y-Kompaktierung  
 $12\lambda \rightarrow 8\lambda$

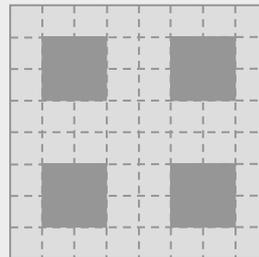
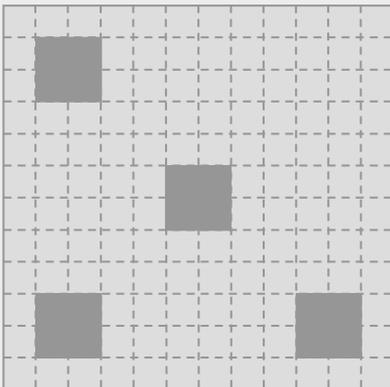


2. x-Kompaktierung  
 $12\lambda \rightarrow 12\lambda$

Ergebnis  
 $12\lambda \times 8\lambda$

## 2D-Kompaktierung

Ausgangslayout

 $12 \lambda \times 12 \lambda$ 

Ergebnis

 $8 \lambda \times 8 \lambda$ 

Entwurfsregeln:

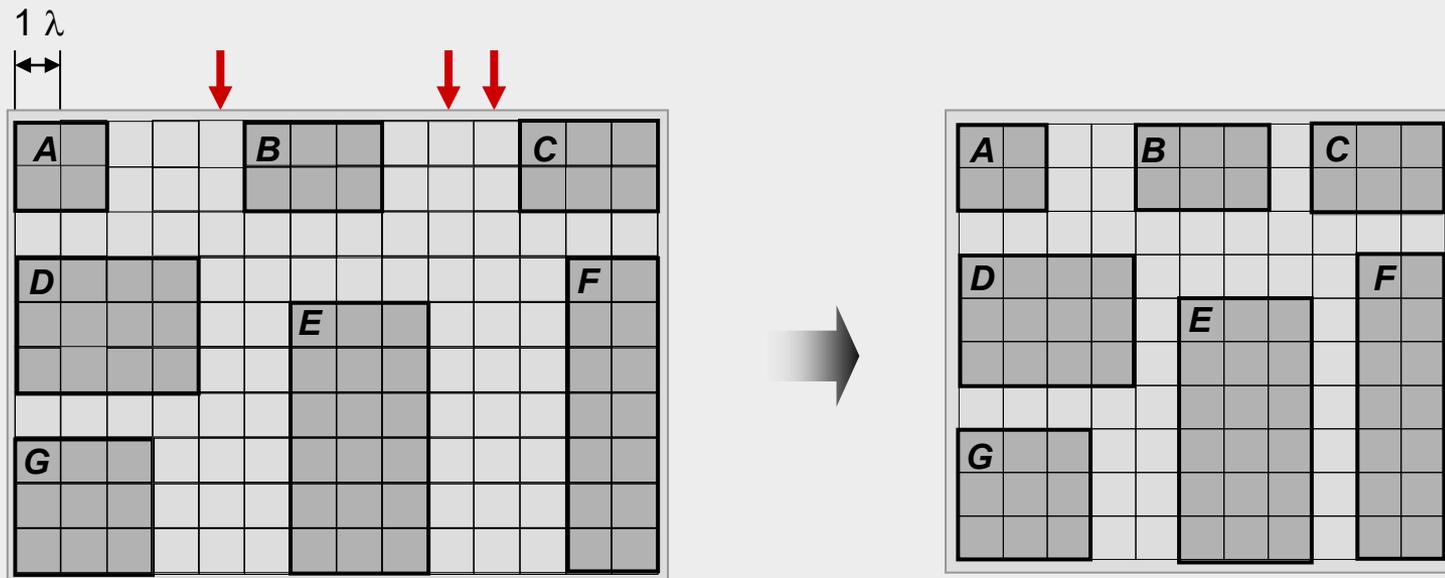
  $2 \lambda \times 2 \lambda$ Mindestabstand  $2 \lambda$ Randabstand  $1 \lambda$ 

2D-Kompaktierung

**x-Kompaktierung:**  $12 \lambda \rightarrow 8 \lambda$ **y-Kompaktierung:**  $12 \lambda \rightarrow 8 \lambda$

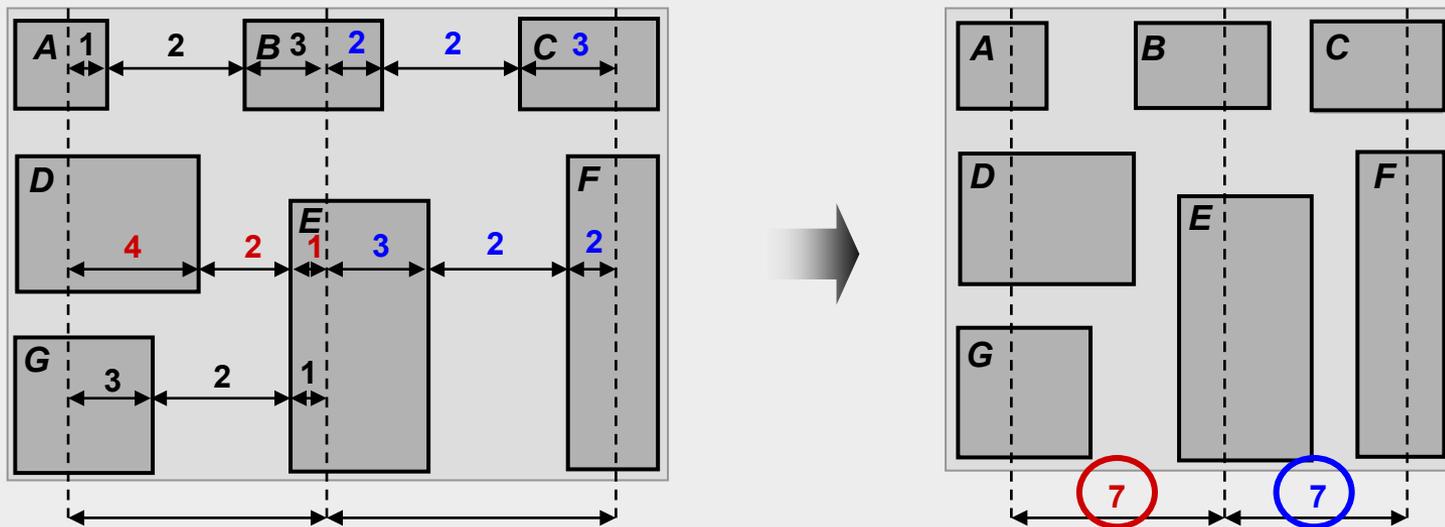
## 8.2 Begriffe, Modelle, Datenstrukturen

- Bei der **rasterbasierten Kompaktierung** werden die Layoutobjekte mittels der Zuordnung zu Rasterpunkten modelliert.
  - **Festes Raster (Fixed grid)**: Rasterabstand entspricht der kleinsten technologisch realisierbaren Layouteinheit



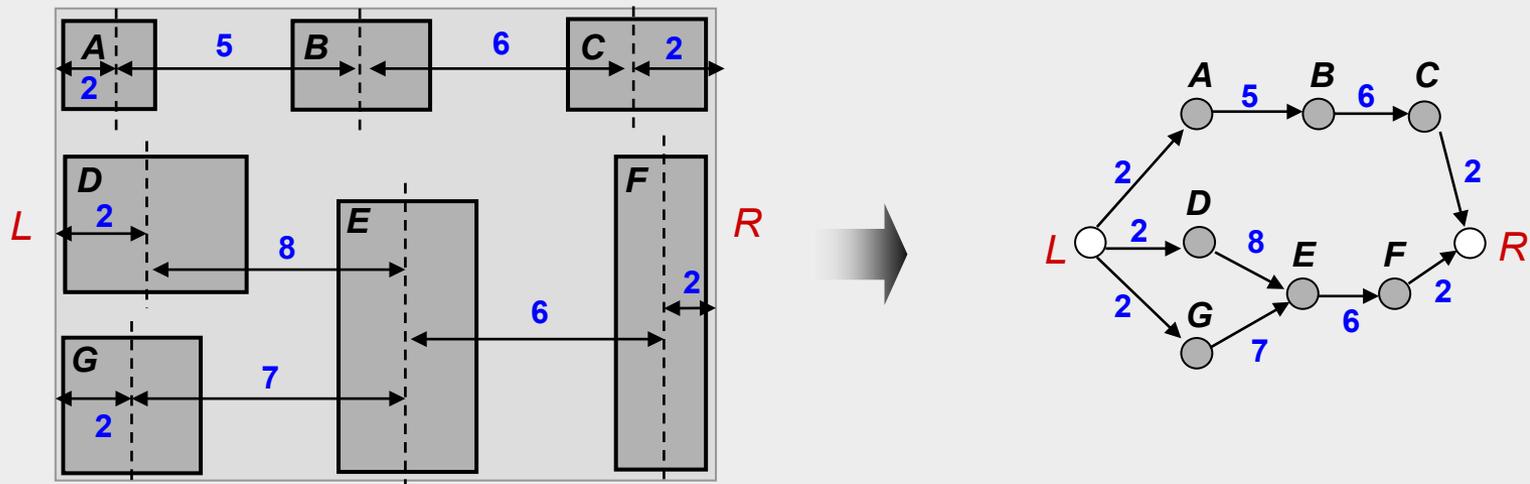
## 8.2 Begriffe, Modelle, Datenstrukturen

- Bei der **rasterbasierten Kompaktierung** werden die Layoutobjekte mittels der Zuordnung zu Rasterpunkten modelliert.
  - **Festes Raster (Fixed grid)**: Rasterabstand entspricht der kleinsten technologisch realisierbaren Layouteinheit
  - **Virtuelles Raster (Virtual grid)**: Abstände der einzelnen Rasterlinien sind nicht fest vorgegeben, sondern entsprechen jeweils den Mindestabständen der sich auf den Rasterlinien befindlichen Layoutelemente

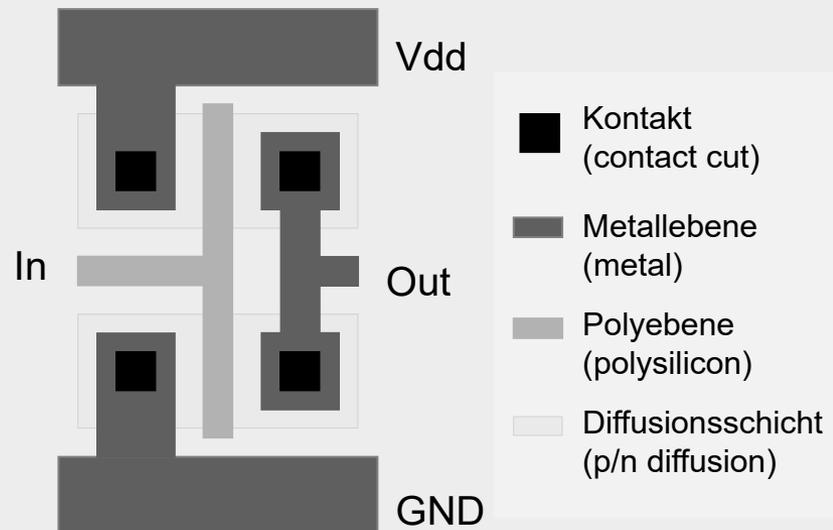


## 8.2 Begriffe, Modelle, Datenstrukturen

- Bei der **graphenbasierten Kompaktierung** wird das Layout in Form eines sog. Abstandsgraphen (Constraint graph) repräsentiert, wobei die Knoten die verschiedenen Layoutobjekte und die Kanten die minimalen Abstandsregeln zwischen diesen Objekten widerspiegeln



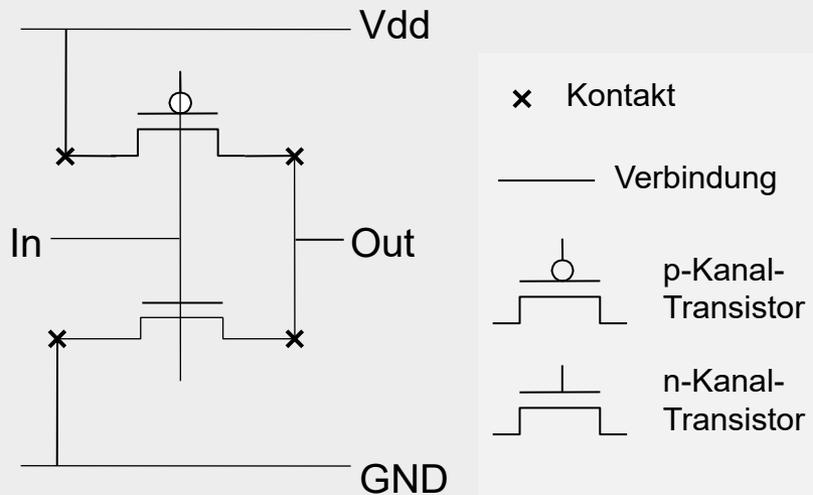
- **Maskenlayout:** Exakte Koordinaten aller Layoutelemente bekannt



Maskenlayout eines Inverters

## 8.2 Begriffe, Modelle, Datenstrukturen

- **Maskenlayout:** Exakte Koordinaten aller Layoutelemente bekannt
- **Symbolisches Layout:** Nur Nachbarschaftsbeziehungen der Layoutelemente bekannt, konkreter Abstand ohne Bedeutung

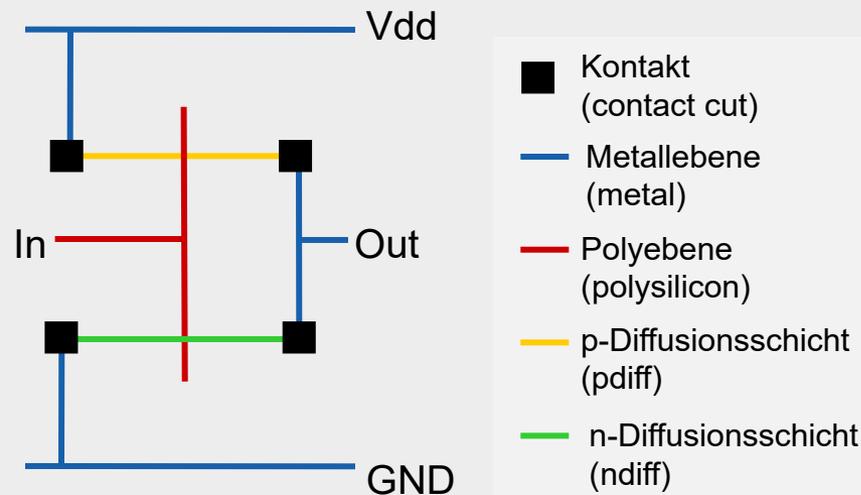


Symbolisches Layout eines Inverters

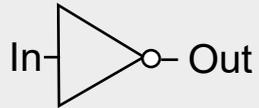
## 8.2 Begriffe, Modelle, Datenstrukturen

- **Maskenlayout:** Exakte Koordinaten aller Layoutelemente bekannt
- **Symbolisches Layout:** Nur Nachbarschaftsbeziehungen der Layoutelemente bekannt, konkreter Abstand ohne Bedeutung
  - Beispiel: **Stick-Diagramm**
  - Symbolische Layoutdarstellung, die alle Komponenten und deren relative Platzierung darstellt
  - Nicht dargestellt: Exakte Platzierung, Transistorgröße, Leiterzuglängen und -breiten

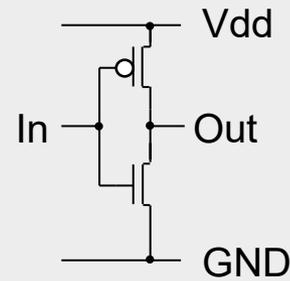
Stick-Diagramm  
eines Inverters



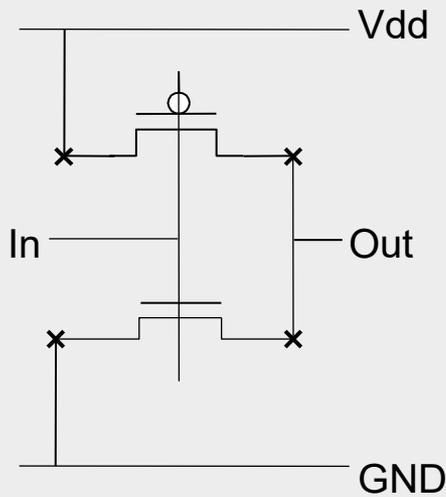
# Beispiel: CMOS-Inverter



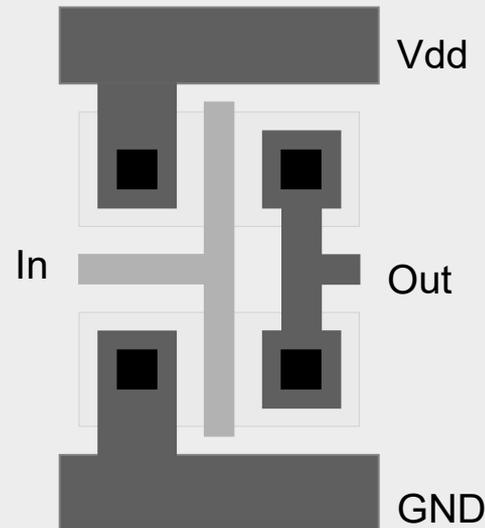
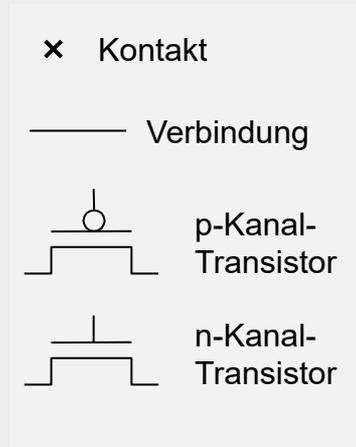
Schaltplansymbol



Transistorschaltung



Symbolisches Layout



Maskenlayout



8.1 Einführung

8.2 Begriffe, Modelle, Datenstrukturen

**→ 8.3 Symbolische Kompaktierung**

8.4 Kompaktierungsalgorithmen

8.4.1 Schnittkompaktierung

8.4.2 Abstandsgraph-Kompaktierung

- Kompaktoren werden auch bei der symbolischen Layoutentwicklung eingesetzt (symbolische Kompaktierung, Symbolic compaction)
- **Symbolische Layoutentwicklung**
  - Basiert auf symbolischer Layoutdarstellung, z.B. Stick-Diagramm
  - Ermöglicht Vereinfachung der Layoutentwurfs durch Umgehung der Vielzahl von Entwurfsregeln und der aufwendigen Überführung von einer Technologie zur nächsten
- **Symbolische Kompaktierung**
  - Überführung eines symbolischen Layouts in eine reale Technologie-Implementierung
  - Kompaktierung dient damit zur Technologie-Anpassung

8.1 Einführung

8.2 Begriffe, Modelle, Datenstrukturen

8.3 Symbolische Kompaktierung

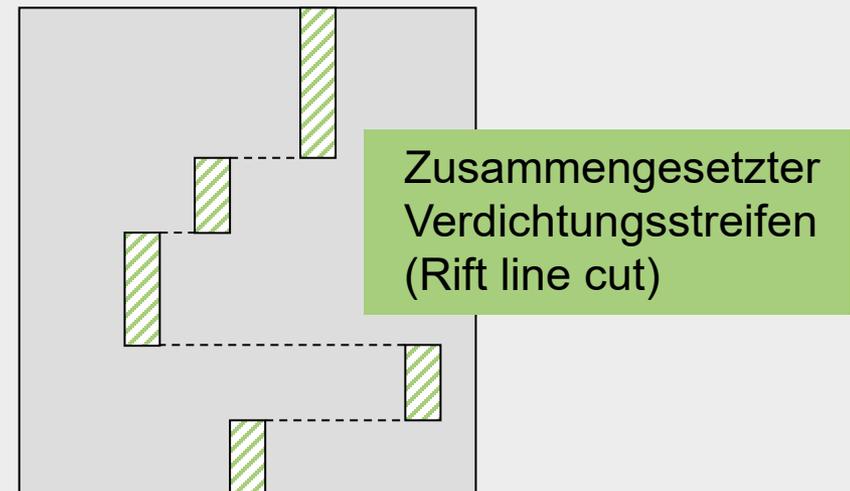
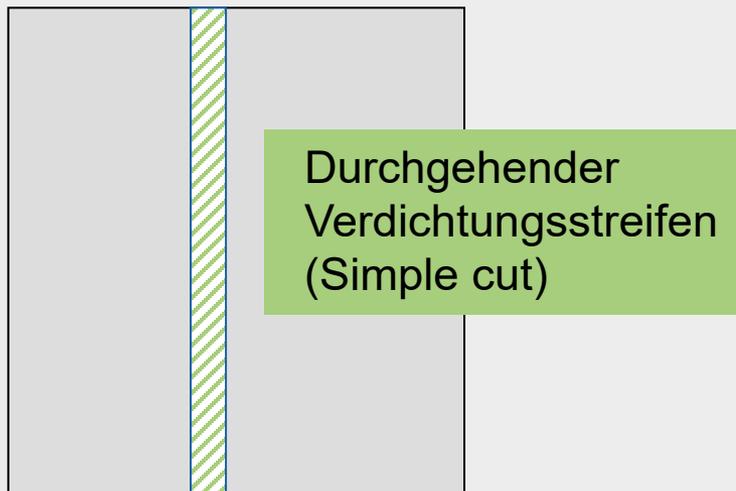
**→** 8.4 Kompaktierungsalgorithmen

8.4.1 Schnittkompaktierung

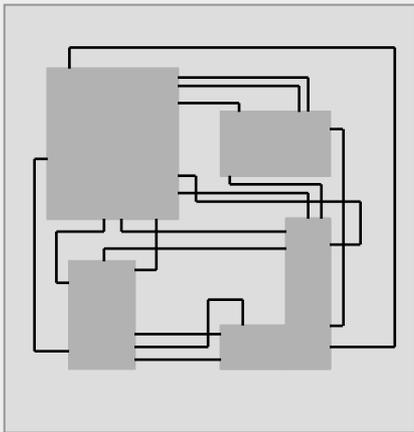
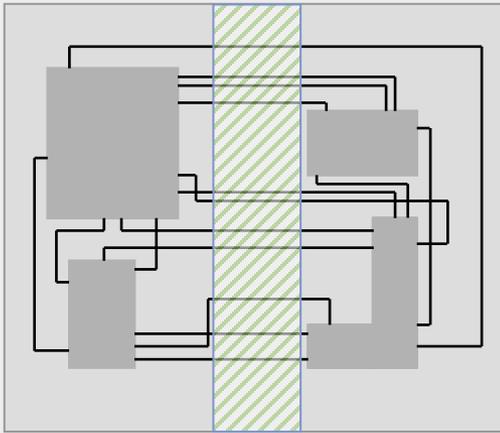
8.4.2 Abstandsgraph-Kompaktierung

## 8.4.1 Schnittkompaktierung

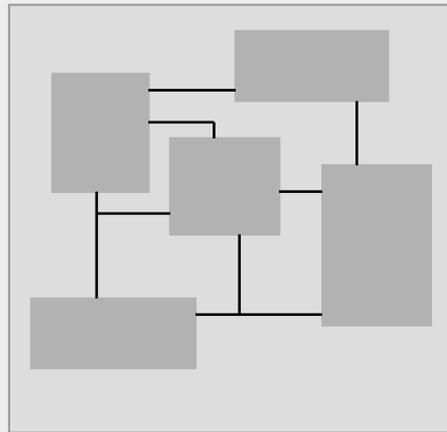
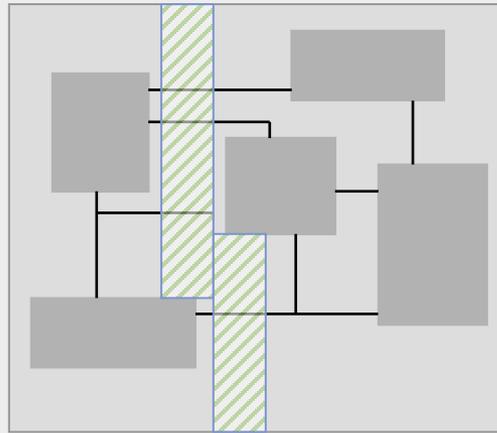
- 1970 erstmals von *Akers, Geyer* und *Roberts* vorgestellt
- Layout enthält **Verdichtungsstreifen**, die frei von platzierten Elementen sind und in denen alle Leiterzüge senkrecht zur Streifenrichtung verlaufen
- Streifenbereich wird aus der Schaltung entfernt und die betreffenden Koordinaten der Schaltungselemente auf die neue Layoutgröße umgerechnet



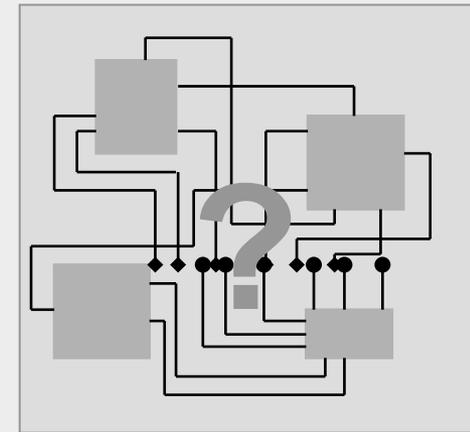
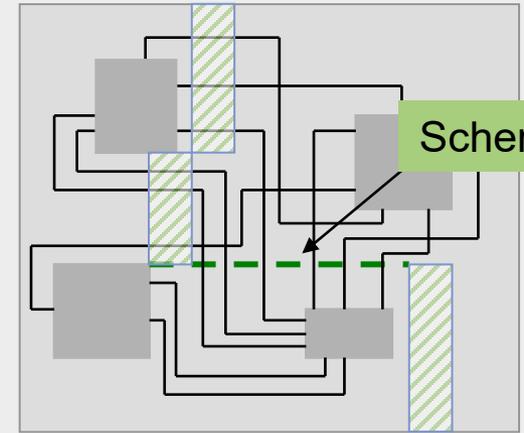
## 8.4.1 Schnittkompaktierung



Durchgehender Verdichtungsstreifen



Zusammengesetzte Verdichtungsstreifen

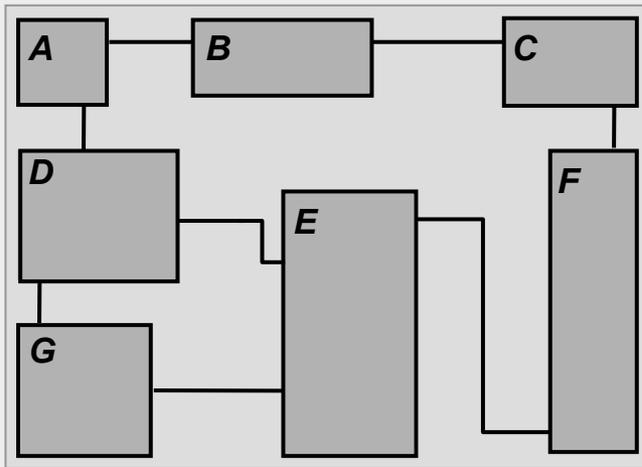


### Algorithmus

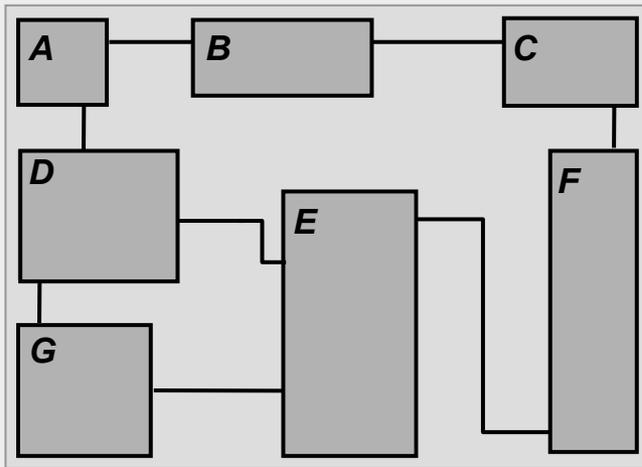
1. Für eine vorgegebene Kompaktierungsrichtung
  - a) Abbildung der Layoutelemente auf einem binären Raster
  - b) Ermitteln von durchgehenden bzw. zusammengesetzten Verdichtungsstreifen
  - c) Entfernen der Verdichtungsstreifen.  
Bei zusammengesetzten Verdichtungsstreifen: Untersuchung der Scherlinie auf Layoutkonflikte.
  - d) Falls bei zwei aufeinander folgenden Durchläufen keine Entfernung von Verdichtungsstreifen möglich ist, weiter mit Schritt 3
2. Wechsel der Kompaktierungsrichtung, weiter mit Schritt 1
3. Rückabbildung der Rastertopologie in reale Layoutkoordinaten, ENDE.

### Implementierung

- Zur Identifizierung der zur Kompaktierung geeigneten Verdichtungsstreifen werden die Rasterpunkte binär, d.h. mit 0 oder 1, gekennzeichnet.
- Rasterpunkte, denen in der jeweiligen Kompaktierungsrichtung nicht zusammenschiebbare Layoutelemente (Bereiche mit Zellenbelegung oder senkrecht zur aktuellen Kompaktierungsrichtung verlaufenden Leiterzügen) zugeordnet sind, markiert man mit 0.
- Anschließend werden alle verbleibenden, für die aktuelle Kompaktierungsrichtung **kompaktierfähigen Layoutbereiche mit 1** gekennzeichnet.



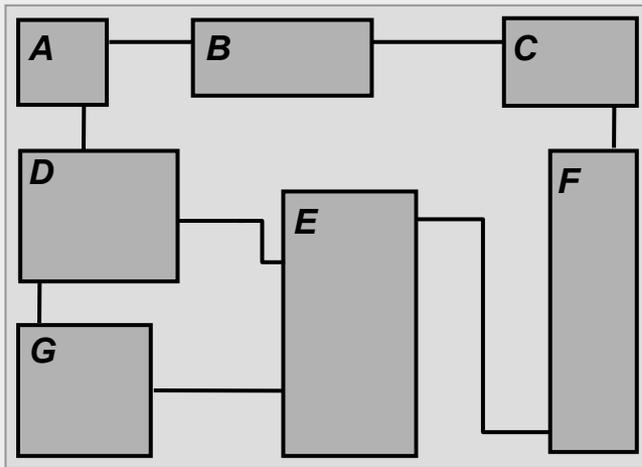
Ausgangslayout; horizontale Kompaktierung



Ausgangslayout

0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
1	0	1	1	1	1	1	0	1	1	1	1	1	0
0	0	0	0	1	1	1	0	1	1	1	1	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0
0	1	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0

Schritt 1a: Abbildung auf binärem Raster;  
 Horizontal nicht „zusammenschiebbar“: 0  
 Horizontal kompaktierfähiger Bereich: 1



Ausgangslayout

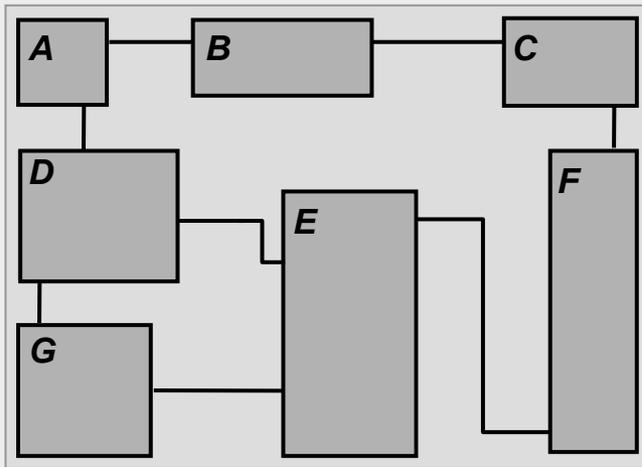
0	0	1	1	0	0	0	0	1	1	1	0	0	0	
0	0	1	1	0	0	0	0	1	1	1	0	0	0	
1	0	1	1	1	1	1	0	1	1	1	1	1	0	
0	0	0	0	1	1	1	0	1	1	1	1	0	0	
0	0	0	0	1	0	0	0	0	1	0	1	0	0	
0	0	0	0	1	0	0	0	0	1	0	1	0	0	
0	1	1	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	1	0	0	0	1	0	1	0	0

Schritt 1a: Abbildung auf binärem Raster

Zusammengesetzte Verdichtungsstreifen      Durchgehende Verdichtungsstreifen

0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
1	0	1	1	1	1	1	0	1	1	1	1	1	0
0	0	0	0	1	1	1	0	1	1	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0
0	1	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0

Schritt 1b: Ermitteln von Verdichtungsstreifen



Ausgangslayout

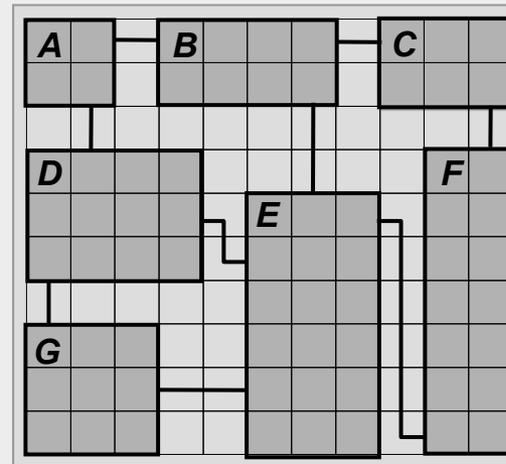
0	0	1	1	0	0	0	0	1	1	1	0	0	0	
0	0	1	1	0	0	0	0	1	1	1	0	0	0	
1	0	1	1	1	1	1	0	1	1	1	1	1	0	
0	0	0	0	1	1	1	0	1	1	1	1	0	0	
0	0	0	0	1	0	0	0	0	1	0	1	0	0	
0	0	0	0	1	0	0	0	0	1	0	1	0	0	
0	1	1	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0	
0	0	0	1	1	1	0	0	0	1	0	1	0	0	
0	0	0	1	1	1	0	0	0	1	0	1	0	0	

Schritt 1a: Abbildung auf binärem Raster

Zusammengesetzte Verdichtungsstreifen (green)  
 Durchgehende Verdichtungsstreifen (orange)

0	0	1	1	0	0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	0
1	0	1	1	1	1	1	0	1	1	1	1	1	0
0	0	0	0	1	1	1	0	1	1	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0
0	1	1	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	0	1	1	1	0	0	0	1	0	1	0	0

Schritt 1b: Ermitteln von Verdichtungsstreifen

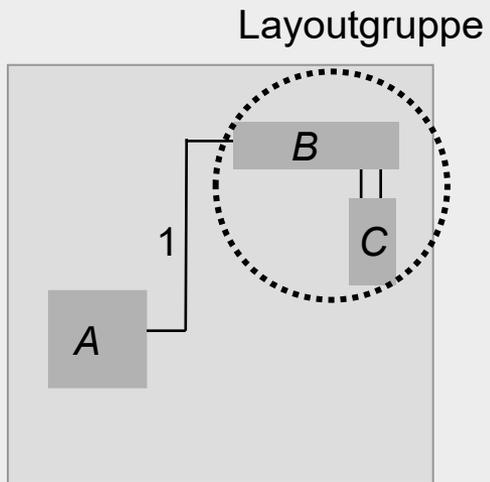


Schritt 1c: Entfernen der Verdichtungsstreifen  
 Schritt 3: Rückabbildung der Rastertopologie

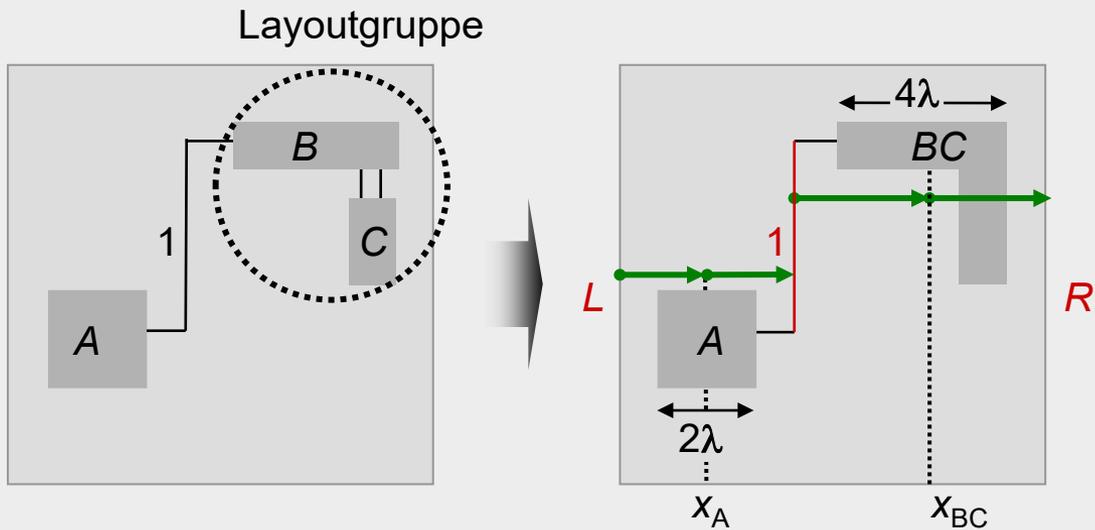
## 8.4.2 Abstandsgraph-Kompaktierung

- Die Abstandsgraph-Kompaktierung beruht auf der Suche nach dem längsten Pfad in einem Graphenmodell, welches die Nachbarschaft von Layoutelementen und deren Mindestabstände abbildet
- Eine **Layoutgruppe** umfasst jeweils die Layoutelemente, die gemeinsam bewegt werden müssen
- Ein **Abstandsgraph** ist ein gerichteter, kantenbewerteter Graph mit folgenden Eigenschaften:
  - Jedes Layoutelement entspricht einem Knoten.
  - Zwischen zwei Knoten befindet sich eine gerichtete Kante, sofern zwischen den Layoutelementen einzuhaltende Abstandsregeln gelten.
  - Die Kantengewichtung entspricht den Abstandsregeln der Kantenknoten.
  - Layoutränder sind ebenfalls als Knoten abzubilden.

Erzeugung eines horizontalen Abstandsgraphen

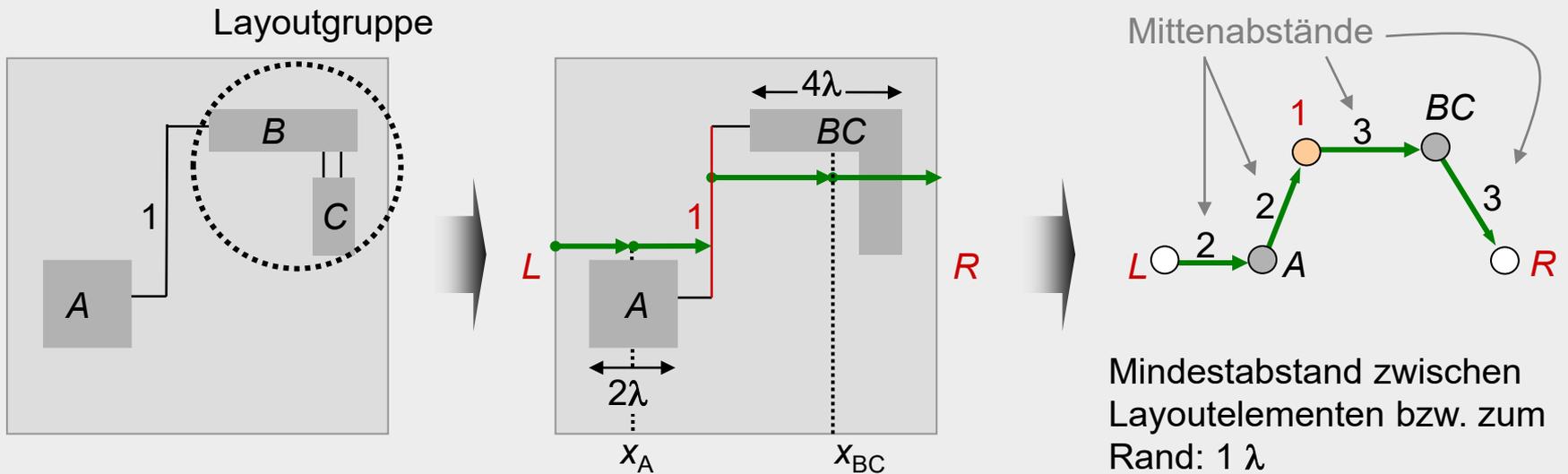


Erzeugung eines horizontalen Abstandsgraphen



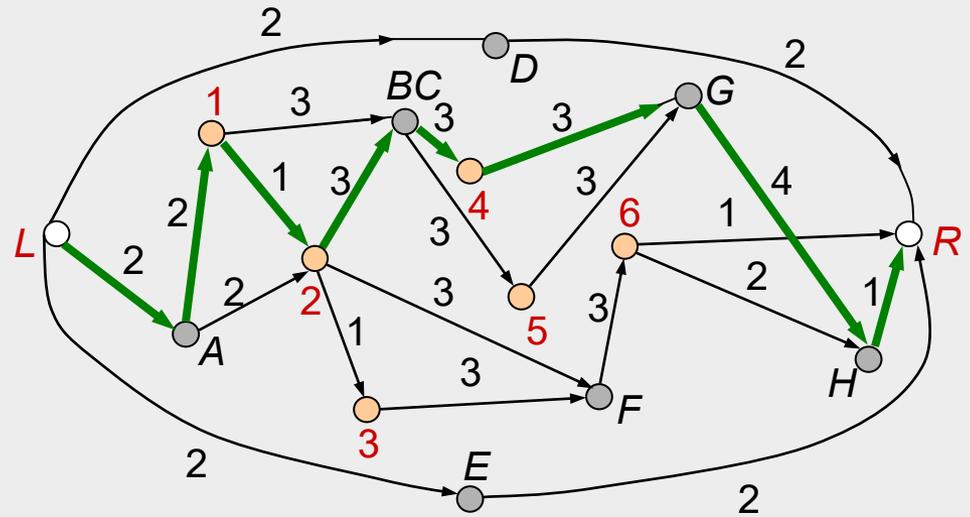
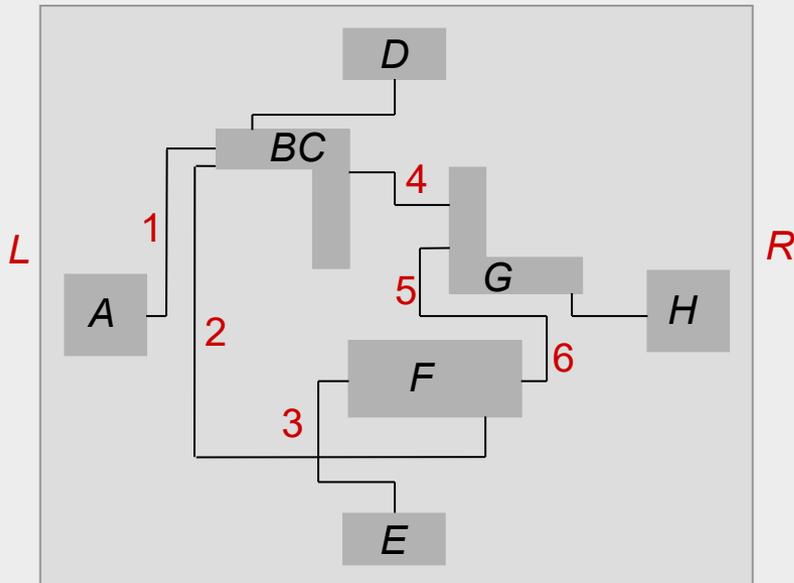
## 8.4.2 Abstandsgraph-Kompaktierung: Abstandsgraph

Erzeugung eines horizontalen Abstandsgraphen



Minimale Layoutbreite:  $2+2+3+3 = 10 \lambda$

## 8.4.2 Abstandsgraph-Kompaktierung: Abstandsgraph – Beispiel



➔ Längster Pfad von L nach R

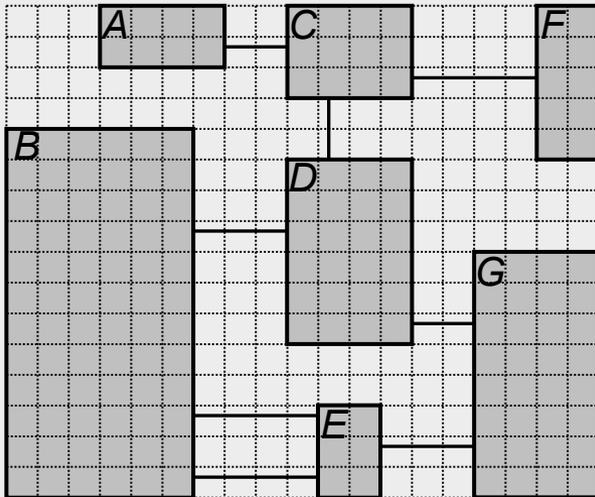
## 8.4.2 Abstandsgraph-Kompaktierung

- Bestimmung des längsten Pfades → minimale Layoutbreite bzw. -höhe
- Layoutelemente entlang dieses Graphen werden mit minimalem Abstand platziert → kompaktiertes Layout
- Layoutsegmente, die nicht auf dem längsten Pfad liegen: Mittelposition
- Für jeden Knoten  $v$  wird dazu eine untere, z.B. linke Grenze  $l(v)$ , und eine obere, z.B. rechte Grenze  $r(v)$ , seiner möglichen horizontalen Positionen ermittelt
  - $l(v) = (\text{längster Weg von linker Kante zum Knoten } v)$
  - $r(v) = (\text{längste Pfadlänge}) - (\text{längster Weg von Knoten } v \text{ zur rechten Kante})$

### Algorithmus

1. Zusammenfassung zu Layoutgruppen; Ordnen sämtlicher Kompaktierungsobjekte nach steigenden x-Koordinaten.
2. Erzeugung eines gerichteten horizontalen Abstandsgraphen  $G_H$ , wobei
  - a) horizontale Nachbarschaften als Kanten  $\{i, j\} \in G_H$  und
  - b) horizontale Kompaktierungsobjekte als Knoten  $(v = 1, 2, \dots, n) \in G_H$ ,
  - c) Mindestabstände als Kantengewichte  $d_{ij}$  abgebildet werden.
3. Berechnung der Grenzen  $l(v)$  und  $r(v)$  der zulässigen horizontalen Positionen jedes Knotens  $v$
4. Das durch den Knoten  $v$  abgebildete Kompaktierungsobjekt wird auf die Position  $x(v)$  gesetzt, mit
  - a)  $x(v) = l(v) = r(v)$  falls  $v$  im längsten Pfad liegt,
  - b)  $x(v) = [l(v) + r(v)] / 2$  falls  $v$  nicht im längsten Pfad liegt.
5. Weiter mit Schritt 1 zur Durchführung einer entgegengesetzten (vertikalen) Kompaktierung bzw. ENDE, falls in zwei Durchgängen keine Verschiebung.

## 8.4.2 Abstandsgraph-Kompaktierung: Beispiel



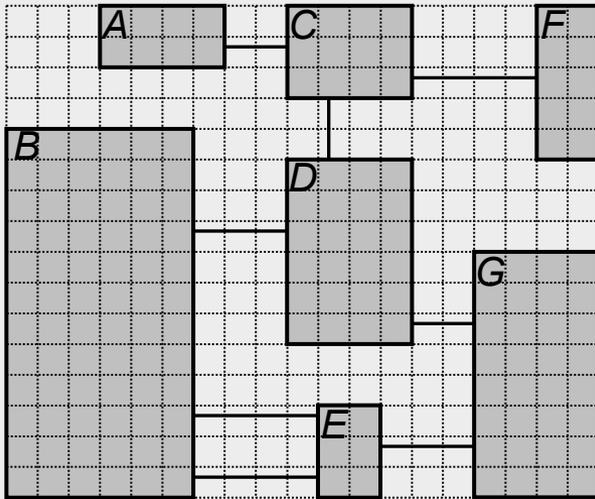
Gegeben:

Mindestabstand eine Gittereinheit ( $1 \lambda$ )

Minimale horizontale Mittenabstände:

$$\begin{array}{lll} d_{AC} = 5 & d_{AD} = 5 & d_{BC} = 6 \\ d_{BD} = 6 & d_{BE} = 5 & d_{CF} = 4 \\ d_{CG} = 5 & d_{DF} = 4 & d_{DG} = 5 \\ d_{EG} = 4 & & \end{array}$$

## 8.4.2 Abstandsgraph-Kompaktierung: Beispiel



Gegeben:

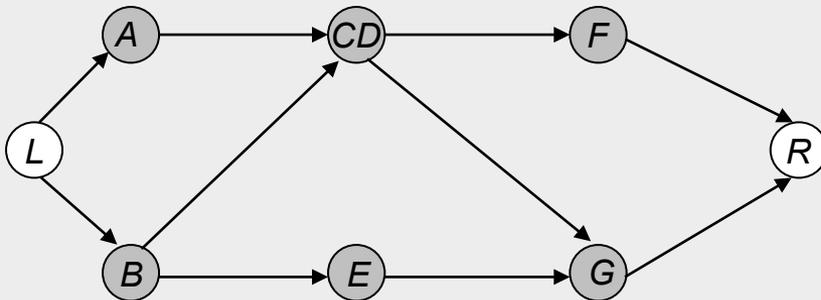
Mindestabstand eine Gittereinheit ( $1 \lambda$ )

Minimale horizontale Mittenabstände:

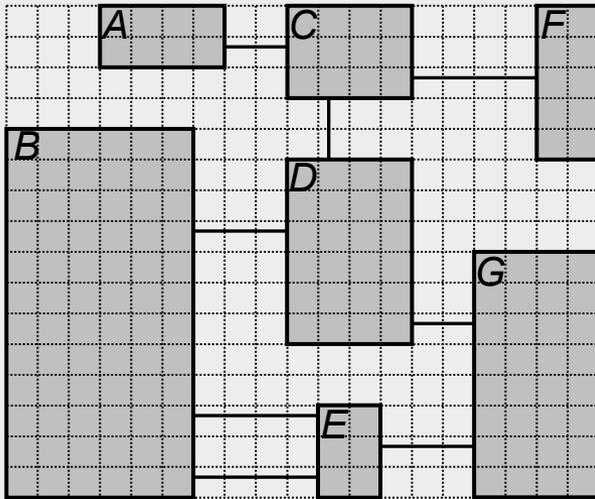
$$\begin{array}{lll}
 d_{AC} = 5 & d_{AD} = 5 & d_{BC} = 6 \\
 d_{BD} = 6 & d_{BE} = 5 & d_{CF} = 4 \\
 d_{CG} = 5 & d_{DF} = 4 & d_{DG} = 5 \\
 d_{EG} = 4 & & 
 \end{array}$$

**Schritte 1 und 2:** Erzeugen des horizontalen Abstandsgraphen mit längstem Pfad

a) Gruppierung von Elementen mit vertikalen Verbindungen: (C,D)



## 8.4.2 Abstandsgraph-Kompaktierung: Beispiel



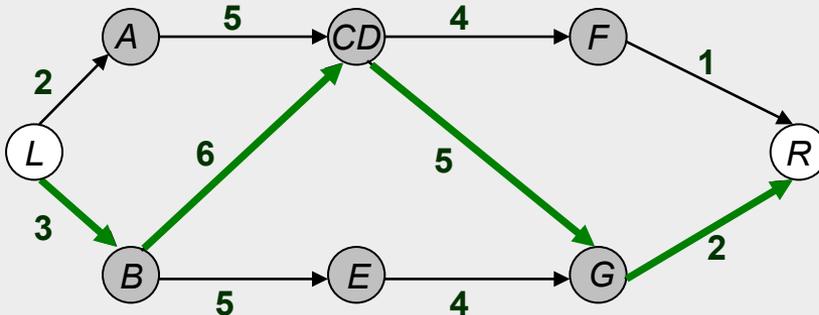
Gegeben:

Mindestabstand eine Gittereinheit ( $1 \lambda$ )

Minimale horizontale Mittenabstände:

$$\begin{array}{lll}
 d_{AC} = 5 & d_{AD} = 5 & d_{BC} = 6 \\
 d_{BD} = 6 & d_{BE} = 5 & d_{CF} = 4 \\
 d_{CG} = 5 & d_{DF} = 4 & d_{DG} = 5 \\
 d_{EG} = 4 & & 
 \end{array}$$

**Schritte 1 und 2:** Erzeugen des horizontalen Abstandsgraphen mit längstem Pfad



a) Gruppierung von Elementen mit vertikalen Verbindungen: (C,D)

b) Minimaler Mittenabstand  $d_{ij}$  bei Gruppen  $g_i$  und  $g_j$ :

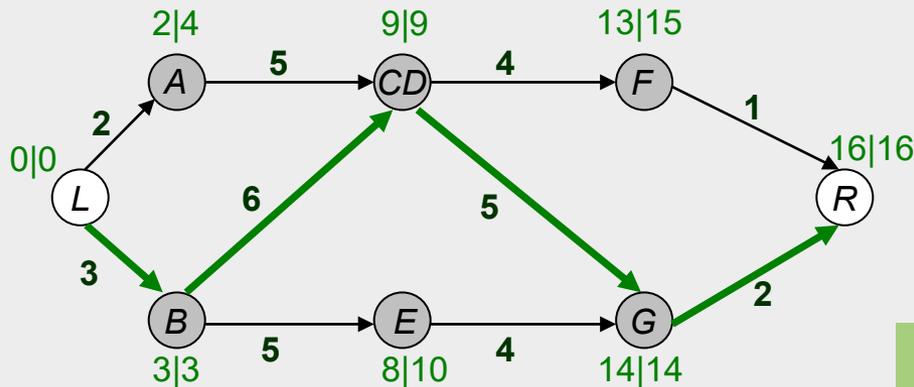
$$d_{ij} = \max_{a \in g_i, b \in g_j} (d_{ab})$$

$$\begin{aligned}
 \text{Beispiel: } d_{A,CD} &= \max(d_{AC}, d_{AD}) \\
 &= \max(5, 5) = 5
 \end{aligned}$$

c) Kennzeichnung des minimalen Mittenabstandes von Gruppen und Elementen durch **Kantengewichte**

## 8.4.2 Abstandsgraph-Kompaktierung: Beispiel

**Schritt 3:** Berechnung der Grenzen  $l(v)$  und  $r(v)$  jedes Knotens  $v$



$$r(v) = (\text{längste Pfadlänge}) - \text{LengthToR}(v)$$

Knoten R:	$l(R) = 16$	$\text{LengthToR}(R) = 0$	$r(R) = 16 - 0 = 16$
Knoten F:	$l(F) = 13$	$\text{LengthToR}(F) = 1$	$r(F) = 16 - 1 = 15$
Knoten G:	$l(G) = 14$	$\text{LengthToR}(G) = 2$	$r(G) = 16 - 2 = 14$
Knoten CD:	$l(CD) = 9$	$\text{LengthToR}(CD) = 7$	$r(CD) = 16 - 7 = 9$
Knoten E:	$l(E) = 8$	$\text{LengthToR}(E) = 6$	$r(E) = 16 - 6 = 10$
Knoten A:	$l(A) = 2$	$\text{LengthToR}(A) = 12$	$r(A) = 16 - 12 = 4$
Knoten B:	$l(B) = 3$	$\text{LengthToR}(B) = 13$	$r(B) = 16 - 13 = 3$
Knoten L:	$l(L) = 0$	$\text{LengthToR}(L) = 16$	$r(L) = 16 - 16 = 0$

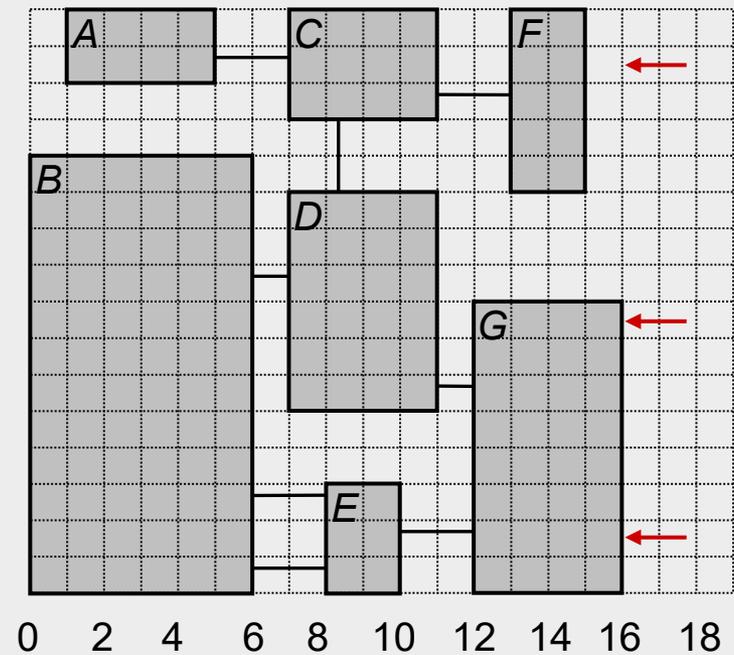
$l(v)$  = längste Pfadlänge von L zu v

$\text{LengthToR}(v)$  = längste Pfadlänge von v zu R

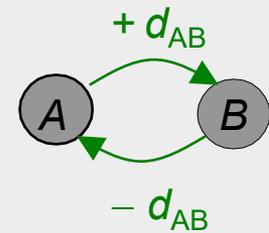
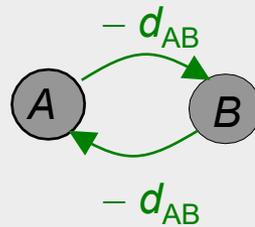
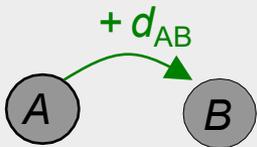
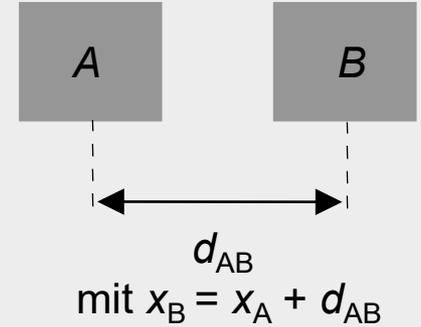
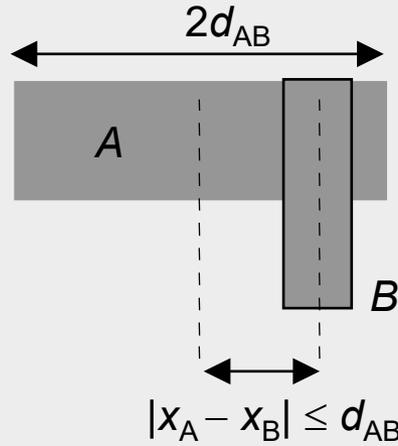
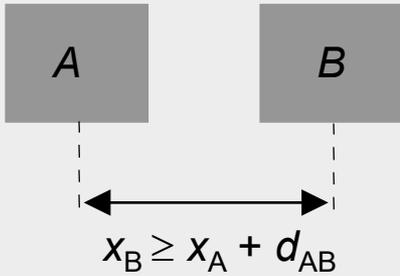
## 8.4.2 Abstandsgraph-Kompaktierung: Beispiel

**Schritt 4:** Berechnung von  $x(v)$  und Anpassung des Layouts

Knoten $R$ :	$l(R) = 16$	$r(R) = 16$	$x(R) = 16$
Knoten $F$ :	$l(F) = 13$	$r(F) = 15$	$x(F) = 14$
Knoten $G$ :	$l(G) = 14$	$r(G) = 14$	$x(G) = 14$
Knoten $CD$ :	$l(CD) = 9$	$r(CD) = 9$	$x(CD) = 9$
Knoten $E$ :	$l(E) = 8$	$r(E) = 10$	$x(E) = 9$
Knoten $A$ :	$l(A) = 2$	$r(A) = 4$	$x(A) = 3$
Knoten $B$ :	$l(B) = 3$	$r(B) = 3$	$x(B) = 3$
Knoten $L$ :	$l(L) = 0$	$r(L) = 0$	$x(L) = 0$



# 8.4.2 Abstandsgraph-Kompaktierung: Modifikationen



Minimaler Mittenabstand

Maximaler Mittenabstand

Fester Mittenabstand

- 8.1 Einführung
- 8.2 Begriffe, Modelle, Datenstrukturen
- 8.3 Symbolische Kompaktierung
- 8.4 Kompaktierungsalgorithmen
  - 8.4.1 Schnittkompaktierung
  - 8.4.2 Abstandsgraph-Kompaktierung

# Zusammenfassung Entwurfsautomatisierung

