

Gates

Selected Fun Problems of the ACM Programming Contest
(Proseminar)

Tobias Meggendorfer

Technische Universität München
Fakultät für Informatik

April 22, 2009

Ausgangssituation

- Chiphersteller will die Effizienz seiner Produkte steigern

Ausgangssituation

- Chiphersteller will die Effizienz seiner Produkte steigern
- Dies wird durch die Optimierung der NAND Gatter Netze gewährleistet

Ausgangssituation

Definition: NAND Gatter

Ausgangssituation

Definition: NAND Gatter

- 2 Eingänge, 1 Ausgang

Ausgangssituation

Definition: NAND Gatter

- 2 Eingänge, 1 Ausgang
- Negierte AND-Verknüpfung (! (X && Y))

Ausgangssituation

Definition: NAND Gatter

- 2 Eingänge, 1 Ausgang
- Negierte AND-Verknüpfung ($!(X \&\& Y)$)

- Belegungstabelle:

X	Y	X NAND Y
0	0	1
0	1	0
1	0	0
1	1	0

Ausgangssituation

Definition: NAND Gatter

- 2 Eingänge, 1 Ausgang
- Negierte AND-Verknüpfung ($!(X \&\& Y)$)

X	Y	X NAND Y
0	0	1
0	1	0
1	0	0
1	1	0

- Belegungstabelle:
- Standardbaustein in der Digitaltechnik, da alle logischen Verknüpfungen nachgebildet werden können

Ausgangssituation

- Chiphersteller will die Effizienz seiner Produkte steigern
- Dies wird durch die Optimierung der NAND Gatter Netze gewährleistet

Ausgangssituation

- Chiphersteller will die Effizienz seiner Produkte steigern
- Dies wird durch die Optimierung der NAND Gatter Netze gewährleistet
- Die Netze können nicht geändert werden

Ausgangssituation

- Chiphersteller will die Effizienz seiner Produkte steigern
- Dies wird durch die Optimierung der NAND Gatter Netze gewährleistet
- Die Netze können nicht geändert werden
- Daher muss die optimale Belegung der Eingänge gefunden werden

Ausgangssituation

Definition: Optimale Belegung von Eingängen

Ausgangssituation

Definition: Optimale Belegung von Eingängen

- So wenig Eingänge wie möglich, trotzdem das gleiche Ergebnis

Ausgangssituation

- Chiphersteller will die Effizienz seiner Produkte steigern
- Dies wird durch die Optimierung der NAND Gatter Netze gewährleistet
- Die Netze können nicht geändert werden
- Daher muss die optimale Belegung der Eingänge gefunden werden

Aufgabenstellung

- Problem:

Aufgabenstellung

- Problem:
 - Die Netze können extrem komplex werden

Aufgabenstellung

- Problem:
 - Die Netze können extrem komplex werden
 - Die Optimierung ist daher von Menschenhand nicht durchführbar

Aufgabenstellung

- Problem:
 - Die Netze können extrem komplex werden
 - Die Optimierung ist daher von Menschenhand nicht durchführbar
- *Aufgabenstellung*: Optimierung durch ein Programm

Allgemeine Form der Ein- und Ausgabe

Eingabe

- n Netze, $n \leq 20$

Allgemeine Form der Ein- und Ausgabe

Eingabe

- n Netze, $n \leq 20$
- m Eingaben, $m \leq 100.000$

Allgemeine Form der Ein- und Ausgabe

Eingabe

- n Netze, $n \leq 20$
- m Eingaben, $m \leq 100.000$
- o Gatter, $o \leq 200.000$

Allgemeine Form der Ein- und Ausgabe

Eingabe

- n Netze, $n \leq 20$
- m Eingaben, $m \leq 100.000$
- o Gatter, $o \leq 200.000$
- Aufbau des Netzes ($2*o$ Zeichen)

Allgemeine Form der Ein- und Ausgabe

Eingabe

- n Netze, $n \leq 20$
- m Eingaben, $m \leq 100.000$
- o Gatter, $o \leq 200.000$
- Aufbau des Netzes ($2*o$ Zeichen)

Ausgabe

Allgemeine Form der Ein- und Ausgabe

Eingabe

- n Netze, $n \leq 20$
- m Eingaben, $m \leq 100.000$
- o Gatter, $o \leq 200.000$
- Aufbau des Netzes ($2*o$ Zeichen)

Ausgabe

- Optimale Belegung der Eingaben (n Zeilen, m Zeichen / Zeile)

Eingabeformat

```
1
3 6
-1 -2 -1 -3 1 2 1 2 4 3 5 5
```

Eingabeformat

1

3 6

-1 -2 -1 -3 1 2 1 2 4 3 5 5

Eingabeformat

1

3 6

-1 -2 -1 -3 1 2 1 2 4 3 5 5

x

x

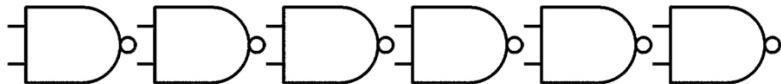
x

Eingabeformat

1

3 **6**

-1 -2 -1 -3 1 2 1 2 4 3 5 5



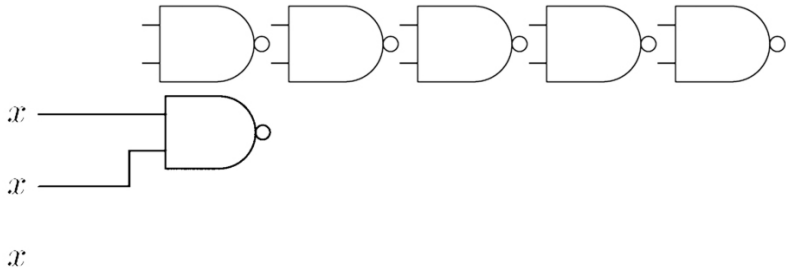
x

x

x

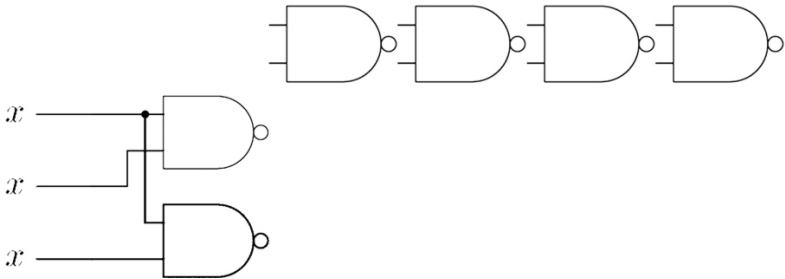
Eingabeformat

1
3 6
-1 -2 -1 -2 1 2 1 2 4 3 5 5



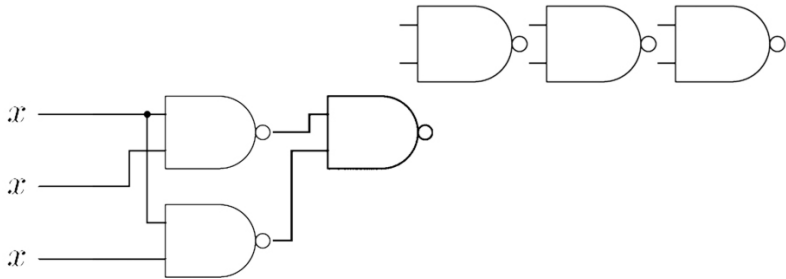
Eingabeformat

1
3 6
-1 -2 **-1** **-3** 1 2 1 2 4 3 5 5



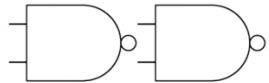
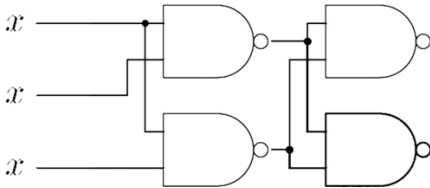
Eingabeformat

1
3 6
-1 -2 -1 -3 **1 2** 1 2 4 3 5 5



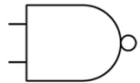
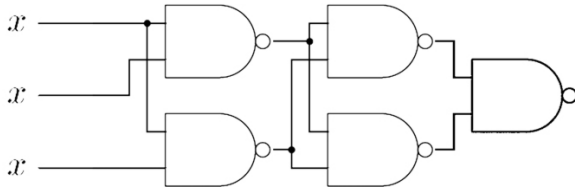
Eingabeformat

1
3 6
-1 -2 -1 -3 1 2 **1 2** 4 3 5 5



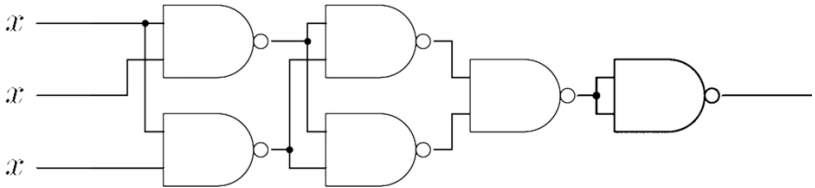
Eingabeformat

1
3 6
-1 -2 -1 -3 1 2 1 2 **4 3** 5 5



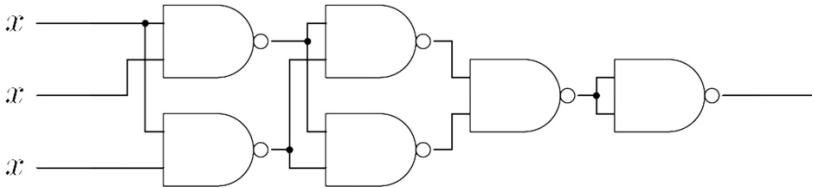
Eingabeformat

1
3 6
-1 -2 -1 -3 1 2 1 2 4 3 **5 5**



Eingabeformat

1
3 6
-1 -2 -1 -3 1 2 1 2 4 3 5 5

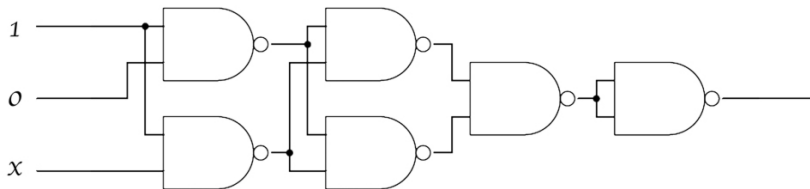


Ausgabeformat

10x

Ausgabeformat

10x



Allgemeiner Ansatz

Ansatz

- Lösen per Bruteforce, d.h. alle möglichen Belegungen auswerten

Allgemeiner Ansatz

Ansatz

- Lösen per Bruteforce, d.h. alle möglichen Belegungen auswerten
- Jede gefundene Belegung wird als Objekt gespeichert

Umsetzung des Netzes und dessen Auswertung

Netz

- Jedes Gatter und jeder Eingang wird durch ein Objekt repräsentiert

Umsetzung des Netzes und dessen Auswertung

Netz

- Jedes Gatter und jeder Eingang wird durch ein Objekt repräsentiert

Auswertung

- Der Wert eines Gatters ist durch die NAND-Verknüpfung der Werte der verbundenen Gatter festgelegt

Umsetzung des Netzes und dessen Auswertung

Netz

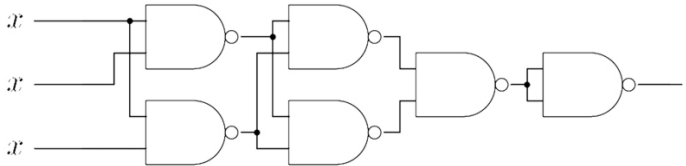
- Jedes Gatter und jeder Eingang wird durch ein Objekt repräsentiert

Auswertung

- Der Wert eines Gatters ist durch die NAND-Verknüpfung der Werte der verbundenen Gatter festgelegt
- Die Werte der Eingaben werden geändert und der Wert des letzten Gatters abgerufen

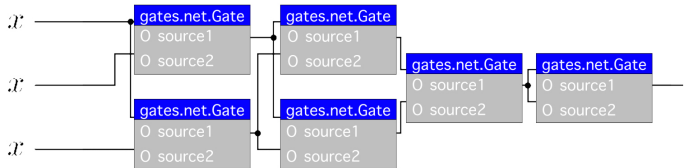
Beispiel: Umsetzung des Netzes

Ausgangssituation



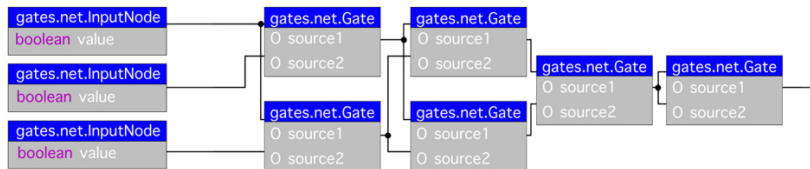
Beispiel: Umsetzung des Netzes

Umsetzung der Gatter



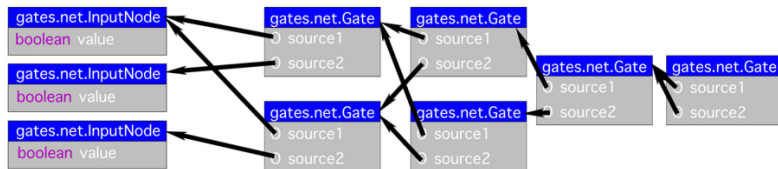
Beispiel: Umsetzung des Netzes

Umsetzung der Eingaben



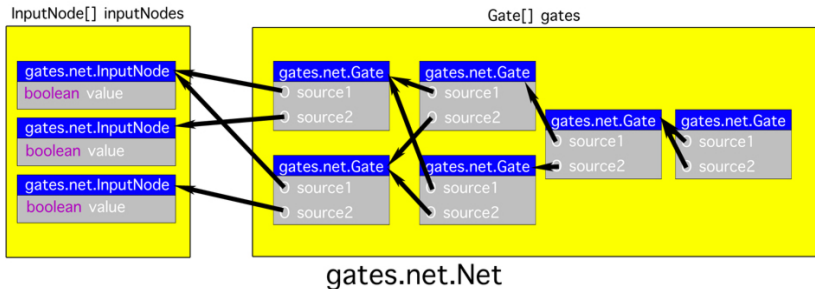
Beispiel: Umsetzung des Netzes

Umsetzung der Vernetzung



Beispiel: Umsetzung des Netzes

Endgültige Umsetzung im Programm



Mögliche Schwierigkeiten bei der Implementierung

- Viele zu testende Belegungen ($3^{\text{Anzahl der Eingaben}}$)

Mögliche Schwierigkeiten bei der Implementierung

- Viele zu testende Belegungen ($3^{\text{Anzahl der Eingaben}}$)
- Großer Speicheraufwand

Wahl der Programmiersprache

Warum Java?

Vorteile von objektorientierten Sprachen

- Einfache Realisierung des Netzes durch Objekte und Zeiger

Wahl der Programmiersprache

Warum Java?

Vorteile von objektorientierten Sprachen

- Einfache Realisierung des Netzes durch Objekte und Zeiger

Besondere Vorteile von Java

- Eigene Erfahrung

Wahl der Programmiersprache

Warum Java?

Vorteile von objektorientierten Sprachen

- Einfache Realisierung des Netzes durch Objekte und Zeiger

Besondere Vorteile von Java

- Eigene Erfahrung
- **Garbage-Collector:** Nicht mehr referenzierte Objekte werden automatisch gelöscht (Hilfreich bei mehreren Netzen)

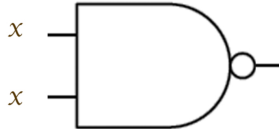
Der algorithmische Kern

Aktuelle Belegung:
Gefundene Belegungen:

Aktueller Wert mit 0:



Aktueller Wert mit 1:



Der algorithmische Kern

Aktuelle Belegung: 0 0

Gefundene Belegungen:

Aktueller Wert mit 0: 1



Aktueller Wert mit 1: 1



Der algorithmische Kern

Aktuelle Belegung: 1 0

Gefundene Belegungen:

Aktueller Wert mit 0: 0



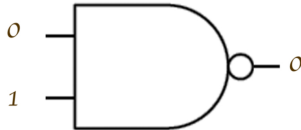
Aktueller Wert mit 1: 0



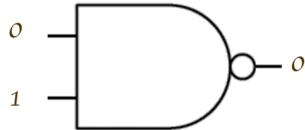
Der algorithmische Kern

Aktuelle Belegung: 0 1
Gefundene Belegungen:

Aktueller Wert mit 0: 0



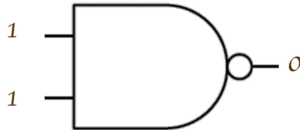
Aktueller Wert mit 1: 0



Der algorithmische Kern

Aktuelle Belegung: 1 1
Gefundene Belegungen:

Aktueller Wert mit 0: 0



Aktueller Wert mit 1: 0



Der algorithmische Kern

Aktuelle Belegung: x 0

Gefundene Belegungen: **x 0**

Aktueller Wert mit 0: 1



Aktueller Wert mit 1: 0

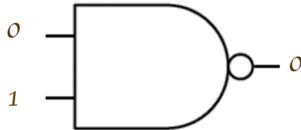


Der algorithmische Kern

Aktuelle Belegung: x 1

Gefundene Belegungen: x 0

Aktueller Wert mit 0: 0



Aktueller Wert mit 1: 0

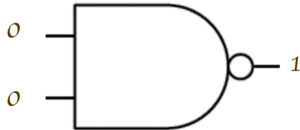


Der algorithmische Kern

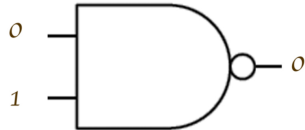
Aktuelle Belegung: 0 x

Gefundene Belegungen: x 0 , **0 x**

Aktueller Wert mit 0: 1



Aktueller Wert mit 1: 0

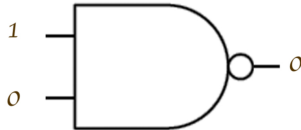


Der algorithmische Kern

Aktuelle Belegung: 1 x

Gefundene Belegungen: x 0 , 0 x

Aktueller Wert mit 0: 0



Aktueller Wert mit 1: 0

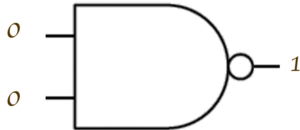


Der algorithmische Kern

Aktuelle Belegung: $x \ x$

Gefundene Belegungen: $x \ 0$, $0 \ x$, **$x \ x$**

Aktueller Wert mit 0: 1



Aktueller Wert mit 1: 0

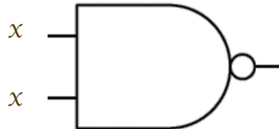


Der algorithmische Kern

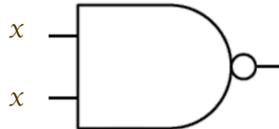
Aktuelle Belegung:

Gefundene Belegungen: x 0 , 0 x , x x

Aktueller Wert mit 0:



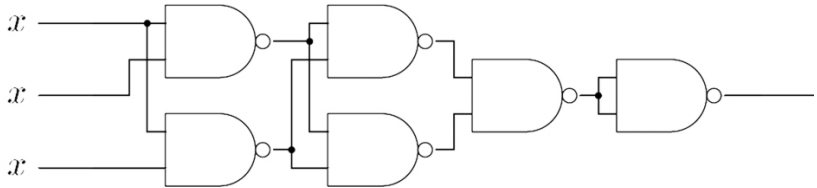
Aktueller Wert mit 1:



Demo

Demo

1
3 6
-1 -2 -1 -3 1 2 1 2 4 3 5 5





Fragen, Unklarheiten, Anmerkungen?



Fragen, Unklarheiten, Anmerkungen?

**Danke für die
Aufmerksamkeit**