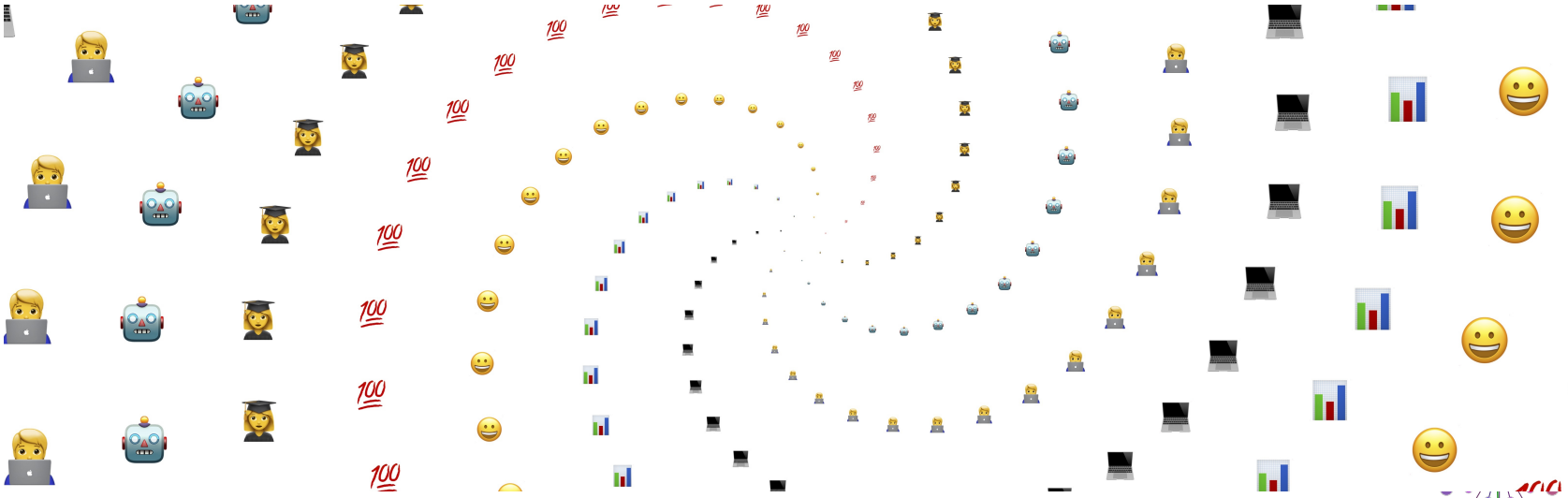
Datenintegrität

Fehler automatisch erkennen und korrigieren

Cyril Blum

Fachschaft Informatik
Kantonsschule im Lee



Rückblick: letztes Mal

- Beim Übertragen von Daten können Fehler auftreten
- Ziel: erkennen, wann eine Nachricht fehlerhaft ist
- Idee: jede Bitfolge steht für eine Nachricht

Nachricht	**Codewort**
`Rot`	`000`
`Blau`	`011`
`Grün`	`101`

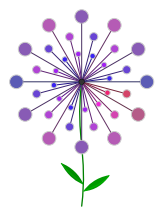
Wie viele Fehler erkennt diese Kodierung?

Rückblick: letztes Mal

- Beim Übertragen von Daten können Fehler auftreten
- Ziel: erkennen, wann eine Nachricht fehlerhaft ist
- Idee: jede Bitfolge steht für eine Nachricht

Nachricht	**Codewort**
`Rot`	`000`
`Blau`	`011`
`Grün`	`101`

1 Bitfehler (z.B. `000` $\rightarrow$ `001`, **ungültig!**)

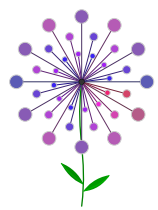


Rückblick: letztes Mal

- Beim Übertragen von Daten können Fehler auftreten
- Ziel: erkennen, wann eine Nachricht fehlerhaft ist
- Idee: jede Bitfolge steht für eine Nachricht

Nachricht	**Codewort**
`Rot`	`000`
`Blau`	`011`
`Grün`	`101`

Was ist der **Abstand** dieser Kodierung?

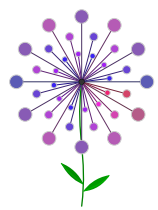


Rückblick: letztes Mal

- Beim Übertragen von Daten können Fehler auftreten
- Ziel: erkennen, wann eine Nachricht fehlerhaft ist
- Idee: jede Bitfolge steht für eine Nachricht

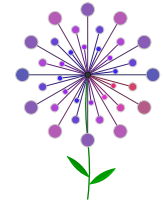
Nachricht	**Codewort**
`Rot`	`000`
`Blau`	`011`
`Grün`	`101`

Der **Abstand einer Kodierung** ist die minimale Distanz zwischen zwei Codewörtern (in diesem Fall 2).



Wichtige Erkenntnis

Eine Kodierung ist k-fehlererkennend, wenn die Kodierung den Mindest-Abstand $k+1$ hat.



Wichtige Erkenntnis

Eine Kodierung ist k-fehlererkennend, wenn die Kodierung den Mindest-Abstand $k+1$ hat.

Z.B.: Die Kodierung 000, 011 und 101 hat den kleinsten Abstand 2, also ist die Kodierung 1-fehlererkennend.

Aufgabe 0.1

Finden Sie für die drei Nachrichten **Rot**, **Blau** und **Grün** eine Kodierung aus Bitfolgen der Länge 5, die **1-fehlerkorrigierend** ist.

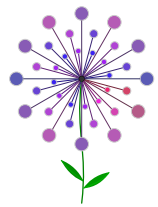
D.h., wenn ein Fehler vorkommt, weiss man nicht nur, dass die Nachricht fehlerhaft ist, sondern man kann das ursprüngliche Codewort aus der fehlerhaften Bitfolge eindeutig rekonstruieren.

Der Abstand zwischen *allen* Codewörtern sollte möglichst gross sein!

Information	**Code**
Rot	00000
Blau	??
Grün	??

Mögliche Lösung

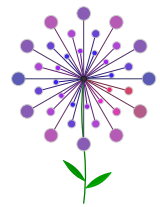
Information	**Code**
Rot	00000
Blau	01110
Grün	10101



1-fehlerkorrigierende Kodierungen

- Ziel: Nicht nur Fehler erkennen, sondern auch **automatisch korrigieren**
- Eine Kodierung ist **1-fehlerkorrigierend**, wenn sie jeden einzelnen Bitfehler beheben kann
- **Voraussetzung:** Der Abstand der Kodierung ist mindestens 3

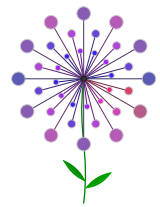
Eine Kodierung ist k-fehlerkorrigierend, wenn der Mindestabstand mindestens $k + 2$ beträgt.



Visualisierung

Gibt es auch eine 1-Fehlerkorrigierende Kodierung mit drei Code-Wörtern der Länge 4?

$\rightarrow$ **Visualisierung schafft Abhilfe**



2D-Hyperwürfel

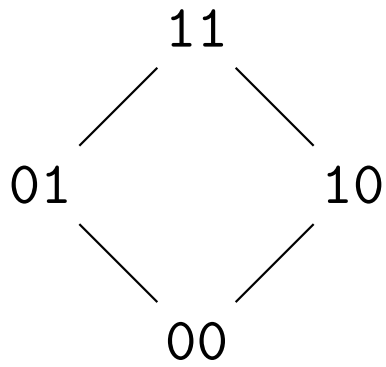


Abbildung: 2D-Hyperwürfel

3D-Hyperwürfel

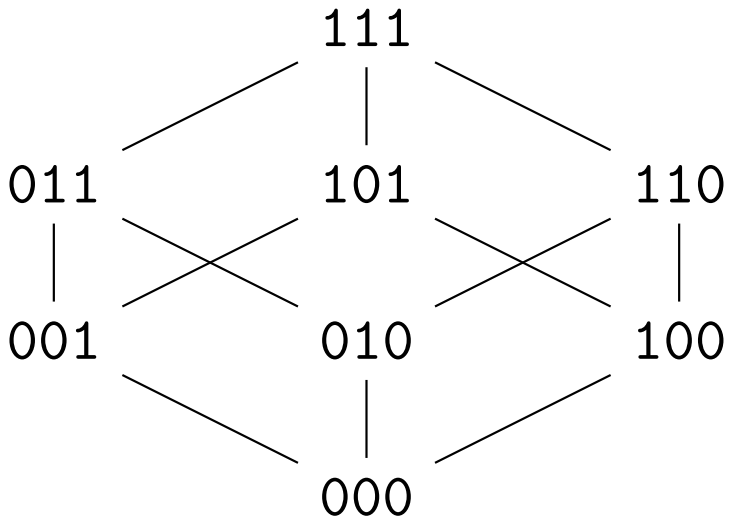


Abbildung: 3D-Hyperwürfel

4D-Hyperwürfel

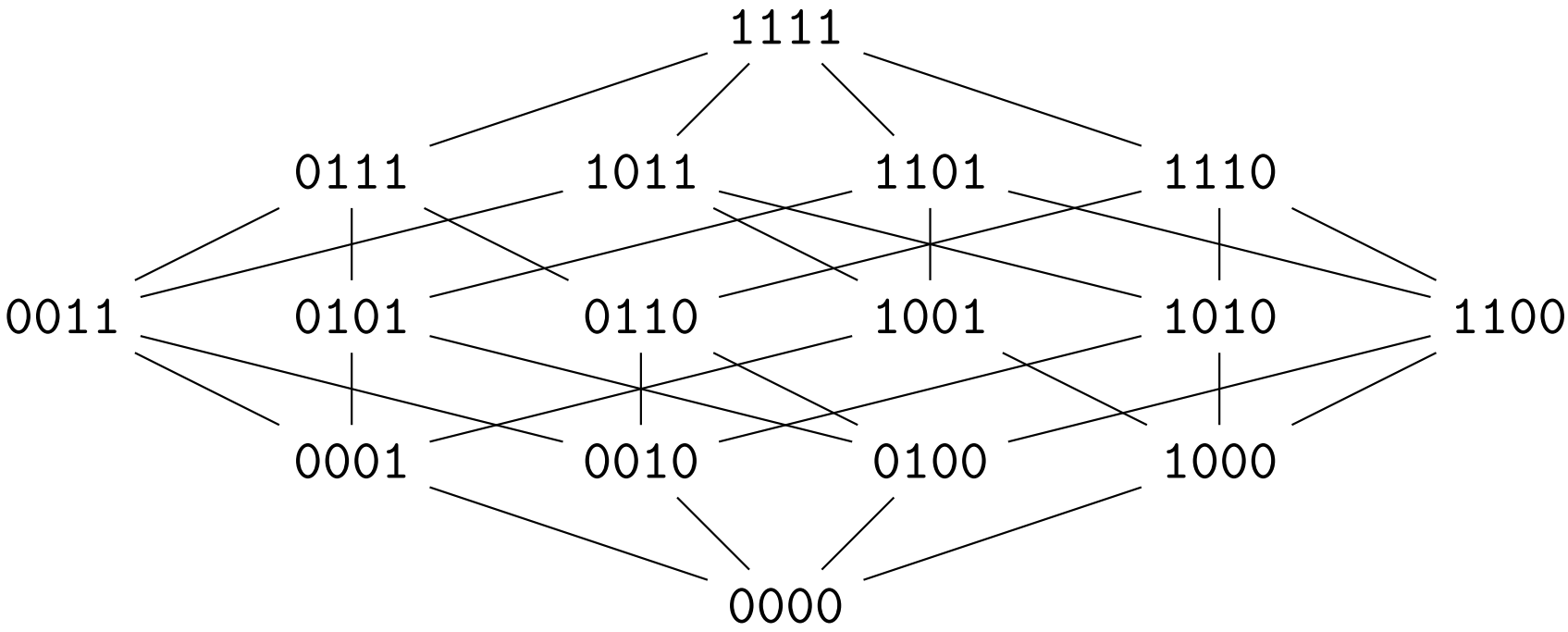
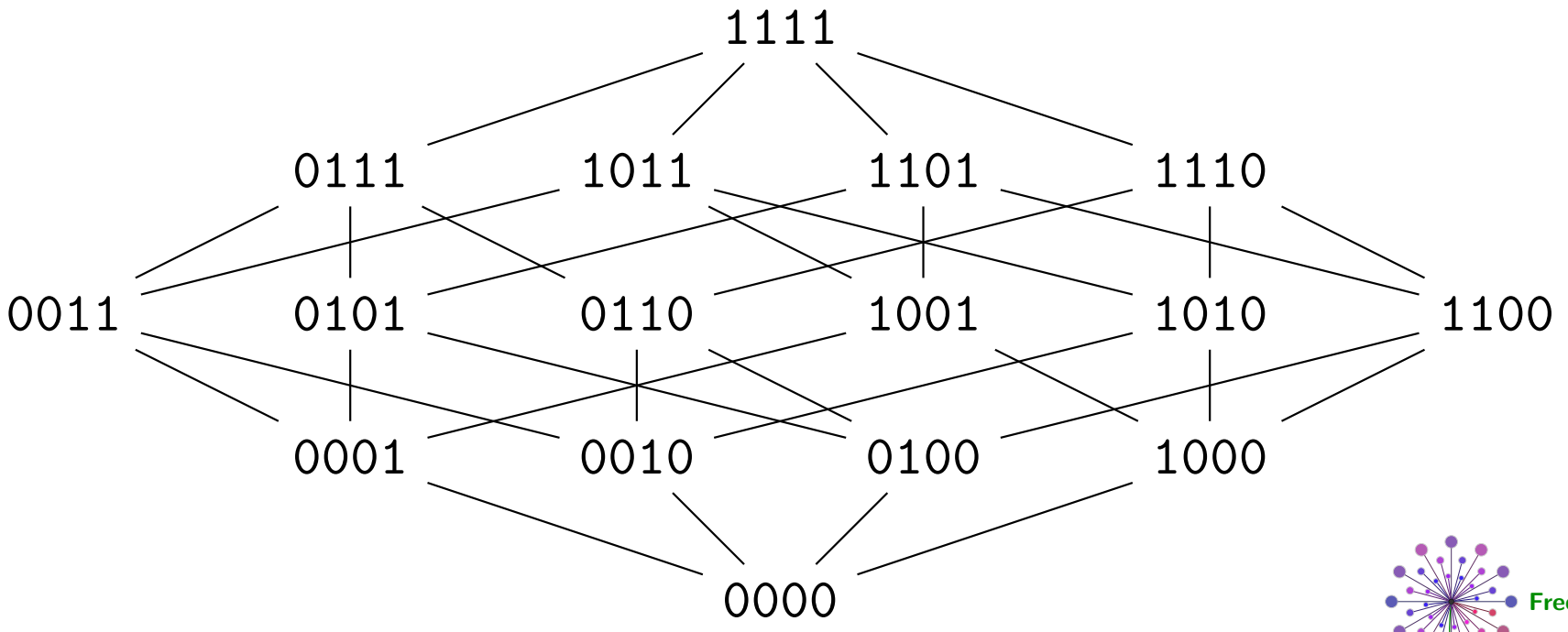


Abbildung: 4D-Hyperwürfel

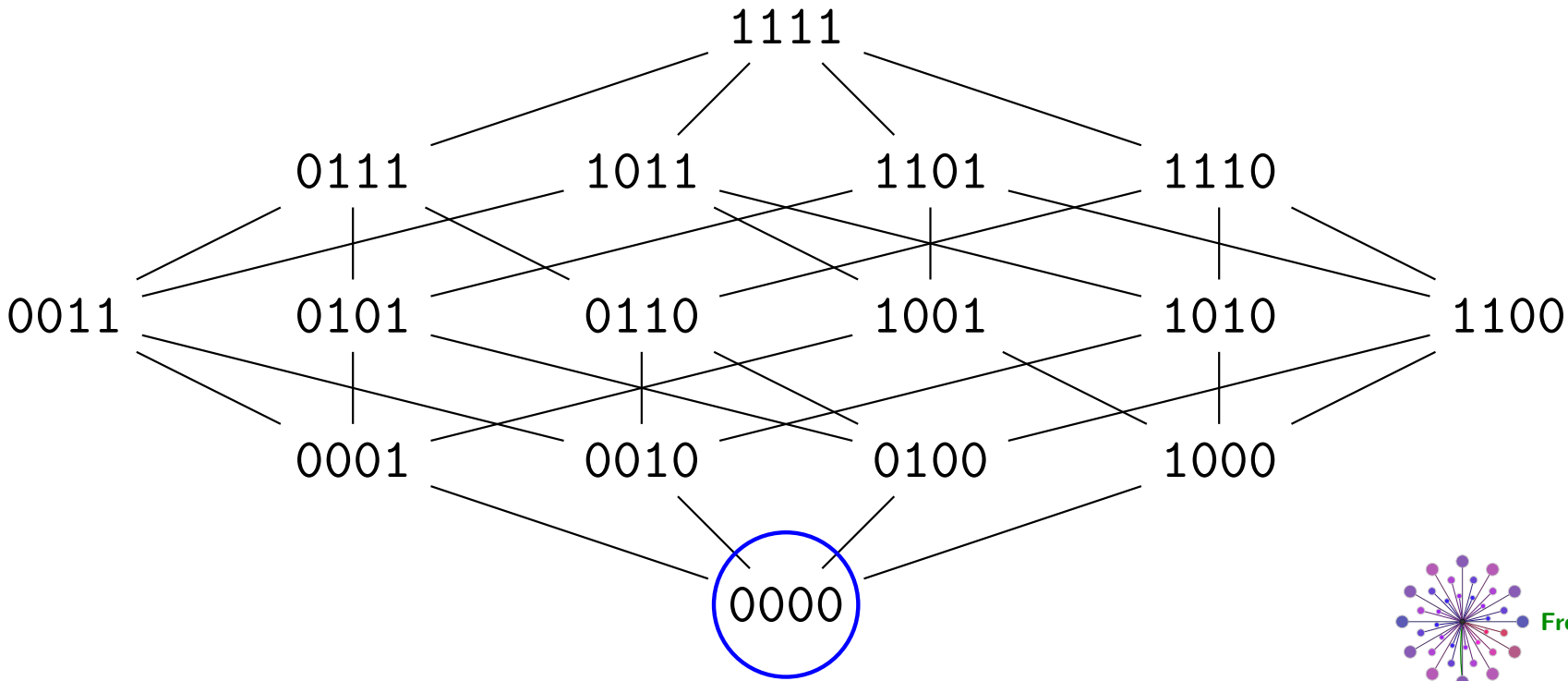
Schritt-für-Schritt-Suche nach 1-fehlerkorrigierender Kodierung (Länge 4)

Eine 1-fehlerkorrigierende Kodierung der Länge 4 muss Mindest-Abstand 3 haben.



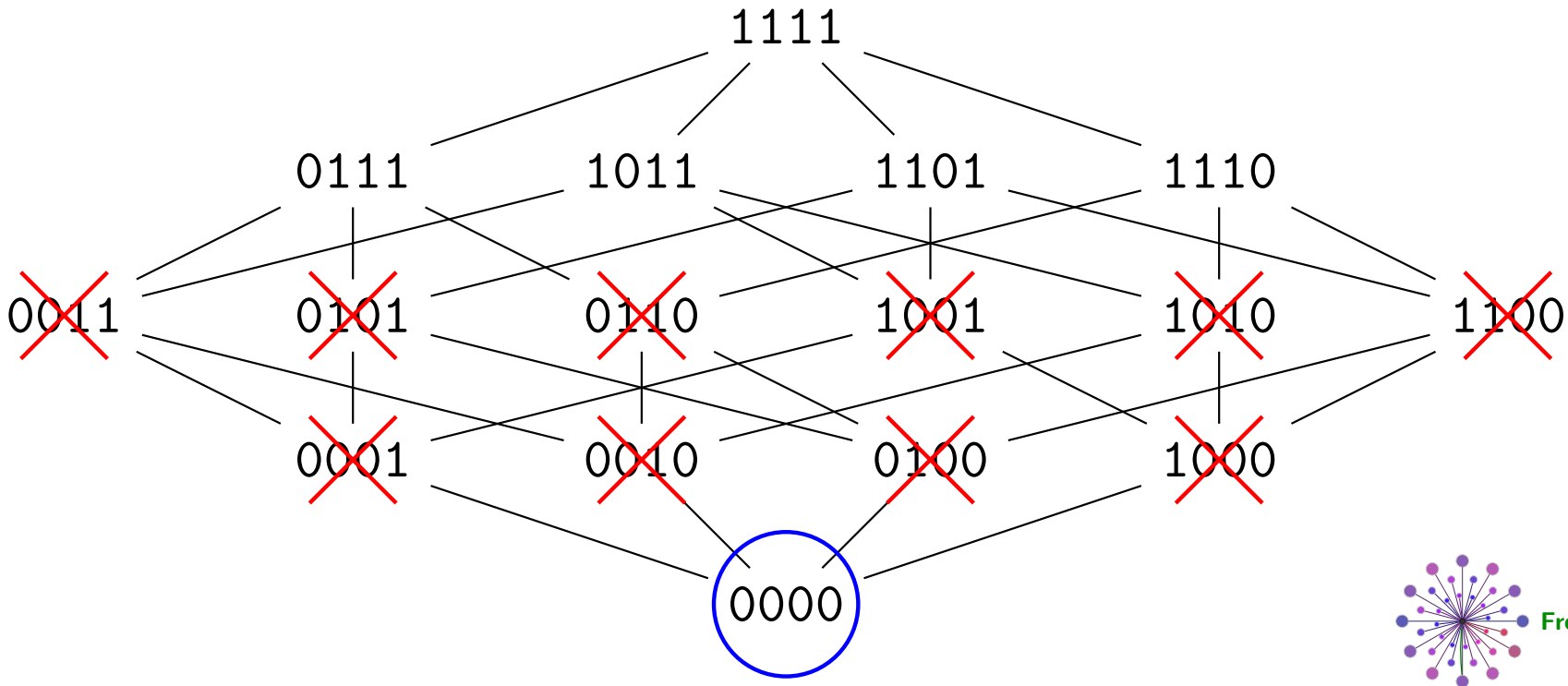
Schritt-für-Schritt-Suche nach 1-fehlerkorrigierender Kodierung (Länge 4)

Eine 1-fehlerkorrigierende Kodierung der Länge 4 muss Mindest-Abstand 3 haben.



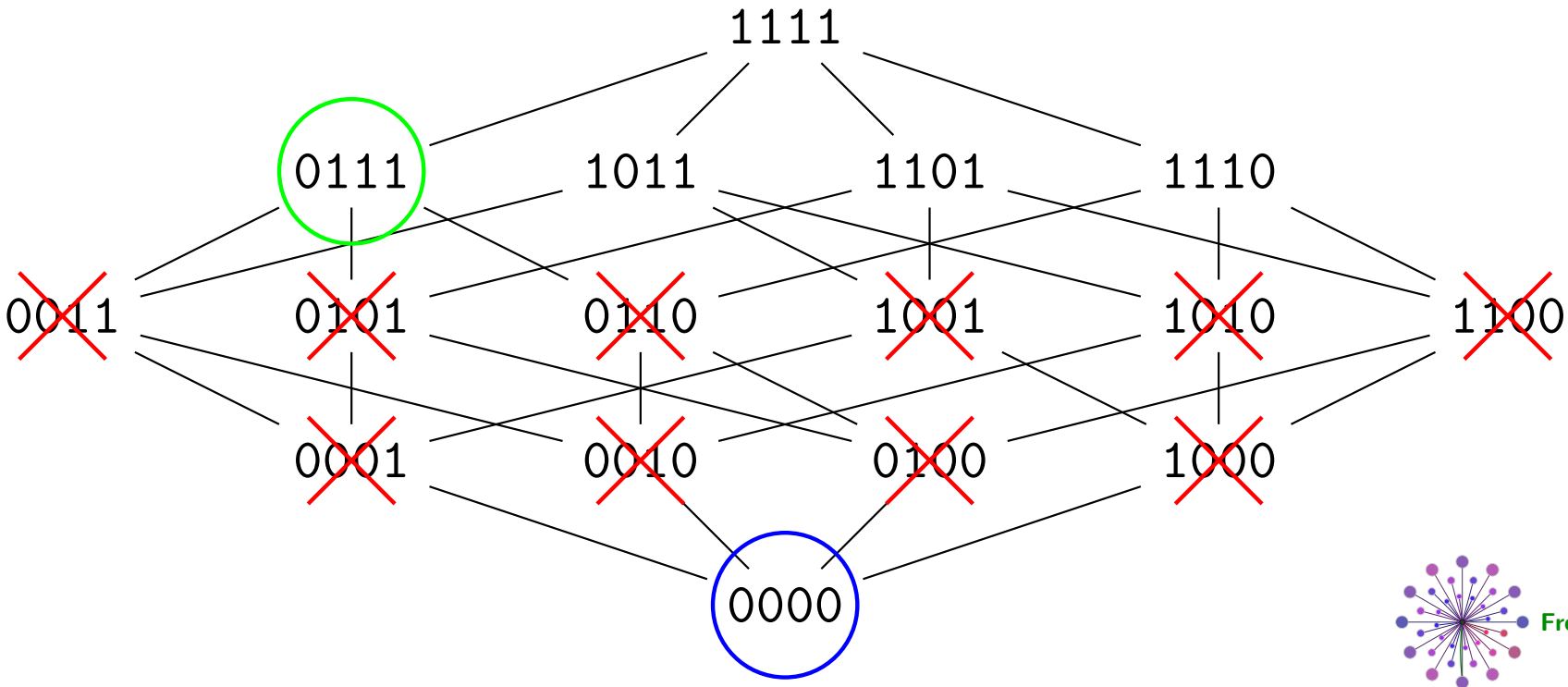
Schritt-für-Schritt-Suche nach 1-fehlerkorrigierender Kodierung (Länge 4)

Eine 1-fehlerkorrigierende Kodierung der Länge 4 muss Mindest-Abstand 3 haben.



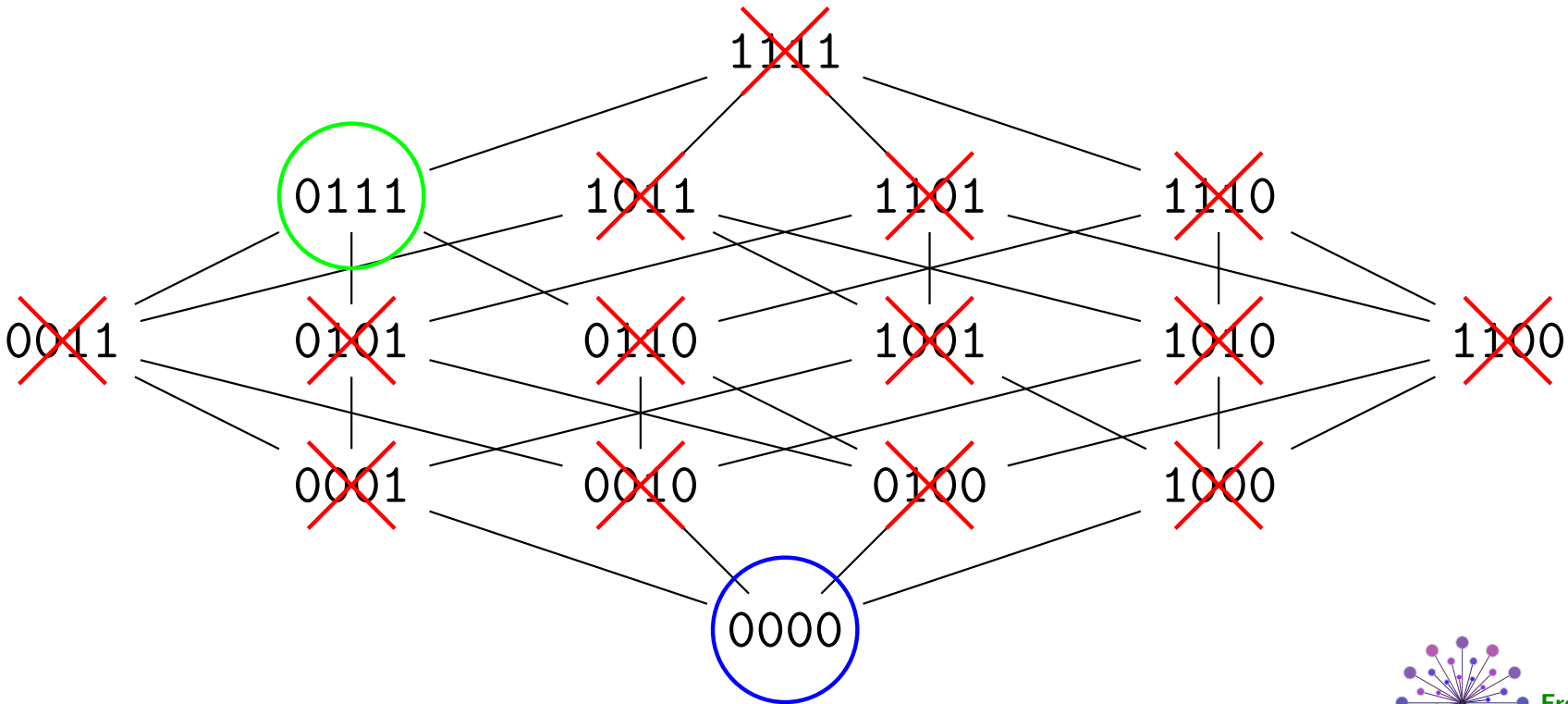
Schritt-für-Schritt-Suche nach 1-fehlerkorrigierender Kodierung (Länge 4)

Eine 1-fehlerkorrigierende Kodierung der Länge 4 muss Mindest-Abstand 3 haben.



Schritt-für-Schritt-Suche nach 1-fehlerkorrigierender Kodierung (Länge 4)

Eine 1-fehlerkorrigierende Kodierung der Länge 4 muss Mindest-Abstand 3 haben.

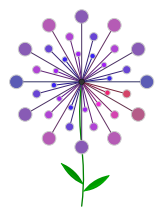


Maximal zwei gültige Codewörter!

Auftrag

Skript

- Definition 1.1 lesen
- 1.13-1.17 (Challenges erst nachher)



FreeFlower