



**educational engineering lab**

Department for Information Technology  
University of Zurich

# Aufgaben

--	--	--

Name

Vorname

Matrikelnummer

**Assessment-Prüfung**

**Formale Grundlagen  
der Informatik I**

## Codes und Informationsgehalt

**Auf wie viele Dezimalziffern genau können vorzeichenlose ganze Zahlen in einem binären Code der Länge 32 bit dargestellt werden?**

## Codes und Informationsgehalt

Auf wie viele Dezimalziffern genau können vorzeichenlose ganze Zahlen in einem binären Code der Länge 32 bit dargestellt werden?

$$\lfloor 32 / \log_2 10 \rfloor = 9$$

## Codes und Informationsgehalt

**Wie gross ist der Informationsgehalt eines 8 bit langen Binärwortes, in dem genau 1 Bit Eins und die restlichen 7 Bit Null gesetzt sind?**

## Codes und Informationsgehalt

Wie gross ist der Informationsgehalt eines 8 bit langen Binärwortes, in dem genau 1 Bit Eins und die restlichen 7 Bit Null gesetzt sind?

**3 bit**

## Codes und Informationsgehalt

**Wie gross ist der Informationsgehalt eines 5 Bit langen Binärwortes, in dem genau 3 Bit Eins und die restlichen 2 Bit Null gesetzt sind?**

## Codes und Informationsgehalt

Wie gross ist der Informationsgehalt eines 5 Bit langen Binärwortes, in dem genau 3 Bit Eins und die restlichen 2 Bit Null gesetzt sind?

**Id 10 = 3.32 bit (es gibt  $5!/(2! \times 3!) = 10$  unterschiedliche Binärwörter)**

## Codes und Informationsgehalt

**Eine Verkehrsampel für Fussgänger signalisiert 50% der Zeitdauer rot, 10% gelb und 40% grün. Wie gross ist der Informationsgehalt jedes dieser drei Signale?**



## Codes und Informationsgehalt

Eine Verkehrsampel für Fussgänger signalisiert 50% der Zeitdauer rot, 10% gelb und 40% grün. Wie gross ist der Informationsgehalt jedes dieser drei Signale?

**rot: 1 bit**

**gelb: 3.32 bit**

**grün: 1.32 bit**

## Boole'sche Algebra

Welche der folgenden Ausdrücke sind mit der Äquivalenz  $x \equiv y$  gleichbedeutend?

- $(x \wedge y) \vee (\neg x \wedge \neg y)$
- $(x \vee y) \wedge (\neg x \vee \neg y)$
- $\neg ((\neg x \wedge y) \vee (x \wedge \neg y))$
- $(\neg x \vee y) \wedge (x \vee \neg y)$

## Boole'sche Algebra

Welche der folgenden Ausdrücke sind mit der Äquivalenz  $x \equiv y$  gleichbedeutend?

- $(x \wedge y) \vee (\neg x \wedge \neg y)$
- $(x \vee y) \wedge (\neg x \vee \neg y)$
- $\neg ((\neg x \wedge y) \vee (x \wedge \neg y))$
- $(\neg x \vee y) \wedge (x \vee \neg y)$

## Boole'sche Algebra

Welche der folgenden Schlüsse sind richtig?

- aus  $x \neq y$  folgt  $y \neq x$
- aus  $x \neq y$  folgt  $\neg x \neq \neg y$
- aus  $x \neq \text{true}$  folgt  $x$
- aus  $(x \neq y) \wedge (y \neq z)$  folgt  $x \neq z$
- aus  $\neg x \neq y$  folgt  $x \neq \neg y$

## Boole'sche Algebra

Welche der folgenden Schlüsse sind richtig?

- aus  $x \neq y$  folgt  $y \neq x$
- aus  $x \neq y$  folgt  $\neg x \neq \neg y$
- aus  $x \neq \text{true}$  folgt  $x$
- aus  $(x \neq y) \wedge (y \neq z)$  folgt  $x \neq z$
- aus  $\neg x \neq y$  folgt  $x \neq \neg y$

## Boole'sche Algebra

Welche der folgenden Boole'schen Ausdrücke sind Tautologien?

- $(x \wedge \neg y) \vee (\neg x \wedge y)$
- $(x \wedge \neg y) \vee (\neg x \vee y)$
- $(x \vee \neg y) \vee (\neg x \vee y)$
- $(x \vee \neg y) \wedge \neg (\neg x \wedge y)$

## Boole'sche Algebra

Welche der folgenden Boole'schen Ausdrücke sind Tautologien?

- $(x \wedge \neg y) \vee (\neg x \wedge y)$
- $(x \wedge \neg y) \vee (\neg x \vee y)$
- $(x \vee \neg y) \vee (\neg x \vee y)$
- $(x \vee \neg y) \wedge \neg (\neg x \wedge y)$

## Boole'sche Algebra

Welche der folgenden Formulierungen sind mit der Implikation  $x \Rightarrow y$  gleichbedeutend?

- aus  $x$  folgt  $y$
- $x$  ist notwendig für  $y$
- wenn  $x$  gilt, dann gilt auch  $y$
- $x$  ist hinreichend für  $y$
- $x$  ist schwächer als  $y$



## Boole'sche Algebra

Welche der folgenden Formulierungen sind mit der Implikation  $x \Rightarrow y$  gleichbedeutend?

- aus  $x$  folgt  $y$
- $x$  ist notwendig für  $y$
- wenn  $x$  gilt, dann gilt auch  $y$
- $x$  ist hinreichend für  $y$
- $x$  ist schwächer als  $y$

## Prädikatenlogik

Gegeben ist ein array  $a [0 \dots N-1]$ . Welche der folgenden Zusicherungen beschreiben den Sachverhalt, dass alle Elemente des arrays unterschiedliche Werte haben?

- $\text{All } i : 0 \leq i < N : \text{not } (\text{Ex } j : 0 \leq j < N : a_i = a_j )$
- $\text{All } i, j : 0 \leq i < j < N : a_i \neq a_j$
- $\text{All } i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i = a_j ) = 0$
- $\text{not } (\text{Ex } i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i = a_j ) > 1)$
- $\text{not } (\text{All } i : 0 \leq i < N : \text{All } j : 0 \leq j < N : a_i = a_j)$

## Prädikatenlogik

Gegeben ist ein array  $a [0 \dots N-1]$ . Welche der folgenden Zusicherungen beschreiben den Sachverhalt, dass alle Elemente des arrays unterschiedliche Werte haben?

- All  $i : 0 \leq i < N : \text{not } (\text{Ex } j : 0 \leq j < N : a_i = a_j )$
- All  $i, j : 0 \leq i < j < N : a_i \neq a_j$
- All  $i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i = a_j ) = 0$
- not  $(\text{Ex } i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i = a_j ) > 1)$
- not  $(\text{All } i : 0 \leq i < N : \text{All } j : 0 \leq j < N : a_i = a_j)$

## Prädikatenlogik

Gegeben sind die beiden arrays  $a [0 \dots N-1]$  und  $b [0 \dots N-1]$ . Welche der folgenden Zusicherungen beschreiben den Sachverhalt, dass kein Element des arrays  $a$  im array  $b$  enthalten ist?

- All  $i : 0 \leq i < N : a_i \neq b_i$
- $\neg(\text{Ex } i : 0 \leq i < N : (\text{Ex } j : 0 \leq j < N : a_i = b_j))$
- $\neg(\text{Ex } i : 0 \leq i < N : (\text{All } j : 0 \leq j < N : a_i = b_j))$
- All  $i : 0 \leq i < N : \neg(\text{Ex } j : 0 \leq j < N : a_i = b_j)$

## Prädikatenlogik

Gegeben sind die beiden arrays  $a [0 \dots N-1]$  und  $b [0 \dots N-1]$ . Welche der folgenden Zusicherungen beschreiben den Sachverhalt, dass kein Element des arrays  $a$  im array  $b$  enthalten ist?

- All  $i : 0 \leq i < N : a_i \neq b_i$
- $\neg(\text{Ex } i : 0 \leq i < N : (\text{Ex } j : 0 \leq j < N : a_i = b_j))$
- $\neg(\text{Ex } i : 0 \leq i < N : (\text{All } j : 0 \leq j < N : a_i = b_j))$
- All  $i : 0 \leq i < N : \neg (\text{Ex } j : 0 \leq j < N : a_i = b_j)$

## Prädikatenlogik

Gegeben sind die beiden arrays  $a [0 \dots N-1]$  und  $b [0 \dots N-1]$ . Welche der folgenden Zusicherungen beschreiben den Sachverhalt, dass jedes Element des arrays  $a$  auch in  $b$  enthalten ist?

- All  $i : 0 \leq i < N : (\text{Ex } j : 0 \leq j < N : a_i = b_j )$
- not All  $i, j : 0 \leq i < j < N : a_i \neq b_j$
- All  $i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i = b_j ) > 0$
- not  $(\text{Ex } i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i \neq b_j ) = 0)$
- All  $i : 0 \leq i < N : \text{not } (\text{Ex } j : 0 \leq j < N : a_i \neq b_j)$

## Prädikatenlogik

Gegeben sind die beiden arrays  $a [0 \dots N-1]$  und  $b [0 \dots N-1]$ . Welche der folgenden Zusicherungen beschreiben den Sachverhalt, dass jedes Element des arrays  $a$  auch in  $b$  enthalten ist?

- All  $i : 0 \leq i < N : (\text{Ex } j : 0 \leq j < N : a_i = b_j)$
- not All  $i, j : 0 \leq i < j < N : a_i \neq b_j$
- All  $i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i = b_j) > 0$
- not  $(\text{Ex } i : 0 \leq i < N : (\text{Anz } j : 0 \leq j < N : a_i \neq b_j) = 0)$
- All  $i : 0 \leq i < N : \text{not } (\text{Ex } j : 0 \leq j < N : a_i \neq b_j)$

## Komplexität

Wie verhält sich der Aufwand für die Berechnung der folgenden Java-Funktion in Abhängigkeit von  $n$  im günstigsten und im ungünstigsten Fall?

```
boolean f(int[] a, int n) {  
    int[] b = new int[n];  
    for (int i=0; i<n; i++)  
        if ((a[i]<0)||a[i]>=n||b[a[i]]) return false;  
        else b[a[i]] ++;  
    for (int i=0; i<n; i++)  
        if (b[i] != 1) return false;  
    return true;  
}
```

best case:

- konstant
- logarithmisch
- linear
- quadratisch
- exponentiell

worst case:

- konstant
- logarithmisch
- linear
- quadratisch
- exponentiell



## Komplexität

Wie verhält sich der Aufwand für die Berechnung der folgenden Java-Funktion in Abhängigkeit von  $n$  im günstigsten und im ungünstigsten Fall?

```
boolean f(int[] a, int n) {  
    int[] b = new int[n];  
    for (int i=0; i<n; i++)  
        if ((a[i]<0)||a[i]>=n)||b[a[i]]) return false;  
        else b[a[i]] ++;  
    for (int i=0; i<n; i++)  
        if (b[i] != 1) return false;  
    return true;  
}
```

best case:

- konstant
- logarithmisch
- linear
- quadratisch
- exponentiell

worst case:

- konstant
- logarithmisch
- linear
- quadratisch
- exponentiell

## Relationen

Welche Eigenschaft hat die Relation “x ist älter als y”, die zwischen zwei Personen x und y definiert ist?

- reflexiv
- symmetrisch
- antisymmetrisch
- transitiv

## Relationen

Welche Eigenschaft hat die Relation “x ist älter als y”, die zwischen zwei Personen x und y definiert ist?

- reflexiv
- symmetrisch
- antisymmetrisch
- transitiv

## Relationen

Welche Eigenschaft hat die Relation “ $x$  hat dieselbe Haarfarbe wie  $y$ ”, die zwischen zwei Personen  $x$  und  $y$  definiert ist?

- reflexiv
- symmetrisch
- antisymmetrisch
- transitiv

## Relationen

Welche Eigenschaft hat die Relation “x hat dieselbe Haarfarbe wie y”, die zwischen zwei Personen x und y definiert ist?

- reflexiv
- symmetrisch
- antisymmetrisch
- transitiv

## Arithmetische Ausdrücke

Der arithmetische Ausdruck  $(x+1) * (y-1)$  soll in die Postorder-Darstellung umgewandelt werden.

## Arithmetische Ausdrücke

Der arithmetische Ausdruck  $(x+1) * (y-1)$  soll in die Postorder-Darstellung umgewandelt werden.

$x 1 + y 1 - *$

## Arithmetische Ausdrücke

Der arithmetische Ausdruck  $(x+1) * (y-1)$  soll in die Preorder-Darstellung umgewandelt werden.



## Arithmetische Ausdrücke

Der arithmetische Ausdruck  $(x+1) * (y-1)$  soll in die Preorder-Darstellung umgewandelt werden.

**\* + x 1 - y 1**

## Arithmetische Ausdrücke

Der arithmetische Ausdruck  $(x+1) * (y-1)$  soll als Baum dargestellt werden.

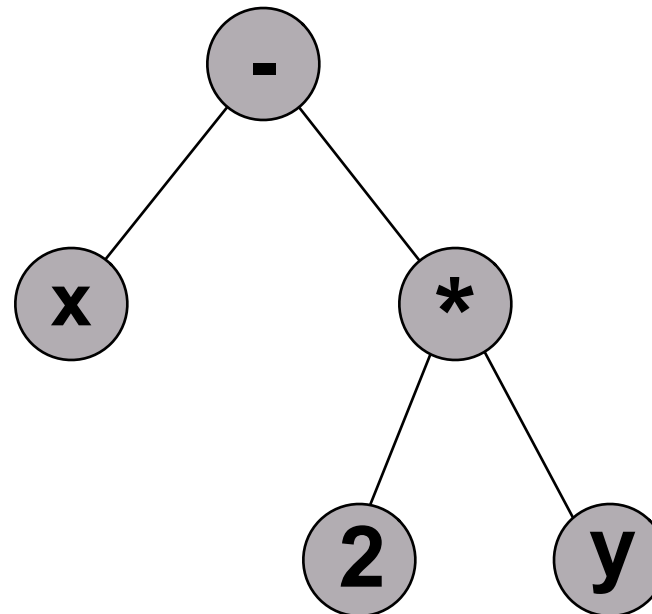
## Arithmetische Ausdrücke

Der arithmetische Ausdruck  $(x+1) * (y-1)$  soll als Baum dargestellt werden.

$* + x 1 - y 1$

## Arithmetische Ausdrücke

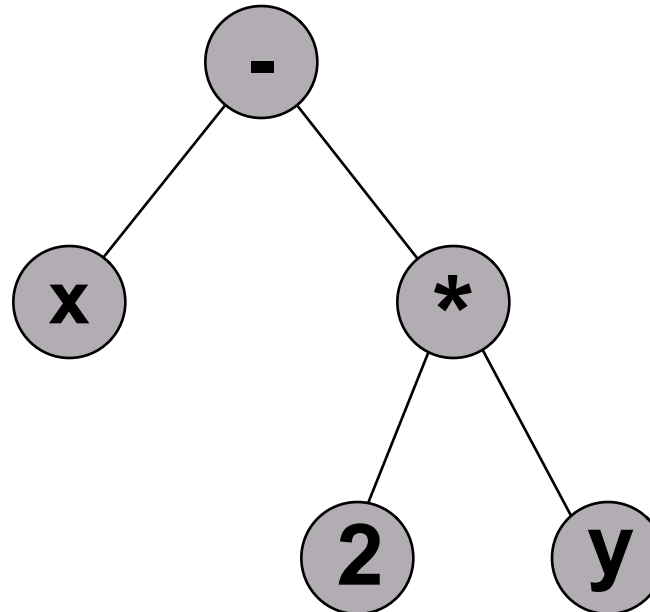
Ein arithmetischer Ausdruck wird durch die folgende Baumstruktur dargestellt:



Wie lautet dieser Ausdruck in Infix-Schreibweise?

## Arithmetische Ausdrücke

Ein arithmetischer Ausdruck wird durch die folgende Baumstruktur dargestellt:

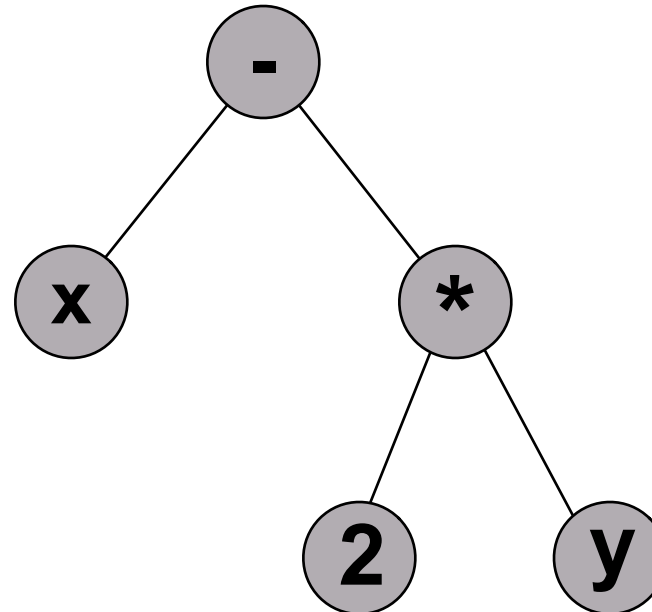


Wie lautet dieser Ausdruck in Infix-Schreibweise?

$$x - 2 * y$$

## Arithmetische Ausdrücke

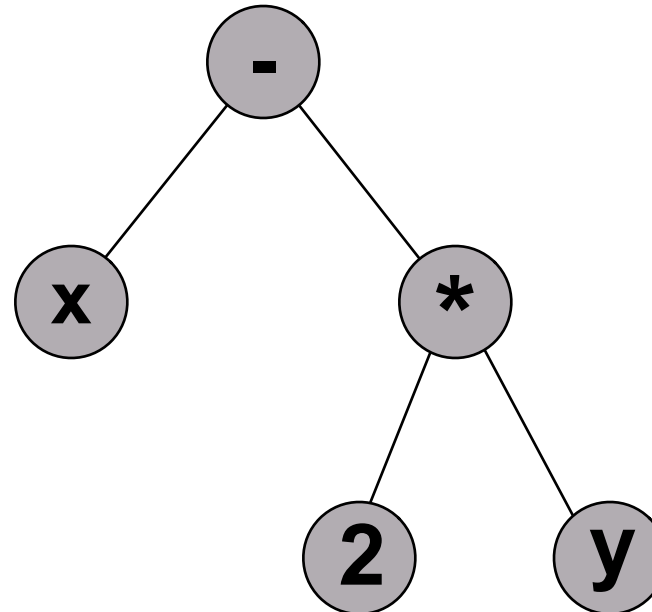
Ein arithmetischer Ausdruck wird durch die folgende Baumstruktur dargestellt:



Wie lautet dieser Ausdruck in Postfix-Schreibweise?

## Arithmetische Ausdrücke

Ein arithmetischer Ausdruck wird durch die folgende Baumstruktur dargestellt:

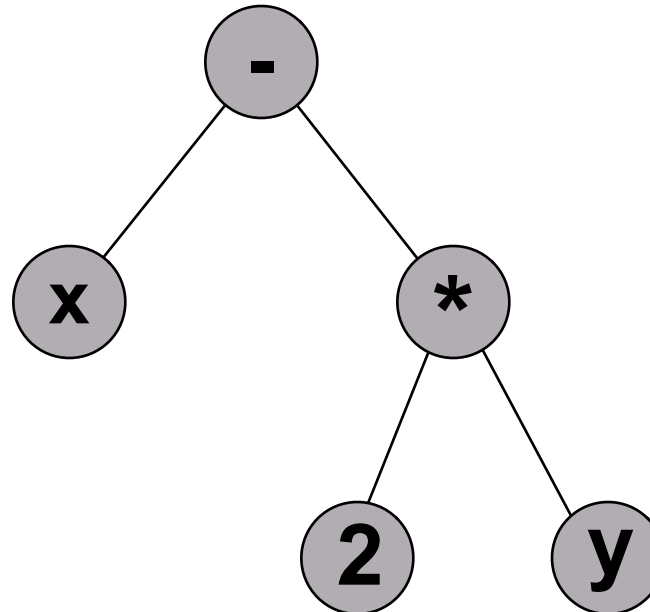


Wie lautet dieser Ausdruck in Postfix-Schreibweise?

**x 2 y \* -**

## Arithmetische Ausdrücke

Ein arithmetischer Ausdruck wird durch die folgende Baumstruktur dargestellt:

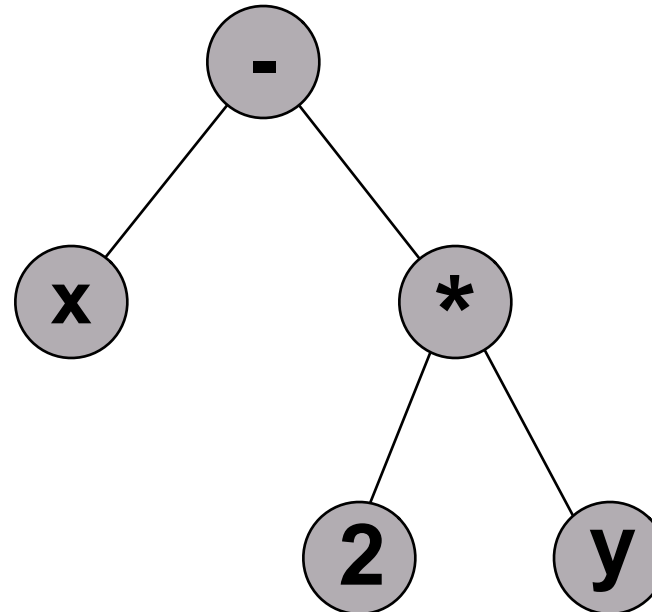


Wie lautet dieser Ausdruck in Prefix-Schreibweise?



## Arithmetische Ausdrücke

Ein arithmetischer Ausdruck wird durch die folgende Baumstruktur dargestellt:



Wie lautet dieser Ausdruck in Prefix-Schreibweise?

**$-x * 2y$**

## Graphen

Welche der folgenden Aussagen sind richtig?

- In einem spannenden Baum ist die Anzahl der Knoten immer grösser als die Anzahl der Kanten.
- In einem Baum sind alle Knoten kritisch.
- In einem zyklensfreien Graphen sind alle Kanten kritisch.
- Bipartite Graphen haben keine Zyklen.

## Graphen

Welche der folgenden Aussagen sind richtig?

- In einem spannenden Baum ist die Anzahl der Knoten immer grösser als die Anzahl der Kanten.
- In einem Baum sind alle Knoten kritisch.
- In einem zyklensfreien Graphen sind alle Kanten kritisch.
- Bipartite Graphen haben keine Zyklen.

## Graphen

Welche der folgenden Aussagen treffen auf vollständige Graphen zu?

- Vollständige Graphen bestehen immer nur aus einer einzigen Komponente
- Vollständige Graphen haben immer einen Hamilton-Zyklus
- Vollständige Graphen haben keine kritischen Knoten
- Vollständige Graphen haben keine kritischen Kanten
- Vollständige Graphen haben einen zweifachen Zusammenhang

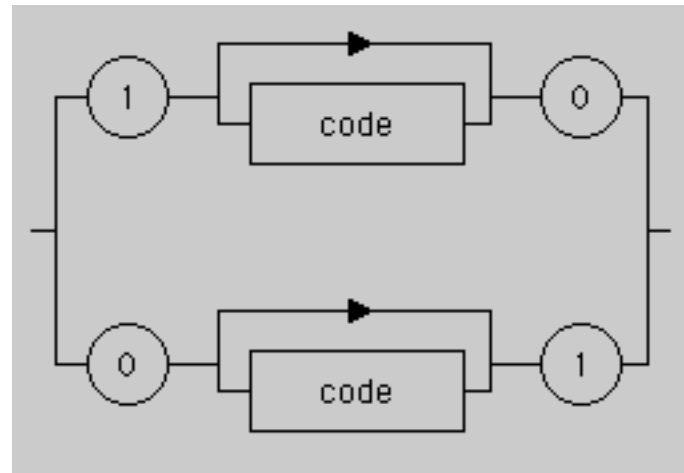
## Graphen

Welche der folgenden Aussagen treffen auf vollständige Graphen zu?

- Vollständige Graphen bestehen immer nur aus einer einzigen Komponente
- Vollständige Graphen haben immer einen Hamilton-Zyklus
- Vollständige Graphen haben keine kritischen Knoten
- Vollständige Graphen haben keine kritischen Kanten
- Vollständige Graphen haben einen zweifachen Zusammenhang

## Syntax

Das folgende Syntaxdiagramm definiert einen code:

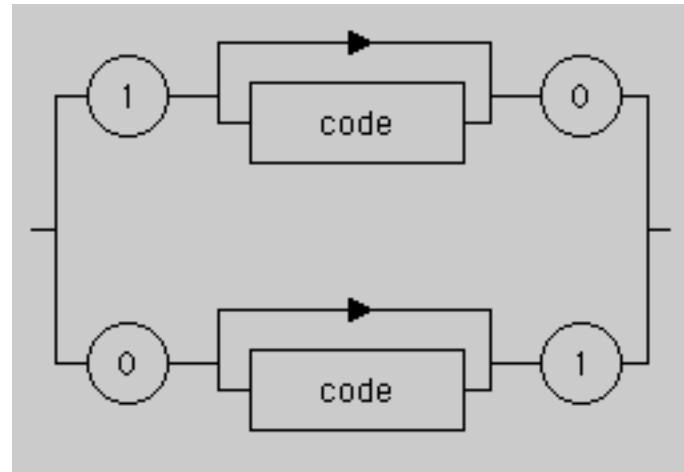


Welche der folgenden Texte sind syntaktisch richtige codes?

- 000
- 0110
- 1010
- 0101
- 01110
- 01010

## Syntax

Das folgende Syntaxdiagramm definiert einen code:

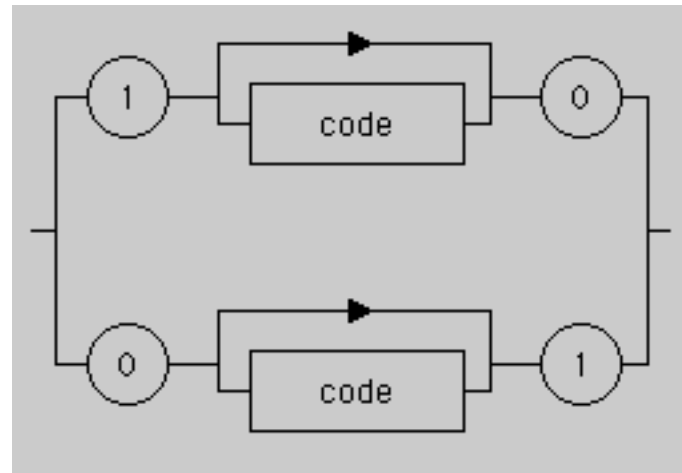


Welche der folgenden Texte sind syntaktisch richtige codes?

- 000
- 0110
- 1010
- 0101
- 01110
- 01010

## Syntax

Das folgende Syntaxdiagramm definiert einen code:



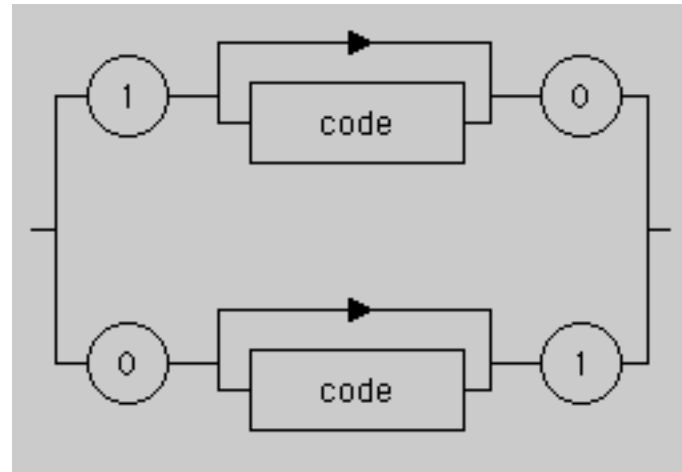
Welche Aussagen treffen auf einen syntaktisch richtigen code zu?

- Jeder code hat eine gerade Wortlänge
- Die Ziffernsumme jedes codes ist ungerade
- Jeder code enthält gleichviele Nullen wie Einsen
- Kein code endet mit der gleichen Ziffer mit der er beginnt
- Das Komplement eines codes ist wieder ein syntaktisch richtiger code



## Syntax

Das folgende Syntaxdiagramm definiert einen code:



Welche Aussagen treffen auf einen syntaktisch richtigen code zu?

- Jeder code hat eine gerade Wortlänge
- Die Ziffernsumme jedes codes ist ungerade
- Jeder code enthält gleichviele Nullen wie Einsen
- Kein code endet mit der gleichen Ziffer mit der er beginnt
- Das Komplement eines codes ist wieder ein syntaktisch richtiger code