

2023-DE-01 Konflikt-Detektor

Body

Anna und Ben wollen einen «Konflikt-Detektor» bauen, der anzeigt, ob sie eine unterschiedliche Meinung haben.

Sie verwenden Einheiten, die in zwei Zuständen sein können: Ja und Nein. Zwei Einheiten können mit einem Kabel verbunden werden.

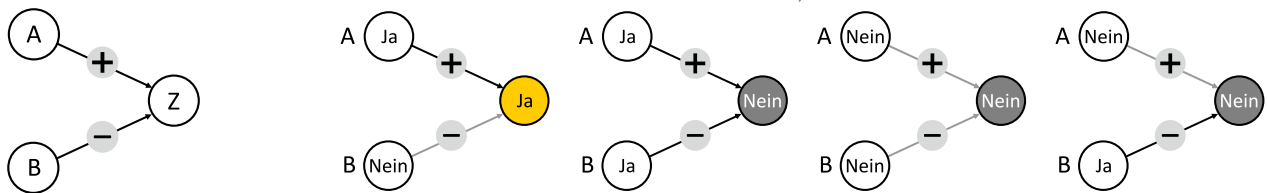
Wenn eine Einheit im

- Zustand Ja ist, sendet sie über alle ausgehenden Kabel ein Signal.
- Zustand Nein ist, sendet sie kein Signal.

Die Kabel werden so eingestellt, dass sie ein Signal als positives (+) oder negatives (−) Signal an die rechts angeschlossene Einheit übermitteln.

Eine angeschlossene Einheit geht in den Zustand Ja, wenn sie mehr positive als negative Signale empfängt, und sonst in den Zustand Nein. Als Eingabe setzt Anna den Zustand der Einheit A und Ben den Zustand der Einheit B.

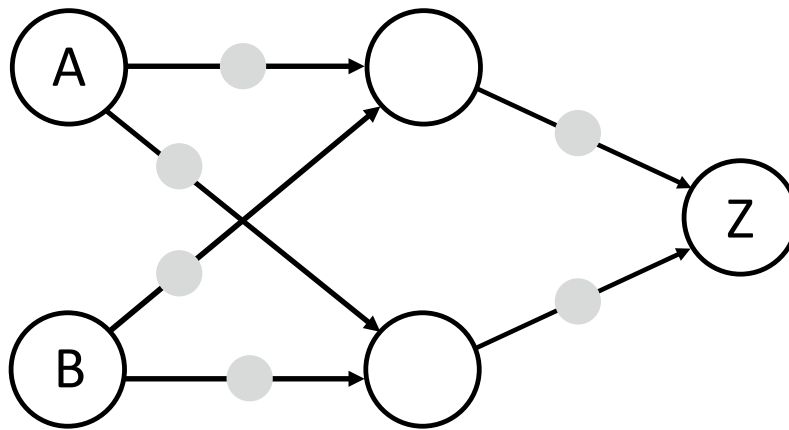
Zuerst bauen Anna und Ben diese Maschine: Sie bemerken, dass die Einheit Z nur dann Ja ist, wenn A Ja und B Nein ist. Das ist nicht das, was sie wollen.



Dann bauen Anna und Ben eine grössere Maschine (unten im Bild) und sind sicher, dass sie der Konflikt-Detektor sein kann: Z soll nur dann Ja sein, wenn A und B in unterschiedlichen Zuständen sind (Ja und Nein bzw. Nein und Ja). Ansonsten soll Z im Zustand Nein sein. Jetzt müssen nur noch die Kabel richtig eingestellt werden.

Question/Challenge - for the brochures

Stelle für jedes Kabel ein, ob es ein Signal positiv (+) oder negativ (−) übermittelt, damit der Konflikt-Detektor korrekt arbeitet.



Interactivity instruction - for the online challenge

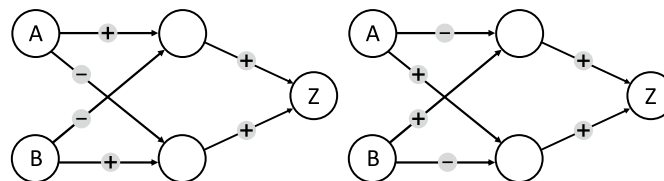
Klicke auf die Signalkabel, um + und - zu ändern. Wenn du fertig bist, klicke auf «Antwort speichern».

Answer Options/Interactivity Description

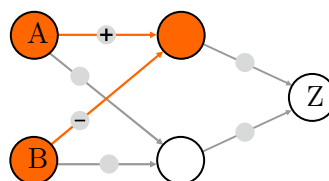
In the picture of the network, each edge has a marker that can take values «+» and «-». Clicking on the edge or the marker toggles between the two values. Initially, all edges have an empty grey box. (DACH: We decided to have the markers preset to -.)

Answer Explanation

Diese beiden Antworten sind richtig:



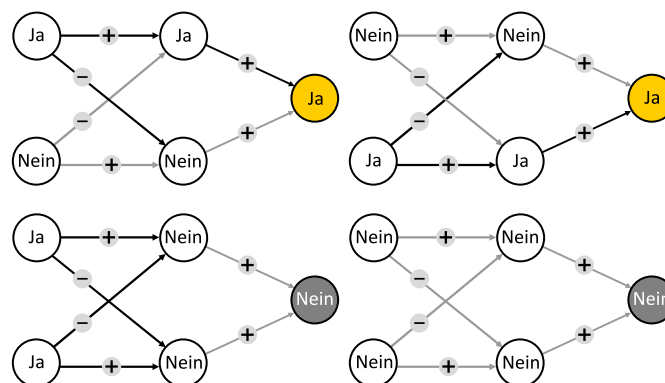
Im Konflikt-Detektor muss die Ausgabe-Einheit genau bei zwei unterschiedlichen Eingaben (A=Ja und B=Nein sowie A=Nein und B=Ja) auf Ja sein. Z kann nur Ja sein, wenn über die zwei eingehenden Kabel mehr positive als negative Signale ankommen. Mindestens eines der Kabel muss also ein positives Signal (+) übermitteln. Nehmen wir einmal an, nur das obere Kabel, das zu Z führt, wird auf + gestellt. Dann muss die Einheit oben Mitte beide gewünschten Eingabekombinationen erkennen können, also in beiden Fällen Ja sein. Zusammen mit den Eingabeeinheiten A und B bildet diese Einheit aber genau so eine Maschine, wie Anna und Ben sie zu Beginn gebaut haben. Sie kann nur in genau einem der gewünschten Fälle Ja sein, und zwar, wenn eines ihrer Kabel auf + und das andere auf - gestellt wird:



Es wird also für jeden der gewünschten Eingabefälle eine eigene Einheit in der Mitte benötigt, eine

für A=Ja und B=Nein, die andere für A=Nein und B=Ja. Die Kabel zur ersten Einheit müssen auf + (Kabel von A) und – (B) gestellt werden, die Kabel zur anderen Einheit auf – (A) und + (B). Welche Einheit in der Mitte welchen Fall übernimmt, ist egal; deshalb gibt es bei den Kabeln von A und B zur Mitte zwei Möglichkeiten. Wenn nun jede Einheit in der Mitte in genau einem gewünschten Fall Ja ist, müssen beide Kabel von der Mitte zu Z auf + gestellt werden; nur dann ist Z=Ja in genau beiden gewünschten Fällen.

Für die erste richtige Antwort zeigt das Bild unten die Funktion des Konflikt-Detektors. Man sieht: Die obere Einheit in der Mitte erkennt den Fall A=Ja und B=Nein, die untere den Fall A=Nein und B=Ja. Die jeweilige Einheit sendet ein positives Signal zu Z, und Z ist Ja. Für die anderen Eingaben (A=Ja und B=Ja sowie A=Nein und B=Nein) sind beide mittleren Einheiten Nein, Z empfängt kein positives Signal und ist damit auch Nein.



This is Informatics

Der Konflikt-Detektor verarbeitet zwei Eingabewerte (Ja oder Nein) und liefert die Ausgabe Ja genau dann, wenn die beiden Eingabewerte unterschiedlich sind. Diese logische Funktion nennt man Exklusiv-Oder (XOR, Kontravalenz). Die erste in dieser Biberaufgabe beschriebene Maschine von Anna und Ben (zwei Schalter und eine Ausgabe-Einheit) ist eine vereinfachte Version eines *Perzeptrons*, das Frank Rosenblatt im Jahr 1957 beschrieben hat. Die Ausgabe-Einheit bildet eine Nervenzelle (Neuron) nach, die Eingabesignale verarbeiten kann und ein Ausgabesignal erzeugt. Mit einem Perzeptron kann man zwar die logischen Operationen Und und Oder implementieren, nicht aber das Exklusiv-Oder. Dazu benötigt man eine weitere Schicht von Schalteinheiten wie in der Lösung dieser Aufgabe. Erst in den 1980er Jahren hat man das erkannt (z.B. Rumelhart, Hinton & Williams 1986) und war dann (später) in der Lage, künstliche neuronale Netze zu programmieren, die ähnlich wie das menschliche Gehirn arbeiten und z. B. Kamerabilder auswerten und Objekte erkennen können. Die Informatik hat Methoden entwickelt, wie grosse neuronale Netze mit vielen Schichten und Einheiten ihre Berechnungen effizient durchführen können. Solche Netze bilden die Grundlage vieler aktueller KI-Systeme.

This is Computational Thinking

Dieser Abschnitt wird in diesem Jahr nicht bearbeitet.

Informatics Keywords and Websites

- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536 : <http://www.cs.toronto.edu/~hinton/absps/naturebp.p>
- Perzeptron, Überblick: <https://de.wikipedia.org/wiki/Perzeptron>
- Tutorial zur Programmierung eines Perzeptrons: https://neuromant.de/2018/11/25/Tutorial_Das-Perzeptron/
- Exklusive-Oder (Kontravalenz): <https://de.wikipedia.org/wiki/Kontravalenz>

Computational Thinking Keywords and Websites

Dieser Abschnitt wird in diesem Jahr nicht bearbeitet.