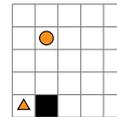


Roro-Robo – Lösungen

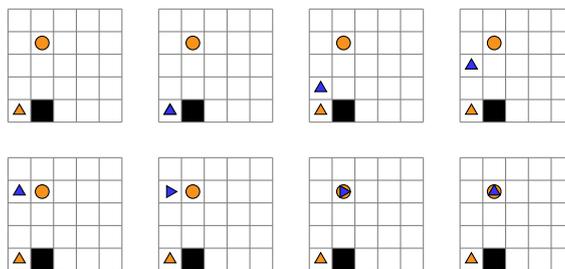
Aufgabe 1 (kleine Roro-Programme* (nur für die Klassen 7/8) [4 Punkte]). Skizziere die einzelnen Schritte, die Roro bei der Ausführung der angegebenen Roro-Programme auf der folgenden Karte durchführt. Wo steht Roro am Ende und in welche Richtung schaut er?

1. $\triangle \triangle \triangle \triangleright \triangle \triangleleft$

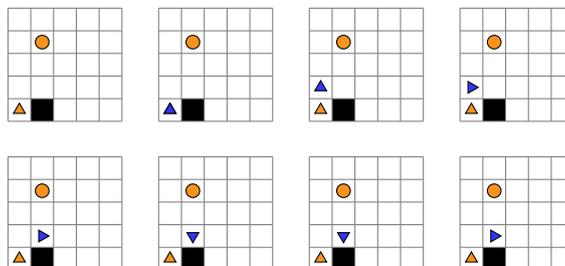
2. $\triangle \triangleright \triangle \triangleright \triangle \triangleleft$



Lösung. Zu 1. Für die Sequenz $\triangle \triangle \triangle \triangleright \triangle \triangleleft$ erhalten wir die folgenden Schritte (und Roro schaut am Ende wieder nach Norden):



Zu 2. Für die Sequenz $\triangle \triangleright \triangle \triangleright \triangle \triangleleft$ erhalten wir die folgenden Schritte (und Roro schaut am Ende nach Osten):



Man beachte, dass Roro bei der fünften Instruktion auf demselben Feld stehenbleibt, da der Weg durch das Hindernis blockiert ist. 

Aufgabe 2 (Lechts und links* (nur für die Klassen 7/8) [4 Punkte]). Roro hat durch einen Unfall verlernt, was das Kommando \triangleright bedeutet. Gib ein Roro-Programm an, das nur \triangleleft und \triangle verwendet und auf der Karte aus Aufgabe 1 (genau am Ende des Programms) das Ziel erreicht. Begründe Deine Antwort!

Thema vom 10. Februar 2017. Einsenden der Lösungen bis 24. März 2017.

Schülerzirkel Mathematik, Fakultät für Mathematik, 93040 Regensburg

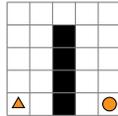
<http://www.mathematik.uni-r.de/schuelerzirkel>, schueler.zirkel@mathematik.uni-regensburg.de

Allgemeine Informationen zur Teilnahme: <http://www.mathematik.uni-r.de/schuelerzirkel>

Allgemeine Hinweise zum Lösen von Aufgaben: <http://www.mathematik.uni-r.de/schuelerzirkel>

Lösung. Indem man jedes Auftreten des Kommandos \blacktriangleright durch $\blacktriangleleft \blacktriangleleft \blacktriangleleft$ ersetzt, kann man jedes gewöhnliche Roro-Programm in eines umwandeln, das auch der verunfallte Roro versteht und dasselbe Ziel erreicht wie das ursprüngliche Programm. Also hat zum Beispiel das Roro-Programm $\blacktriangle \blacktriangle \blacktriangle \blacktriangleleft \blacktriangleleft \blacktriangleleft \blacktriangle \blacktriangleleft$ die gewünschte Eigenschaft. \square

\square **Aufgabe 3 (In der Kürze ... * [4 Punkte]).** Gib ein Roro-Programm mit möglichst wenig Instruktionen an, mit dem Roro auf der folgenden Karte das Ziel erreicht. Begründe auch, warum es nicht mit noch weniger Instruktionen geht!



Lösung. Man kann leicht nachprüfen, dass das Roro mit dem folgenden Programm das Ziel erreicht:

$\blacktriangle \blacktriangle \blacktriangle \blacktriangle \blacktriangleright \blacktriangle \blacktriangle \blacktriangle \blacktriangle \blacktriangleright \blacktriangle \blacktriangle \blacktriangle \blacktriangle$

Dieses Programm besteht aus 14 Instruktionen. Aus folgendem Grund gibt es kein Programm, mit dem Roro das Ziel erreicht, das weniger Instruktionen enthält:

Wegen des Hindernisses muss Roro das mittlere Feld in der obersten Zeile passieren. Da sich Roro nur auf nördlich/südlich/östlich/westlich benachbarte Felder bewegen kann, benötigt man mindestens sechs \blacktriangle -Instruktionen, um vom Start zum obersten mittleren Feld zu gelangen. Da das oberste mittlere Feld jedoch nicht vom Start aus durch sechs \blacktriangle -Instruktionen erreichbar ist (Roro muss mindestens einmal die Richtung ändern), ist mindestens eine weitere Instruktion nötig. Insgesamt benötigt Roro also mindestens sieben Instruktionen, um vom Start zum obersten mittleren Feld zu kommen. Mit demselben Argument sieht man, dass nochmal mindestens sieben weitere Instruktionen nötig sind, um vom obersten mittleren Feld das Ziel zu erreichen.

Also sind insgesamt mindestens $7 + 7 = 14$ Instruktionen nötig. \square

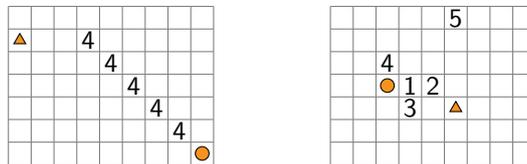
\square **Aufgabe 4 (Palindromprogramme** [4 Punkte]).** Ein Roro-Programm ist ein *Palindromprogramm*, wenn es mit seiner Spiegelung übereinstimmt. Zum Beispiel ist das Programm $\blacktriangleright \blacktriangle \blacktriangleleft$ ein Palindromprogramm, aber $\blacktriangleright \blacktriangle \blacktriangleright$ ist keines.

1. Gib ein Palindromprogramm an, mit dem Roro auf der Karte aus Aufgabe 1 das Ziel erreicht.
2. Gibt es auch ein Palindromprogramm dieser Art, wenn wir zusätzlich verlangen, dass Roro am Schluss nach Westen schaut? Begründe Deine Antwort!

Lösung. Zu 1. Zum Beispiel ist $\blacktriangleright \blacktriangle \blacktriangleleft \blacktriangle \blacktriangle \blacktriangle \blacktriangleright \blacktriangle \blacktriangleleft$ ein solches Palindromprogramm (es ist tatsächlich ein Palindromprogramm und man überprüft leicht, dass Roro damit das Ziel erreicht).

Zu 2. Nein, ein solches Palindromprogramm kann es nicht geben, denn: Jedes Palindromprogramm enthält dieselbe Anzahl von \blacktriangleright und \blacktriangleleft (denn durch die Palindrom-eigenschaft gehört zu jedem \blacktriangleleft ein \blacktriangleright und umgekehrt; ist die Länge des Programms ungerade, so muss die mittlere Instruktion \blacktriangle sein, da sonst die Spiegelungseigenschaft nicht erfüllt ist). Also wird jede Drehung von Roro in einem Palindromprogramm in einer späteren Instruktion wieder rückgängig gemacht, d.h. Roro schaut am Ende in dieselbe Richtung wie am Anfang. In unserem Fall schaut Roro daher am Ende wieder nach Norden und nicht nach Westen. \square

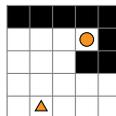
- \square **Aufgabe 5** (Repetitio est mater studiorum** [4 Punkte]). Gerät Roro auf ein Feld, auf dem eine Zahl n steht, so führt Roro zunächst die letzten n Instruktionen aus, die er vor dem Erreichen dieses Feldes ausgeführt hat, und fährt erst dann mit seinem Programm (bzw. noch ausstehenden Wiederholungen) fort. Gib jeweils ein Programm mit höchstens acht Instruktionen an, mit dem Roro auf den folgenden beiden Karten das Ziel (am Ende des Programms) erreicht.



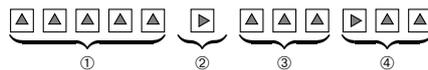
Lösung. Zur linken Karte. Zum Beispiel hat das Programm $\blacktriangle \blacktriangleright \blacktriangle \blacktriangle \blacktriangleright \blacktriangle \blacktriangleleft \blacktriangle$ die gewünschte Eigenschaft: Wenn Roro diese acht Instruktionen ausgeführt hat, steht er auf der obersten 4 und schaut nach Osten. Dann werden jeweils die letzten vier Instruktionen (also $\blacktriangleright \blacktriangle \blacktriangleleft \blacktriangle$) wiederholt und Roro nähert sich so „Stufe für Stufe“ dem Ziel.

Zur rechten Karte. Eine mögliche Lösung ist zum Beispiel $\blacktriangle \blacktriangleleft \blacktriangle \blacktriangleright \blacktriangle$. \square

- \square **Aufgabe 6** (Vergesslichkeit*** (empfohlen ab Klasse 9) [4 Punkte]). Roro ist leider manchmal vergesslich. Daher kann es passieren, dass er vergisst, eine der Instruktionen in einem Programm auszuführen. Schreibe ein Roro-Programm, mit dem Roro auch dann auf der folgenden Karte sicher das Ziel erreicht, wenn er höchstens eine der Instruktionen des Programms vergisst. Begründe kurz, warum Dein Programm die gewünschte Eigenschaft besitzt!



Lösung. Ein Beispiel für ein solches Programm ist:



Wir begründen nun, warum dieses Programm gegen das Vergessen von höchstens einer Instruktion resistent ist:

- Falls Roro keine Instruktion vergisst, erreicht er damit das Ziel (einfach zu überprüfen; aufgrund der Hindernisse, die das Ziel umgeben, wird er sich am Ende nicht vom Ziel wegbewegen).
- Falls Roro eine der Instruktionen aus ① vergisst, wird er trotzdem am Ende von ① genau unterhalb der Hindernisse ankommen (da ja eigentlich ein \triangle „zu viel“ ist) und erreicht dann wie im vorigen Fall das Ziel.
- Falls Roro die Instruktion ② vergisst, bleibt er während der Instruktionen aus ③ auf demselben Feld stehen (wegen des Hindernisses) und erreicht dann mit den verbleibenden Instruktionen das Ziel.
- Falls Roro eine der Instruktionen aus ③ vergisst, erreicht er trotzdem das Ziel (da ja eigentlich ein \triangle „zu viel“ ist).
- Falls Roro eine der Instruktionen aus ④ vergisst, befindet er sich bereits am Ziel und wird dort (wegen der Hindernisse) auch bleiben. 

Turtle-Graphik

Die Programmierung der Bewegungen von Roro ist an die Programmierung der Bewegungen in sogenannter Turtle-Graphik angelehnt. In Turtle-Graphik gibt es normalerweise zusätzlich noch die Möglichkeit, mit einem „Stift“ die Bewegungen aufzuzeichnen, und weitere Kontrollstrukturen, die Rekursion etc. ermöglichen.

Weiterführende Links

https://en.wikipedia.org/wiki/Turtle_graphics

[https://en.wikipedia.org/wiki/Logo_\(programming_language\)](https://en.wikipedia.org/wiki/Logo_(programming_language))