Kara

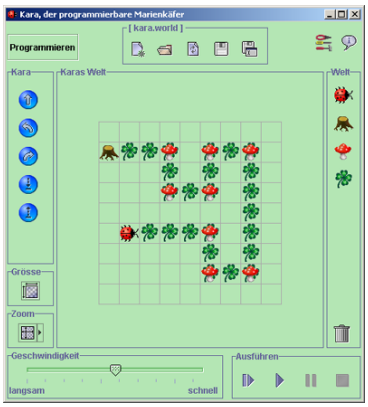
Wie funktioniert Kara?

In der Welt des Marienkäfers...

... gibt es:

- **unbewegliche Baumstümpfe,**
- **Pilze, die Kara verschieben und**
- **Kleeblätter, die Kara legen und aufnehmen kann**

... und natürlich Kara selbst!



Kara, der Marienkäfer...

... hat **Sensoren**, mit denen er seine Umwelt wahrnimmt:


stehe ich vor einem Baumstumpf?


ist links von mir ein Baumstumpf?


ist rechts von mir ein Baumstumpf?


stehe ich vor einem Pilz?


stehe ich auf einem Kleeblatt?

... versteht einige **Befehle**, die er folgsam ausführt:


mache einen Schritt vorwärts!


drehe um 90° nach links!


drehe um 90° nach rechts!

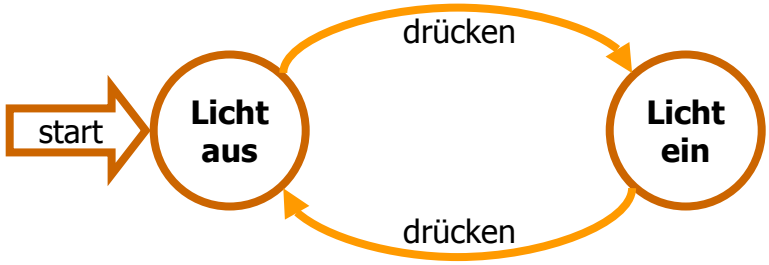

lege ein Kleeblatt hin!


nimm ein Kleeblatt auf!

Ein ganz einfacher Automat: Lichtschalter

Ein Lichtschalter hat nur zwei Zustände, ein und aus.

Ein „Sensor" meldet, wenn der Schalter betätigt wird.



Ein wenig komplexer: Videogerät

Getränkeautomaten

Dieser Automat...

- **akzeptiert nur**

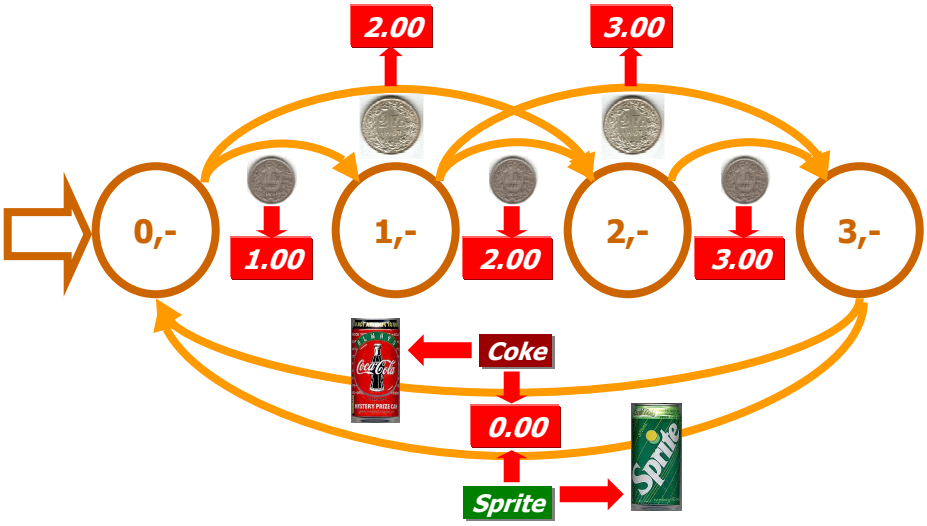
- **zeigt eingeworfenen Betrag an**

0.00

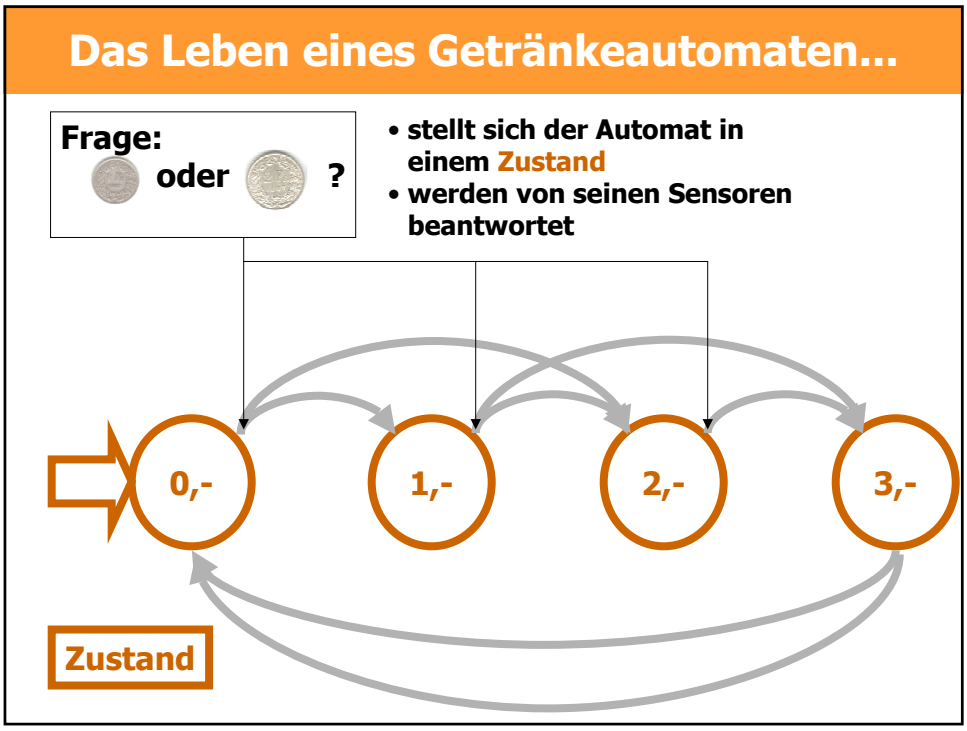
- **gibt nur**   **aus**

für 3,-

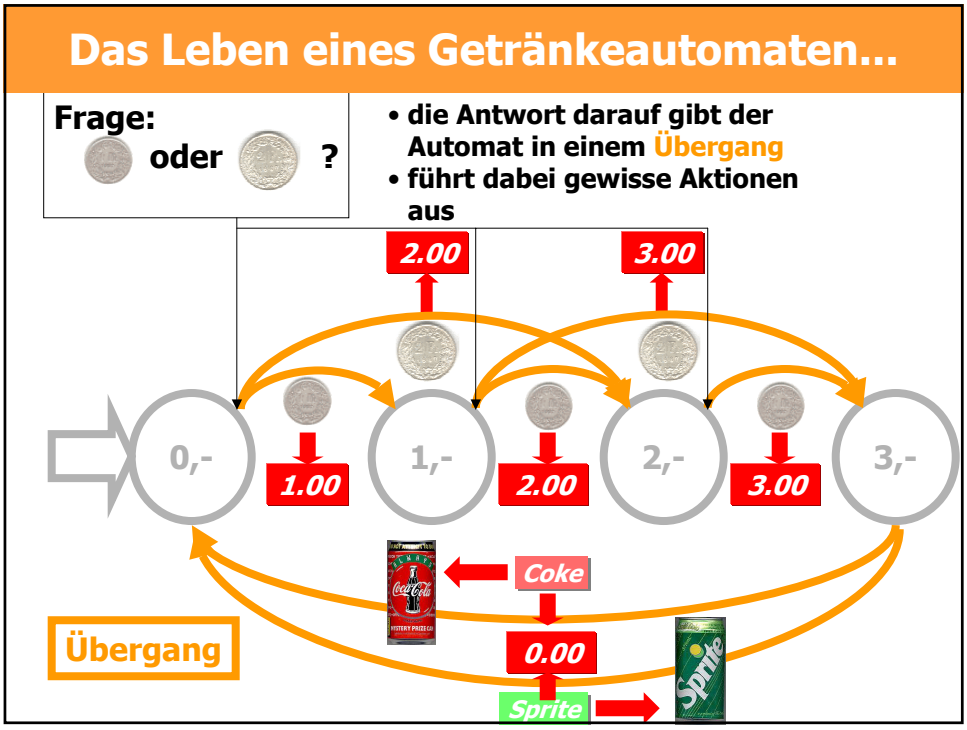
Das Leben eines Getränkeautomaten...



Das Leben eines Getränkeautomaten...

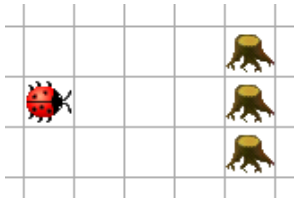


Das Leben eines Getränkeautomaten...

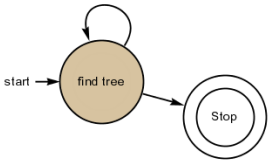


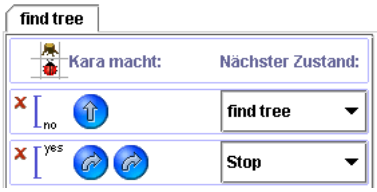
Kara sucht einen Baumstumpf !

Die Aufgabe:
bis zum nächsten Baumstumpf laufen, dann 180° Drehung!



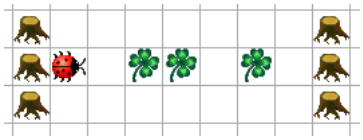
Das Program:
ein Zustand „suche Baum“,
ein Sensor „stehe vor Baum“



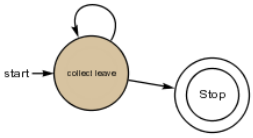


Kara, der Blättersammler !

Die Aufgabe:
alle Blätter bis zum nächsten Baumstumpf aufnehmen!

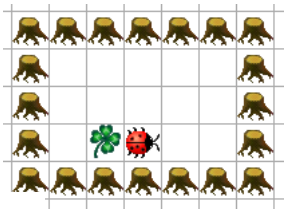


Das Programm:
ein Zustand „collect leaves“,
benötigt zwei Sensoren

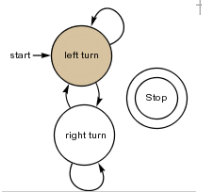


Kara, der Wächter (1)

Die Aufgabe:
Raum linksherum bis Kleeblatt,
dann rechtsherum bis Kleeblatt,
dann linksherum... laufen



Das Programm:
zwei Zustände:
einer für Linksdrehung,
einer für Rechtsdrehung



Kara, der Wächter (2)

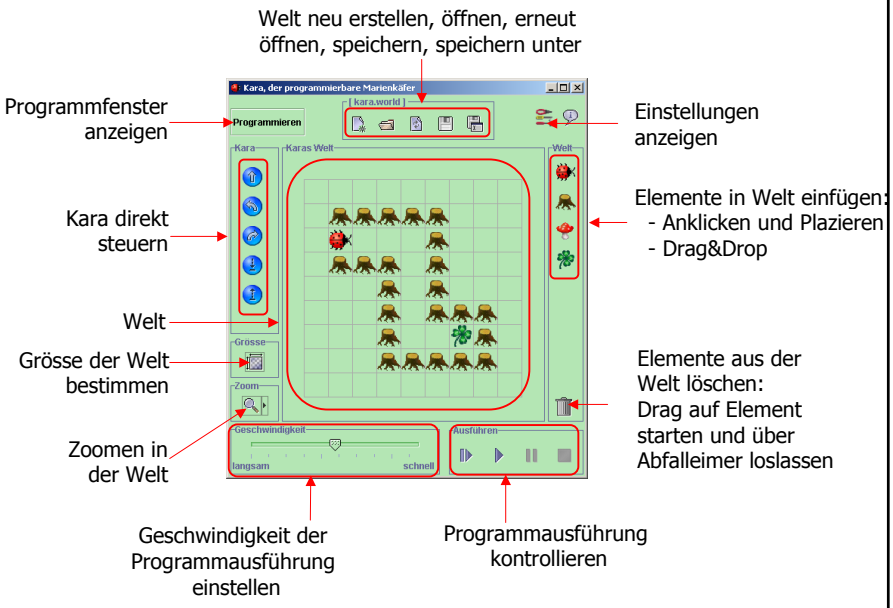
Das Programm: ein Zustand !

Welche Lösung ist „besser"?
- Lesbarkeit des Programms?
- Anzahl Befehle: Weniger = besser?
- Anzahl Zustände: Weniger = besser?

⇒ **Eine Frage des persönlichen Programmierstils!**

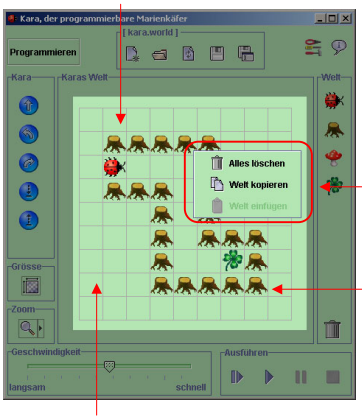
☺ ☺ ☺

Getting started: das Weltfenster von Kara



Getting started: die Welt von Kara

Sichtbaren Weltausschnitt verschieben:
- linke Maustaste gedrückt halten und Maus bewegen
  (geht nur, falls Welt grösser als sichtbarer Bereich)



Weltmenu durch Rechtsklick öffnen

Objekte in der Welt mit Drag&Drop verschieben

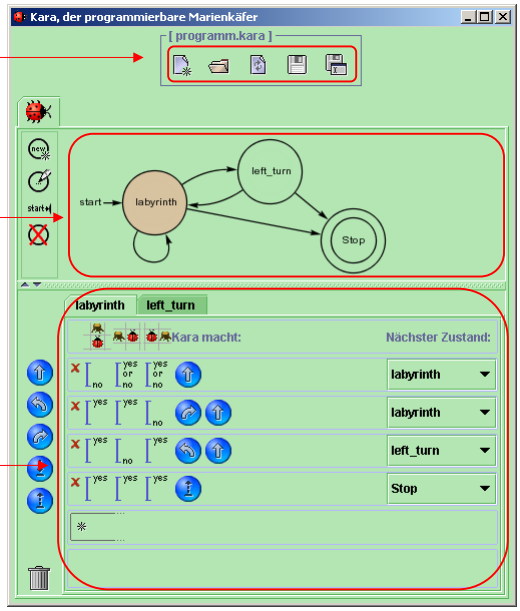
Zoomen in der Welt:
- mittlere Maustaste gedrückt halten und Maus bewegen, oder
- linke Maustaste und CTRL gedrückt halten und Maus bewegen

Getting started: das Programmfenster von Kara

Programm neu erstellen, öffnen, erneut öffnen, speichern, speichern unter

Automatenansicht

Zustandsansicht

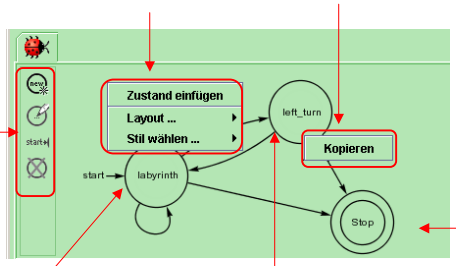


Getting started: die Automatenansicht

Automatenmenu durch Rechtsklick in Hintergrund öffnen

Zustandsmenu durch Rechtsklick auf Zustand öffnen

Zustand erstellen, bearbeiten, als Startzustand setzen, löschen



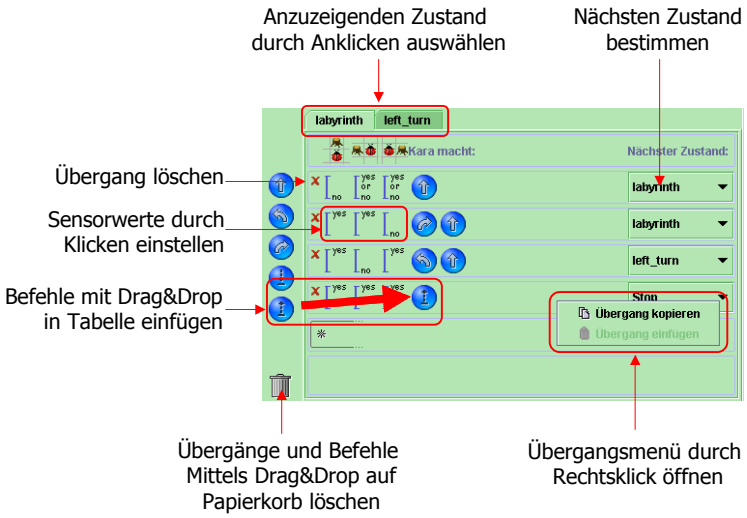
Zoomen und verschieben der Ansicht gleich wie in der Welt

Zustand verschieben:
Zustand in der Mitte anklicken und Maus bei gedrückter Taste bewegen

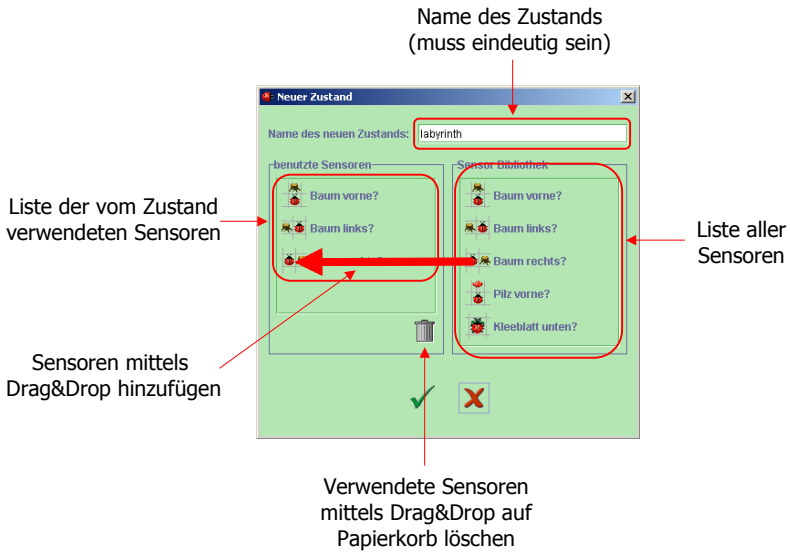
Neuen Übergang erstellen:

1. Maustaste über Rand des Zustandes drücken
2. Bei gedrückter Taste Maus über Zielzustand bewegen
3. Taste loslassen

Getting started: die Zustandsansicht



Getting started: Zustände erzeugen/bearbeiten

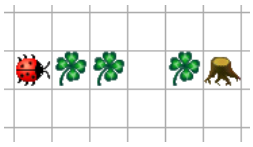


Aufgaben für Kara

KARA: EINSTIEGSAUFGABEN

Hinweis**: Die Abbildungen der Welten bei den Aufgaben sind als Beispiel zu verstehen. Ihre Programme sollten die Aufgaben so lösen, dass Kara die Aufgabe auch in einer **gleich strukturierten Welt** lösen kann. Zum Beispiel soll es bei der ersten Aufgabe nicht darauf ankommen, wo die Kleeblätter liegen oder wie weit der Baum entfernt ist. Kara weiss nur, dass geradeaus vor ihm ein Baum ist, ohne Pilz zwischen ihm und dem Baum. Analog sind die Bilder bei den übrigen Aufgaben zu verstehen. **Wichtig: Testen Sie Ihre Programme immer an mehreren verschiedenen Welten, insbesondere bei den Aufgaben 2 und 3!

Aufgabe 1 – Kara, der Blätterfetischist



Schreiben Sie ein Programm, das Kara bis zum nächsten Baum führt. Kara weiss, dass geradeaus vor ihm ein Baum ist. Liegt auf einem Feld ein Blatt, soll Kara es aufnehmen; liegt auf einem Feld kein Blatt, eines hinlegen. Bei dem Baum angekommen ist das Programm zu beenden.

Aufgabe 2a – Kara, der Tunnelsucher I

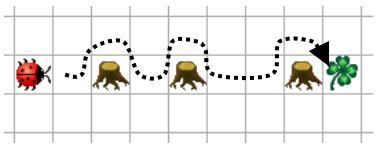
Kara sucht den Eingang eines geraden Tunnels (Feld 2a). Er weiss, dass dieser Tunnel geradeaus vor ihm liegt. Schreiben Sie ein Programm, das ihn auf dem ersten Feld im Tunnelinnern anhalten lässt. *Aber Achtung: manche Tunnels haben zunächst eine einseitige Gallerie, manche links, manche rechts!*

Aufgabe 2b – Kara, der Tunnelsucher II

Kara will den Ausgang des Tunnels finden (Feld 2b). Dazu muss er zunächst den Tunnel durchqueren. Schreiben Sie ein Programm, das ihn auf dem ersten Feld nach dem Tunnel anhalten lässt – er soll nicht bis zum Ende der Gallerie laufen! **Hinweis:** Die Lösung erfordert zwei Zustände! ***Überlegen Sie Sich, warum ein ein Zustand nicht ausreichen kann!***

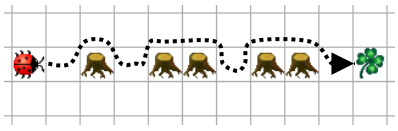
Bemerkung: Den STOP-Zustand zähle ich nie mit beim Zählen der Zustände!

Aufgabe 3a – Kleeblattsuche im Wald I



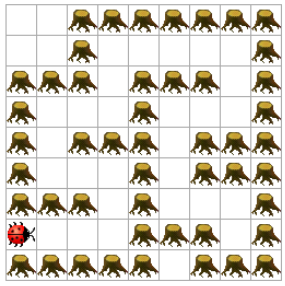
Kara sucht ein Kleeblatt. Er weiss, dass eines geradeaus vor ihm liegt – er muss nur um die Bäume herumlaufen. Glücklicherweise stehen nie zwei Bäume nebeneinander. Schreiben Sie ein Programm, das ihn bis zum Kleeblatt führt!

Aufgabe 3b – Kleeblattsuche im Wald II



Erweitern Sie Ihr Programm von Aufgabe 3a so, dass Kara auch mit mehreren nebeneinander stehenden Bäumen fertig wird! **Hinweis:** Die Lösung dieser Aufgabe erfordert zwei Zustände! Warum reicht ein Zustand nicht aus?

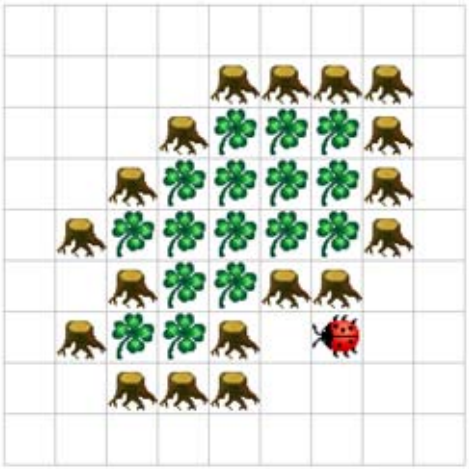
Aufgabe 4 – das Labyrinth



Kara steht in einem Labyrinth aus Bäumen. Schreiben Sie ein Programm, das ihn bis zum Ende des Labyrinths führt! Dort soll er ein Kleeblatt hinlegen; damit ist die Programm zu beenden.
Überlegen Sie Sich, wie „stabil“ Ihr Programm ist: was ist, wenn eine Wand ein „Loch“ hat (ein Baum fehlt)? Funktioniert es trotzdem noch? Unter welchen Bedingungen?

KARA, DER KLEEBLATT-BEWACHER!

Kara hat sich einen leckeren Vorrat von Kleeblättern angelegt. Dieser ist von Bäumen umringt. Damit sich niemand an seinen Kleeblättern vergreift, beschliesst Kara, diesen Wald zu patroullieren, ihn immer wieder abzulaufen. So sieht der „Vorratswald“ aus [followwall.world]:



Die Aufgabe (Schwierigkeitsgrad: 2 von 5)

Programmieren Sie Kara so, dass er endlos um diesen „Wald“ läuft! Sie können selbst entscheiden, ob Sie ihn links- oder rechtsherum laufen lassen.

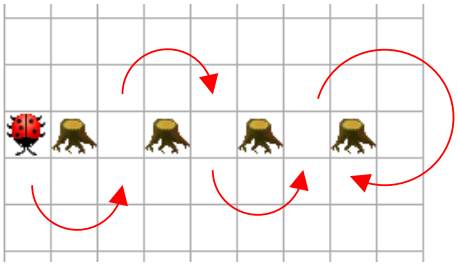
Hinweis: Ein Zustand reicht aus für die Lösung der Aufgabe. Versuchen Sie, mit möglichst wenig Sensoren auszukommen, und möglichst wenig Übergänge in ihrem Zustand zu haben!

Varianten

- Falls Ihr Programm mehr als einen Zustand hat: Schreiben Sie eine „optimierte“ Fassung, die nur einen Zustande hat!
- Erweitern Sie Ihr Programm so, dass Kara zuerst geradeaus bis zum nächsten Baum läuft und erst dann mit dem Patroullieren beginnt. Dazu müssen Sie ihn vor Programmbeginn allerdings so setzen, dass er geradeaus laufend überhaupt einen Baum findet!

KARA, DER SLALOM-SKATER

Inline-Skates faszinieren Kara! Er möchte üben, Slalom zu „fahren“ zwischen Bäumen [slalom.world]:



Starten mit einer Linksdrehung, dann zwischen den Bäumen elegant in eine Rechtsdrehung übergehen, dann zwischen den nächsten Bäumen wiederum in einer Linksdrehung... der Anfang des Slaloms ist eingezeichnet; so weiter ad infinitum! Oder bis Kara schwindlig wird!

Die Aufgabe (Schwierigkeitsgrad: 2 von 5)

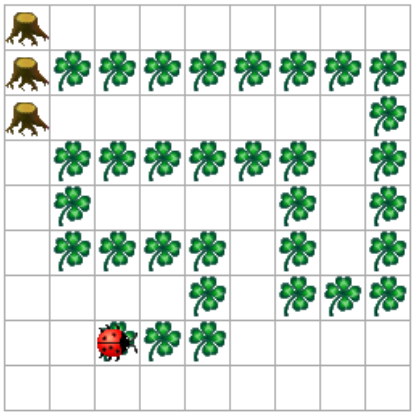
Programmieren Sie Kara so, dass er diesen Slalom fahren kann! Wie lang der Parcour ist (wieviele Bäume nebeneinander stehen), weiss Kara zu Beginn natürlich nicht! Auch soll es ihm egal sein, ob die Bäume horizontal oder vertikal nebeneinander stehen!

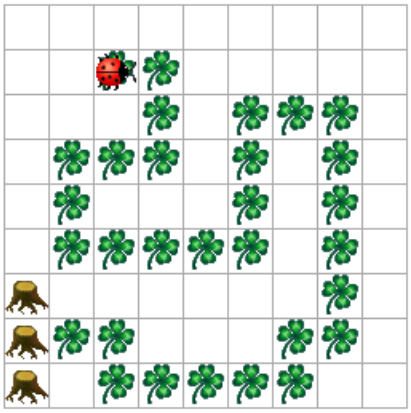
Zusatzfrage

Ihre Lösung hat mindestens zwei Zustände. Warum kann es mit weniger nicht gehen?

PacMan – oder wie verfolge ich eine Spur?

Kara möchte das uralte PC-Spiel PacMan spielen – in einer vereinfachten Version. Er soll eine Spur von Kleeblättern „auffressen“. Er weiss, dass diese Spur nie entlang eines Baumes geht – sie endet an einem Baum! So sehen „Spur-Welten“ aus [pacman1.world, pacman2.world]:





Die Aufgabe (Schwierigkeitsgrad: 3 von 5)

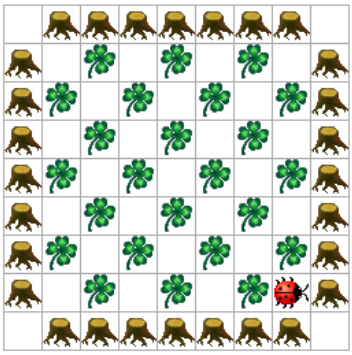
Programmieren Sie Kara so, dass er die Spur von Kleeblättern „auffrisst“! Da Sie wissen, dass die Spur nie entlang eines Baumes geht, kann das Programm beendet werden, sobald Kara auf einem Kleeblatt vor einem Baum steht. Sie können selbst bestimmen, ob Sie auf einem Kleeblatt oder davor starten wollen.

Hinweis I: Die Lösung ist trickreich! Überlegen Sie Sich zuerst auf Papier genau, in welchen Situationen Kara sich finden kann, und was er tun muss!

Hinweis II: Testen Sie Ihr Programm an mehreren verschiedenen Spuren! Sonst laufen Sie Gefahr, dass Ihr Programm nur für genau einen Spezialfall von Spur funktioniert. Das ist beim Programmieren immer eine sehr grosse Gefahr und kann leicht zu Problemen beim Einsatz eines Programmes führen!

KARA, DER PARKETTLEGER

Kara möchte sich als Parkettleger betätigen. Er will ein schachbrettmässiges Muster legen [chessboard.world]:



Die Aufgabe (Schwierigkeitsgrad: 2 von 5)

Programmieren Sie Kara so, dass er ein solches Muster innerhalb der Bäume erzeugt! Es gibt zwei grundsätzlich verschiedene Möglichkeiten, ein solches Muster zu erzeugen. Können Sie sich denken, welche zwei Varianten möglich sind?

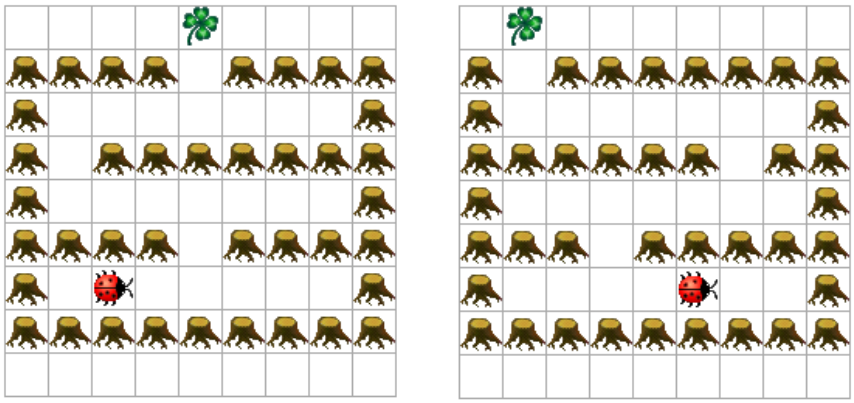
Hinweis: Wählen Sie die Weltgrösse so, dass Ihr Programm möglichst einfach wird. Je nach Verfahren sind die Anforderungen an die Weltgrösse unterschiedlich. Hilft es Ihrem Programm, wenn die Länge und Breite der Welt ungerade oder gerade sind? Oder hilft es zu wissen, dass Länge und Breite gleich gross sind?

Zusatzfrage

Wie gross ist die kleinste Welt, die Ihr Programm bearbeiten kann?

EINFACHES LABYRINTH – KLEEBLATT-SUCHE

Kara sitzt in einem Labyrinth fest. Er möchte raus, denn beim Ausgang des Labyrinths wartet ein leckeres Kleeblatt auf ihn! Zwei Beispiele von Labyrinth-Welten [levellabyrinth1.world, levellabyrinth2.world]:



Jede horizontale Baumreihe ausser der untersten hat genau einen Ausgang. Diesen muss Kara jeweils finden. Hinter dem letzten Ausgang wartet das Kleeblatt auf ihn!

Die Aufgabe (Schwierigkeitsgrad: 3 von 5)

Programmieren Sie Kara so, dass er das Kleeblatt findet! Er soll das Kleeblatt aufnehmen; damit soll das Programm enden. Die Startposition und Startrichtung in der untersten Labyrinthzeile können sie frei wählen. Sie müssen aber Ihr Programm daran anpassen!

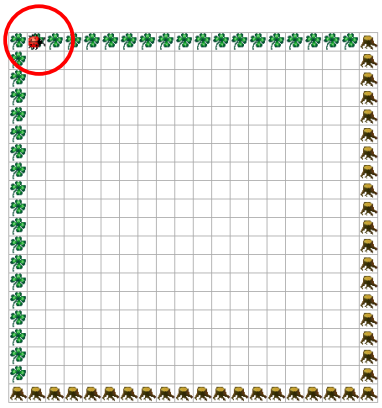
Hinweis: Testen Sie Ihr Programm an mehreren verschiedenen Labyrinthen! Sonst laufen Sie Gefahr, dass Ihr Programm nur für genau einen Spezialfall von Labyrinth funktioniert. Das ist beim Programmieren immer eine sehr grosse Gefahr und kann leicht zu Problemen beim Einsatz eines Programmes führen!

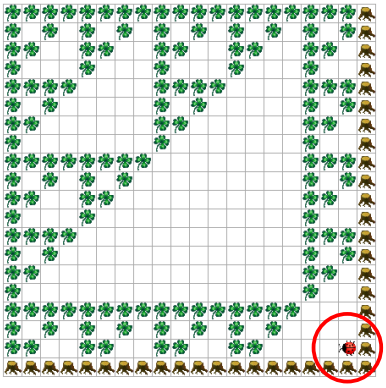
Zusatzfragen

- Wie verhält sich Ihr Programm, wenn es mehr als ein Loch in einer Baumreihe hat? Funktioniert es immer noch? Warum oder warum nicht?
- Falls Ihr Programm drei Zustände hat: Versuchen Sie, mit zwei Zuständen auszukommen! Ginge es auch mit einem einzigen Zustand?

KARA, DER MATHEMATIKER

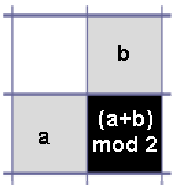
Kara möchte das berühmte Pascal-Dreieck zeichnen, bestehend aus Kleeblättern [pascal.world; pascal_done.world]:





Die Aufgabe (Schwierigkeitsgrad: 4 von 5)

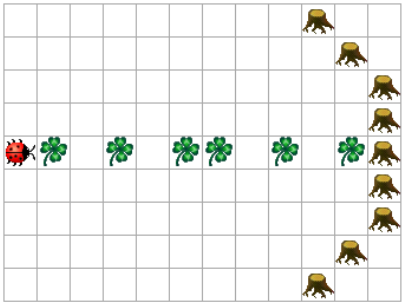
Programmieren Sie Kara so, dass er ein binäres Pascal-Dreieck zeichnet. Das heisst, jede Zahl wird „modulo 2“ abgebildet: eine gerade Zahl wird als freies Feld dargestellt, eine ungerade Zahl als Feld mit einem Kleeblatt drauf. Der Einfachkeit halber kippen wir das Dreieck in die linke obere Ecke der Welt. Die obige Abbildung zeigt die Ausgangslage für diese Aufgabe und das von Kara gezeichnete Pascal-Dreieck. Wie die inneren Felder berechnet müssen, zeigt folgende Abbildung:



Hinweis: Denken Sie daran: die Zustände sind Kara’s Gedächtnis! Diese Tatsache müssen Sie bei dieser Aufgabe ausnützen. Mit weniger als drei Zuständen dürften Sie nicht auskommen.

KARA UND DIE AUSSERIRDISCHEN – MUSTER IN DER WELT (1)

Kara glaubt an Ausserirdische. Er ist überzeugt, dass bestimmte Muster von Kleeblättern in seiner Welt auf ihre baldige Ankunft hinweisen! Nur weiss er nicht so recht, welches Muster er erwarten soll. Er stellt sich vor, dass es eines von drei Mustern ist. Betrachten wir als erstes folgende Welt [pattern010.world]:



Erste Aufgabe (Schwierigkeitsgrad: 3 von 5)

Programmieren Sie Kara so, dass er prüft, ob in der Zeile, in der er steht, folgendes Muster vorkommt:

Er soll solange das Muster suchen, bis er an einem Baum ankommt. Dann gibt er die Suche auf. Sobald er das Muster findet, soll er seine Freude darüber signalisieren, indem er sich links dreht und in die darüberliegende Zeile begibt. Danach kann er das Programm beenden. Findet er es nicht, geht er enttäuscht nach rechts in die darunterliegende Zeile.

Hinweis I: Für ein Muster, das drei Felder „lang“ ist, brauchen Sie drei Zustände. Überlegen Sie Sich zuerst auf Papier, wie Sie diese Zustände als Gedächtnis einsetzen können!

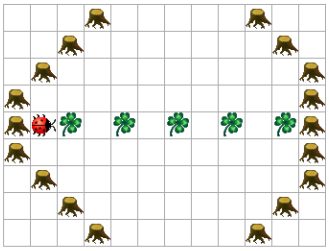
Hinweis II: Testen Sie Ihr Programm an mehreren verschiedenen Bildern! Testen Sie, ob Ihr Programm funktioniert, wenn das Muster vorkommt, wenn es nicht vorkommt, wenn es nur teilweise vorkommt und an dem Baum „unterbrochen“ wird**. Ihr Programm muss in allen Fällen die korrekte Antwort geben!**

Variante

Versuchen Sie, andere Muster zu suchen! Geben Sie Sich ein Muster vor, und versuchen Sie, es zu suchen. Je länger das Muster, desto mehr Zustände brauchen Sie! Warum?

KARA UND DIE AUSSERIRDISCHEN – MUSTER IN DER WELT (2)

Kara meint, ein anderes Muster könnte ihm auch einen Hinweis geben. Allerdings müsste dieses Muster beliebig oft wiederholt sein in einer Zeile. Die Welt sieht wie folgt aus [pattern01r.world]:



Zweite Aufgabe (Schwierigkeitsgrad: 3 von 5)

Programmieren Sie Kara so, dass er prüft, ob in der Zeile, in der er steht, folgendes Muster „aneinandergereiht“ ist:

Anders gesagt, die Zeile soll (wie im Bild) abwechslungsweise Felder ohne und mit Kleeblättern aufweisen. Sie muss ohne Kleeblatt beginnen und mit einem Kleeblatt enden – dann ist das Muster eine gewisse Anzahl mal aufgetreten! Im obigen Bild ist es fünf mal vertreten.

Kommt Kara zum Schluss, dass die Zeile das richtige Muster aufweist, soll er seine Freude darüber signalisieren, indem er sich links dreht und in die darüberliegende Zeile begibt. Findet er es nicht, geht er enttäuscht nach rechts in die darunterliegende Zeile.

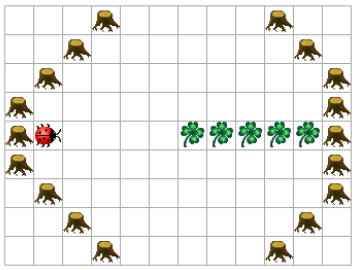
Hinweis: Testen Sie Ihr Programm an mehreren verschiedenen Bildern! Testen Sie, ob Ihr Programm funktioniert, wenn die Bildzeile dem beschriebenen Aufbau nicht genügt. **Ihr Programm muss in jedem Fall die korrekte Antwort geben!**

Variante

Versuchen Sie, andere aneinandergereihte Muster zu suchen! Geben Sie Sich ein Muster vor, und versuchen Sie, es zu suchen.

KARA UND DIE AUSSERIRDISCHEN – MUSTER IN DER WELT (3)

Kara meint, noch ein weiteres Muster könnte ihm einen Hinweis geben. Das Muster wäre: zuerst eine beliebige Anzahl Felder ohne Kleeblätter, dann die gleiche Anzahl Felder mit Kleeblätter [pattern0n1n.world]:



Dritte Aufgabe (Schwierigkeitsgrad: 3 von 5)

Kara hat allerdings seine Zweifel, ob seine Fähigkeiten ausreichen, ein derartiges Muster zu erkennen. Vor allem weiss er nicht von vornherein, wie lange die Zeile ist. Vielleicht lebt er ja in einer viel viel längeren Welt! Auch will er die Welt nicht verändern, denn sonst verärgert er womöglich die Ausserirdischen.

Was meinen Sie: Reichen Kara's Fähigkeiten aus, zu prüfen, ob die Zeile ein solches Muster enthält? Gehen Sie wie folgt vor:

1. Überlegen Sie Sich zunächst, auf Papier, eine Lösung für den Fall, dass die Zeile genau 4 Felder lang ist. Kein Problem, oder?

2. Überlegen Sie Sich nun, wie eine Lösung aussieht, wenn Sie wissen, dass die Zeile **maximal 4** Felder aufweist!

   Hinweis: Die Lösung braucht 4 Zustände.

3. Programmieren Sie Ihre Lösung von (3)! Auch hier gilt: Testen Sie Ihr Programm an möglichst vielen Mustern!!

   Hinweis: Schwierigkeitsgrad: 3-4.

4. Versuchen Sie, daraus abzuleiten, wo das Problem mit ganz langen Zeilen liegt!

Tja, wenn sich die Ausserirdischen mit einem derartigen Muster anmelden... Pech gewesen!

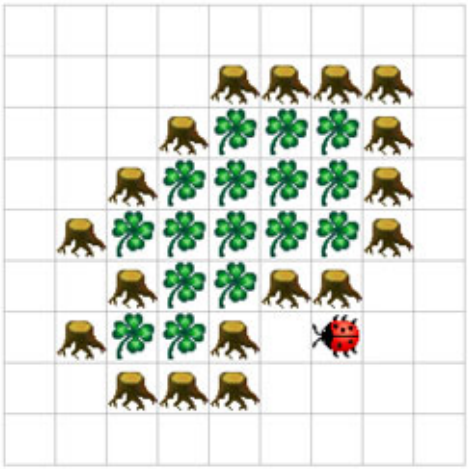
Freiwillige Zusatzaufgabe (Schwierigkeitsgrad: 5 von 5)

Wenn Kara erlaubt wird, die Welt zu verändern, so kann er die Aufgabe lösen! Er könnte zum Beispiel die Zeile ober- oder unterhalb der „Musterzeile" als Gedächtnis, als „Speicher" benutzen. Aber Achtung: die Lösung braucht 5 Zustände und ist ziemlich aufwendig!

Lösungen der Aufgaben

LÖSUNG: KARA, DER KLEEBLATT-BEWACHER!

Kara hat sich einen leckeren Vorrat von Kleeblättern angelegt. Dieser ist von Bäumen umringt. Damit sich niemand an seinen Kleeblättern vergreift, beschliesst Kara, diesen Wald zu patroullieren, ihn immer wieder abzulaufen. So sieht der „Vorratswald“ aus [followwall.world]:



Die Aufgabe (Schwierigkeitsgrad: 2 von 5)

Programmieren Sie Kara so, dass er endlos um diesen „Wald“ läuft! Sie können selbst entscheiden, ob Sie ihn links- oder rechtsherum laufen lassen.

Hinweis: Ein Zustand reicht aus für die Lösung der Aufgabe. Versuchen Sie, mit möglichst wenig Sensoren auszukommen, und möglichst wenig Übergänge in ihrem Zustand zu haben!

Varianten

- Falls Ihr Programm mehr als einen Zustand hat: Schreiben Sie eine „optimierte“ Fassung, die nur einen Zustande hat!
- Erweitern Sie Ihr Programm so, dass Kara zuerst geradeaus bis zum nächsten Baum läuft und erst dann mit dem Patroullieren beginnt. Dazu müssen Sie ihn vor Programmbeginn allerdings so setzen, dass er geradeaus laufend überhaupt einen Baum findet!

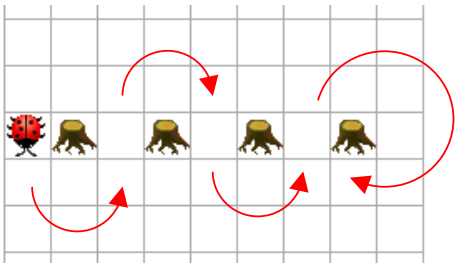
Lösung (mit Rechtsdrehung)

Das ist alles! In jeder Situation stellt das Programm sicher, dass nach der Ausführung der Befehle entweder rechts von Kara oder rechts hinter Kara wieder ein Baum ist. So folgt Kara endlos der „Wand“ aus Bäumen...

Damit kann das Prinzip der Invariante demonstriert werden: Kara stellt mit jedem Zustandsübergang sicher, dass die Bedingung „rechts von Kara oder rechts hinter Kara ist ein Baum“ erfüllt ist. So kann man sich von der Korrektheit des Programmes überzeugen – wer will, kann den Beweis natürlich auch formalisieren nach Dijkstra!

LÖSUNG: KARA, DER SLALOM-SKATER

Inline-Skates faszinieren Kara! Er möchte üben, Slalom zu „fahren“ zwischen Bäumen [slalom.world]:



Starten mit einer Linksdrehung, dann zwischen den Bäumen elegant in eine Rechtsdrehung übergehen, dann zwischen den nächsten Bäumen wiederum in einer Linksdrehung... der Anfang des Slaloms ist eingezeichnet; so weiter ad infinitum! Oder bis Kara schwindlig wird!

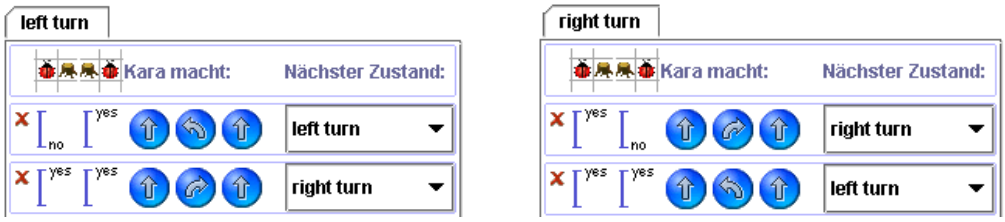
Die Aufgabe (Schwierigkeitsgrad: 2 von 5)

Programmieren Sie Kara so, dass er diesen Slalom fahren kann! Wie lang der Parcour ist (wieviele Bäume nebeneinander stehen), weiss Kara zu Beginn natürlich nicht! Auch soll es ihm egal sein, ob die Bäume horizontal oder vertikal nebeneinander stehen!

Zusatzfrage

Ihre Lösung hat mindestens zwei Zustände. Warum kann es mit weniger nicht gehen?

Lösung



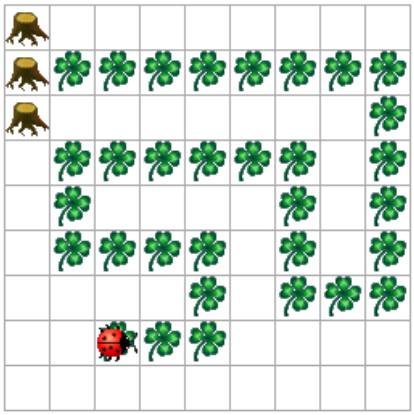
Start: Wenn Kara wie in der Abbildung steht, in „left turn“. Schaut er nach oben, hat also den Baum rechts von sich, dann in „right turn“.

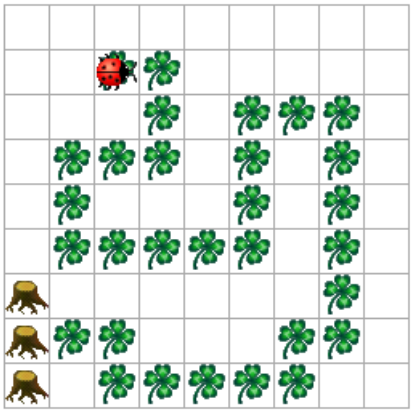
Lösung Zusatzfrage

Die Antwort auf die Zusatzfrage steckt eigentlich schon in der Aufgabenstellung: „*Starten mit einer Linksdrehung, dann zwischen den Bäumen elegant in eine Rechtsdrehung übergehen, dann zwischen den nächsten Bäumen wiederum in einer Linksdrehung...*“. Die beiden Zustände sind Kara’s **Gedächtnis**. Ohne sie kann er sich nicht merken, ob er *zwischen* zwei Bäumen eine Links- oder Rechtsdrehung vollziehen muss!

LÖSUNG: PACMAN – ODER WIE VERFOLGE ICH EINE SPUR?

Kara möchte das uralte PC-Spiel PacMan spielen – in einer vereinfachten Version. Er soll eine Spur von Kleeblättern „auffressen“. Er weiss, dass diese Spur nie entlang eines Baumes geht – sie endet an einem Baum! So sehen „Spur-Welten“ aus [pacman1.world, pacman2.world]:





Die Aufgabe (Schwierigkeitsgrad: 3 von 5)

Programmieren Sie Kara so, dass er die Spur von Kleeblättern „auffrisst“! Da Sie wissen, dass die Spur nie entlang eines Baumes geht, kann das Programm beendet werden, sobald Kara auf einem Kleeblatt vor einem Baum steht. Sie können selbst bestimmen, ob Sie auf einem Kleeblatt oder davor starten wollen.

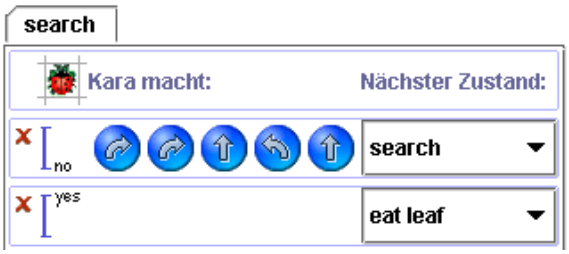
Hinweis I: Die Lösung ist trickreich! Überlegen Sie Sich zuerst auf Papier genau, in welchen Situationen Kara sich finden kann, und was er tun muss!

Hinweis II: Testen Sie Ihr Programm an mehreren verschiedenen Spuren! Sonst laufen Sie Gefahr, dass Ihr Programm nur für genau einen Spezialfall von Spur funktioniert. Das ist beim Programmieren immer eine sehr grosse Gefahr und kann leicht zu Problemen beim Einsatz eines Programmes führen!

Zusatzaufgabe

Falls Ihr Programm mehr als einen Zustand hat: Schreiben Sie eine „optimierte“ Fassung, die nur einen Zustande hat!

Lösung mit zwei Zuständen



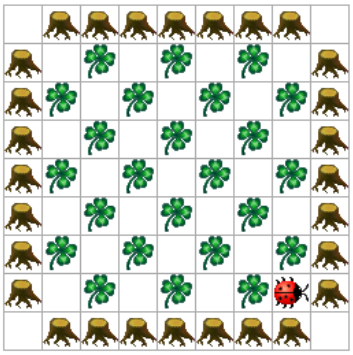
Start: in „eat leaf“, wenn Kara auf Kleeblatt startet; in „search“, wenn er davor steht.

Lösung mit nur einem Zustand

Bemerkung: „Optimiert“ ist hier nur die Anzahl Zustände – einer wurde eingespart. Sie lassen sich problemlos kombinieren, weil sie ganz andere Situationen abdecken. Die Anzahl Befehle hat sich nicht geändert.

Eine weitere, einiges schwierigere Aufgabe wäre es, Spuren zu „fressen“, die ganzen Baumreihen entlang gehen dürfen. Vielleicht nimmt ja jemand die Herausforderung an?

LÖSUNG: KARA, DER PARKETTLEGER



Kara möchte sich als Parkettleger betätigen. Er will ein schachbrettmässiges Muster legen [chessboard.world]:

Die Aufgabe (Schwierigkeitsgrad: 2 von 5)

Programmieren Sie Kara so, dass er ein solches Muster innerhalb der Bäume erzeugt! Es gibt zwei grundsätzlich verschiedene Möglichkeiten, ein solches Muster zu erzeugen. Können Sie sich denken, welche zwei Varianten möglich sind?

Hinweis: Wählen Sie die Weltgrösse so, dass Ihr Programm möglichst einfach wird. Je nach Verfahren sind die Anforderungen an die Weltgrösse unterschiedlich. Hilft es Ihrem Programm, wenn die Länge und Breite der Welt ungerade oder gerade sind? Oder hilft es zu wissen, dass Länge und Breite gleich gross sind?

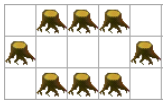
Zusatzfrage

Wie gross ist die kleinste Welt, die Ihr Programm bearbeiten kann?

Lösung 1: Kara läuft horizontal (oder vertikal) hin und her

Die Welt muss horizontal 2+1+n*2 (n>=1) Felder haben: zwei für die Bäume, 1 für die Startposition und jeweils zwei für jeden Zustandsübergang. Auf diese Weise kommt die Lösung mit nur zwei Zuständen aus. Vertikal muss mindestens eine Zeile zwischen den Bäumen frei sein.

Weltgrösse: Somit sieht die kleinstmögliche Welt, 5 auf 3 Felder gross, wie folgt aus:

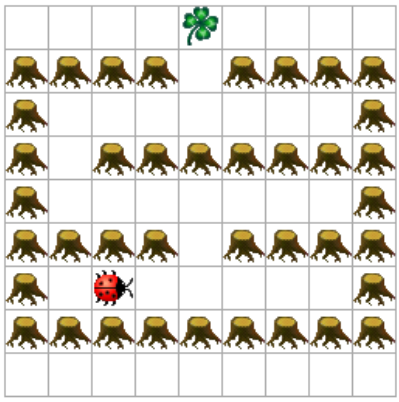


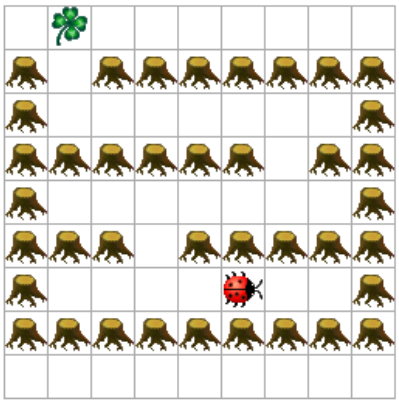
Lösung 2: Kara läuft diagonal hin und her

Start: Im linken unteren Ecken der Welt, nach rechts schauend, im Zustand „no leaves“.
Weltgrösse: Die kleinste Welt ist 4x4 Felder gross.

LÖSUNG: EINFACHES LABYRINTH – KLEEBLATT-SUCHE

Kara sitzt in einem Labyrinth fest. Er möchte raus, denn beim Ausgang des Labyrinths wartet ein leckeres Kleeblatt auf ihn! Zwei Beispiele von Labyrinth-Welten [levellabyrinth1.world, levellabyrinth2.world]:





Jede horizontale Baumreihe ausser der untersten hat genau einen Ausgang. Diesen muss Kara jeweils finden. Hinter dem letzten Ausgang wartet das Kleeblatt auf ihn!

Die Aufgabe (Schwierigkeitsgrad: 3 von 5)

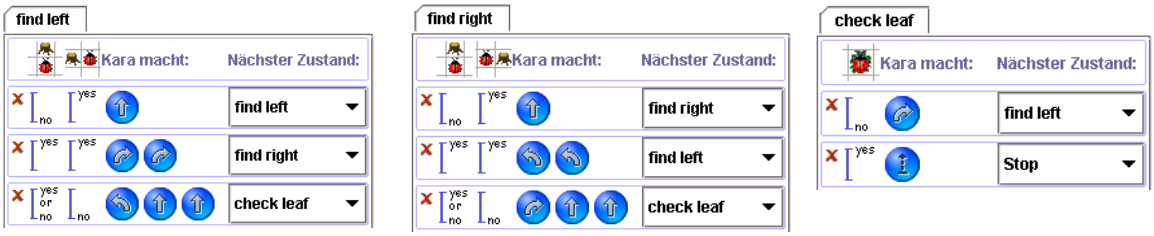
Programmieren Sie Kara so, dass er das Kleeblatt findet! Er soll das Kleeblatt aufnehmen; damit soll das Programm enden. Die Startposition und Startrichtung in der untersten Labyrinthzeile können sie frei wählen. Sie müssen aber Ihr Programm daran anpassen!

Hinweis: Testen Sie Ihr Programm an mehreren verschiedenen Labyrinthen! Sonst laufen Sie Gefahr, dass Ihr Programm nur für genau einen Spezialfall von Labyrinth funktioniert. Das ist beim Programmieren immer eine sehr grosse Gefahr und kann leicht zu Problemen beim Einsatz eines Programmes führen!

Zusatzfragen

- Wie verhält sich Ihr Programm, wenn es mehr als ein Loch in einer Baumreihe hat? Funktioniert es immer noch? Warum oder warum nicht?
- Falls Ihr Programm drei Zustände hat: Versuchen Sie, mit zwei Zuständen auszukommen! Ginge es auch mit einem einzigen Zustand?

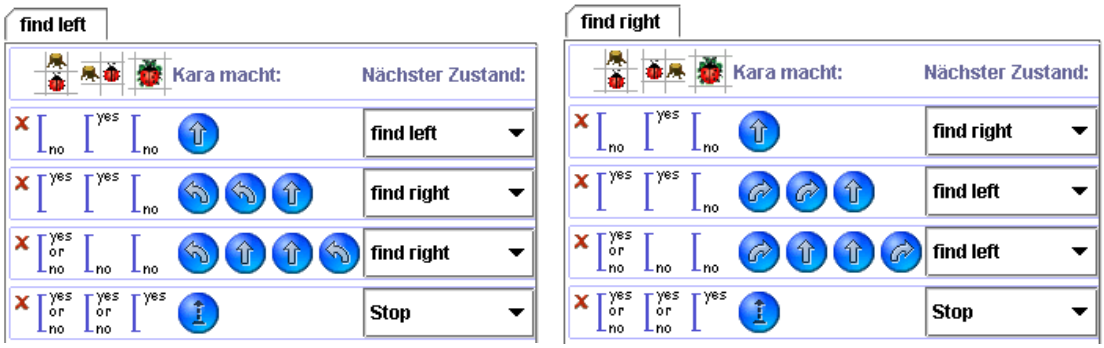
Lösung mit drei Zuständen



Start: „find left“, wenn Kara nach rechts schaut; „find right“, wenn er nach links schaut.

Anmerkung: Es ist egal, ob in „check leaf“ bei „leaf on ground? no“ nach „find left“ oder „find right“ gegangen wird. Kara muss nur in der entsprechenden Richtung gedreht werden.

Optimierte Lösung mit zwei Zuständen



Start: „find left“, wenn Kara nach rechts schaut; „find right“, wenn er nach links schaut.

Anmerkung: Man könnte noch in einem der beiden Zuständen den Sensor „leaf on ground?“ einsparen.

Zu den Zusatzfragen

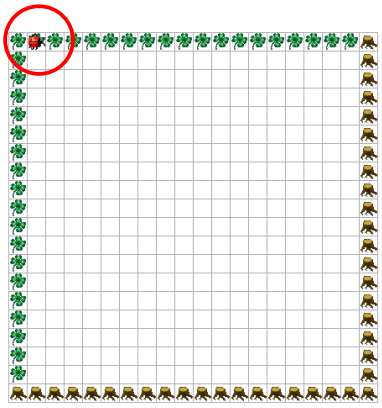
Löcher in den horizontalen Baumreihen stören das Programm nicht – solange die oberste Baumreihe nur genau bei dem Kleeblatt ein Loch hat! Oder aber die Löcher sind an das Programm angepasst so positioniert, dass sie den Programmablauf nicht stören!

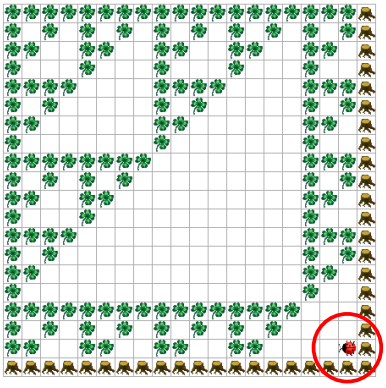
Mit einem Zustand kann es nicht gehen. Kara kann mit nur einem Zustand bei einem „Loch“ in der Baumreihe zu seiner Linken oder Rechten nicht wissen, ob er daran schon mal vorbeigekommen ist oder nicht, ob er hindurch soll oder nicht!

Wenn hingegen die Aufgabenstellung vereinfacht würde: die Löcher sind immer am Ende einer Baumreihe, dann würde ein Zustand ausreichen!

LÖSUNG: KARA, DER MATHEMATIKER

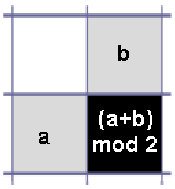
Kara möchte das berühmte Pascal-Dreieck zeichnen, bestehend aus Kleeblättern [pascal.world; pascal_done.world]:





Die Aufgabe (Schwierigkeitsgrad: 4 von 5)

Programmieren Sie Kara so, dass er ein binäres Pascal-Dreieck zeichnet. Das heisst, jede Zahl wird „modulo 2“ abgebildet: eine gerade Zahl wird als freies Feld dargestellt, eine ungerade Zahl als Feld mit einem Kleeblatt drauf. Der Einfachkeit halber kippen wir das Dreieck in die linke obere Ecke der Welt. Die obige Abbildung zeigt die Ausgangslage für diese Aufgabe und das von Kara gezeichnete Pascal-Dreieck. Wie die inneren Felder berechnet müssen, zeigt folgende Abbildung:



Hinweis: Denken Sie daran: die Zustände sind Kara’s Gedächtnis! Diese Tatsache müssen Sie bei dieser Aufgabe ausnützen. Mit weniger als drei Zuständen dürften Sie nicht auskommen.

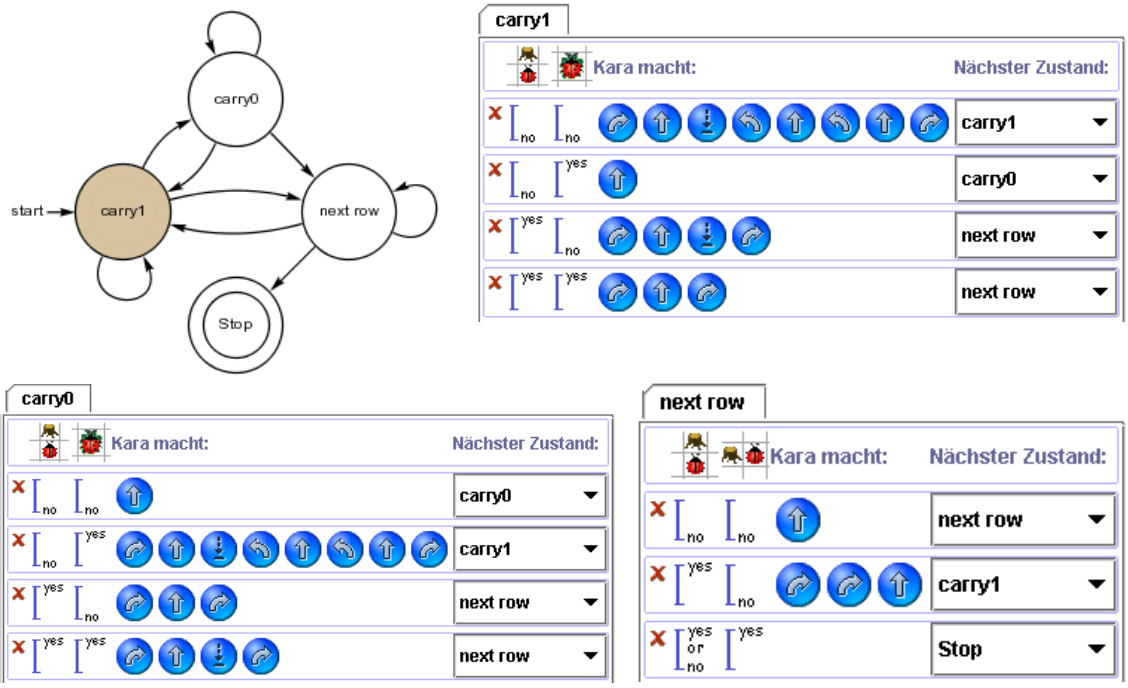
Lösung

Die Berechnung der Werte im Pascal-Dreieck, das heisst, das Plazieren von Kleeblättern durch Kara, ist im Prinzip einfach: Kara berechnet das Pascal-Dreieck zeilenweise von links nach rechts. Erreicht Kara bei seiner Berechnung das Ende einer Zeile, markiert durch einen Baum, so läuft er im Zustand `next row` an den Anfang der nächsten Zeile.

Wie berechnet Kara die Werte in den einzelnen Zellen des Pascal-Dreiecks? Kara kann über den Sensor „auf Kleeblatt?“ feststellen, ob er sich auf einem Feld mit einem Kleeblatt (also einer ungeraden Zahl) befindet oder nicht (das Feld **b** in obiger Abbildung links). Dann muss er noch wissen, ob sich auf dem Feld unmittelbar rechts hinter ihm (das Feld **a** in obiger Abbildung; die eingekreisten Felder rechts) ein Kleeblatt befindet oder nicht. Diese Information wird dadurch abgespeichert, dass sich Kara in einem der beiden folgenden Zustände befindet:

- `carry0`: Auf dem Feld hinten rechts von Kara befindet sich kein Kleeblatt (also eine gerade Zahl)
- `carry1`: Auf dem Feld hinten rechts von Kara befindet sich ein Kleeblatt (also eine ungerade Zahl)

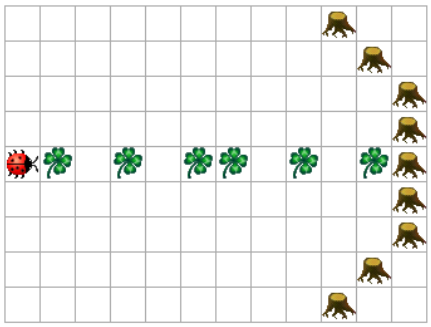
Befindet sich Kara im Zustand `carry0` und steht auf einem Feld mit keinem Kleeblatt, so bleibt das Feld rechts von Kara leer und Kara macht einen Schritt vorwärts. Falls er auf einem Feld mit einem Kleeblatt steht, so resultiert für das Feld rechts von Kara eine ungerade Zahl. Kara legt dort ein Kleeblatt hin und macht dann einen Schritt vorwärts. Entsprechend verhält sich Kara im Zustand `carry1`. Folgende Abbildung zeigt das ganze Programm.



LÖSUNG:

KARA UND DIE AUSSERIRDISCHEN – MUSTER IN DER WELT (1)

Kara glaubt an Ausserirdische. Er ist überzeugt, dass bestimmte Muster von Kleeblättern in seiner Welt auf ihre baldige Ankunft hinweisen! Nur weiss er nicht so recht, welches Muster er erwarten soll. Er stellt sich vor, dass es eines von drei Mustern ist. Betrachten wir als erstes folgende Welt [pattern010.world]:



Erste Aufgabe (Schwierigkeitsgrad: 3 von 5)

Programmieren Sie Kara so, dass er prüft, ob in der Zeile, in der er steht, folgendes Muster vorkommt:

Er soll solange das Muster suchen, bis er an einem Baum ankommt. Dann gibt er die Suche auf. Sobald er das Muster findet, soll er seine Freude darüber signalisieren, indem er sich links dreht und in die darüberliegende Zeile begibt. Danach kann er das Programm beenden. Findet er es nicht, geht er enttäuscht nach rechts in die darunterliegende Zeile.

Hinweis I: Für ein Muster, das drei Felder „lang“ ist, brauchen Sie drei Zustände. Überlegen Sie Sich zuerst auf Papier, wie Sie diese Zustände als Gedächtnis einsetzen können!

Hinweis II: Testen Sie Ihr Programm an mehreren verschiedenen Bildern! Testen Sie, ob Ihr Programm funktioniert, wenn das Muster vorkommt, wenn es nicht vorkommt, wenn es nur teilweise vorkommt und an dem Baum „unterbrochen“ wird**. Ihr Programm muss in allen Fällen die korrekte Antwort geben!**

Variante

Versuchen Sie, andere Muster zu suchen! Geben Sie Sich ein Muster vor, und versuchen Sie, es zu suchen. Je länger das Muster, desto mehr Zustände brauchen Sie! Warum?

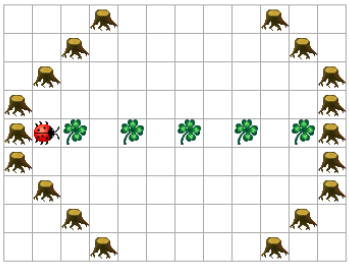
Lösung

Mit den Zuständen merkt sich Kara, welchen Teil des Musters er bereits erkannt hat. In „start“ fängt das Programm an. In „0“ weiss Kara, dass er bereits ein leeres Feld hinter sich hat. In „01“ weiss er, dass er ein leeres Feld und eines mit Kleeblatt hinter sich hat. Liegt daher in „01“ kein Kleeblatt auf dem Feld, weiss er dass er das Muster erkannt hat!

LÖSUNG:

KARA UND DIE AUSSERIRDISCHEN – MUSTER IN DER WELT (2)

Kara meint, ein anderes Muster könnte ihm auch einen Hinweis geben. Allerdings müsste dieses Muster beliebig oft wiederholt sein in einer Zeile. Die Welt sieht wie folgt aus [pattern01r.world]:



Zweite Aufgabe (Schwierigkeitsgrad: 3 von 5)

Programmieren Sie Kara so, dass er prüft, ob in der Zeile, in der er steht, folgendes Muster „aneinandergereiht“ ist:

Anders gesagt, die Zeile soll (wie im Bild) abwechslungsweise Felder ohne und mit Kleeblättern aufweisen. Sie muss ohne Kleeblatt beginnen und mit einem Kleeblatt enden – dann ist das Muster eine gewisse Anzahl mal aufgetreten! Im obigen Bild ist es fünf mal vertreten.

Kommt Kara zum Schluss, dass die Zeile das richtige Muster aufweist, soll er seine Freude darüber signalisieren, indem er sich links dreht und in die darüberliegende Zeile begibt. Findet er es nicht, geht er enttäuscht nach rechts in die darunterliegende Zeile.

Hinweis: Testen Sie Ihr Programm an mehreren verschiedenen Bildern! Testen Sie, ob Ihr Programm funktioniert, wenn die Bildzeile dem beschriebenen Aufbau nicht genügt. **Ihr Programm muss in jedem Fall die korrekte Antwort geben!**

Variante

Versuchen Sie, andere aneinandergereihte Muster zu suchen! Geben Sie Sich ein Muster vor, und versuchen Sie, es zu suchen.

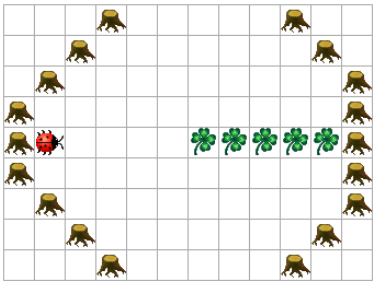
Lösung

Dieses Programm funktioniert ähnlich wie das der ersten Aufgabe. Im Zustand „0“ weiss Kara, dass ein leeres Feld hinter ihm liegt. Steht er auf einem Kleeblatt vor einem Baum, erfüllt die Zeile die Suchbedingung.

LÖSUNG:

KARA UND DIE AUSSERIRDISCHEN – MUSTER IN DER WELT (3)

Kara meint, noch ein weiteres Muster könnte ihm einen Hinweis geben. Das Muster wäre: zuerst eine beliebige Anzahl Felder ohne Kleeblätter, dann die gleiche Anzahl Felder mit Kleeblätter [pattern0n1n.world]:



Dritte Aufgabe (Schwierigkeitsgrad: 3 von 5)

Kara hat allerdings seine Zweifel, ob seine Fähigkeiten ausreichen, ein derartiges Muster zu erkennen. Vor allem weiss er nicht von vornherein, wie lange die Zeile ist. Vielleicht lebt er ja in einer viel viel längeren Welt! Auch will er die Welt nicht verändern, denn sonst verärgert er womöglich die Ausserirdischen.

Was meinen Sie: Reichen Kara's Fähigkeiten aus, zu prüfen, ob die Zeile ein solches Muster enthält? Gehen Sie wie folgt vor:

1. Überlegen Sie Sich zunächst, auf Papier, eine Lösung für den Fall, dass die Zeile genau 4 Felder lang ist. Kein Problem, oder?

2. Überlegen Sie Sich nun, wie eine Lösung aussieht, wenn Sie wissen, dass die Zeile **maximal 4** Felder aufweist!

   Hinweis: Die Lösung braucht 4 Zustände.

3. Programmieren Sie Ihre Lösung von (3)! Auch hier gilt: Testen Sie Ihr Programm an möglichst vielen Mustern!!

   Hinweis: Schwierigkeitsgrad: 3-4.

4. Versuchen Sie, daraus abzuleiten, wo das Problem mit ganz langen Zeilen liegt!

Tja, wenn sich die Ausserirdischen mit einem derartigen Muster anmelden... Pech gewesen!

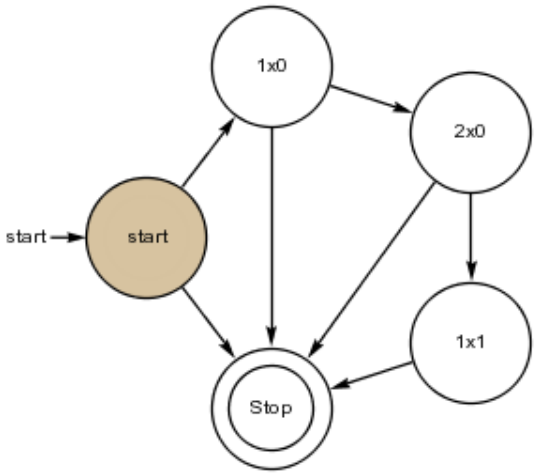
Freiwillige Zusatzaufgabe (Schwierigkeitsgrad: 5 von 5)

Wenn Kara erlaubt wird, die Welt zu verändern, so kann er die Aufgabe lösen! Er könnte zum Beispiel die Zeile ober- oder unterhalb der „Musterzeile" als Gedächtnis, als „Speicher" benutzen. Aber Achtung: die Lösung braucht 5 Zustände und ist ziemlich aufwendig!

Lösung

Zu 1. Die Lösung reduziert sich in diesem Fall auf „prüfe Muster“, leicht abgewandelt von der ersten Such-Aufgabe.

Zu 2, 3. Mit einer oberen Schranke gibt es eine relativ elegant, quasi symmetrische Lösung. Der Einfachkeit halber hier nur die Übersicht (patterns0n1n_max4.kara):



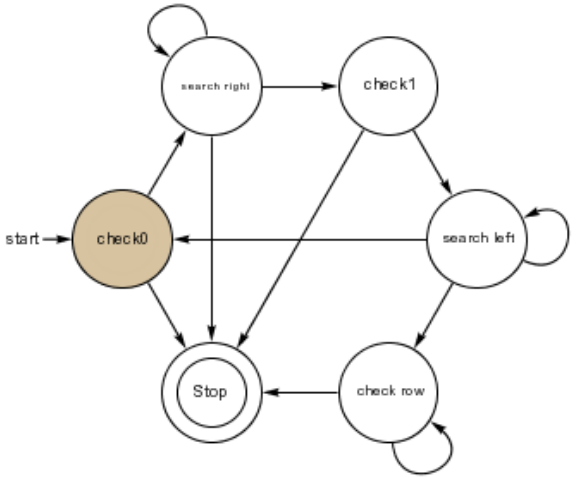
Die Idee ist einfach. Die Zustände merken sich, welchen Teil des Musters sie schon erkannt haben. Sie wissen auch, welcher Teil noch vor ihnen liegen muss. Diese Lösung kann leicht verallgemeinert werden für beliebige obere Schranken.

Zu 4. Es ist aber keine praktikable Lösung. Da Kara kein Gedächtnis in Form eines internen Zählers (Variable) hat, müssen Zustände die Rolle des Gedächtnises übernehmen. Für 6 Felder kann ein Automat mit 6 Zuständen erstellt werden. Es gilt: für *n* Gelder kann ein Automat mit *n* Zuständen erstellt werden.

Ein endlicher Automat kann nur reguläre Ausdrücke erkennen; das Muster „*n* leere Felder, *n* Felder mit Kleeblättern“, wobei *n* mindestens eins sein muss, ist kein regulärer Ausdruck – zumindest, wenn n gegen unendlich gehen darf. Hier könnte die ganze Theorie von der Mächtigkeit regulärer Ausdrücke und kontextfreier Grammatiken gezeigt werden...

Lösung der Zusatzaufgabe

Darf Kara die Welt verändern, so kann er die Aufgabe lösen. Die Welt dient ihm dann als Gedächtnis, als „Speicher“. Denn wäre die Welt unendlich lang, so könnte Kara jede Berechnung der Welt anstellen, weil er dann eine Turingmachine realisieren könnte... Hier geht’s aber etwas einfacher! Der Einfachheit halber hier die Lösung [pattern0n1n.kara] mit der Übersicht:



Start: Kara muss auf dem leeren Feld starten, dass vor dem ersten Feld mit Kleeblatt liegt, und muss in dessen Richtung schauen.

Die Idee ist folgende. In der Zeile oberhalb der Musterzeile markiert Kara die Felder, die dem Muster entsprechen. Zu Beginn erwartet er ein leeres Feld. Da er dieses hat, markiert er das Feld oberhalb mit einem Kleeblatt. Dann läuft er nach rechts auf der Suche nach dem ersten nicht markierten Feld. Unterhalb von diesem muss ein Kleeblatt sein. Ist dem so, markiert er das Feld oberhalb, läuft nach links auf der Suche nach einem unmarkierten Feld, prüft das Feld unter diesem Feld...

Die Zeile erfüllt das Muster genau dann, wenn bei Programmende die obere Zeile leer ist!