EDUCETH
Das Bildungsportal der ETH Zürich

Publikation von www.educeth.ch

Kara

Ein Leitprogramm von Judith Zimmermann, ETH Zürich

Inhalt und Lernziel:
Kara ist ein Marienkäfer, der in einer einfachen grafischen Welt auf dem Bildschirm lebt. Er kann programmiert werden, um in seiner Welt Aufgaben zu erledigen, zum Beispiel Kleeblätter zu sammeln. Die Schüler/innen erhalten in einer einfachen Umgebung einen Einstieg in die Programmierung.

Unterrichtsmethode:
Das Leitprogramm ist ein Selbststudienmaterial in Form eines Heftes oder eines elektronischen Angebotes. Es enthält alle notwendigen Unterrichtsinhalte, Übungen, Arbeitsanleitungen und Tests, die die Schüler/innen brauchen, um ohne Lehrperson lernen zu können.

Fachliches Review:
Juraj Hromkovic, Informationstechnologie und Ausbildung, ETH Zürich

Fachdidaktisches Review:
Juraj Hromkovic, Informationstechnologie und Ausbildung, ETH Zürich

Publiziert auf EducETH:
28. April 2006

Rechtliches:
Die vorliegende Unterrichtseinheit darf ohne Einschränkung heruntergeladen und für Unterrichtszwecke kostenlos verwendet werden. Dabei sind auch Änderungen und Anpassungen erlaubt. Der Hinweis auf die Herkunft der Materialien (ETH Zürich, EducETH) sowie die Angabe der Autorinnen und Autoren darf aber nicht entfernt werden.

Publizieren auf EducETH?
Möchten Sie eine eigene Unterrichtseinheit auf EducETH publizieren? Auf folgender Seite finden Sie alle wichtigen Informationen: http://www.educeth.ch/autoren

Weitere Informationen:
Weitere Informationen zu dieser Unterrichtseinheit und zu EducETH finden Sie im Internet unter http://www.educ.ethz.ch oder unter http://www.educeth.ch.

SCIENCE MEETS SCHOOL
EDUCETH.CH

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

ETH Institut für Verhaltenswissenschaften, Departement für Informatik ETH Zürich

Leitprogramm Kara

Einstieg in das Programmieren mit Automaten

Schulstufe: Gymnasium **Dauer: ca. 8 Stunden**

Autorin: Judith Zimmermann **Betreuer: Prof. Juraj Hromkovic**

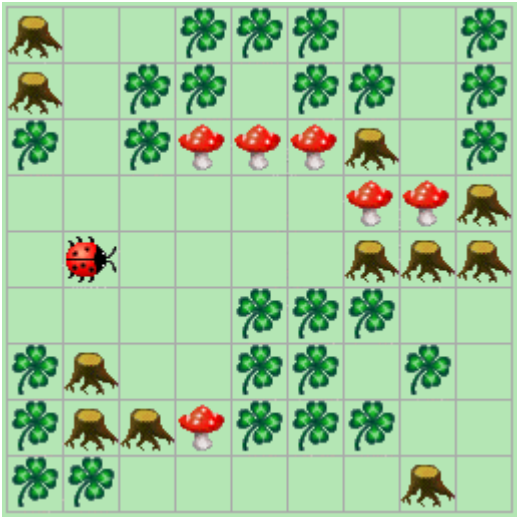
Fassung vom 23.01.07 **Keine Schulerprobung**

Die Kara-Umgebung und alle Aufgaben, die Kara in diesem Leitprogramm gestellt werden, wurden von Raimond Reichert entwickelt. Ebenso die Beispiele von alltäglichen Automaten.

Dank geht an Martina Zimmermann und Raimond Reichert für ihre hilfreichen Hinweise.

Teil 1

Einführung



Kara ist ein Marienkäfer. Seine Welt ist eine Wiese, die in quadratische Felder aufgeteilt ist. Kara kann von Feld zu Feld über die Wiese gehen. Dort können Kleeblätter und Pilze wachsen oder Baumstümpfe stehen. Kara soll mit deiner Hilfe verschiedene Aufträge ausführen und beispielsweise alle Blätter einsammeln.

Die Aufgabe besteht nun darin, Kara so zu programmieren, dass er selbständig seine Aufträge erfüllt. Kara folgt Befehlen und kann damit an sein Ziel gesteuert werden.

Man kann Kara die Anweisung geben, einen Schritt nach vorne zu gehen. Aber er könnte vor einem Baumstrumpf stehen und würde sich dann den Kopf stossen.
Um das zu verhindern, versteht Kara einige Fragen, die er mit Ja oder Nein beantworten kann. Verneint er die Frage, ob ein Baum auf dem Feld vor ihm steht, kann man ihm die Anweisung geben, einen Schritt vorwärts zu gehen. Falls dort doch ein Baum steht, muss man ihn um das Hindernis herumführen. Zum Beantworten der Fragen benutzt Kara *Sensoren*. Mit Hilfe der Sensoren kann Kara z. B. feststellen, ob das Feld vor ihm frei ist, ohne dass er das Feld betreten muss.

Kara programmiert man durch Wenn-dann-Aussagen.
Beispiele: Wenn Kara nicht vor einem Baum steht, dann soll er einen Schritt nach vorne gehen. Wenn Kara auf einem Blatt steht, dann soll er es aufnehmen.

Eine Folge solcher Wenn-dann-Aussagen ergibt ein Programm.
Wenn man es startet, kann man am Bildschirm Karas Weg über die Wiese mitverfolgen. Verhält sich Kara in jeder Situation so wie gedacht? Kann er die Aufgabe lösen? Nach dem gleichen Prinzip funktionieren auch bekannte Geräte: Videorekorder, Kaffeemaschinen, Getränkeautomaten, etc.

Am Ende dieses Leitprogramms hast du vieles gelernt:

- Du hast eine Vorstellung dafür entwickelt, was ein Programm ist, wie es abläuft und auf äussere Faktoren reagieren kann. Das erleichtert dir den Umgang mit vorhandener Software.
- Beim Programmieren schärfst du dein Verständnis für Logik. Nur durch korrektes Erkennen der Situation und geschicktes Anleiten von Kara kann er die Aufgaben lösen.

Inhaltsverzeichnis

Teil 1 ..... *2*
Einführung *2*
Inhaltsverzeichnis *3*
Arbeitsanleitung *4*
1. Kara und seine Umgebung *5*
1.1 Karas Welt **6**
2. Alltägliche Automaten *11*
2.1 Was Kara und Yoga gemeinsam haben **12**
2.2 Getränkeautomaten **13**
3. Kara programmieren *19*
3.1 Kara auf Papier **21**
3.2 Kara am Computer **24**
4. Aufgabensammlung *36*
Kara, der Kleeblatt-Bewacher **37**
Kara, der Slalom-Skater **38**
Kara, der Parkettleger **39**
Einfaches Labyrinth – Kleeblatt-Suche **40**
PacMan – oder wie verfolgt man eine Spur? **41**
Kara und die Ausserirdischen **42**
Muster in der Welt (1) **42**
Muster in der Welt (2) **43**
Muster in der Welt (3) **44**
Kara, der Mathematiker **45**
Teil 2 ..... *54*
Kapiteltests und Lösungen für die Lehrperson *54*
Anhang *60*
Material für die Lernenden **60**
Quellenangaben **61**

Arbeitsanleitung

Mit diesem Leitprogramm lernt man, Kara zu programmieren. Jeweils ein Kapitel wird selbständig bearbeitet und anschliessend beim Lehrer/ bei der Lehrerin mit dem Kapiteltest abgeschlossen. Die einzelnen Kapitel sind immer gleich strukturiert:

Übersicht
Was wird in diesem Kapitel behandelt? Was gibt es zu tun?

Lernziele
Was kann man nach dem Bearbeiten des Kapitels?

In Abschnitten mit neuem Lernstoff gibt es diese Symbole:

Information

Wissenssicherung

Aufgabe

Lernkontrolle
Selbstprüfung, ob alles verstanden wurde.

Lösungen
zu den Aufgaben der Lernkontrolle.

Nach der Lernkontrolle kannst du dich bei deiner Lehrerin oder deinem Lehrer für den **Kapiteltest** melden.

Unklarheiten?
Falls du ein Kapitel nicht verstanden hast, lies es nochmals gründlich durch. Wenn du dann noch Fragen hast, besprich sie zuerst mit einem Mitschüler/einer Mitschülerin und dann erst mit der Lehrperson.

Muss das ganze Leitprogramm bearbeitet werden?
Kapitel 1, 2 und 3 sind obligatorisch, müssen also bearbeitet werden.
Kapitel 4 ist eine Sammlung von Aufgaben mit Schwierigkeitsgraden zwischen 1 (leicht) und 5 (schwer). Du musst nicht alle Aufgaben aus Kapitel 4 bearbeiten. Wenn du Zeit hast, kannst du einzelne Aufgaben auswählen und versuchen zu lösen.

1. Kara und seine Umgebung

Übersicht

Inhalt
In diesem Kapitel wird Kara und seine Umgebung vorgestellt: Welche Befehle kennt Kara? Was findet er auf seiner Wiese und welche Eigenschaften haben diese Dinge? Um mit Kara und seiner Umgebung vertraut zu werden, wird er schrittweise und von Hand gesteuert, damit er kleine Aufgaben lösen kann.

Aufgabe
Lies zunächst die Abschnitte der Reihe nach durch, mach dich mit Karas Welt vertraut und spiel damit ein wenig. Dabei ist auch mehr erlaubt, als in den Aufgaben vorgegeben ist.

Lernziele

- Du kennst Kara und seine Umgebung und kannst ihm eine Welt zusammenstellen.
- Du kannst Kara von Hand steuern und weisst, was Sensoren sind.

1.1 Karas Welt

Das Programm wird mit einem Doppelklick der linken Maustaste auf das Kara-Programm gestartet. Wähle nun die Umgebung „Kara – Programmieren mit Automaten“ (gelb markiert). Es erscheint auf dem Bildschirm:

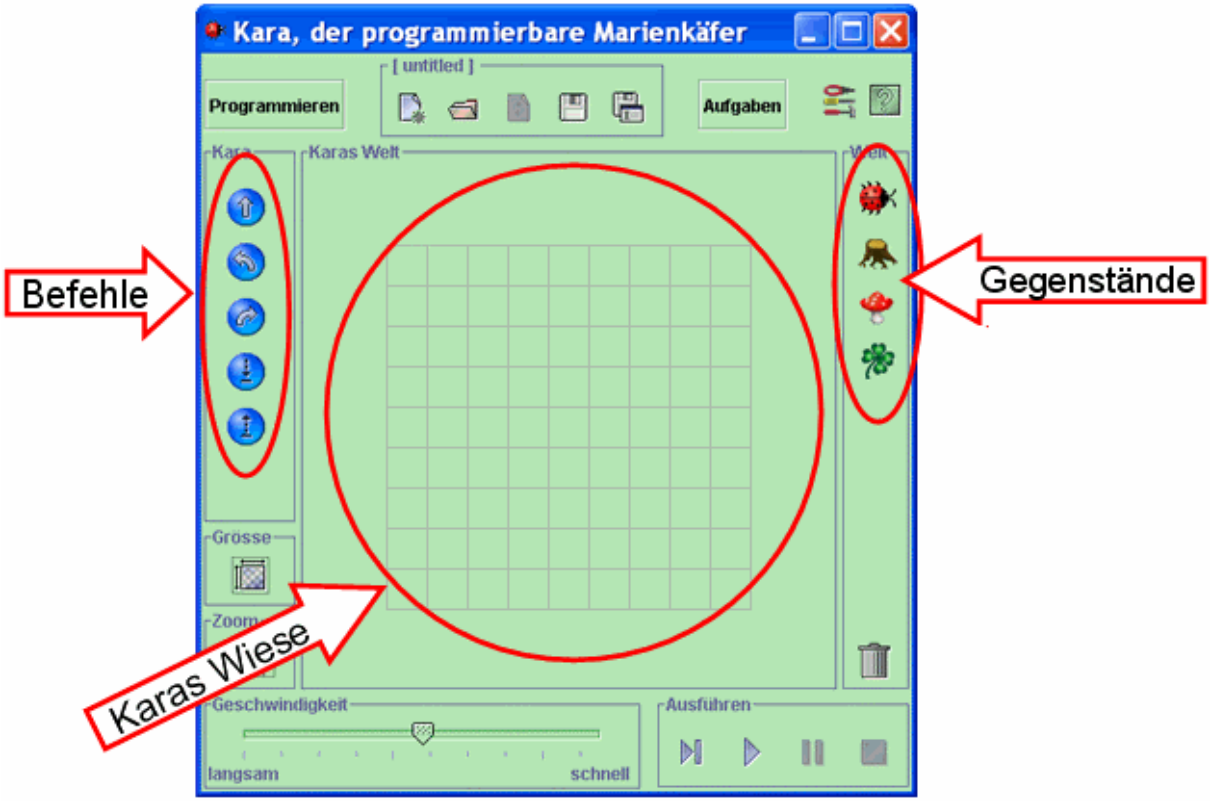


Abbildung 1: Karas Welt. Im Zentrum ist Karas Wiese zu sehen, die in Quadrate aufgeteilt ist. Rechts im Bild sind die Gegenstände in seiner Welt abgebildet. In jedem Viereck können ein oder mehrere Gegenstände platziert werden. Links stehen alle Befehle, die Kara ausführen kann.

Gegenstände

Man kann die Welt frei gestalten und beliebig viele Gegenstände auf die Wiese legen.

Das ist Kara. Kara kann in der Welt nur einmal vorkommen.

Kara kann Kleeblätter aufnehmen und ablegen. Er hat immer einen grossen Sack Kleeblätter dabei.

Der Baumstrumpf braucht ein Feld für sich alleine. Kara kann nicht über ihn klettern, sondern muss ihm ausweichen.

Pilze können zusammen mit Kleeblättern auf einem Feld stehen. Kara kann nicht über Pilze krabbeln. Er kann einen einzelnen Pilz verschieben, mehrere jedoch nicht.

Ziehe mit Drag-and-Drop (Gegenstand auswählen, mit gedrückter Maustaste zum Ziel ziehen, loslassen) den gewählten Gegenstand auf die Wiese. Willst du einen Gegenstand von der Wiese löschen, ziehe ihn mit Drag-and-Drop in den Mülleimer.

Befehle
Kara kennt die Befehle von der linken Seite der Wiese und führt sie folgsam aus.

Kara hüpft auf das vor ihm liegende Feld.
Falls ein Baumstumpf darauf steht, kann er nicht vorwärts gehen.

Kara bleibt auf seinem Feld stehen und dreht sich nach links.

Kara bleibt auf seinem Feld stehen und dreht sich nach rechts.

Kara legt ein Kleeblatt ab, wenn das Feld leer ist.
Liegt dort schon ein Kleeblatt, dann erscheint ein Hinweis.

Kara nimmt ein Kleeblatt vom Feld auf.
Falls dort kein Kleeblatt liegt, erscheint ebenfalls ein Hinweis.

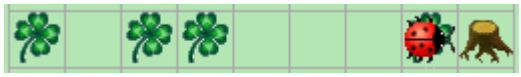
Mit der nächsten Aufgabe übt man, Kara zu bedienen.

Wissenssicherung 1.1

Schaffe diese Welt für Kara:

Kara muss zum Baumstumpf gehen. Unterwegs soll er jedes Kleeblatt aufnehmen. Liegt kein Kleeblatt auf einem Feld, soll er eines hinlegen. Benutze dazu die Befehle vom linken Rand.

Zum Schluss soll es so aussehen:



Lass Kara nun einen Schritt geradeaus gehen. Was passiert?

Sensoren
Wenn man Kara von Hand steuert, übernimmt man für ihn das Sehen. In Aufgabe 1.1 wurde jedes Mal überprüft, ob er auf einem Kleeblatt steht. Dann gab man ihm den Befehl, ein Kleeblatt abzulegen oder aufzunehmen. Anschliessend kontrollierte man, ob ein Baum seinen Weg versperrte oder er einen Schritt weiter gehen kann.

Kara soll sich nicht nur in einer einzigen Welt auskennen, sondern in jeder beliebigen. Dazu muss er so programmiert werden, dass er seinen Auftrag in verschiedenen Welten ausführen kann. Kara soll z. B. auf dem Weg zum Baum unterschiedlich viele Kleeblätter aufnehmen können, abhängig von der Anzahl der vorhandenen Blätter. Um diese Aufgabe zu lösen, kann Kara die unten aufgeführten Fragen mit Hilfe der Sensoren mit Ja oder Nein beantworten. Man kann Kara beispielsweise fragen, ob er auf einem Kleeblatt steht oder nicht.

	Steht Kara vor einem Baumstumpf?
	Ist links von Kara ein Baumstumpf?
	Ist rechts von Kara ein Baumstumpf?
	Steht Kara vor einem Pilz?
	Steht Kara auf einem Kleeblatt?

Aufgabe 1.2

Welche Sensoren würdest du einsetzen, damit Kara den Auftrag aus der Wissenssicherung 1.1 erledigen kann?

Aufgabe 1.3

Das ist die aktuelle Welt:

Kara startet in der angezeigten Position und führt dann nacheinander diese Aktionen aus:

Vervollständige die Tabelle. Überlege, was Karas Sensoren bei jedem Schritt in der aktuellen Welt sehen. Schreibe bei einem Nein als Antwort ein ×, bei einem Ja ein ✔ in die Tabelle.

Sensoren	Start	nach 1 Schritt	nach 2 Schritten	nach 3 Schritten	nach 4 Schritten	nach 5 Schritten
Baum vorne?	×					
Baum links?	✔					
Baum rechts?	×					
Pilz vorne?	×					
Kleeblatt unten?	✔					

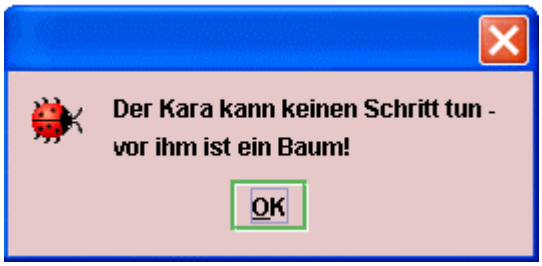
Kapiteltest

Wenn du weisst, wie man eine Welt für Kara baut und wie er von Hand gesteuert wird, gehe für den Kapiteltest zum Lehrer/zur Lehrerin.

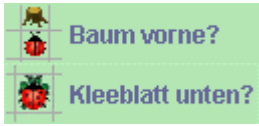
Lösungen Kapitel 1

Wissenssicherung 1.1
Diese Befehlsfolge bringt Kara ans Ziel:

Macht Kara noch einen Schritt vorwärts, läuft er vor den Baumstumpf.



Aufgabe 1.2
Man muss für die richtigen Anweisungen wissen, ob Kara auf einem Kleeblatt steht und ob er schon bei einem Baum angekommen ist. Benötigt werden also die beiden Sensoren:



Aufgabe 1.3

Sensoren	Start	nach 1 Schritt	nach 2 Schritten	nach 3 Schritten	nach 4 Schritten	nach 5 Schritten
Baum vorne?	×	×	×	×	×	✔
Baum links?	✔	×	×	×	×	×
Baum rechts?	×	×	✔	×	×	×
Pilz vorne?	×	×	×	×	×	×
Kleeblatt unten?	✔	×	×	✔	×	✔

2. Alltägliche Automaten

Übersicht

Worum geht es?
In diesem Kapitel werden verschiedene Automaten vorgestellt. Der Begriff „Automat“ bedeutet hier jedoch nicht das Gleiche wie in der Alltagsprache. Als Automat wird die Steuerung eines Getränke- oder Geldautomaten bezeichnet. Im nächsten Kapitel wird dieses Prinzip zum Programmieren von Kara verwendet.

Automaten basieren auf Wenn-dann-Aussagen. Wenn konkrete Bedingungen erfüllt sind, dann werden bestimmte Aktionen ausgeführt. Beispiel: Wenn bei einem Getränkeautomaten genügend Geld eingeworfen und Cola gewählt wurde, dann soll er eine Dose Cola auswerfen.

Automaten haben auch ein Gedächtnis. Der Getränkeautomat weiss, wie viel Geld bereits eingeworfen wurde.

Was tust du?
Bearbeite das Kapitel abschnittsweise und versuche, jede Aufgabe sofort zu lösen und nicht erst am Schluss des Kapitels, denn die Aufgaben bauen aufeinander auf. Lege dir einige Blätter Entwurfpapier bereit. Der Computer wird nicht gebraucht.

Lernziele

Nach dem Bearbeiten dieses Kapitels..

- weisst du, wie ein Automat funktioniert.
- kennst du die Begriffe: Zustand, Übergang, Input, Aktionen, Startzustand.

2.1 Was Kara und Yoga gemeinsam haben

Kara wird mit Hilfe von Automaten programmiert. Aber zuerst ein kleiner Exkurs in eine Yoga-Stunde, um dabei Automaten kennen zu lernen.

Beim Yoga nimmt man eine spezielle Stellung ein, zum Beispiel die Grussposition (Position 1) in Abbildung 2. Man bleibt für einige Zeit in dieser Position, bis man aufgefordert wird, die Stellung zu wechseln. In der neuen Stellung wartet man auf die nächste Anweisung, usw.

In das Wenn-dann-Schema übertragen heisst es: Wenn der Yogi in Position 1 steht und der Lehrer die Anweisung gibt, die Arme über den Kopf zu nehmen, dann soll der Yogi in die Position 2 wechseln.

Bei der Wahl der nächsten Anweisung muss die Lehrperson die aktuelle Körperhaltung des Yogi beachten. Steht der Yogi z. B. schon auf einem Bein und bekommt die Anweisung, ein Bein zu heben, dann landet er unsanft auf dem Boden.

Ein Ablauf im Yoga kann so aussehen:

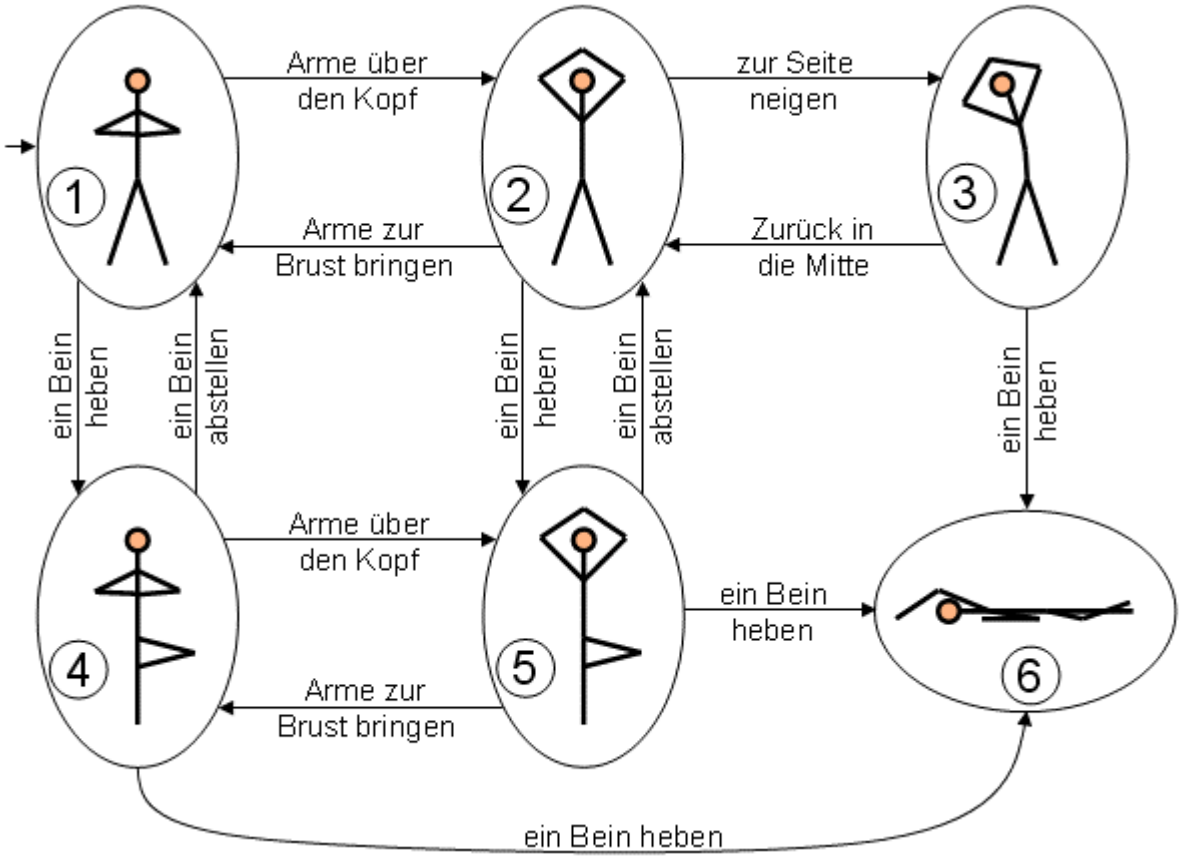


Abbildung 2: Eine Yogastunde

Eine Yogastunde wie in Abbildung 2 und ein Automat haben vieles gemeinsam.

Automat	**Yogalektion**
Ein Automat bleibt in einem Zustand und wartet darauf, über einen äusseren Sensor einen …	Der Yogi bleibt in einer Position und wartet auf die nächste …
Input zu erhalten. Ausgelöst durch diesen Input wählt er den …	Anweisung der Lehrperson. Sagt der Guru zum Beispiel „Arme über den Kopf nehmen“, folgt der Yogi dem …
passenden Übergang. Er führt die gespeicherten...	Pfeil mit der Beschriftung „Arme über den Kopf nehmen“. Dann …
Aktionen aus und wechselt entweder in einen anderen Zustand oder bleibt im alten Zustand.	führt er die Anweisung des Lehrers aus: Der Yogi hebt die Arme in die Höhe und nimmt die neue Position ein.
Jeder Automat hat einen Startzustand, in dem er mit der Arbeit beginnt.	Damit die Yogis auch mit den Übungen beginnen können, wenn der Guru zu spät kommt, beginnen sie immer in der Grusspposition.

Im nächsten Abschnitt wird ein erster Automat aus dem Alltag genauer betrachtet.

2.2 Getränkeautomaten

Einen Getränkeautomaten hat wohl jeder schon einmal benutzt. Wie kann man ihn steuern?

Abbildung 3: Getränkeautomat. Die Durstigen werfen Geld in den Automaten und wählen das gewünschte Getränk. Der Automat gibt auf der Kreditanzeige den eingeworfenen Betrag an. Wenn ein Getränk gewählt wurde und genügend Kredit vorhanden ist, wirft er die gewünschte Dose aus und setzt die Kreditanzeige auf Null.

Der Getränkeautomat hat diese Funktionen:
Er akzeptiert nur 1- und 2-Franken-Stücke.
Er zeigt den eingeworfenen Betrag an.
Er gibt nur Cola und ein weiteres Getränk aus.
Eine Dose kostet 3.- Franken,

Das Innenleben des Automaten wird nun etwas genauer betrachtet.
Der Automat muss vier verschiedene Situationen unterscheiden können: Null, ein, zwei und drei Franken Kredit. Zur Unterscheidung hat er vier verschiedene Zustände. Jeder Zustand steht für eines der möglichen Guthaben.

Abbildung 4: Die vier Zustände des Getränkeautomaten

Die Kunden können auch mehr Geld einwerfen. Um es zu vereinfachen wird aber angenommen, dass die Kunden maximal drei Franken einwerfen.

Im folgenden Schritt kommen die Übergänge hinzu. Pfeile also, die den Übergang von einem Zustand zum Nächsten kennzeichnen. Auf den Übergängen wird zuerst der notwendige Input eingetragen, z. B. Kunde wirft 2.- sFr ein. Durch einen Schrägstrich abgetrennt steht die Automatenaktion daneben, im Beispiel die Anzeige von 2.- sFr auf dem Display.

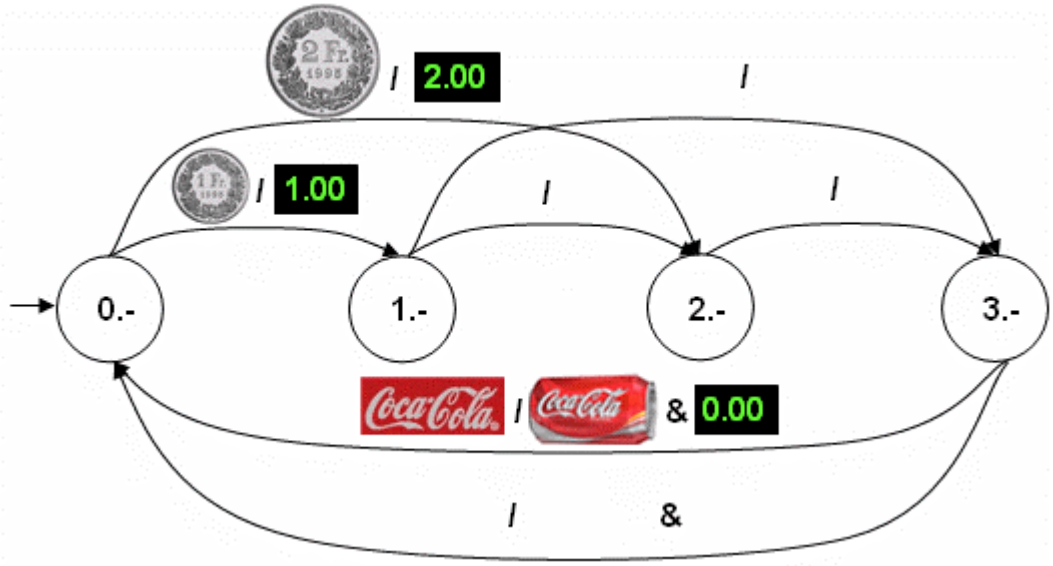


Abbildung 5: Getränkeautomat mit Übergängen. Für diesen Automaten sind alle relevanten Übergänge eingetragen. Drei der sieben Übergänge sind schon vollständig beschriftet.

Aus dem Zustand „0.- sFr Kredit“ (ganz links) heraus gibt es zwei Übergänge:
Wird ein 1-Franken-Stück eingeworfen, wird der Übergang in den Zustand „1.- Franken Kredit“ genommen. Zusätzlich muss auf dem Display „1.00“ angezeigt werden.
Wird ein 2-Franken-Stück eingeworfen, kommt ein anderer Übergang zum Zug. Es wird in den Zustand „2.- Franken Kredit“ gewechselt. Das Display zeigt „2.00“ an.

Der dritte, beschriftete Übergang führt aus dem Zustand „3.- Franken Kredit“ in den Zustand „0.- Franken Kredit“. Dazu muss der Kunde ein Getränk gewählt haben. Der Automat gibt eine Dose aus und setzt das Display auf „0.00“.

Wissenssicherung 2.1

In Abbildung 5 sind nicht alle Übergänge beschriftet. Vervollständige das Schema des Getränkeautomaten. Du kannst selbst wählen, was für ein weiteres Getränk dieser Automat verkauft. Welcher Zustand ist der Startzustand?
Verfasse drei Wenn-dann-Aussagen zum Getränkeautomaten.

Aufgabe 2.2

Zeichne einen Automaten, der einen Lichtschalter steuert.
Ein Sensor meldet dem Automaten, wenn der Schalter gedrückt wurde.

Die Aufgabe ist gut gelöst, wenn:

- der Automat korrekt arbeitet.
- die Zustände aussagekräftige Namen haben.
- jeder Übergang eingezeichnet und mit dem Input und der Aktion beschriftet ist.

Aufgabe 2.3

Zeichne einen Automaten für die Steuerung eines Videorecorders. Der Videorecorder hat der Einfachheit halber nur die folgenden Tasten: ▶ ▶▶ ■.

Für diese Aufgabe soll jeder mögliche Übergang eingezeichnet werden. Für alle Zustände und für jeden möglichen Input muss es somit einen Übergang geben.
Unter Umständen führt ein Übergang wieder in einen Zustand zurück. Bisher wurden zur besseren Übersicht nur die wichtigsten Übergänge eingezeichnet.

Die Aufgabe ist dann gut gelöst, wenn:

- der Automat korrekt arbeitet.
- die Zustände aussagekräftige Namen haben.
- jeder Übergang eingezeichnet und mit dem Input und der Aktion beschriftet ist.
- der Startzustand angegeben ist.

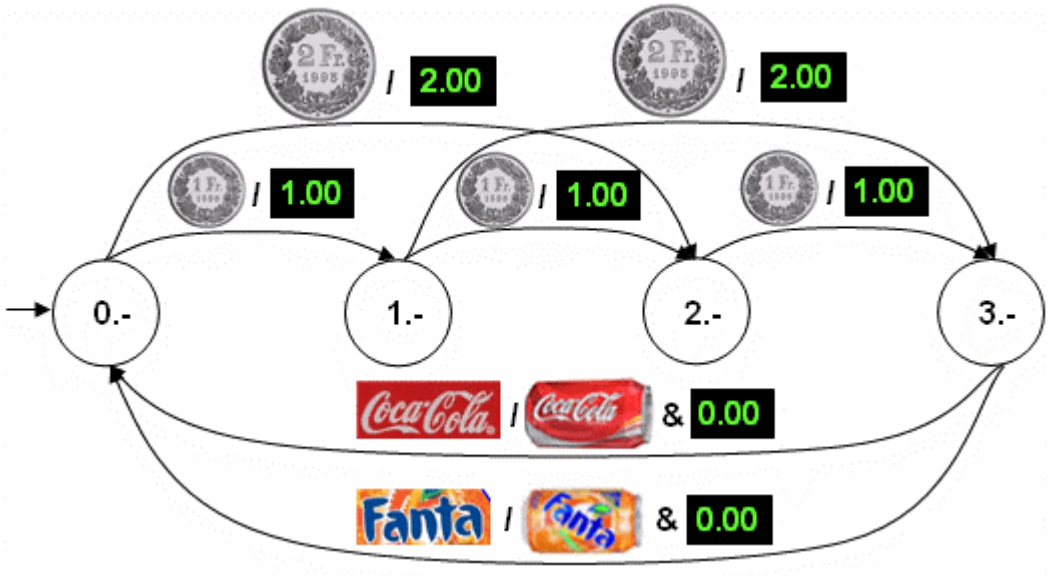
Kapiteltest

Wenn dir die Begriffe Zustand, Übergang, Input, Aktionen und Startzustand klar sind, kannst du zur Lehrperson gehen und den Kapiteltest absolvieren.

Lösungen Kapitel 2

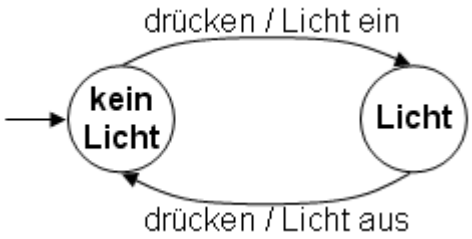
Wissenssicherung 2.1

a) + b) Als zweites Getränk wurde Fanta gewählt.

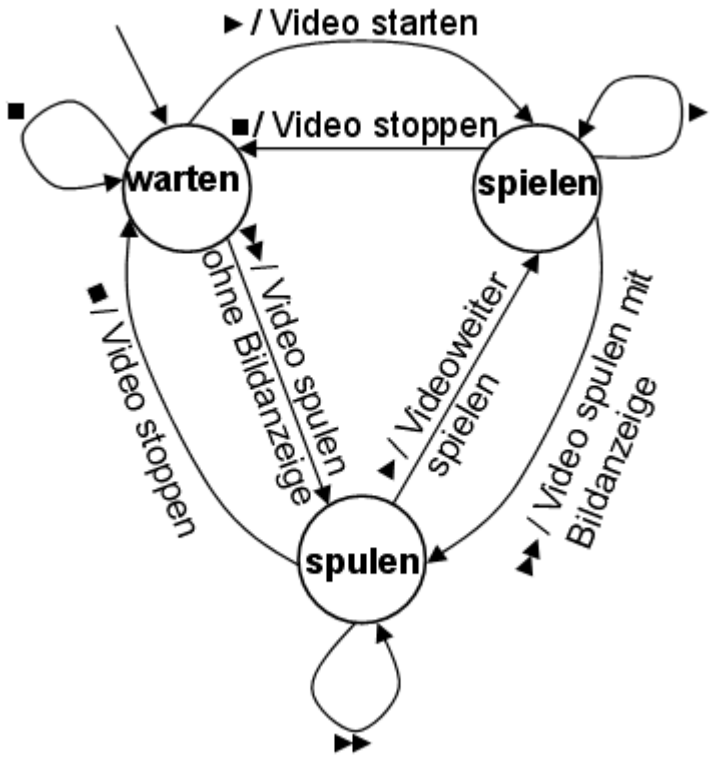


c) Es gibt sehr viele Möglichkeiten. Einige Beispiele:
Wenn der Automat im Zustand „0.- Franken Kredit“ ist und jemand 1.- sFr einwirft, dann soll der Automat „1.00“ auf dem Display anzeigen und in den 1- Franken-Kredit-Zustand wechseln.
Wenn der Automat im Zustand „1.- Franken Kredit“ ist und jemand einen 2.- sFr einwirft, dann soll der Automat „3.00“ auf dem Display anzeigen und in den 3-Franken-Kredit-Zustand wechseln.
Wenn der Automat im Zustand „3.- Franken Kredit“ ist und jemand auf den Cola-Knopf drückt, dann soll der Automat eine Cola-Dose auswerfen, „0.00“ auf dem Display anzeigen und in den Zustand „0.- Franken Kredit“ wechseln.

Aufgabe 2.2



Aufgabe 2.3



Es gibt auch andere Möglichkeiten, diese Aufgabe zu lösen. Wenn du nicht sicher bist, ob deine Lösung gut ist, besprich sie mit deinem Lehrer oder deiner Lehrerin.

3. Kara programmieren

Übersicht

Worum geht es?
In diesem Kapitel wird dargelegt, wie man Kara mit Hilfe eines Automaten programmieren kann. Es ist wichtig, dass Kara seine Aufgabe jeweils in unterschiedlichen, aber gleich strukturierten Welten erfüllen kann.
Kara kann z. B. alle vor ihm liegenden Blätter aufnehmen, unabhängig von ihrer Anzahl und Verteilung auf den Feldern. Die Welten in den Aufgaben dienen als Beispiele.

Was tust du?
Lies die Abschnitte der Reihe nach durch und bearbeite die Aufgaben direkt. Damit du dein Programm testen kannst, sind mehrere Welten vorgegeben. Du kannst natürlich selbst noch zusätzliche Welten aufbauen. Lass dein Programm in allen Welten laufen und überprüfe, ob Kara seine Aufgabe erledigen kann.

Lernziele

Nach dem Bearbeiten dieses Kapitels,

- kannst du Kara so programmieren, dass er einfache Aufgaben selbständig lösen kann.
- weisst du, wie man Fehler im Programm finden und beheben kann.

Kara soll nun programmiert werden, dazu wird für jede Aufgabe ein Automat geschrieben, der Kara durch seine Welt steuert. Der Automat ist Karas Gehirn. Er bekommt von den Sensoren Informationen über die Umgebung. Das ist der Input für den Automaten. Die Aktionen des Automaten sind Befehle an Kara, die er ausführen soll.

Sensoren

	Steht Kara vor einem Baumstumpf?
	Ist links von Kara ein Baumstumpf?
	Ist rechts von Kara ein Baumstumpf?
	Steht Kara vor einem Pilz?
	Steht Kara auf einem Kleeblatt?

Befehle

Kara hüpft auf das Feld vor ihm. Falls ein Baum darauf steht, kann er nicht vorwärts gehen.

Kara bleibt auf dem Feld stehen und dreht sich nach links.

Kara bleibt auf dem Feld stehen und dreht sich nach rechts.

Kara legt ein Kleeblatt ab, falls das Feld leer ist. Liegt dort schon ein Kleeblatt, dann erscheint ein Hinweis.

Kara nimmt das Kleeblatt vom Feld auf. Falls kein Kleeblatt auf dem Feld liegt, erscheint ebenfalls ein Hinweis.

3.1 Kara auf Papier

Kara, der Blättersammler
Aufgabe: Schreibe ein Programm, das Kara geradeaus bis zum nächsten Baumstumpf führt. Kara weiss, dass irgendwo vor ihm ein Baumstumpf steht. Liegt auf einem Feld ein Blatt, soll er es aufnehmen. Wenn er am Baum ankommt, ist das Programm beendet.
Achtung: Nur wenn ein Baumstumpf vor Kara liegt, läuft das Programm korrekt!

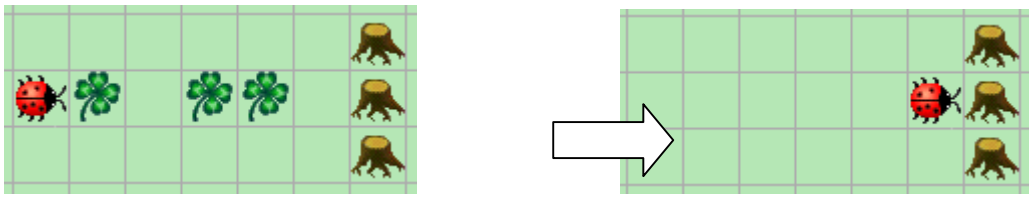


Abbildung 6: Das Bild links zeigt die Ausgangslage. Auf dem Bild rechts sieht man die Welt, nachdem Kara seinen Auftrag erledigt hat.

Zuerst soll der Automat auf einem Blatt Papier entworfen werden. Dann definiert man Sensoren. Zum Lösen dieser Aufgabe ist es unwichtig, ob Bäume links oder rechts von Kara sind. Man braucht nur wissen, wann Kara am Baum angekommen ist und ob er auf einem Kleeblatt steht oder nicht.

Welche Zustände kommen vor?
Man braucht zwei Zustände: einen Zustand zum Sammeln der Kleeblätter und den Stoppzustand. Der Stoppzustand soll erreicht sein, sobald Kara vor dem Baum steht. Die bisher betrachteten Automaten haben die Arbeit nie beendet, sondern einfach auf neue Benutzer gewartet. Bei Kara ist es jedoch anders. Er soll melden, wenn er mit der Arbeit fertig ist. Das erreichet man, indem er nach beendeter Arbeit in den Stoppzustand wechselt.

Abbildung 7: Die beiden Zustände des Automaten für den Blättersammler. So lange Kara mit der Arbeit nicht fertig ist, soll er im Zustand „Blätter sammeln“ bleiben. Sobald die Arbeit verrichtet ist, soll er in den Stoppzustand wechseln.

Welche Übergänge werden benötigt?
Für alle denkbaren Kombinationen der Sensoren werden zunächst Übergänge hinzugefügt. Eine möglicher Input durch die beiden Sensoren ist: Vor Kara steht kein Baum und Kara steht nicht auf einem Kleeblatt. Es entspricht der ersten Zeile in Abbildung 8.

×	×
×	✔
✔	×
✔	✔

Abbildung 8: In dieser Tabelle sind alle möglichen Kombinationen der Inputs aufgelistet, die Kara über die gewählten Sensoren erhalten kann. Ein Kreuz bedeutet, dass der Sensor „Nein“ meldet. Ein Haken bedeutet „Ja“.

Nun überlegt man, wie Kara in jedem der vier Fälle reagieren soll und welchen Zustand er anschliessend einnehmen muss. Solange die Aufgabe ungelöst ist, darf nicht in den Stoppzustand gewechselt werden. Wenn der Auftrag erledigt ist, muss hingegen in den Stoppzustand übergegangen werden.

		Karas Aktionen	nächster Zustand
×	×		Blätter sammeln
×	✔		Blätter sammeln
✔	×	-	Stopp
✔	✔		Stopp

Abbildung 9: Diese Tabelle zeigt, welche Aktionen Kara aufgrund der Inputs durch die Sensoren ausführen soll. Die erste Zeile der Tabelle zeigt: Wenn Kara weder vor einem Baum, noch auf einem Kleeblatt steht, dann soll er einen Schritt nach vorne machen.

Die Zustände aus Abbildung 7 und die Übergänge aus Abbildung 8 ergeben den folgenden Automaten:

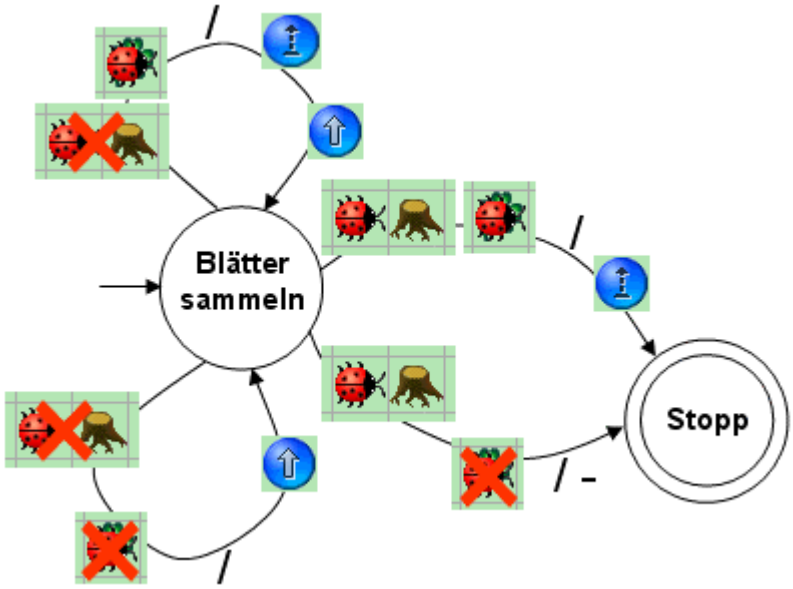
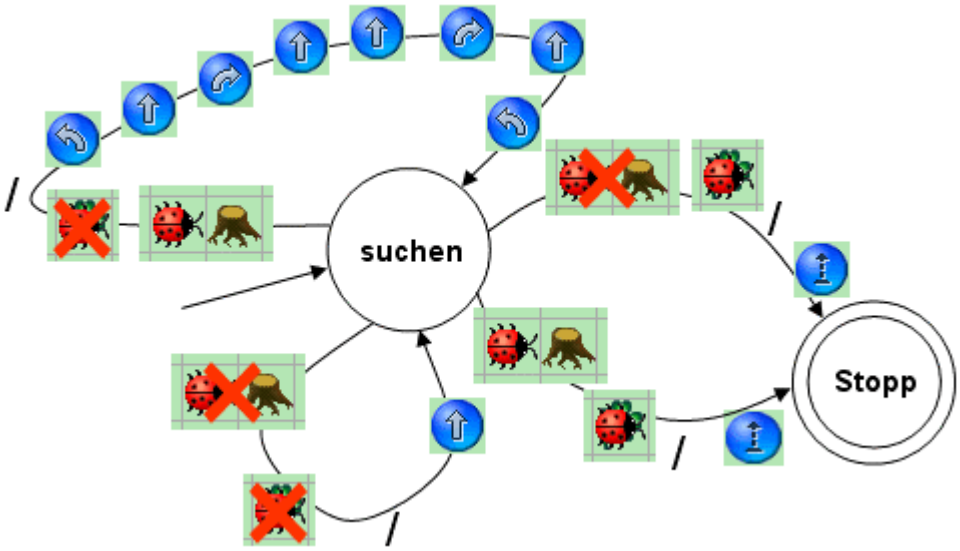


Abbildung 10: Automat, der Kara anleitet, geradeaus bis zum nächsten Baum zu gehen und unterwegs alle Blätter aufzunehmen.

Wissenssicherung 3.1

In dieser Aufgabe soll ein bestehender Automat analysiert werden.



Gegeben ist die Wiese unten. Zeichne Karas Weg ein, wenn er durch den oben abgebildeten Automaten gesteuert wird.

Hinweis: Wenn Kara während eines Ausweichmanövers gegen einen Baum läuft, wird eine Fehlermeldung ausgegeben und der Automat wechselt in den Stoppzustand.

Im folgenden Abschnitt wird Kara mit diesem Automaten der Aufgabe 3.1 programmiert, so dass er selbständig um die Bäume gehen kann und das gesuchte Blatt im Wald findet.

3.2 Kara am Computer

Aufgabe 3.2

Auf den folgenden Seiten wirst du Schritt für Schritt angeleitet, den Automaten aus Aufgabe 3.1 zu programmieren. Lies die Abschnitte der Reihe nach durch und führe die mit dem Symbol ➢ gekennzeichneten Arbeitsschritte direkt aus.

Die Lösungen zu dieser Aufgabe sind direkt im Text eingebaut. So kann man jederzeit sein Ergebnis kontrollieren. Die komplette Lösung steht auf Seite 32/33.

Hinweis: Automat und Programm bedeuten für Kara das Gleiche.

➢ Starte Kara und wähle „Kara – Programmieren mit Automaten“. Klicke dann oben links auf „programmieren“ und das Programmierfenster für Kara wird geöffnet.

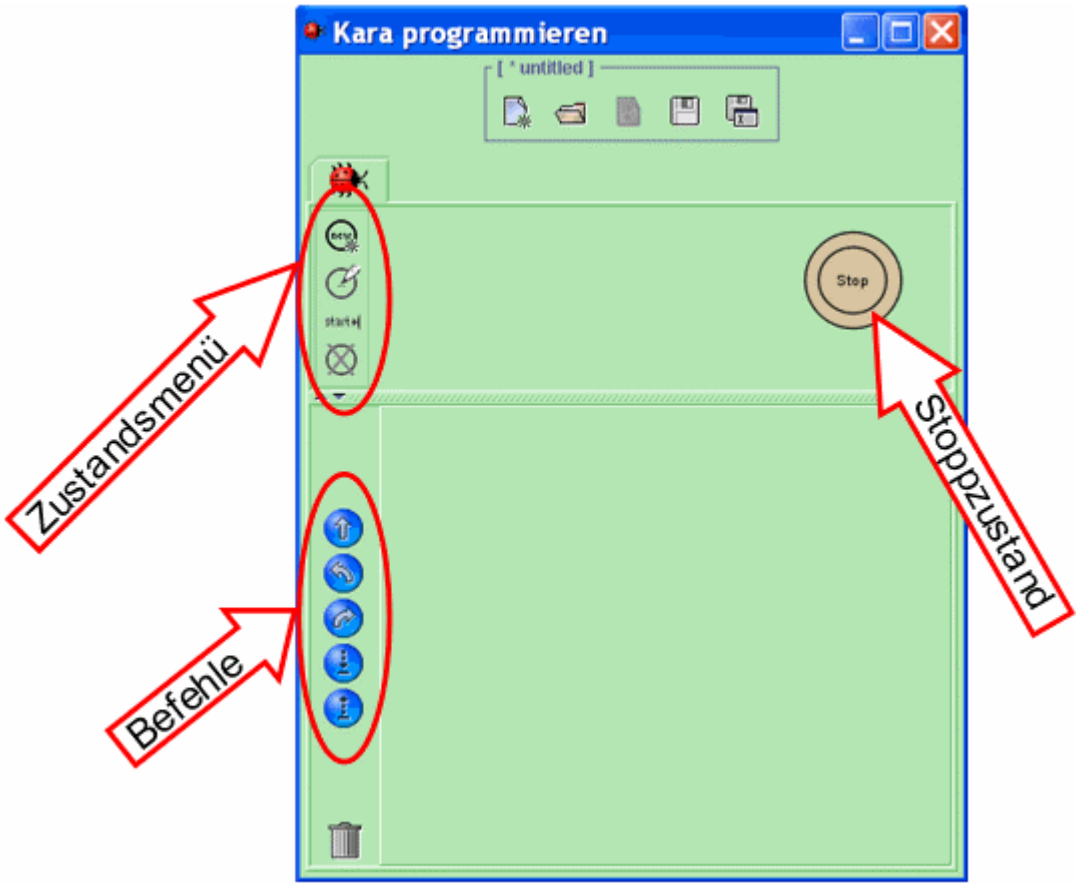


Abbildung 11: Der Stoppzustand ist standardmässig bereits eingetragen und der Automat kann dort einzeichnet werden. Links befinden sich alle Operationen, die für einen Automaten notwendig sind.

Zustände einfügen
Die folgenden Operationen stehen zur Verfügung, um einen Automaten zu zeichnen:

	Um einen **neuen Zustand** zu erhalten, klickt man mit der Maus auf dieses Symbol. Es geht ein Editor auf, in dem man den Zustand entwerfen kann.
	Einen Zustand kann man **verändern**, indem man ihn zuerst mit der Maus markiert und dann durch einen weiteren Mausklick auf dieses Symbol den Zustandseditor öffnet.
	Jeder Automat muss einen **Startzustand** haben. Markiere einen Zustand und klicke mit der Maus auf dieses Symbol, um die Auswahl als Startzustand zu definieren.
	Ein Mausklick auf dieses Symbol **löscht** den zuvor mit der Maus ausgewählten Zustand.
	Einen **Übergang** zwischen zwei Zuständen erhält man, wenn man im Ausgangszustand in den äusseren Ring klickt und dann die Maustaste gedrückt hält. Der Mauszeiger wird anschliessend in den Folgezustand gezogen und man lässt die Maustaste wieder los.

Zeichne einen neuen Zustand mit einem linken Mausklick auf das Symbol .

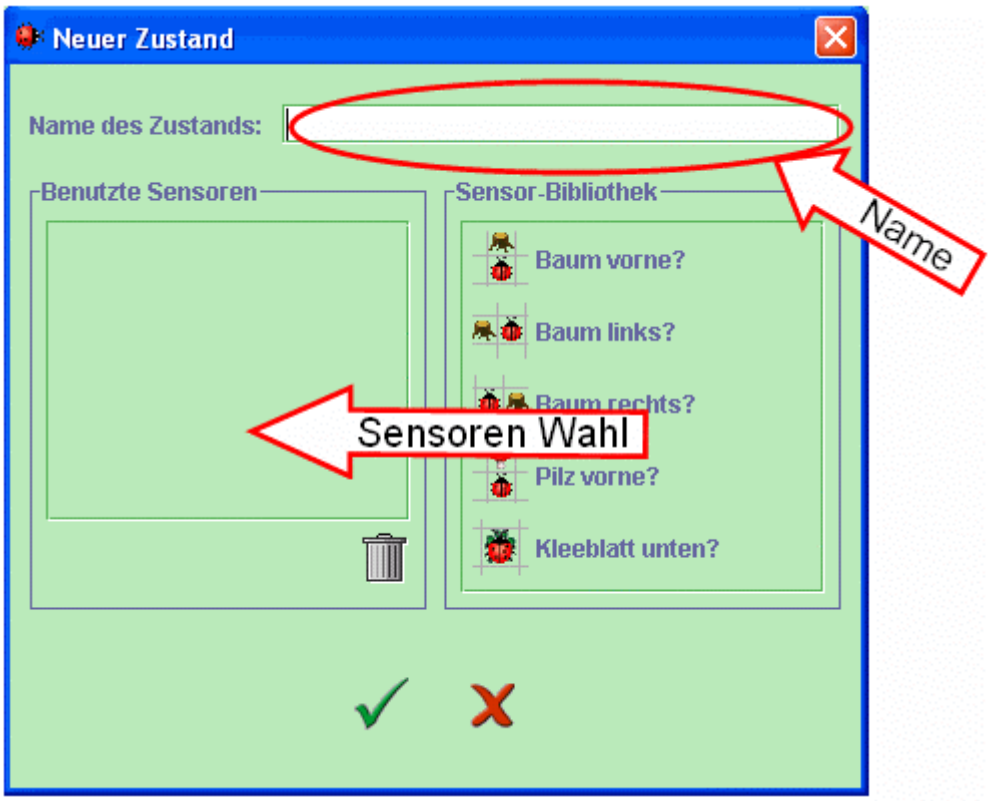


Abbildung 12: Zustandseditor. Die gewählten Sensoren erscheinen im Übergangseditor des Zustandes. Der Übergangseditor liegt im Programmeditor unterhalb des Feldes, in dem der Automat gezeichnet wird (siehe Abbildung 13).

- Gib dem neuen Zustand im Namensfeld einen eindeutigen Namen, z. B. „Suchen“.
- Wähle die benötigten Sensoren aus. Ziehe sie mit Drag-and-Drop in den Rahmen unter „Benutzte Sensoren“. Für das Blättersammeln muss man wissen, ob Kara vor einem Baum steht oder auf einem Kleeblatt. Abbildung 13 zeigt, wo man die gewählten Sensoren im Programmeditor findet.

Im Programmeditor ist jetzt ein zusätzlicher Zustand zu sehen. Es fehlen aber noch alle Übergänge, die erst im nächsten Schritt eingefügt werden.

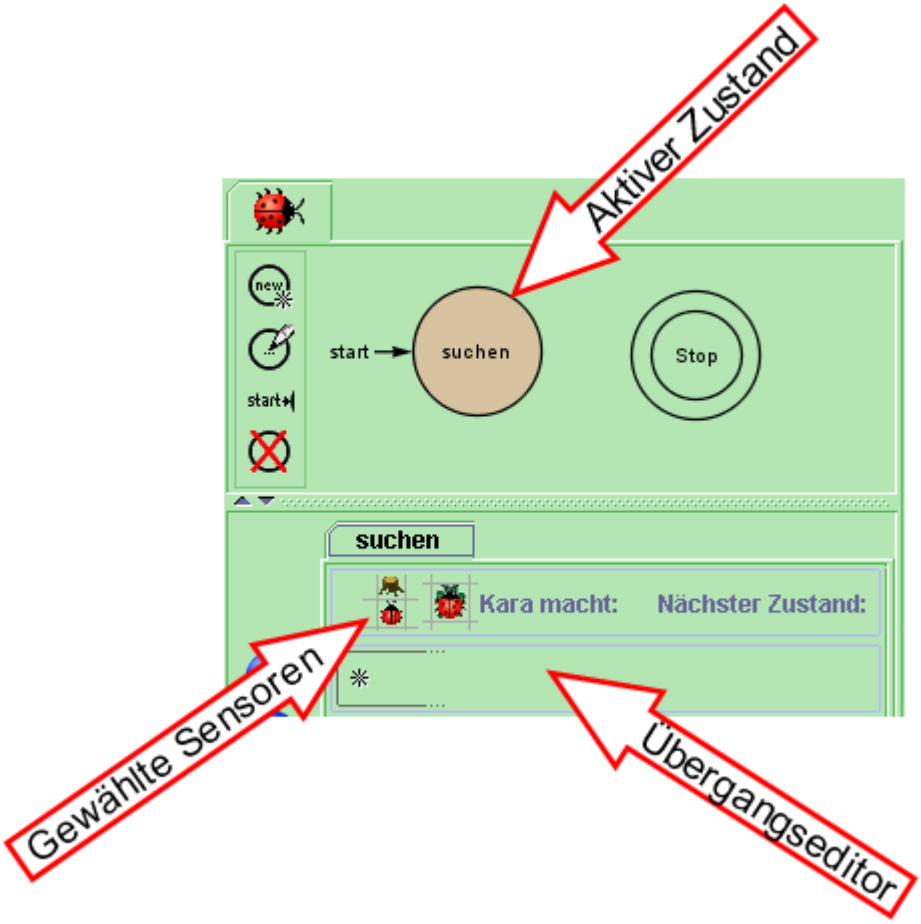


Abbildung 13: Diese Abbildung zeigt den Programmeditor mit dem Übergangseditor für den aktiven Zustand. Im Übergangseditor werden die gewählten Sensoren angezeigt.

- Klicke abwechselnd den Zustand „Suchen“ und den Stoppzustand an. Wie verändert sich der Übergangseditor?

Hinweis: Im Stoppzustand benötigt man keine Sensoren und Übergänge mehr, weil die Arbeit für Kara abgeschlossen ist.

Übergänge
Die Übergänge werden gezeichnet, indem man auf den äusseren Rand des Startzustands klickt und den Mauszeiger auf den Folgezustand zieht, ohne dabei die Maustaste loszulassen.

Hinweis: In der Ansicht des Automaten wird immer nur ein Übergang dargestellt. Im Übergangseditor werden alle Übergänge angezeigt.

➢ Übernimm alle Übergänge des Automaten aus der Wissenssicherung 3.1.

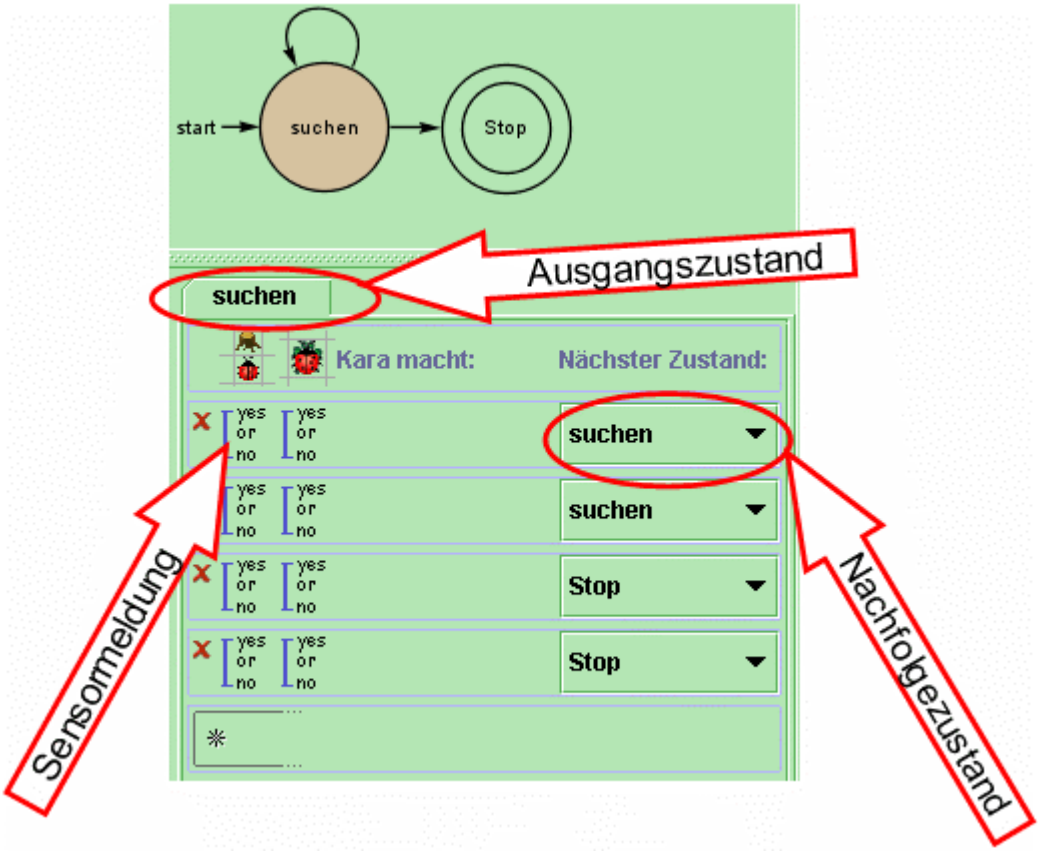


Abbildung 14: Im unteren Teil des Programmeditors ist der Übergangseditor zu sehen. Dort kann man die Inputs (Bedingungen, Meldungen der Sensoren) und Aktionen (Anweisungen an Kara) der einzelnen Übergänge festlegen. Eine Zeile entspricht einem Übergang. Der Übergang beginnt im Ausgangszustand und endet im Nachfolgezustand. Nachfolgezustände sind nicht für alle Übergänge identisch.

Jetzt fehlen nur noch die Angaben zum Input. Sie definieren, wann ein bestimmter Übergang benutzt wird und welche Aktionen darauf folgen.

Input und Aktionen
Soll Kara vor einem Baum stehen bleiben oder nicht, wenn er den Übergang der ersten Zeile nimmt? Und soll er im Zustand „Suchen“ bleiben? Zur Beschriftung der Übergänge mit Input und Aktionen benutzt man den Übergangseditor.

Mit einem Mausklick auf den blauen Balken kann die Belegung in yes oder no geändert werden. Ist die Meldung des Sensors nicht entscheidend, kann auch yes or no beibehalten werden. Anschliessend können die Befehle in der gewünschten Reihenfolge der Sensorbelegung rechts mit Drag-and-Drop eingefügt werden.

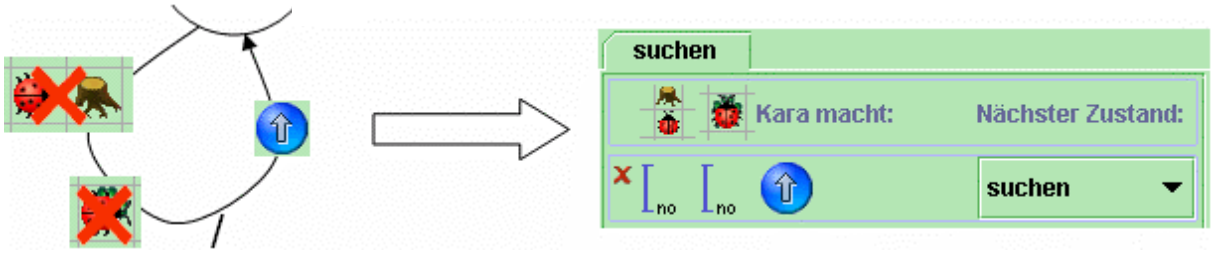


Abbildung 15: Kopie eines Übergangs vom Papier in das Kara-Programm

➢ Ergänze die Übergänge mit allen Sensorbelegungen und den Anweisungen für Kara.

Nun ist der Automat fertig und du kannst Kara damit laufen lassen. Wechsle dazu wieder in das Fenster mit Karas Welt. Am unteren Rand findet man die folgenden Menüpunkte:

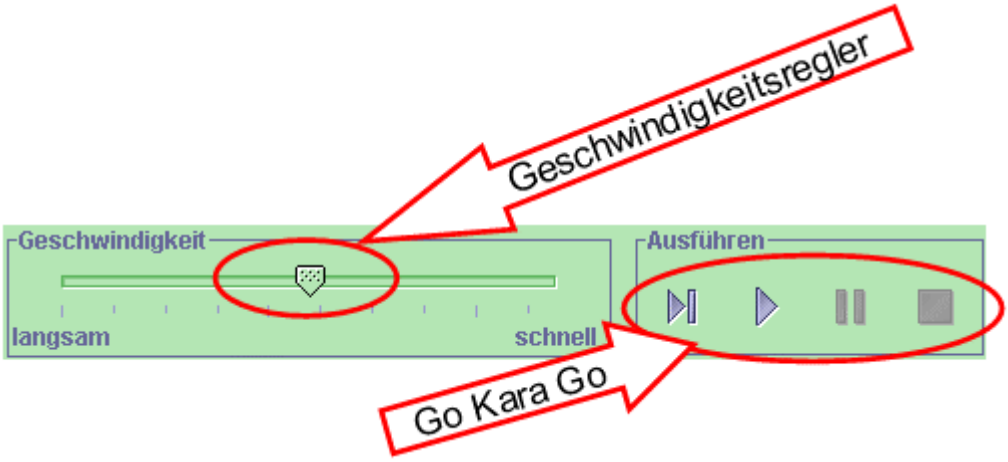


Abbildung 16: Mit diesem Menü kann Karas aktuelles Programm gestartet werden. Mit dem linken Symbol unter „Ausführen“ kann man sich Schritt für Schritt durch das Programm klicken. Das ist besonders bei der Fehlersuche hilfreich.

Lass Kara in den folgenden Welten laufen und teste, ob dein Programm richtig funktioniert.

Welt 1:

Welt 2:

Welt 3:

Speichern der Welten und Programme
Zum Speichern von Welten und Programmen benutzt man im entsprechenden Fenster am oberen Rand dieses Menü:

Abbildung 17: Menü zum Verwalten der Welten und Programme

Erstellt eine neue, leere Welt oder neue, leere Programme.

Öffnet eine bestehende Welt oder ein bestehendes Programm.

Speichert eine Welt oder ein Programm.

Speichert Welt oder Programm unter neuem Namen.

Das bestehende Programm soll gespeichert werden, bevor ein neues Programm geschrieben wird.

Klicke auf das Symbol für Speichern:
Es öffnet sich ein Fenster. Gib einen Namen ein und speichere das Programm.

Damit ist die Aufgabe 3.2 abgeschlossen.

Wissenssicherung 3.3

Übertrage den Automaten von Abbildung 9 ins Kara-Programm. Er leitet Kara geradeaus bis zum nächsten Baum und lässt ihn unterwegs alle Blätter aufnehmen.

Hinweis: Benutze einen neuen Programmeditor. .

Teste das Programm anschliessend in den folgenden Welten:

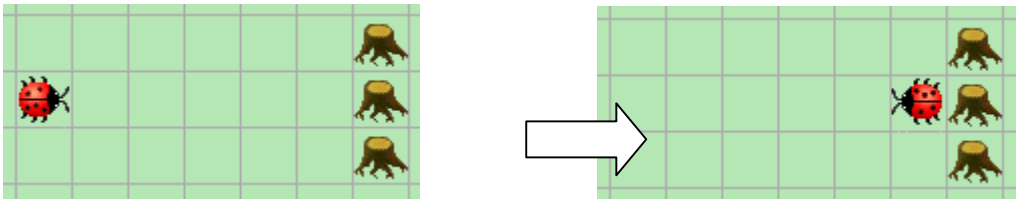
Welt 1:

Welt 2:

Welt 3:

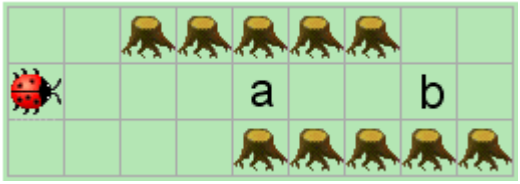
Aufgabe 3.4

Schreibe ein Programm, das Kara geradeaus bis zum nächsten Baum führt. Dort angekommen, soll er sich um 180° drehen.



Aufgabe 3.5 a

Kara sucht den Eingang eines geraden Tunnels (Feld a). Schreibe ein Programm, das ihn auf dem ersten Feld im Inneren anhalten lässt.
Aber Achtung: Einige Tunnel haben zunächst eine einseitige Wand – manche links, manche rechts.



Teste das Programm in unterschiedlichen Welten. Bist du sicher, dass Kara die Aufgabe in allen Welten mit einem Tunnel lösen kann?

Aufgabe 3.5 b

Kara will den Ausgang des Tunnels finden (Feld b). Dazu muss er zunächst den Tunnel durchqueren. Schreibe ein Programm, das ihn auf dem ersten Feld nach dem Tunnel anhalten lässt. Er soll nicht bis zum Ende der Reihe laufen!

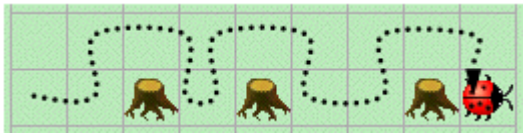
Hinweis: Die Lösung erfordert zusätzlich zum Stoppzustand zwei weitere Zustände.

Kapiteltest

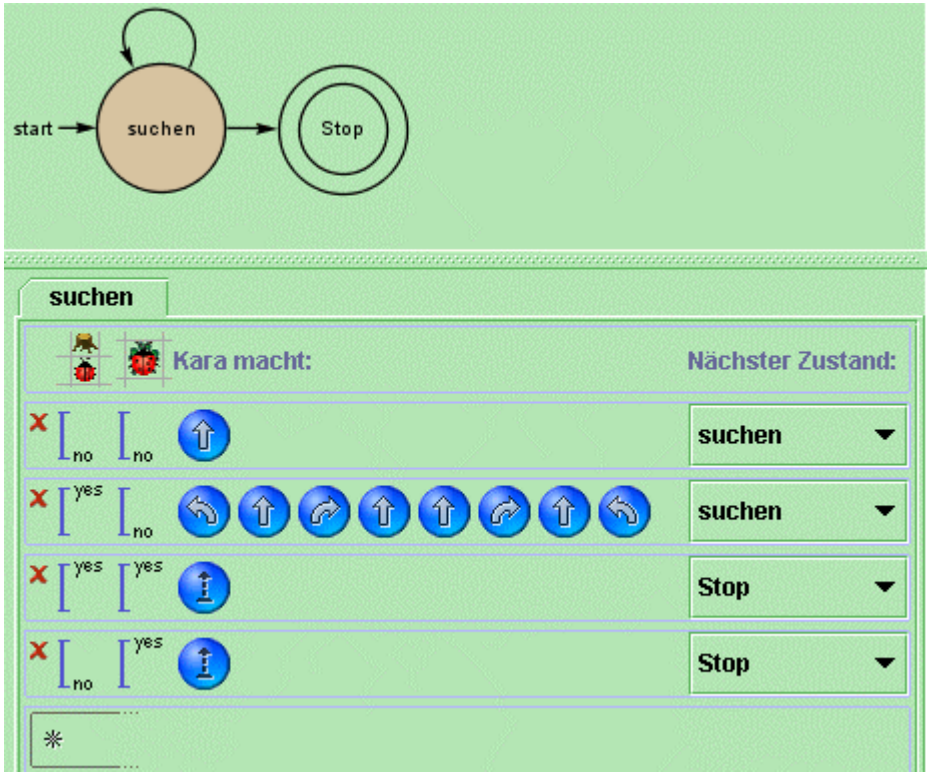
Wenn dir klar ist, wie die Automaten aus den Aufgaben funktionieren, kannst du dich bei der Lehrperson zum Kapiteltest melden.

Lösungen Kapitel 3

Wissenssicherung 3.1

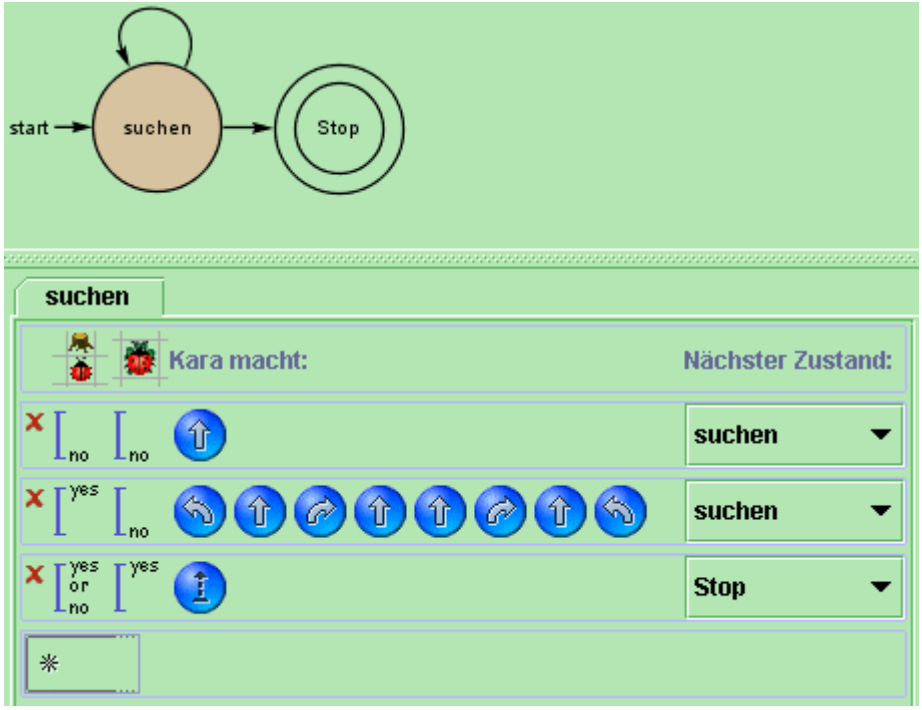


Aufgabe 3.2 a



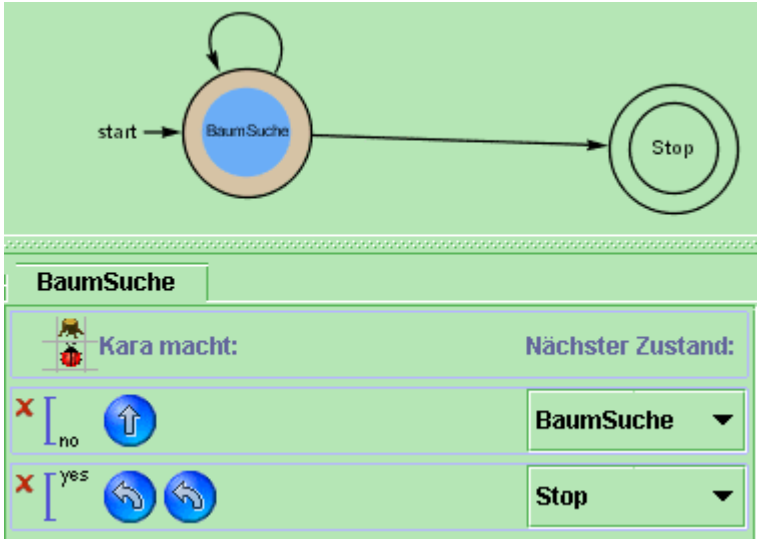
In den beiden letzten Übergängen gibt es keinen Unterschied zwischen den auszuführenden Aktionen. Wann immer Kara ein Kleeblatt findet, soll er es aufnehmen und die Arbeit abschliessen. Egal, ob er vor einem Baum steht oder nicht. Deshalb kann man die beiden letzten Übergänge so zusammenfassen, wie es in Lösung 3.2. b zu sehen ist.

Aufgabe 3.2 b

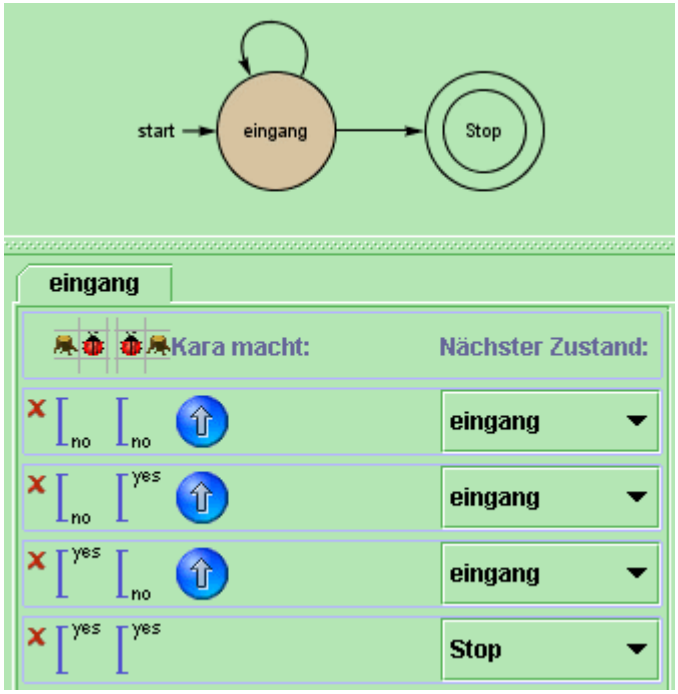


Wissenssicherung 3.3

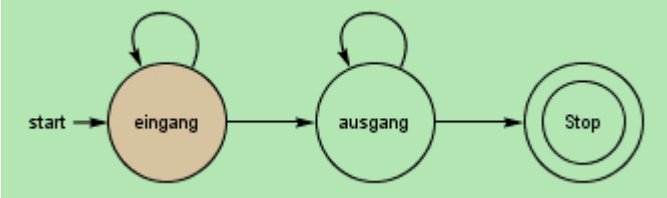
Aufgabe 3.4



Aufgabe 3.5 a



Aufgabe 3.5 b


start
eingang
ausgang
Stop

eingang
Kara macht:
Nächster Zustand:
no
no
eingang
no
yes
eingang
yes
no
eingang
yes
yes
ausgang

ausgang
Kara macht:
Nächster Zustand:
no
no
Stop
no
yes
Stop
yes
no
Stop
yes
yes
ausgang

4. Aufgabensammlung

Übersicht

Worum geht es?
Dieses Kapitel ist nicht mehr als Leitprogramm verfasst. Du weisst nun, wie Kara programmiert wird und kannst selbständig Aufgaben lösen.

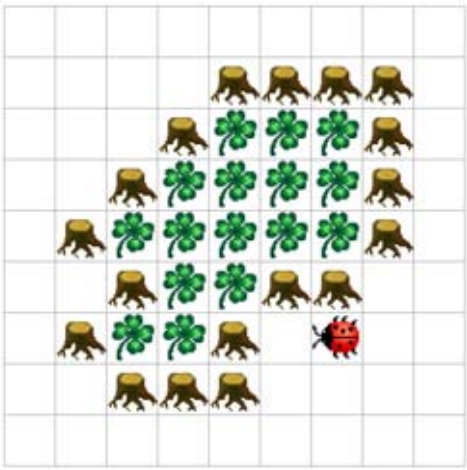
Was tust du?
Auf den folgenden Seiten gibt es eine Aufgabensammlung. Die Aufgaben sind mit einem Schwierigkeitsgrad zwischen 1 und 5 markiert, wobei 1 am einfachsten ist. Du kannst selbst einzelne Aufgaben auswählen und bearbeiten. Die Lösungen stehen ab Seite 46.

Lernziele

Nach dem Lösen einiger Aufgaben wirst du dich bald mit Automaten vertraut fühlen.

Kara, der Kleeblatt-Bewacher

Kara hat einen leckeren Vorrat an Kleeblättern gesammelt, der von Bäumen umringt ist. Damit niemand seine Kleeblätter stiehlt, bewacht Kara den Vorrat, indem er aussen um ihn herumgeht. So sieht der „Vorratswald“ aus:



Aufgabe, Schwierigkeitsgrad 2

Programmiere Kara so, dass er endlos um diesen „Wald“ läuft. Entscheide selbst, ob er links oder rechts herum läuft.

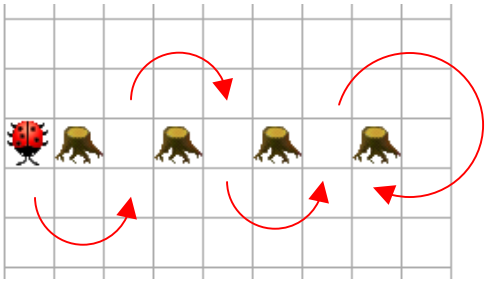
Hinweis: Ein Zustand reicht für die Lösung der Aufgabe aus. Versuche, mit möglichst wenigen Sensoren und Übergängen auszukommen.

Varianten

Falls dein Programm mehr als einen Zustand hat: Schreibe eine optimierte Fassung mit einem einzigen Zustand.
Erweitere das Programm so, dass Kara zuerst geradeaus bis zum nächsten Baum läuft und erst dann mit dem Patrouillieren beginnt. Zu Programmbeginn muss er in Richtung Baum stehen, wenn er geradeaus geht.

Kara, der Slalom-Skater

Inline-Skates faszinieren Kara. Er möchte üben, zwischen Bäumen Slalom zu fahren:



Der Parcours: Starten mit einer Linksdrehung, dann zwischen den Bäumen elegant in eine Rechtsdrehung übergehen. Zwischen den nächsten Bäumen wieder eine Linksdrehung, usw. Der Anfang des Slaloms ist eingezeichnet.

Aufgabe, Schwierigkeitsgrad 2

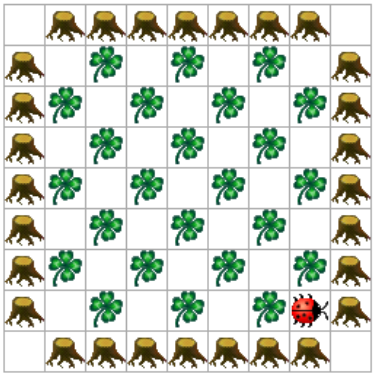
Scheibe ein Programm für Kara, mit dem er diesen Slalom fahren kann. Er kennt zu Beginn die Parcourslänge und die Anzahl der Bäume nicht. Es soll die Aufgabe sowohl bei horizontaler, als auch bei vertikaler Anordnung der Bäume meistern.

Zusatzfrage

Warum muss die Lösung mindestens zwei Zustände haben?

Kara, der Parkettleger

Kara möchte sich als Parkettleger betätigen und ein Schachbrett-Muster legen.



Aufgabe, Schwierigkeitsgrad 2

Programmiere Kara so, dass er ein solches Muster innerhalb der Baumreihen erzeugt. Es gibt zwei grundsätzlich verschiedene Möglichkeiten ein solches Muster zu bilden. Welche beiden Varianten sind denkbar?

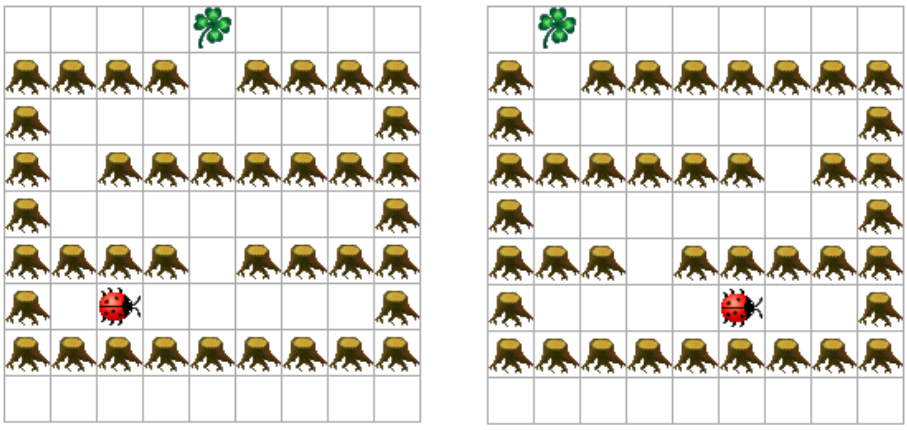
Hinweis: Wähle eine Weltgrösse, die das Programm vereinfacht. Abhängig vom Verfahren sind die Anforderungen an die Grösse unterschiedlich: Ist es für dein Programm wichtig, wenn Länge und Breite der Welt aus einer ungeraden oder geraden Anzahl Feldern bestehen? Oder hilft das Wissen, dass Länge und Breite gleich gross sind?

Zusatzfrage

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

Einfaches Labyrinth – Kleeblatt-Suche

Kara hat sich in einem Labyrinth verlaufen und möchte zum Ausgang, denn dort wartet ein leckeres Kleeblatt auf ihn. Zwei Beispiele von Labyrinth-Welten:



Mit Ausnahme der unteren Reihe hat jede horizontale Baumreihe genau einen Ausgang, den Kara stets finden muss. Nach dem letzten Ausgang wartet das Kleeblatt auf ihn.

Aufgabe, Schwierigkeitsgrad 3

Programmiere Kara so, dass er das Kleeblatt findet. Das Programm endet, wenn er es aufgenommen hat. Die Startposition und Startrichtung in der untersten Labyrinthzeile ist frei wählbar, muss aber im Programm berücksichtigt werden.

Hinweis: Teste das Programm an verschiedenen Labyrinthen. Sonst kann es sein, dass es nur mit einem bestimmten Labyrinth funktioniert. Speziallösungen sind beim Programmieren immer ein sehr grosses Risiko, das leicht zu Problemen beim Einsatz eines Programms führen kann.

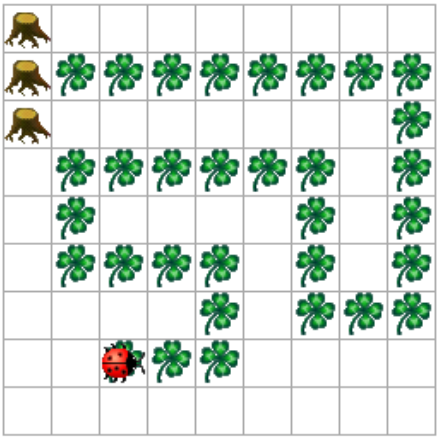
Zusatzfragen

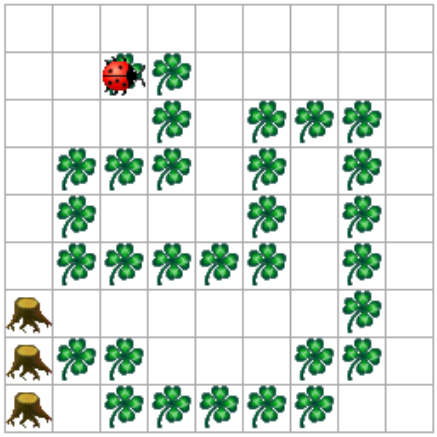
Wie verhält sich dein Programm, wenn es mehr als einen Ausgang in einer Baumreihe gibt? Funktioniert es dann noch? Warum oder warum nicht?

Falls dein Programm drei Zustände hat, dann versuche sie auf zwei zu reduzieren. Reicht auch ein einziger Zustand aus?

PacMan – oder wie verfolgt man eine Spur?

Kara möchte das etwas veralterte Computerspiel „PacMan“ in einer vereinfachten Version spielen und eine Kette von Kleeblättern „auffressen“. Er weiss, dass diese Spur nie entlang von Bäumen führt, aber sie an einem Baum endet.
So sehen „Spur-Welten“ aus:





Aufgabe, Schwierigkeitsgrad 3
Programmiere Kara so, dass er die Spur von Kleeblättern „auffrisst“. Das Programm endet, wenn Kara ein Kleeblatt aufnimmt, das vor einem Baum liegt. Du kannst selbst bestimmen, ob er auf oder vor einem Kleeblatt starten soll.

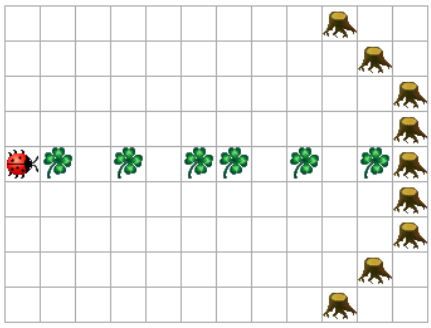
Hinweis I: Die Lösung ist verzwickt. Skizziere zuerst auf Papier, in welchen Situationen sich Kara befinden kann und was er dann tun muss.

Hinweis II: Teste dein Programm an mehreren verschiedenen Spuren. Sonst kann es sein, dass das Programm nur in genau einen Spezialfall funktioniert.

Kara und die Ausserirdischen

Muster in der Welt (1)

Kara glaubt an Ausserirdische. Er ist überzeugt, dass bestimmte Kleeblattmuster in seiner Welt auf ihre bevorstehende Ankunft hinweisen. Er weiss nur nicht, auf welches er warten soll und denkt, es ist eines von drei Mustern.
Gegeben ist diese erste Welt:



Aufgabe, Schwierigkeitsgrad 3
Kara soll in seiner Zeile prüfen, ob dieses Muster vorkommt. Programmiere ihn entsprechend.

Er soll das Muster suchen, bis er zu einem Baum kommt und dann die Suche beenden. Findet er die Anordnung, soll er seine Freude zeigen, indem er sich nach links dreht und eine Zeile höher geht. Dann endet das Programm. Findet er das Muster nicht, geht er enttäuscht nach rechts eine Zeile tiefer.

Hinweis I: Man braucht drei Zustände für ein Muster, das drei Felder umfasst. Überlege zuerst mittels Papier, wie du diese Zustände als Gedächtnis einsetzen kannst.

Hinweis II: Überprüfe dein Programm in mehreren verschiedenen Welten.
Teste die Funktion,

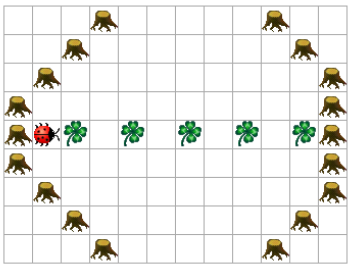
- wenn das Muster vorkommt.
- wenn es nicht vorkommt.
- wenn es nur teilweise vorkommt und durch einem Baum unterbrochen wird.

Das Programm muss in allen Fällen die korrekte Antwort geben.

Variante
Teste das Programm an einem selbst entworfenen Muster. Je komplexer das Muster, desto mehr Zustände brauchst du. Warum ist das so?

Muster in der Welt (2)

Kara meint, auch ein anderes Muster kann ihm einen Hinweis geben. Dieses Muster muss in einer Zeile beliebig oft stehen. Die Welt sieht wie folgt aus:



Aufgabe, Schwierigkeitsgrad 3

Programmiere Kara so, dass er in seiner Zeile prüft, ob in dieses Muster wiederholt vorkommt.

Die Zeile soll, wie in der Abbildung, abwechselnd Felder mit und ohne Kleeblatt aufweisen. Sie muss ohne Kleeblatt beginnen und mit einem Kleeblatt enden, denn dann ist das Muster mehrfach vorhanden. In der Abbildung kommt es fünfmal vor.

Erkennt Kara das richtige Muster in der Zeile, dann freut er sich. Er dreht sich nach links und geht eine Zeile höher. Findet er es nicht, geht er enttäuscht nach rechts in die darunter liegende Zeile.

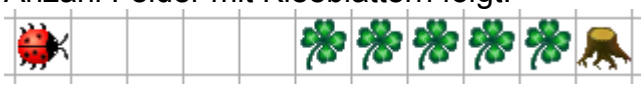
Hinweis: Teste dein Programm in mehreren verschiedenen Weltenbildern. Prüfe auch die Funktion, wenn die Bildzeile der beschriebenen Anordnung nicht genügt. Das Programm muss in jedem Fall die korrekte Antwort geben.

Variante

Probiere andere aneinandergereihte Muster aus. Entwirf selbst ein Muster und versuche es zu finden.

Muster in der Welt (3)

Kara meint, noch ein weiteres Muster könnte ihm einen Hinweis geben. Das Muster besteht aus einer beliebigen Anzahl Felder ohne Kleeblätter, denen dann die gleiche Anzahl Felder mit Kleeblättern folgt:



Aufgabe, Schwierigkeitsgrad 3- 4

Kara hat allerdings Zweifel, ob seine Fähigkeiten ausreichen, ein solches Muster zu erkennen. Er weiss zu Beginn nicht, wie lang die Zeile ist. Vielleicht lebt er ja in einer viel längeren Welt? Er möchte die Welt nicht verändern, denn sonst verärgert er vielleicht die Ausserirdischen.

Glaubst du, dass Karas Fähigkeiten ausreichen, um zu prüfen, ob eine Zeile dieses Muster enthält?

a) Überlege dir zunächst auf Papier eine Lösung für den Fall, dass die Zeile genau vier Felder lang ist. Kein Problem, oder?
b) Wie lautet die Lösung, wenn die Zeile <u>maximal</u> vier Felder hat?
   Tipp: Du brauchst vier Zustände.
c) Schreibe anschliessend das Programm für Aufgabe b).
   (Schwierigkeitsgrad: 3-4)
d) Versuche daraus abzuleiten, worin das Problem mit ganz langen Zeilen liegt.

Auch hier gilt: Teste dein Programm an möglichst vielen Mustern!

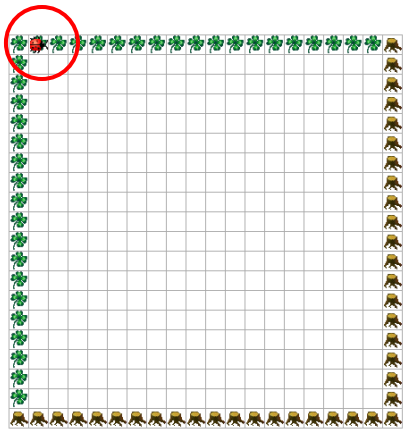
Wenn sich die Ausserirdischen mit einem derartigen Muster anmelden: Pech gehabt!

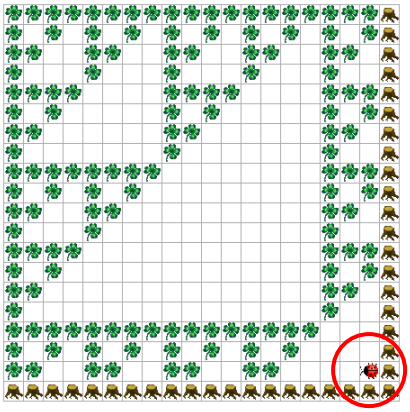
Freiwillige Zusatzaufgabe, Schwierigkeitsgrad 5

Erlaubt man Kara, die Welt zu verändern, kann er die Aufgabe lösen. Er könnte z. B. die Zeilen über oder unter der Zeile mit dem gesuchten Muster als Gedächtnis („Speicher“) benutzen. Aber Achtung: Die Lösung benötigt fünf Zustände und ist ziemlich aufwendig.

Kara, der Mathematiker

Kara möchte das berühmte Pascal-Dreieck aus Kleeblättern zeichnen:



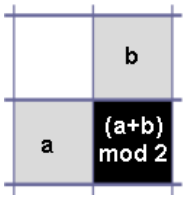


Aufgabe, Schwierigkeitsgrad 4

Programmiere 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.

Um es zu vereinfachen, wird das Dreieck in die linke, obere Ecke der Welt gekippt.

Die Abbildungen oben zeigen die Ausgangslage für diese Aufgabe und das von Kara zu zeichnende Pascal-Dreieck. Wie die inneren Felder berechnet werden müssen, verdeutlicht das folgende Schema:



<u>Hinweis:</u> Die Zustände sind Karas Gedächtnis. Diese Tatsache muss man bei dieser Aufgabe ausnutzen. Mit weniger als drei Zuständen kommt man nicht aus.

Lösungen der Aufgabensammlung

Lösung: Kara, der Kleeblatt-Bewacher

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

Lösung: Kara, der Slalom-Skater

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

Lösung der 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. Zwischen den nächsten Bäumen wieder eine Linksdrehung“.

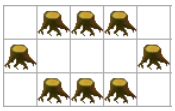
Die beiden Zustände sind Karas Gedächtnis. Ohne sie kann er sich nicht merken, ob er zwischen zwei Bäumen eine Links- oder Rechtsdrehung vollziehen muss.

Lösung: Kara, der Parkettleger

a) Kara läuft horizontal oder vertikal.

Die Welt muss horizontal 2+1+n × 2 für (n ≤ 1) Felder haben: zwei für die Bäume, eines für die Startposition und jeweils zwei für jeden Zustandsübergang. So kommt die Lösung mit nur zwei Zuständen aus. Vertikal muss mindestens eine Zeile zwischen den Bäumen frei sein.

Weltgrösse: Demnach sieht die kleinstmögliche Welt, 5 × 3 Felder gross, so aus:



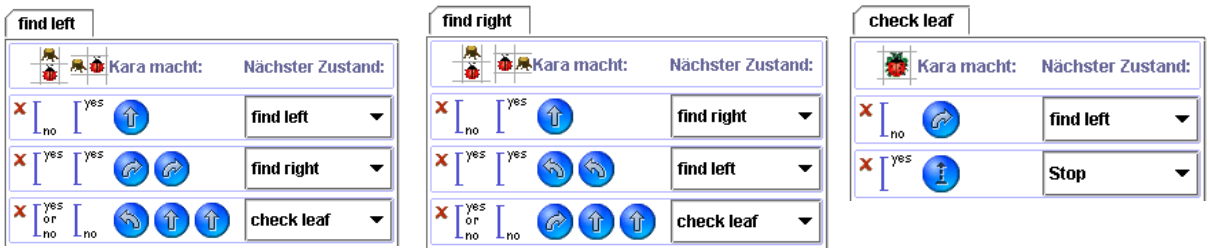
b) Kara läuft diagonal hin und her.

Start: In der linken, unteren Ecke der Welt, nach rechts schauend, im Zustand „no leaves“.

Weltgrösse: Die kleinste Welt ist 4 × 4 Felder gross.

Lösung: Einfaches Labyrinth – Kleeblatt-Suche

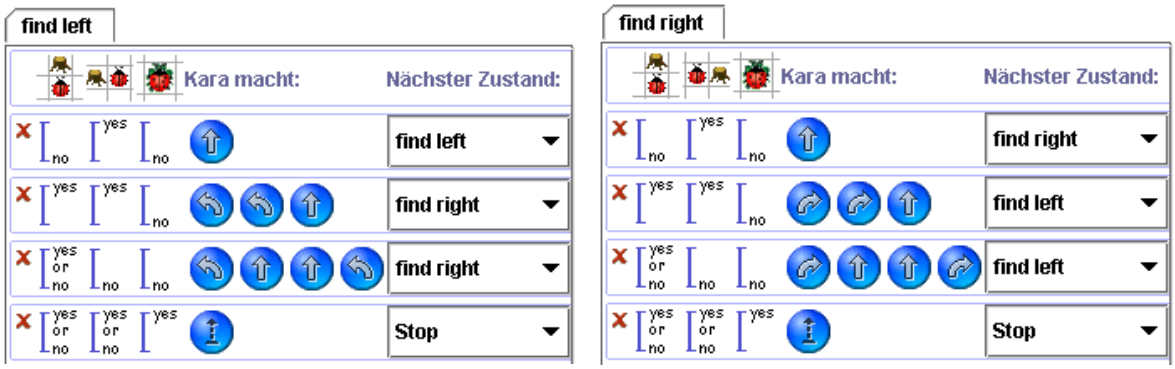
a) Lösung mit drei Zuständen



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 die entsprechende Richtung gedreht werden.

b) Optimierte Lösung mit zwei Zuständen



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

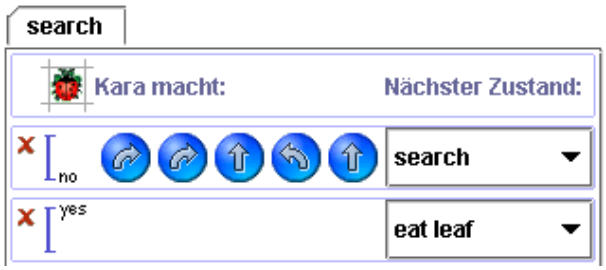
Anmerkung: Man könnte noch in einem der beiden Zustände den Sensor `leaf on ground?` einsparen.

Zusatzfragen

Lücken in den horizontalen Baumreihen stören das Programm nicht. Jedoch darf die oberste Baumreihe nur genau vor dem Kleeblatt einen Durchgang haben. Lücken stören den Programmablauf auch nicht, wenn sie an das Programm angepasst sind. Mit einem einzigen Zustand funktioniert das nicht, denn dann kann Kara bei einer Lücke in der Baumreihe zu seiner Linken oder Rechten nicht wissen, ob er daran schon vorbeigekommen ist und ob er hindurch soll.
Würde man hingegen die Aufgabenstellung vereinfachen, indem man die Durchgänge immer am Ende einer Baumreihe einrichtet, dann reichte ein Zustand aus.

Lösung: PacMan – oder wie verfolge ich eine Spur?

a) Lösung mit zwei Zuständen



Start: In `eat leaf`, wenn Kara auf Kleeblatt startet. In `search`, wenn er davor steht.

b) Lösung mit einem Zustand

„Optimiert“ ist hier nur die Anzahl der Zustände, denn einer wurde gespart. Die Zustände lassen sich problemlos kombinieren, weil sie ganz andere Situationen abdecken. Die Anzahl der Befehle hat sich nicht geändert.

Eine weitere, schwierigere Aufgabe ist es, Spuren zu „fressen“, die an ganzen Baumreihen entlang führen. Vielleicht nimmt ja jemand die Herausforderung an?

Lösungen: Kara und die Ausserirdischen
Muster in der Welt (1)

Mittels der Zustände 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` hat er ein leeres Feld und eines mit Kleeblatt hinter sich. Deshalb hat er das Muster erkannt, wenn er in `01` kein Kleeblatt auf dem Feld findet.

Muster in der Welt (2)

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.

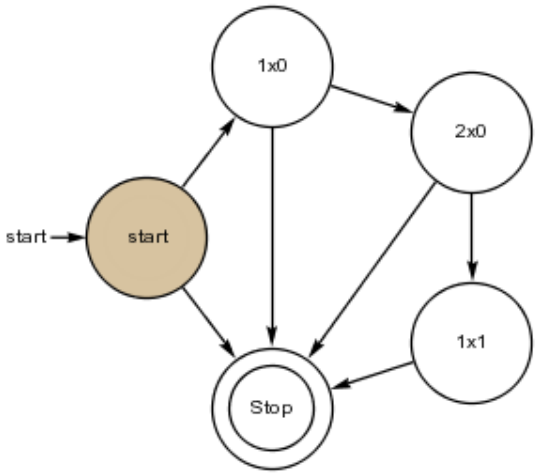
Muster in der Welt (3)

Zu a)

Die Lösung reduziert sich in diesem Fall auf „prüfe Muster“, leicht abgewandelt von der ersten Suchaufgabe.

Zu b und c)

Mit einer oberen Schranke gibt es eine relativ elegante, quasi symmetrische Lösung. Zur Verdeutlichung hier nur die Übersicht:



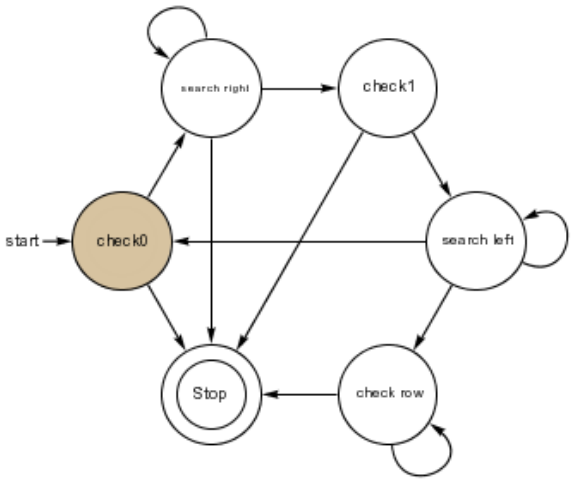
Die Idee ist einfach. Die Zustände merken sich, welchen Teil des Musters sie schon erkannt haben. Sie wissen auch, welcher Teil noch vor ihnen liegen muss. Diese Lösung kann für beliebige obere Schranken leicht verallgemeinert werden.

Zu d)

Es ist keine praktikable Lösung. Da Kara kein Gedächtnis in Form eines internen Zählers (Variable) hat, müssen Zustände die Rolle des Gedächtnisses übernehmen. Für sechs Felder kann ein Automat mit sechs Zuständen erstellt werden. Es gilt: für *n* Felder 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“ für $n \geq 1$ ist kein regulärer Ausdruck, 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, dann kann er die Aufgabe lösen. Die Welt dient ihm dabei als Gedächtnis, als „Speicher“. Denn wäre die Welt unendlich lang, könnte Kara jede Berechnung der Welt anstellen, weil er dann eine Turingmaschine realisieren könnte. Hier geht es aber etwas einfacher. Der Einfachheit halber hier die Lösung [pattern0n1n.kara] mit der Übersicht:



Start: Kara muss auf dem leeren Feld starten, das vor dem ersten Feld mit Kleeblatt liegt. Er muss Richtung Kleeblatt schauen.

In der Zeile oberhalb der Musterzeile markiert Kara die Felder, die dem Muster entsprechen. Zu Beginn erwartet er ein leeres Feld. Weil er es hat, markiert er das Feld oberhalb mit einem Kleeblatt. Auf der Suche nach dem ersten, unmarkierten Feld läuft er nach rechts. Unterhalb davon müsste ein Kleeblatt sein. Wenn er es findet, markiert er das Feld oberhalb. Dann läuft nach links auf der Suche nach einem unmarkierten Feld, prüft das Feld unter diesem Feld, usw.

Die Zeile erfüllt das Kriterium an das Muster genau dann, wenn bei Programmende die obere Zeile leer ist.

Lösung: Kara, der Mathematiker

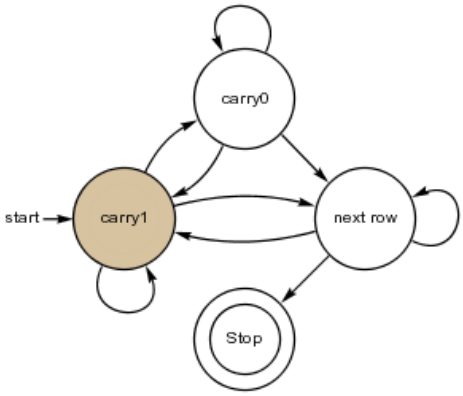
Die Berechnung der Werte im Pascal-Dreieck, also das Platzieren von Kleeblättern durch Kara, ist im Prinzip einfach: Kara berechnet das Pascal-Dreieck zeilenweise von links nach rechts. Erreicht Kara bei seiner Berechnung das Ende einer Zeile, das durch einen Baum markiert ist, läuft er im Zustand `next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

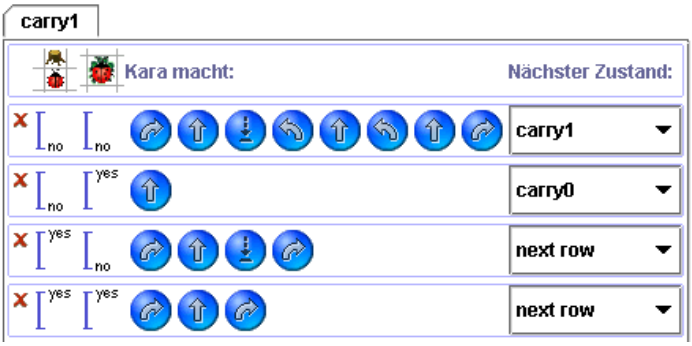
Wie berechnet Kara die Werte in den einzelnen Zellen des Pascal-Dreiecks?
Kara kann über den Sensor „auf Kleeblatt?“ feststellen, ob er auf einem Feld mit Kleeblatt steht, also einer ungeraden Zahl, siehe Abbildung oben links.
Zusätzlich muss er wissen, ob sich auf dem Feld unmittelbar rechts hinter ihm ein Kleeblatt befindet, in den Abbildungen Feld a und die eingekreisten Felder.

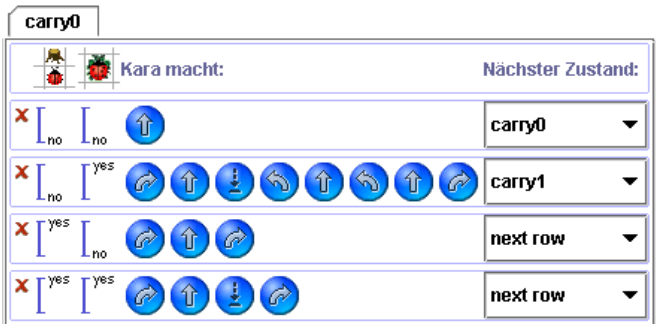
Diese Information wird durch einen der beiden folgenden Zustände gespeichert:

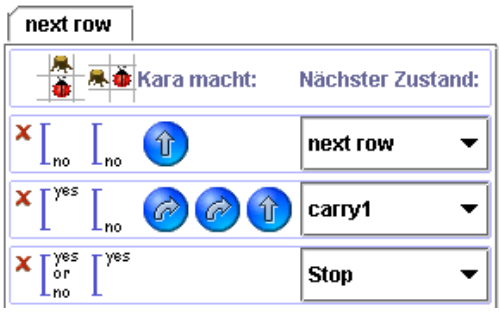
- `carry0:` Auf dem Feld hinten rechts von Kara befindet sich kein Kleeblatt, eine gerade Zahl.
- `carry1:` Auf dem Feld hinten rechts von Kara befindet sich ein Kleeblatt, eine ungerade Zahl.

Befindet sich Kara im Zustand `carry0` und steht auf einem Feld ohne Kleeblatt, so bleibt das Feld rechts von Kara leer und er macht einen Schritt vorwärts. Falls er auf einem Feld mit einem Kleeblatt steht, so ist dem Feld rechts von Kara eine ungerade Zahl zugeordnet. Kara legt dort ein Kleeblatt ab und macht dann einen Schritt vorwärts. Entsprechend verhält sich Kara im Zustand `carry1`.
Folgende Abbildungen zeigen das ganze Programm.









Teil 2

Kapiteltests und Lösungen für die Lehrperson

	Kapiteltest 1

Aufgabe 1
Beschreibe in eigenen Worten, was du unter Sensoren verstehst. Wofür werden sie eingesetzt?

Aufgabe 2
Gegeben ist die folgende Welt:

Kara führt die diese Bewegungen aus:

Fülle die nachstehende Tabelle aus, indem du ein Kreuz bei negativen und einen Haken für positive Antworten vergibst.

Schritte	0	1	2	3	4	5	6	7	8
Baum vorne?									
Baum links?									
Baum rechts?									
Pilz vorne?									
Kleeblatt unten?									

Lösungen Kapiteltest 1

Aufgabe 1
Über die Sensoren kann Kara Informationen über seine Umgebung erfassen. Er kann zum Beispiel feststellen, ob direkt vor ihm dem Feld ein Baum steht oder nicht.

Man benötigt diese Informationen, um Kara generell programmieren können. Er kann dadurch seine Aufgaben in unterschiedlichen, aber gleich strukturierten Welten lösen. Ein Beispiel: Kara soll alle Kleeblätter bis zu einem Baum aufnehmen können, unabhängig von ihrer Menge und Anordnung

Aufgabe 2

Schritte	0	1	2	3	4	5	6	7	8
Baum vorne?	✔	×	×	×	×	×	×	×	×
Baum links?	✔	✔	×	×	×	×	×	×	✔
Baum rechts?	×	×	✔	×	×	✔	×	×	✔
Pilz vorne?	×	×	✔	✔	✔	✔	✔	✔	✔
Kleeblatt unten?	×	×	✔	×	×	✔	✔	✔	×

Kapiteltest 2

Aufgabe 1
Erkläre mit eigenen Worten die Begriffe: Zustand, Übergang, Input, Aktionen und Startzustand.

Aufgabe 2
Notiere drei Wenn-dann-Aussagen zum Videorecorder-Automat aus Aufgabe 2.3.

Lösungen Kapiteltest 2

Aufgabe 1
Zustand: Ein Automat befindet sich immer in einem Zustand und wartet auf neuen Input. Die Zustände stellen das Gedächtnis des Automaten dar. Er kann sich zum Beispiel merken, wie viel Geld ein Käufer bereits in den Getränkeautomaten eingeworfen hat.

Input: Ein Automat reagiert nur auf neuen Input von aussen. Oft erfolgt dieser Input über Sensoren.

Aktionen: Auf konkrete Inputs reagiert der Automat mit Aktionen. Zum Beispiel soll der Getränkeautomat die Aktion „Ausgabe einer Cola“ ausführen, wenn der Käufer auf den Cola-Knopf drückt. Es dürfen nur Aktionen ausgeführt werden, die dem Übergang aus dem aktuellen Zustand zugeordnet sind. Also muss der Getränkeautomat im Zustand „3.- Franken-Kredit“ sein, damit eine Cola ausgegeben wird.

Übergang: Übergänge sind die Verbindungen zwischen den Zuständen. Auf den Übergängen sind Input und Aktionen vermerkt. Damit ein konkreter Übergang angewendet wird, muss der notwendige Input erfolgt sein. Der Automat befindet sich anschliessend im Zustand am Ende des Übergangs.

Startzustand: Nach dem Start befindet sich ein Automat in diesem Zustand.

Aufgabe 2
Verschiedene Antworten sind möglich, hier drei Beispiele:

Wenn sich der Automat im Zustand „warten“ befindet und den Input „abspielen“ erhält, dann soll er in den Zustand „abspielen“ wechseln und das Video wird abgespielt.

Wenn sich der Automat im Zustand „abspielen“ befindet und den Input „spulen“ erhält, dann soll er in den Zustand „spulen“ wechseln und das Video mit Bild spulen.

Wenn sich der Automat im Zustand „warten“ befindet und den Input „spulen“ erhält, dann soll er in den Zustand „spulen“ wechseln und das Video ohne Bild spulen.

Kapiteltest 3

Aufgabe 1
Zeichne deinen Automaten aus Aufgabe 3.5 b auf ein Blatt Papier oder rufe die Lehrperson direkt an deinen Computer. Erkläre, wie dein Automat Schritt für Schritt die folgende Aufgabe löst:

Aufgabe 2
Warum braucht man für Aufgabe 3.5 b mehr als einen Zustand?

Lösungen Kapiteltest 3

Aufgabe 1
Hier soll der Schüler/ die Schülerin den eigenen Lösungsweg erklären und anhand des gezeigten Beispiels noch einmal Schritt für Schritt überdenken.

Aufgabe 2
Ein Zustand reicht nicht aus, weil Kara auf eine einseitige Wand flexibel reagieren muss, denn mal befindet er sich vor dem Tunnel, mal hat er ihn hinter sich. Kara muss sich also merken, ob er noch vor oder schon im Tunnel ist. Dafür braucht er zwei Zustände.

Anhang

Material für die Lernenden

Damit die Schülerinnen und Schüler Kara programmieren können, muss das Softwarepaket „Kara“ auf den Computern installiert werden.
Auf EducETH finden Sie eine genaue Anleitung:
http://www.educeth.ch/lehrpersonen/informatik/unterrichtsmaterialien_inf/programmieren/kara_umgebung/index

Auf EducETH finden Sie auch viele zusätzliche Anregungen und Aufgabestellungen im Zusammenhang mit Kara:
http://www.educeth.ch/lehrpersonen/informatik/unterrichtsmaterialien_inf/programmieren/unterrichtsmaterial_kara/index

Mehr Hintergrundsinformationen enthält das Buch: „Programmieren mit Kara. Ein spielerischer Zugang zur Informatik“ von Reichert u.a.

Zusätzliche Materialien sind nicht notwendig.

Quellenangaben

Alle Aufgaben, die Kara in diesem Leitprogramm gestellt werden und die Beispiele von alltäglichen Automaten stammen von Raimond Reichert. Zu finden sind sie in den unten aufgelisteten Publikationen.

Raimond Reichert: Ein Leitfaden zu Kara.
http://www.educeth.ch/lehrpersonen/informatik/unterrichtsmaterialien_inf/programmieren/unterrichtsmaterial_kara/leitfaden_kara.pdf

Raimond Reichert: Einführungsvortrag in die Programmierung von Kara.
http://www.educeth.ch/lehrpersonen/informatik/unterrichtsmaterialien_inf/programmieren/unterrichtsmaterial_kara/einfuehrungsvortrag_kara.pdf

R. Reichert, J. Nievergelt, W. Hartmann: Programmieren mit Kara.
Springer, Berlin 2003. ISBN 3540403620