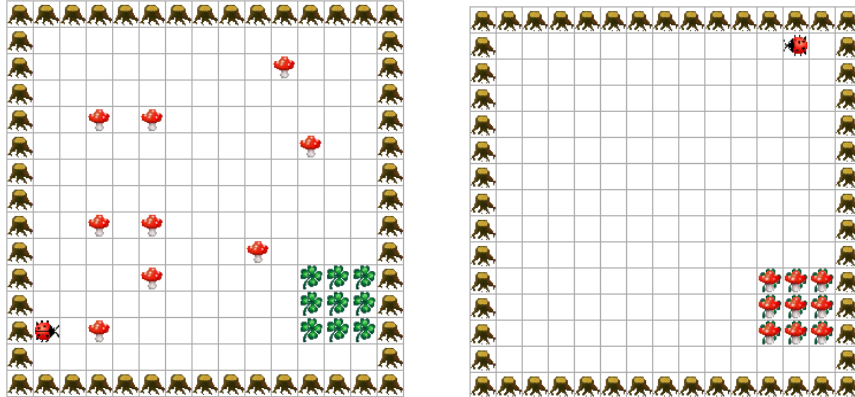


LÖSUNG: SOKOBAN!

Kara möchte eine vereinfachte Version des Japanischen Strategiespiels Sokoban spielen [sokoban.world; sokoban_done.world]:



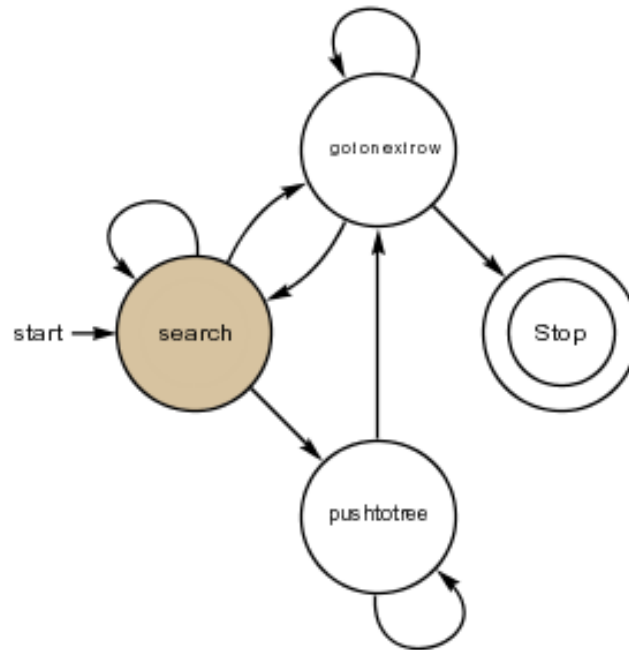
Die Aufgabe (Schwierigkeitsgrad: 5 von 5)

Programmieren Sie Kara so, dass er die Pilze auf die Kleeblätter schiebt! Lassen Sie Kara wie in der Abbildung gezeigt links unten starten. Damit das Programm nicht noch komplexer wird, können Sie die Pilze so verteilen, dass Kara sich problemlos zwischen den Pilzen bewegen kann. So ist es wohl hilfreich, immer eine Zeile leer zu lassen zwischen Zeilen, in denen Pilze stehen.

Hinweis: Wenn Sie ein wenig „tricky“ programmieren, so kann die Aufgabe mit 3 Zuständen gelöst werden. Weniger dürfte nur schwer möglich sein.

Lösung

Die Lösung ist ein wenig „tricky“. Die Zustände `search` und `gotonextrow` suchen nach dem nächsten Pilz, der in das Ziel geschoben werden soll. Dieses „Schieben“ erledigt der Zustand `pushtotree`. Die Idee in diesem Zustand ist, dass Kara immer auf dem Feld vor dem Pilz ist (in der Richtung, in der geschoben wird). So kann Kara erkennen, wenn er vor einem Baum steht, und sich entsprechend neu positionieren. `pushtotree` merkt auch, wenn Kara auf den Kleeblättern angelangt ist.



pushtotree

Kara macht:	Nächster Zustand
 yes or no no no no	pushtotree ▼
 yes yes no no	pushtotree ▼
 yes yes no yes	gotonextrow ▼
 yes no yes yes	gotonextrow ▼
 yes no yes no	pushtotree ▼

Figure 1 illustrates two examples of state transition diagrams for the game of Go. The left diagram shows a state labeled 'gotonextrow' with a board position where Black is to move. The right diagram shows a state labeled 'search' with a board position where White is to move. Both diagrams show transitions to 'Nächster Zustand' (Next State) based on player actions (Kara macht:).