

# Grundlagen der Künstlichen Intelligenz

Prof. Dr. M. Helmert  
Dr. M. Wehrle, S. Sievers  
Frühjahrssemester 2015

Universität Basel  
Fachbereich Informatik

## Übungsblatt 3

Abgabe: 20. März 2015

### Aufgabe 3.1 (4 Punkte)

Das *Pancake-Problem* (*Pfannkuchen-Problem*) der Grösse  $n$  ist folgendermassen definiert: Es gibt  $n$  Pfannkuchen unterschiedlicher Länge  $1, \dots, n$ , die auf einem Stapel liegen. Mit der Aktion  $\text{turn}_i$ ,  $i \in \{1, \dots, n\}$ , werden die obersten  $i$  Pfannkuchen gewendet (so wendet beispielsweise  $\text{turn}_2$  die obersten beiden Pfannkuchen). Das Ziel ist es, die Pfannkuchen mit möglichst wenigen Schritten so zu wenden, dass alle Pfannkuchen auf dem Stapel der Grösse nach geordnet sind (der längste Pfannkuchen soll unten liegen, der kürzeste oben). Die folgende Abbildung zeigt ein Beispiel.



Formalisieren Sie den Zustandsraum des Pancake-Problems aus dem obigen Beispiel. Spezifizieren Sie hierzu die Menge der Zustände, der Aktionen mit entsprechenden Aktionskosten und der Transitionen. Geben Sie weiterhin den Anfangszustand und die Menge der Zielzustände an.

### Aufgabe 3.2 (0+6+2 Punkte)

Bei dieser Aufgabe handelt es sich um eine Programmieraufgabe. Wir erwarten, dass Sie Ihre Implementierung selbständig, das heisst ohne Anwendung von fremdem Code z.B. aus dem Internet erstellen. Uns ist bewusst, dass Programmieraufgaben aufwendiger sind als die üblichen theoretischen Aufgaben und helfen bei technischen Schwierigkeiten und Verständnisproblemen gerne weiter. Bitte wenden Sie sich dazu *mit genügend zeitlichem Abstand zum Abgabetermin* an Cédric Geissmann, Silvan Sievers oder Martin Wehrle.

- Auf der Vorlesungsseite befinden sich Beispielimplementierungen in Java für einen trivialen Zustandsraum sowie für den Zustandsraum des Blocks-World-Problems aus der Vorlesung. Die Zustände des trivialen Zustandsraums bestehen aus den Integern 1,2 und 3, die man mit entsprechenden Aktionen inkrementieren und dekrementieren kann. Beide Beispielimplementierungen implementieren die bereitgestellten Interfaces `State` und `Action` sowie das Black-Box-Interface `StateSpace`, das Sie in der Vorlesung kennengelernt haben. Machen Sie sich mit diesen Implementierungen vertraut. Testen Sie die Implementierungen mit der Klasse `StateSpaceTest`, die vom initialen Zustand aus auf zufällige Art Nachfolgezustände berechnet.
- Implementieren Sie nach dem Vorbild von (a) den Zustandsraum für das Pancake-Problem. Implementieren Sie hierzu die bereitgestellten Interfaces `State`, `Action` und `StateSpace`.
- Implementieren Sie nach dem Vorbild der Blocks-World-Implementierung von (a) eine Methode `buildFromCmdline`, die ein Input-Pancake parst (als Datei oder direkt von der Konsole) und einen entsprechenden Zustandsraum erzeugt. Ein Input-Pancake kann in einer Zeile kodiert werden, die aus einer Permutation von Zahlen besteht, die den Anfangszustand des Pancakes beschreibt. Testen Sie ihre Implementierung, indem Sie die bereitgestellte Klasse `StateSpaceTest` entsprechend erweitern.

Reichen Sie bitte den Code inklusive Hinweisen zu Kompilierung und Aufruf bei courses ein (sofern nicht selbsterklärend).

*Die Übungsblätter dürfen in Gruppen von zwei Studierenden bearbeitet werden. Bitte schreiben Sie beide Namen auf Ihre Lösung.*