Fachbereich Informatik
Wesley Terpstra
Dr.-Ing. Ilia Petrov
Prof. Alejandro Buchmann, Ph. D.
Sommersemester 2009

TECHNISCHE UNIVERSITÄT DARMSTADT

Grundlagen der Informatik II

2. Praktikumsaufgabe

Nun da ein geeignetes Zeitfenster ausgewählt worden ist, muss eine Einsatzgruppe zusammen gestellt werden, die den Auftrag erledigt. Die Dienststelle musste leider feststellen, dass die Auswahl eines geeigneten Teams nicht so einfach ist wie ursprünglich angenommen. Als zuständiger IT-Experte sind Sie ausgewählt worden, das Problem zu lösen.

Um den Erfolg der Mission zu sichern, muss das Team über eine Reihe von Fertigkeiten verfügen. Glücklicherweise gibt es für jede Fertigkeit (genau) zwei Agenten, die entsprechend ausgebildet sind. Zum Beispiel: Sowohl Fred als auch Hugo können einen Atomsprengkopf entschärfen. Unglücklicherweise kommen manche Agenten nicht miteinander aus. Zum Beispiel: Amy und Fred haben gerade eine unangenehme Scheidung hinter sich. Außerdem benötigen einige Agenten die Fertigkeiten eines anderen Agenten, um ihre Aufgabe zu erfüllen. Zum Beispiel: Fred kann einen Sprengkopf entschärfen, aber er benötigt Lara, um das nukleare Material zu sichern.

Dank Ihrem ausgefeilten Training haben Sie bereits erkannt, dass diese Anforderungen immer ein Paar von Agenten betreffen, zum Beispiel “Fred OR Hugo”. Leicht zu überprüfen ist, dass “!Amy OR !Fred” potentielle Streitigkeiten vermeiden hilft. “!Fred OR Lara” regelt den dritten Fall.

Aus Sicherheits- und Datenschutzgründen hat die Dienststelle sich dazu entschlossen, Ihnen keine spezifischen Missionsparameter mitzuteilen. Stattdessen sollen Sie ein Programm schreiben, dass eine mögliche Lösung für jeden Parametersatz finden kann bzw. ermitteln kann, wenn keine Lösung möglich ist.

Aus Ihrer bisherigen Erfahrung vermuten Sie, dass die Aufgabe gelöst werden kann, in dem das Problem graphentheoretisch formuliert wird. Eine topologische Sortierung der starken Zusammenhangskomponenten sollte in der Tat ausreichen! Im Folgenden finden Sie einige Notizen, welche die Vorgehensweise beschreiben. Die Schritte 1-3 sind in den Vorlesungsfolien erklärt.

Sie sollten wie immer Ihre Undercoverkontakte in der GDI2-Vorlesung nutzen, um die Aufgabe zu diskutieren. Aus Sicherheitsgründen müssen Sie darauf achten, dass Sie die Lösung vollständig verstanden und selbst implementiert haben.

Zu implementieren: StrikeForce.java für die Teamzusammenstellung!

Die Lösung ist bis zum **24. Mai 2009 23:59** einzureichen!

Achtung: Diese Anweisungen sind mit besonderer Sorgfalt zu behandeln und werden sich automatisch selbst zerstören!

Schritt 0: Laden Sie die Sourcecode-Vorgaben von `https://www.dvs.tu-darmstadt.de/teaching/inf2/2009-de/Problem2.zip` herunter.

Schritt 1: Implementieren Sie Graphumkehrung.

Schritt 2a: Implementieren Sie Tiefensuche (DFS).

Schritt 2b: Implementieren Sie Topologisches Sortieren mit DFS.

Hinweis: Die umgekehrte Ergebnisreihenfolge der DFS ist eine topologische Sortierung.

Hinweis: Benutzen Sie auf gar keinen Fall .contains() bei einer Liste (List). Es wird die asymptotische Eigenschaft Ihrer Implementierung ändern und führt wahrscheinlich dazu, dass Sie die 30 Sekunden Beschränkung überschreiten. Benutzen Sie .contains() stattdessen bei einer Menge (Set) oder Abbildung (Map).

Schritt 3: Implementieren Sie die Identifizierung von starken Zusammenhangskomponenten.

Hinweis: Führen Sie DFS auf dem *umgekehrten* Graphen aus. Wahlen Sie Startknoten des DFS in topologischer Ordnung des ursprünglichen Graphen. Jeder gefundene Baum enthält die Knoten einer starken Zusammenhangskomponente.

Hinweis: Dies liefert automatisch die gesuchten Komponenten in topologischer Ordnung.

Schritt 4: Konvertieren Sie die Missionsparameter (Constraints) in einen Implikationsgraphen.

Hinweis: Jeder Agent erscheint als zwei Knoten: “Bob” und “!Bob”.

Hinweis: Jeder Constraint erzeugt zwei gerichtete Kanten.

Schritt 5: Finden Sie eine Lösung, die den Constraints genügt.

Hinweis: Untersuchen Sie die Komponenten in umgekehrter topologischer Reihenfolge.

Hinweis: Setzen Sie die Teamzugehörigkeit gemäß des Eintrags in der Komponente.

Hinweis: Wenn eine Komponente sowohl “!Bob” als auch “Bob” enthält, gibt es keine Lösung.

Hinweis: Entscheiden Sie, welcher Knoten Priorität hat, falls sich ein Agent in zwei verschiedenen Komponenten befindet.

Zusätzliche Informationen über dieses Problem (2SAT genannt) finden Sie unter:

a) `http://en.wikipedia.org/wiki/2SAT`
b) `http://www.cs.rpi.edu/~goldberg/2sat.pdf`
c) `http://www.cl.cam.ac.uk/~mr10/qbf2slds.pdf`
d) Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. *Introduction to Algorithms*, 2nd edition. Kapitel 22.3, 22.4, 22.5.