

Proseminar Theoretische Informatik
Elmar Eder, Universität Salzburg
Blatt 2, Aufgaben für 17.3.2021

Aufgabe 1 Sei A die Aussage "2 ist ungerade" und B die Aussage "3 ist eine Primzahl" (Beispiel aus der Vorlesung). Geben Sie für jede der Aussagen $\neg A$, $A \wedge B$, $A \vee B$, $A \Rightarrow B$ und $A \Leftrightarrow B$ an, ob sie wahr oder falsch ist!

Aufgabe 2 In Prolog wird ein Prädikat durch seinen Namen gefolgt von einem Schrägstrich und seiner Stelligkeit bezeichnet. So gibt es zum Beispiel in SWI-Prolog (swipl) und in GNU-Prolog (gprolog) ein zweistelliges Prädikat `member` und ein dreistelliges Prädikat `append`. Diese werden bezeichnet mit `member/2` bzw. `append/3`. Machen Sie sich zunächst mit diesen beiden Prolog-Prädikaten vertraut, indem Sie unter anderen die Anfragen

```
?- member(X, [a,b,c]).  
?- member(a,L).  
?- append([a,b],[c,d,e],L).  
?- append(L1,L2,[a,b,c,d]).  
?- append(L1,L1,L).  
?- append(L1,L2,L).
```

eingeben und indem Sie in SWI-Prolog z.B. `?- help(member/2).` aufrufen bzw. in GNU-Prolog im Manual nachschlagen! Implementieren Sie dann diese beiden Prädikate als selbst definierte Prolog-Prädikate `element/2` bzw. `konkatenation/3` ohne Verwendung der eingebauten Prädikate!

Aufgabe 3 Bei dem Spiel *Nim* spielen zwei Spieler gegeneinander. Es liegen mehrere Haufen von Steinen auf einem Tisch. Die Spieler ziehen abwechselnd. Jeder Zug besteht darin, aus einem Haufen ein oder mehr Steine zu entfernen. Wer als erster nicht mehr ziehen kann, weil alle Haufen leer sind, hat verloren.

Geben Sie einen Algorithmus an, der für einen Spieler, der gerade am Zug ist, feststellen kann, ob es für ihn möglich ist, auch gegen einen guten Gegner zu gewinnen! Außerdem soll der Algorithmus in diesem Fall einen Zug ausgeben, der dem Spieler den Gewinn garantiert. Implementieren Sie Ihren Algorithmus in Prolog! Bauen Sie damit ein Nim-Programm, das gegen einen menschlichen Gegner optimal spielen kann! Testen Sie Ihr Programm für einige Spielstellungen mit wenigen Steinen!

Hinweis: Wenn G ein Prolog-Ziel ist, dann ist $\backslash+G$ die *negation as failure* von G . Das Ziel $\backslash+G$ gelingt genau dann, wenn das Ziel G fehlschlägt. Bei einem Aufruf von $\backslash+G$ versucht Prolog zunächst G zu beweisen. Wenn dies gelingt, schlägt der Aufruf von $\backslash+G$ fehl. Andernfalls gelingt der Aufruf von $\backslash+G$, ohne dass dabei irgendwelche Variablen ersetzt werden.

Aufgabe 4 Gibt es für das Spiel Schach einen Algorithmus, der ähnlich wie in der vorigen Aufgabe, in endlicher Zeit feststellen kann, ob es bei einer gegebenen Schachstellung

einen Zug gibt, der dem ziehenden Spieler den Gewinn garantiert? Man sagt dann, es sei *entscheidbar*, ob es einen solchen Zug gibt. Begründen Sie Ihre Antwort!

Aufgabe 5 Geben Sie für $\neg((A \vee \neg B) \Leftrightarrow \neg(A \wedge C))$ die Wahrheitstafel an!

Aufgabe 6 Geben eine Wahrheitstafel für alle zweistelligen aussagenlogischen Verknüpfungen an!

Aufgabe 7 Unter welchen Bedingungen ist die Aussageform $\forall x \exists y ((x + y)^2 = z)$ über dem Individuenbereich der reellen Zahlen wahr?

Aufgabe 8 Ein Häftling wird zum Tode verurteilt. Ihm wird mitgeteilt, dass er nächste Woche gegen Mittag hingerichtet wird, aber den Tag der Hinrichtung nicht vorher wissen wird. Der Häftling überlegt sich: Am Sonntag kann ich nicht hingerichtet werden, da ich ja sonst am Sonntag vormittag schon den Hinrichtungstag wissen würde. Am Samstag kann daher auch nicht der Hinrichtungstag sein, da ich ja den Sonntag bereits ausgeschlossen habe und ich daher am Samstag vormittag wissen würde, dass der Hinrichtungstag der Samstag ist. Mit der gleichen Argumentation kann ich den Freitag ausschließen, dann den Donnerstag, u.s.w. bis zum Montag. Ich kann also ganz beruhigt sein. Ich werde gar nicht hingerichtet. Plötzlich kommt am Mittwoch mittag der Henker und holt den Häftling ab, und der Häftling hat tatsächlich vorher nicht gewusst, an welchem Tag er hingerichtet wird. Woran liegt es, dass hier ein Widerspruch auftaucht, und gibt es einen solchen Widerspruch auch in der modernen Logik?