Zusammenfassung

Den Kontext dieser Arbeit bilden wissensbasierte Systeme, die einen Benutzer interaktiv und unter Realzeitbedingungen sowohl bei der Lösung komplexer Entscheidungs- und Planungsprobleme, als auch bei der Planausführung unterstützen. Dabei werden Probleme betrachtet, die die Interpretation und Planung zeitlicher Abläufe notwendig machen. In der Künstlichen Intelligenz sind zahlreiche logikbasierte Verfahren zur Lösung dieser Probleme entwickelt worden. Der Nachteil dieser Verfahren ist, daß sie in der Regel zu langsam sind, um Realzeitbedingungen zu erfüllen. Ziel dieser Arbeit ist die Entwicklung von Optimierungsverfahren, die die Effizienz dieser Verfahren steigern.

Wir verwenden den Repräsentationsformalismus, der üblicherweise in der logischen Programmierung benutzt wird, d.h. definite und normale Programme. In diesem Rahmen wird eine neue Klasse von Programmen eingeführt, sogenannte Kettenprogramme, mit denen sich komplexe zeitliche Sachverhalte sehr einfach darstellen lassen ohne auf temporallogische Konstrukte zurückgreifen zu müssen.

Um die Effizienz dieser Programme zu steigern, d.h. um die Zeit, die für ihre Auswertung durch Vorwärtsinferenzen benötigt wird zu reduzieren, werden mehrere Optimierungsverfahren entwickelt. Die erste Methode transformiert logische Programme mit dem Ziel redundante Berechnungen bei Vorwärtsinferenzen zu vermeiden. Für definite Kettenprogramme werden zwei Vorwärtsinferenzprozeduren entwickelt, die die syntaktischen Merkmale dieser Programme ausnutzen. Während die erste Methode auf das Originalprogramm angewendet wird, nutzt die zweite Methode zusätzlich Eigenschaften des transformierten Programms aus. Die Analyse der Methoden liefert ein Kriterium zur Auswahl einer der beiden Inferenzprozeduren. Die Tauglichkeit dieses Kriteriums und die Erfüllung der Realzeitbedingung werden durch eine experimentelle Auswertung bestätigt.

Kettenprogramme sind ein Spezialfall stratifizierter Programme. Die Transformationsmethode ist auch auf diese weit aus größere Programmklasse anwendbar. Für diesen allgemeineren Fall entwickeln wir Dekompositionsmethoden, die das transformierte Programm in Komponenten zerlegen. Wir weisen empirisch nach, daß die sequentielle Auswertung der Komponenten um ein vielfaches schneller ist als die Auswertung des Originalprogramms.

Alle bisher aufgezeigten Optimierungsmethoden steigern die Effizienz ohne die Semantik der Programme zu verändern. Im Gegensatz dazu ist der Zweck der letzten Optimierungsmethode die Veränderung der Semantik, um die Qualität des durch die logischen Programme repräsentierten Wissen zu verbessern und so bessere Problemlösungen zu ermöglichen. Dieses Optimierungsverfahren ist eine Lernmethode, die sich besonders zum Lernen von Wissen über zeitliche Prozesse eignet.

Alle Methoden werden anhand der Daten einer Fallstudie aus dem Bereich Robotik ausgewertet. Die Ergebnisse belegen die praktische Relevanz der Verfahren. Ihre Anwendbarkeit geht jedoch, insbesondere aufgrund der in dieser Arbeit theoretisch nachgewiesenen generellen Eigenschaften, weit über die Fallstudie hinaus.

Abstract

The context of this thesis are knowledge-based systems that support a user with solving complex decision and planning problems as well as executing plans. The systems are required to solve the above tasks interactively and under real-time constraints. In addition, it is assumed that the problems to be solved require the interpretation and planning of temporal processes. In the field of Artificial Intelligence, numerous logic-based methods were developed to solve these tasks. The disadvantage of these methods is that they are generally too slow to meet real-time constraints. In this thesis, we present optimization methods that improve the efficiency of the above methods such that real-time requirements are met.

We assume the representation formalism that is commonly used in the field of Logic Programming (i.e., definite and normal programs). Within this framework we introduce a new class of programs (general chain programs) that allow us to represent complex relations between time intervals in a very simple way and without special operators typically used in temporal logics.

In order to improve the efficiency of logic programs (i.e., the time needed by a forward inference procedure to evaluate them), we develop several optimization methods. The first one transforms a logic program in order to avoid redundant computations that occur during the evaluation of the program via forward inferences. For definite general chain programs, we develop two optimized forward inference procedures which exploit their syntactical features. Whereas the first method is applied to the original program, the second method is applied to the transformed program. The analysis of the methods yields a criterion for choosing one of the methods in different situations. An empirical evaluation confirms the appropriateness of the criterion and the satisfaction of real-time constraints for both methods.

General chain programs are a specific class of stratified programs. We prove that the transformation method is applicable to this larger class of programs. For this more general case, we develop decomposition methods that split up a program into components. An empirical evaluation for definite programs shows that the sequential evaluation of the transformed and decomposed program is always faster than the evaluation of the original program.

All aforementioned methods improve the efficiency without changing the semantics of a logic program. In contrast, the last optimization method changes the semantics in order to improve the quality of the knowledge represented by the logic program. In this way, better problem solving solutions will be found by the system that uses the knowledge. The learning method is suitable for learning knowledge about temporal processes.

All optimization methods are evaluated with the data of a robotics case study. The results confirm the practical relevance of the methods. However, their applicability goes far beyond the case study. This is implied by the general properties of the methods that are theoretically proven in this thesis.