Abstract. In automated verification of protocols, one source of complications is that channels may have unbounded capacity, in which case a naive model of the protocol is no longer finite state. Symbolic techniques have therefore been developed for representing the contents of unbounded channels. In this paper, we survey some of these techniques and apply them to a simple leader election protocol. We consider protocols with entities modeled as finite state machines which communicate by sending messages from a finite alphabet over unbounded channels; this is a framework for which many techniques have been developed. We also consider a more general model in which messages may belong to an unbounded domain of values which may be compared according to a total ordering relation: the motivation is to study protocols with timestamps or priorities. We show how techniques from the previous setting can be extended to this more general model, but also show that reachability quickly becomes undecidable if channels preserve the ordering between messages.

1 Introduction

Protocol verification has, since 25 years, been a driving application for the development of automated verification techniques. State space exploration techniques were developed in this context [36,33], and it is one of the important application areas for current model checking tools such as SPIN [25] and UPPAAL [27].

Protocol verification involves the construction of a model of a protocol, which can be subject to analysis, e.g., by a model checking tool. The model should abstract from less relevant details of the protocol in order to facilitate the analysis. Typically, a protocol model consists of a number of processes, which communicate over channels of some kind. In many cases, the channels are a significant source of problems for the analysis. If communication channels are large or unbounded, naive model-checking cannot be performed exhaustively. When modeling a protocol, one must therefore be careful to model the channels in a way which suits subsequent analysis. Many model checkers do not support unbounded channels.

In general, unbounded channels have infinitely many states, and must therefore be represented symbolically in automated verification. Different representations have been proposed for different types of channels. In this paper, we
will survey some of these symbolic techniques and illustrate them on a simple example.

First, we consider a model of protocols with entities modeled as finite state machines which communicate by sending messages from a finite alphabet over unbounded channels. If channels are unordered, this model can be represented by Petri Nets. FIFO ordered channels has been considered rather extensively in protocol verification [8,23,15,24,30,31,34]. Since this model can simulate Turing machines [12], we will devote most attention to a weaker model in which the FIFO channels can spontaneously lose messages at any time. We will illustrate two types of symbolic representations and their use in checking whether some states are reachable. The first type represents \textit{upward closed} sets. It is useful in backward reachability analysis, since it gives a lower bound on which messages must be in each channel for a certain set of states to be reachable. This representation was used in [5] to decide the reachability problem, and in [5,21] to decide the termination problem. The second type (called Simple Regular Expressions in [3]), dually represents \textit{downward closed} sets. This representation is useful in forward reachability analysis, since it gives an upper bound on which messages may be in the channel (we can never be sure whether a message is in the channel, since it can be lost). Analogous symbolic representations of upward closed sets have also been used for unordered channels and related models like Petri nets and broadcast systems [20,4,22,19,17].

We also consider how the techniques illustrated in the first part may be extended to models with an infinite set of message values, on which a limited set operations can be performed. We focus on the case where the domain of message values is equipped with a total ordering, which may be used in guards. The motivation is to study protocols with timestamps or priorities. We present a negative result showing that with lossy FIFO channels one can simulate perfect FIFO channels already if tests for equality and inequality are allowed on messages. For unordered channels, however, the backward reachability analysis presented in the first part of the paper can be used as a decision procedure. Using ideas from our earlier work [6,7] we can show that the techniques carry over to handle messages on which an ordering relation is defined.

This small survey is organized as follows. In the next section, we define protocols with a finite set of messages and present a simple leader election protocol as a running example. Forward and backward reachability analysis with associated symbolic representations are presented in Section 3. In Section 4, we extend the model to an infinite set of messages, and present an undecidability result for lossy FIFO channels, and an extension to messages with an ordering relation. Section 6 contains discussion and conclusion.

2 Protocols with a Finite Set of Messages

In this section, we present our first protocol model: finite-state processes which communicate over unbounded channels using a finite message alphabet. In this model, a program has two parts: a control part and a channel part. The con-