1. History

Prior to discussing what I see as desirable and achievable features of the next generation of interactive theorem provers, I want to say something about the history of my own work and that of my colleagues, which forms the basis for the view of the future I sketch in the remainder of this paper. Simple uses of an interactive theorem prover for the teaching of elementary logic began more than twenty years ago. I remember well our first demonstrations with elementary-school children in 1963. For a number of years we concentrated on teaching elementary logic and algebra to bright elementary- and middle-school children. We felt at the time that this was the right level of difficulty to reach for in terms of computer capacity and resources that could be devoted to the endeavor. All of this early work was done on one of the low-serial-number PDP-1's, which John McCarthy and I jointly purchased from grants at Stanford in 1963. This early work has been described in Suppes and Binfords (1965) and Suppes (1972).

By the late sixties it became clear that we could aim at something a step more advanced, and by 1972 I was able to convert the elementary-logic course at Stanford to a course taught entirely at computer terminals. By that time we had introduced a better and more powerful interactive theorem prover. That course is probably the longest-running show anywhere on earth having an interactive theorem prover used on a regular basis day in and day out by large numbers of persons. Approximately 100 students each term at Stanford enroll for this course and it is given every term. Access to the computer is pretty much around the clock seven days a week so that at almost any time of the day or night a routine use of our interactive theorem prover is taking place. The content of the course in elementary logic is comparable to that of my text (Suppes, 1957). It is obvious enough that the theorem-proving demands of such an elementary course are not very severe. To update also the computer framework, by the early seventies we had moved from a PDP-1 to a PDP-KA10.

The next natural move up was to a course in axiomatic set theory, roughly corresponding to my text in the subject (Suppes, 1960/1972). Here there were nontrivial theorems to be proved, above all the classical organization of a mathematical subject into a long sequence of theorems, with no hope of individual theorems' being proved from scratch directly from the axioms. Since 1974 this
The course also has been offered every term as a course in computer-assisted instruction, with students' getting all of the instruction at terminals. The enrollment is much smaller than that of the logic course. The average enrollment each term is eight or nine students, but the enrollment is greater than it was in the days before the course was computerized. By the time this course was introduced we had moved to a PDP-K10, and a little later we were able to add a second K10. We are still running both courses on the two K10s running as a dual processor. We have ported the logic course to a number of other systems and it is running in 15 or 20 places around the world, but the set theory is a much more elaborate course. It is probably a matter of at least two man-years to convert the course to a portable framework. The details of the set-theory course are described in several articles in Suppes (1981) and more recently in McDonald and Suppes (1984). I shall not recapitulate more details than you will want to hear about this course. It is organized in terms of somewhat more than 600 theorems. Depending upon the grades students seek they prove somewhere between 30 and more than 50 theorems. Some of the theorems are too hard to require them to prove in a beginning course of this kind. After all, one of the theorems on transfinite induction is essentially the main content of von Neumann's dissertation. I also do not want to give the wrong impression about the finished character of the setup. I think that the interactive theorem prover we are using has many good features but it is still awkward to prove the hardest theorems in the sequence. Much work remains to be done to make the proofs of those theorems as natural and easy as they really should be.

Roughly speaking, I would describe the main features of the interactive theorem prover used in the set-theory course under three headings. First, elementary rules of inference, roughly the kind we associate with first-order logic, are available and can be used by the students. Second, students can call a resolution theorem prover that will run for a few seconds of machine time. What they give as input to the theorem prover are the definitions or previous theorems that are to be used to infer a desired formula. Third, a goal structure is provided for helping the student in determining the structure of his proof. A goal structure represents the kind of expert knowledge that is not yet perfected in all respects but that is used extensively by the students and provides a good deal of meaningful assistance.

There are several important remarks to be made about the way in which the students use the theorem prover. First, the most frequently used rule is the use of a previous theorem to make a fairly direct inference. Second, contrast must be sharply drawn between the highly interactive nature of the way the students use the theorem prover and the way proofs can be printed out under a review function at the end. The interactive phase of creating the proof looks from the outside world like a mess. It has the kind of highly interactive discourse structure that is not easy