Notes on ISAC’s Dialogues

Walther Neuper
neuper@ist.tugraz.at
March 19, 2008

There is a novel precondition for dialog design: ISAC’s math-engine is powerful enough to feature a dialog between partners on an equal base – both of the partners, the learner and the system, can do a step in the calculation, both can reject or accept a step of the other partner!

Stepwise calculation can be broken down to minimal parts of interaction; such a part concerns a step from the current formula \( f \) applying a tactic \( tac \) which yields the resulting formula \( f' \) (the derivation of \( f \)), i.e. \( f \xrightarrow{tac} f' \). These parts are called ’dialog atoms’ and ordered by descending ’activity’ of the learner:

1. \([X1]^{1}\) given \( f \), input the next formula \( f' \)
2. given a partial \( f' \) (supplied by ISAC), complete \( f' \) such that it is a derivation of \( f \)
3. given \( f \), input a tactic \( tac \) to be applied to \( f \)
4. given \( f \), select \( tac \) from a list (supplied by ISAC) to be applied to \( f \). There are several possibilities for ISAC to compile the list:
   - \([X1]\) take all tactics from the current method (regardless the applicability of the tactic)
   - take all applicable tactics from the method
   - \([X2]^{2}\) take all applicable atomic rewrite tactics from the method (i.e. rewrite tactics applying only one theorem or one calculation).
   - ...
5. given \( f \) and a partial \( tac \), complete the \( tac \) (i.e. a theorem, a substitution, etc.) such that it can be applied to \( f \)

---

1Items marked with \([X1]\) have been implemented in ISAC by December 2003.
2Items marked with \([X2]\) have been implemented in ISAC by February 2008, inspired by a field test. An additional requirement was to present the theorems as simple as possible, e.g. without questionmarks and without conditions.
6. given \( f \), \( tac \), and a partial \( f' \), complete \( f' \) such that it is the result of applying \( tac \) to \( f \)

7. given \( f \) and \( f' \), input \( tac \) such that \( f' \) is the result of \( f \) applying \( tac \)

8. given \( f \) and \( f' \), select \( tac \) from a list (supplied by ISAC) such that \( f' \) is the result of \( f \) applying \( tac \)

9. given \( f \), \( f' \) and a partial \( tac \), complete \( tac \) such that \( f' \) is the result of \( f \) applying \( tac \)

10. given \( f \), hit a button to get a \( tac \) to apply

11. given (an applicable) \( tac \), hit a button to get \( f' \)

12. [X1] given \( f \), hit a button to get \( f' \)

[Neu01], p.112 ff, tries basic considerations to establish a symmetry in the dialog between the user and the system: (1.) is symmetric to (12.), etc.

The surprising number of dialog atoms states a real challenge for ISAC’s dialog design, which we hope to accomplish in cooperation with experts in didactics and learning theory.

References