Constructive Logic (15-317), Spring 2022
Recitation 14: Session Types and Review

Recall from lecture that we translated a finite-state transducer which compressed runs of b into single instances of b into an ordered program in two different ways — first as an ordered forward logic program, then as a concurrent program in the subsingleton fragment of ordered logic. We will look here at some further examples of this type.

Task 1. Write a transducer over the alphabet a, b which produces ab for every occurrence of ab in the input and erases all other symbols.

1. Present it in the form of ordered inference rules (for a forward ordered logic program).
2. Present it in the form of a well-typed concurrent program in the subsingleton fragment of ordered logic.

Task 2. Reconsider the transducers for compressing runs of b's, given here as a set of ordered inference rules. We present here the version without an explicit final state.

$$\frac{a\ q_0}{q_0\ a} \qquad \frac{b\ q_0}{q_1\ b} \qquad \frac{\$\ q_0}{\$}$$

$$\frac{a\ q_1}{q_0\ a} \qquad \frac{b\ q_1}{q_1} \qquad \frac{\$\ q_1}{\$}$$

In our encoding as a program Q_0 of type $\mathit{string} \vdash Q_0 : \mathit{string}$ we treated letters as messages and states as processes. No explicit representation of the final state is necessary with the rules above.

Define a dual encoding where symbols of the alphabet and endmarkers are represented processes and states as messages.

1. Define an appropriate type *state* so that $\mathit{state} \vdash P_a : \mathit{state}$ where P_a is the process representation for the alphabet symbol a.
2. For each symbol a of the transducer alphabet, define the process P_a.
3. Give the type of the process $P_\$$ representing the endmarker \$. It may make sense to represent a final state with an explicit message, but you may also find it simpler not to.
4. Define the process $P_\$$ for the endmarker.
5. Define the initial configuration for the string *babb* and initial state q_0.
6. Describe the final configuration for the given example string and initial state (once the program has run to completion).
7. Consider how to compose two transducers encoded in this form. How does this compare to the composition of transducers in the original encoding given in lecture (via cut)?

1 Review

You should ask any questions you may have about the material of this course, either as preparation for the final, or for general interest. Below is a high-level list of the topics covered in the course, which you might expect to see come up to some extent on the final exam.

1. Proof Theory:
    - Natural Deduction
    - Harmony
    - Verifications and Uses
    - Proof terms
    - Sequent Calculus
    - Cut and Identity
    - Proof normalization/cut elimination
2. Proof Search:
    - Reduced Sequent Calculus
    - Inversion and the inversion calculus G4IP
    - Logic Programming
    - Prolog
    - Backwards Chaining
    - Forwards Chaining and Forwards Logic Programming
    - Focusing
3. Other logics and extensions to constructive logic:
    - Quantifiers
    - Heyting Arithmetic
    - Classical Logic
    - Modal Logic
    - Linear Logic
    - Ordered Logic
    - Ordered/Linear Logic proofs as concurrent programs
    - (Forward) Ordered/Linear Logic Programs