

ITP Exercise 7

1 Final Project

You should all have started working on your final project by now. The last lecture will take place Monday, 12th June. Practical sessions will continue till 23rd June. There will be an additional practical session on 19 June during the slot normally used for the lecture. The final project is at the very latest due on the 23rd. If other agreements have been reached in person, these take precedence. Despite the late deadline, you are still required to turn up to at least one practical session each week. I highly recommend attending more than one each week.

2 Exercise 7

Since you are working on the final project, exercise 7 is optional. You are not required to work it and there is no deadline.

Define the reflexive, transitive closure of a relation using 3 different methods as shown below. `RTC_REL` is defining it using the inductive relation library. `RTC_DIRECT` is using a simple higher order logic definition and `RTC_REC` uses recursion on natural numbers to define it. Show that all three definitions are defining the same function, i. e. show $(RTC_REL = RTC_DIRECT) \wedge (RTC_REL = RTC_REC) \wedge (RTC_REC = RTC_DIRECT)$.

```
val (RTC_REL_rules, RTC_REL_ind, RTC_REL_cases) = Hol_reln ‘
  (!x y.    R x y                               ==> RTC_REL R x y) /\
  (!x.      RTC_REL R x x                       RTC_REL R x x) /\
  (!x y z.  RTC_REL R x y /\ RTC_REL R y z ==> RTC_REL R x z)‘

val RTC_DIRECT_def = new_definition ("RTC_DIRECT",
  ‘‘RTC_DIRECT R = \ (a:'a) (b:'a). !P.
    ((!x. P x x) /\ (!x y. R x y ==> P x y) /\
     (!x y z. (P x y /\ P y z) ==> P x z)) ==>
    (P a b)‘‘);

val RTC_REC_NUM_def = Define ‘
  (RTC_REC_NUM 0 R x y <=> (x = y)) /\
  (RTC_REC_NUM (SUC n) R x y <=> (?z. R x z /\ RTC_REC_NUM n R z y))‘

val RTC_REC_def = Define ‘
  RTC_REC R x y = ?n. RTC_REC_NUM n R x y‘;
```